Generative Adversarial Neural Networks

April 19, 2025

This document is the requested PDF conversion of the Jupyter Notebook, including all output cells.

Authors:

Fabio Cozzuto Student ID: 002214965

Johan Mogollon Student ID: 002359844

Contributions:

- Fabio Cozzuto: All code, experiments, and analysis
- Johan Mogollon: All code, experiments, and analysis

Course: CS551 - Deep Learning

Professor: Dr. Mohammed Ayoub Alaoui Mhamdi

1 PART 1: Deep Convolutional GAN

1.1 Environment Setup & Imports

[1]:
```
# Ensure Jupyter can import our GAN modules
import os, sys
sys.path.insert(0, os.path.abspath('.'))
sys.path.append('.')

# --- Standard Libraries ---
import warnings
warnings.filterwarnings("ignore")

# --- Data Handling ---
import numpy as np
from PIL import Image
import math
np.math = math

# --- PyTorch ---
import torch
import torch.nn as nn
```

```
import torch.nn.functional as F
import torch.optim as optim
from torch.utils.data import DataLoader
from torchvision import transforms
from torch.utils.tensorboard import SummaryWriter

# --- Local Modules ---
from data_loader import get_data_loader, CustomDataSet
from models import DCGenerator, DCDiscriminator, CycleGenerator, conv, deconv,␣
 ↪ResnetBlock
from utils import to_var, to_data, create_dir

# --- Visualization ---
import matplotlib.pyplot as plt
import imageio # For saving images
import matplotlib.image as mpimg

# --- Argument Parsing ---
import argparse

# --- Other ---
import glob

# Set random seed
SEED = 11
np.random.seed(SEED)
torch.manual_seed(SEED)
if torch.cuda.is_available():
  torch.cuda.manual_seed(SEED)

os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

# Check for GPU availability
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
```

```
Using device: cuda
```

1.2 Implement Data Augmentation [10 points]

We implemented the augmentations in the following code:

```python
def get_data_loader(data_path, opts):
    """Creates data loaders.
    """
    basic_transform = transforms.Compose([
        transforms.Resize(opts.image_size, Image.BICUBIC),
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
    ])

    if opts.data_aug == 'basic':
        transform = basic_transform
    elif opts.data_aug == 'deluxe':
        load_size = int(1.1 * opts.image_size)
        osize = [load_size, load_size]
        transform = transforms.Compose([
            transforms.Resize(osize, Image.BICUBIC),
            transforms.RandomCrop(opts.image_size),
            transforms.RandomHorizontalFlip(),
            transforms.ColorJitter(
                brightness=0.3, contrast=0.3, saturation=0.3, hue=0.1),
            transforms.RandomAffine(
                degrees=10, translate=(0.05, 0.05), scale=(0.95, 1.05), shear=5),
            transforms.RandomPerspective(distortion_scale=0.2, p=0.5),
            transforms.ToTensor(),
            transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
        ])
        pass

    dataset = CustomDataSet(os.path.join('data/', data_path), opts.ext, transform)
    dloader = DataLoader(dataset=dataset, batch_size=opts.batch_size, shuffle=True, num_workers=opts.num_workers)

    return dloader
```

1.3 Implement the Discriminator of the DCGAN [10 points]

1.3.1 Padding Calculation for DCGAN Discriminator

Question: With kernel size (K=4) and stride (S=2), what padding (P) halves the spatial dimensions?

Answer: We want each layer to reduce the spatial dimensions by a factor of 2, without clipping important features. That means that we want to control the padding. So, we have the convolution output formula:

$$O = \left\lfloor \frac{I + 2P - K}{S} \right\rfloor + 1$$

Where: - (I) = input size - (O) = output size - (K = 4) (kernel size) - (S = 2) (stride) - (P) = padding

We want to obtain this:

$$output\_size = \frac{input\_size}{2}$$

So we solve as follows:

$$\left\lfloor \frac{I + 2P - 4}{2} \right\rfloor + 1 = \frac{I}{2} \Rightarrow 2P = 2 \Rightarrow P = 1$$

[ ]:
```python
# We can do the same calculations with the following code:
```

```python
input_size = 64  # Example input size, this will vary per layer
kernel_size = 4
stride = 2
padding = 1
output_size = (input_size - kernel_size + 2 * padding) / stride + 1


print(f"Given kernel_size={kernel_size}, stride={stride}, the required padding␣
 ↪is: {padding}")
print(f"Example: Input size = {input_size}, Output size = {output_size}")
```

```
Given kernel_size=4, stride=2, the required padding is: 1
Example: Input size = 64, Output size = 32.0
```

1.3.2 DCDiscriminator class in the models.py file

We completed the code for DCDsicriminator as you can see in the following image:

```python
class DCDiscriminator(nn.Module):
    """Defines the architecture of the discriminator network.
       Note: Both discriminators D_X and D_Y have the same architecture in this assignment.
    """
    def __init__(self, conv_dim=64, norm='batch'):
        super(DCDiscriminator, self).__init__()

        ########################################
        ##   FILL THIS IN: CREATE ARCHITECTURE   ##
        ########################################

        self.conv1 = conv(3, conv_dim, kernel_size=4, stride=2, padding=1, norm=None)
        self.conv2 = conv(conv_dim, conv_dim * 2, kernel_size=4, stride=2, padding=1, norm=norm)
        self.conv3 = conv(conv_dim * 2, conv_dim * 4, kernel_size=4, stride=2, padding=1, norm=norm)
        self.conv4 = conv(conv_dim * 4, conv_dim * 8, kernel_size=4, stride=2, padding=1, norm=norm)
        self.conv5 = nn.Sequential(
            nn.AdaptiveAvgPool2d(1),
            nn.Conv2d(conv_dim * 8, 1, kernel_size=1, stride=1, padding=0, bias=False)
        )

    def forward(self, x):
        ########################################
        ##   FILL THIS IN: FORWARD PASS   ##
        ########################################

        out = F.leaky_relu(self.conv1(x), 0.2)
        out = F.leaky_relu(self.conv2(out), 0.2)
        out = F.leaky_relu(self.conv3(out), 0.2)
        out = F.leaky_relu(self.conv4(out), 0.2)
        out = self.conv5(out)
        out = torch.sigmoid(out)
        return out.view(out.size(0), -1).mean(1)
```

1.4 Implement the Generator of the DCGAN [10 points]

1.4.1 DCGenerator class in the models.py file

```python
class DCGenerator(nn.Module):
    def __init__(self, noise_size, conv_dim):

        ########################################
        ##   FILL THIS IN: CREATE ARCHITECTURE   ##
        ########################################

        self.deconv1 = deconv(noise_size, conv_dim * 8, kernel_size=4, stride=1, padding=0, norm='batch')
        self.deconv2 = deconv(conv_dim * 8, conv_dim * 4, kernel_size=4, stride=2, padding=1, norm='batch')
        self.deconv3 = deconv(conv_dim * 4, conv_dim * 2, kernel_size=4, stride=2, padding=1, norm='batch')
        self.deconv4 = deconv(conv_dim * 2, conv_dim, kernel_size=4, stride=2, padding=1, norm='batch')
        self.deconv5 = nn.Sequential(
            nn.ConvTranspose2d(conv_dim, 3, kernel_size=4, stride=2, padding=1),
            nn.Tanh()
        )

    def forward(self, z):
        """Generates an image given a sample of random noise.

            Input
            -----
                z: BS x noise_size x 1 x 1   -->  16x100x1x1

            Output
            ------
                out: BS x channels x image_width x image_height  -->  16x3x32x32
        """


        ########################################
        ##   FILL THIS IN: FORWARD PASS   ##
        ########################################

        out = F.relu(self.deconv1(z))
        out = F.relu(self.deconv2(out))
        out = F.relu(self.deconv3(out))
        out = F.relu(self.deconv4(out))
        out = F.tanh(self.deconv5(out))
        return out
```

1.5 Experiments

1.5.1 Implement the DCGAN Training Loop [10 points]

Discriminator

```
for batch in train_dataloader:

    real_images, labels = batch
    real_images, labels = utils.to_var(real_images), utils.to_var(labels).long().squeeze()

    ##############################################
    ###         TRAIN THE DISCRIMINATOR         ####
    ##############################################

    d_optimizer.zero_grad()

    # FILL THIS IN
    # 1. Compute the discriminator loss on real images
    D_real_loss = criterion(D(real_images), torch.ones(real_images.size(0)).to(real_images.device))

    # 2. Sample noise
    noise = sample_noise(opts.noise_size)

    # 3. Generate fake images from the noise
    fake_images = G(noise)

    # 4. Compute the discriminator loss on the fake images
    D_fake_loss = criterion(D(fake_images.detach()), torch.zeros(fake_images.size(0)).to(fake_images.device))

    D_total_loss = D_real_loss + D_fake_loss
    if iteration % 2 == 0:
        D_total_loss.backward()
        d_optimizer.step()
```

Generator

For this part we filled the code and carefully add the logger to work with TensorBoard

```
##########################################
###          TRAIN THE GENERATOR        ###
##########################################

g_optimizer.zero_grad()

# FILL THIS IN
# 1. Sample noise
noise = sample_noise(opts.noise_size)

# 2. Generate fake images from the noise
fake_images = G(noise)

# 3. Compute the generator loss
G_loss = criterion(D(fake_images), torch.ones(fake_images.size(0)).to(fake_images.device))

G_loss.backward()
g_optimizer.step()


# Print the log info
if iteration % opts.log_step == 0:
    logger.add_scalar('D/real_loss', D_real_loss.item(), iteration)
    logger.add_scalar('D/fake_loss', D_fake_loss.item(), iteration)
    logger.add_scalar('G/loss', G_loss.item(), iteration)
    print('Iteration [{:4d}/{:4d}] | D_real_loss: {:6.4f} | D_fake_loss: {:6.4f} | G_loss: {:6.4f}'.format(
          iteration, total_train_iters, D_real_loss.item(), D_fake_loss.item(), G_loss.item()))
```

1.5.2 Train the DCGAN [10 points]

The following code traiin the DCGAN, so this is the first execution we can do to understand the model and to see if there is some kind of error.

[14]:
```
!python vanilla_gan.py --num_epochs=100
```

```
Namespace(image_size=64, conv_dim=32, noise_size=100, num_epochs=100,
batch_size=16, num_workers=0, lr=0.0003, beta1=0.5, beta2=0.999,
data='cat/grumpifyBprocessed', data_aug='deluxe', ext='*.png',
checkpoint_dir='./checkpoints_vanilla',
sample_dir='output/./vanilla\\grumpifyBprocessed_deluxe', log_step=10,
sample_every=200, checkpoint_every=400)
data/cat/grumpifyBprocessed\*.png
204
                     G
---------------------------------------
DCGenerator(
  (deconv1): Sequential(
    (0): ConvTranspose2d(100, 256, kernel_size=(4, 4), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv2): Sequential(
    (0): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2),
padding=(1, 1), bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv3): Sequential(
    (0): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv4): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv5): Sequential(
    (0): ConvTranspose2d(32, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1))
    (1): Tanh()
  )
)
---------------------------------------
                     D
---------------------------------------
DCDiscriminator(
  (conv1): Sequential(
```

```
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
---------------------------------------
Iteration [  10/1300] | D_real_loss: 0.6178 | D_fake_loss: 0.7502 | G_loss:
0.6525
Iteration [  20/1300] | D_real_loss: 0.6142 | D_fake_loss: 0.6889 | G_loss:
0.7106
Iteration [  30/1300] | D_real_loss: 0.6136 | D_fake_loss: 0.6380 | G_loss:
0.7735
Iteration [  40/1300] | D_real_loss: 0.6197 | D_fake_loss: 0.5993 | G_loss:
0.8131
Iteration [  50/1300] | D_real_loss: 0.5597 | D_fake_loss: 0.5798 | G_loss:
0.8414
Iteration [  60/1300] | D_real_loss: 0.5511 | D_fake_loss: 0.5561 | G_loss:
0.8760
Iteration [  70/1300] | D_real_loss: 0.5666 | D_fake_loss: 0.5429 | G_loss:
0.8949
Iteration [  80/1300] | D_real_loss: 0.5230 | D_fake_loss: 0.5236 | G_loss:
0.9362
Iteration [  90/1300] | D_real_loss: 0.4994 | D_fake_loss: 0.5078 | G_loss:
0.9562
Iteration [ 100/1300] | D_real_loss: 0.4876 | D_fake_loss: 0.5042 | G_loss:
0.9720
Iteration [ 110/1300] | D_real_loss: 0.5292 | D_fake_loss: 0.5037 | G_loss:
```

```
0.9872
Iteration [ 120/1300] | D_real_loss: 0.4859 | D_fake_loss: 0.4822 | G_loss:
1.0049
Iteration [ 130/1300] | D_real_loss: 0.4613 | D_fake_loss: 0.4800 | G_loss:
1.0250
Iteration [ 140/1300] | D_real_loss: 0.4790 | D_fake_loss: 0.4701 | G_loss:
1.0304
Iteration [ 150/1300] | D_real_loss: 0.5439 | D_fake_loss: 0.4753 | G_loss:
1.0636
Iteration [ 160/1300] | D_real_loss: 0.5240 | D_fake_loss: 0.7516 | G_loss:
0.8212
Iteration [ 170/1300] | D_real_loss: 0.7106 | D_fake_loss: 0.6148 | G_loss:
0.9410
Iteration [ 180/1300] | D_real_loss: 0.6782 | D_fake_loss: 0.6707 | G_loss:
0.8691
Iteration [ 190/1300] | D_real_loss: 0.6560 | D_fake_loss: 0.6227 | G_loss:
0.9067
Iteration [ 200/1300] | D_real_loss: 0.6389 | D_fake_loss: 0.5957 | G_loss:
0.9406
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000200.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000200.png
Iteration [ 210/1300] | D_real_loss: 0.6120 | D_fake_loss: 0.6410 | G_loss:
0.7780
Iteration [ 220/1300] | D_real_loss: 0.6320 | D_fake_loss: 0.6819 | G_loss:
0.8670
Iteration [ 230/1300] | D_real_loss: 0.6197 | D_fake_loss: 0.6466 | G_loss:
0.8606
Iteration [ 240/1300] | D_real_loss: 0.6379 | D_fake_loss: 0.6505 | G_loss:
0.8493
Iteration [ 250/1300] | D_real_loss: 0.5951 | D_fake_loss: 0.6552 | G_loss:
0.8564
Iteration [ 260/1300] | D_real_loss: 0.6986 | D_fake_loss: 0.6543 | G_loss:
0.8097
Iteration [ 270/1300] | D_real_loss: 0.6528 | D_fake_loss: 0.6498 | G_loss:
0.8544
Iteration [ 280/1300] | D_real_loss: 0.6566 | D_fake_loss: 0.6575 | G_loss:
0.7644
Iteration [ 290/1300] | D_real_loss: 0.6537 | D_fake_loss: 0.6390 | G_loss:
0.8233
Iteration [ 300/1300] | D_real_loss: 0.6749 | D_fake_loss: 0.6065 | G_loss:
0.7766
Iteration [ 310/1300] | D_real_loss: 0.6958 | D_fake_loss: 0.6174 | G_loss:
0.8497
Iteration [ 320/1300] | D_real_loss: 0.6245 | D_fake_loss: 0.7284 | G_loss:
0.8159
Iteration [ 330/1300] | D_real_loss: 0.5676 | D_fake_loss: 0.6776 | G_loss:
0.8000
Iteration [ 340/1300] | D_real_loss: 0.6663 | D_fake_loss: 0.6813 | G_loss:
```

```
0.8646
Iteration [ 350/1300] | D_real_loss: 0.6642 | D_fake_loss: 0.6103 | G_loss:
0.8398
Iteration [ 360/1300] | D_real_loss: 0.6668 | D_fake_loss: 0.5932 | G_loss:
0.7523
Iteration [ 370/1300] | D_real_loss: 0.7290 | D_fake_loss: 0.6219 | G_loss:
0.7837
Iteration [ 380/1300] | D_real_loss: 0.6581 | D_fake_loss: 0.6134 | G_loss:
0.8027
Iteration [ 390/1300] | D_real_loss: 0.6969 | D_fake_loss: 0.6002 | G_loss:
0.7968
Iteration [ 400/1300] | D_real_loss: 0.6334 | D_fake_loss: 0.6706 | G_loss:
0.7645
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000400.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000400.png
Iteration [ 410/1300] | D_real_loss: 0.6471 | D_fake_loss: 0.6601 | G_loss:
0.7701
Iteration [ 420/1300] | D_real_loss: 0.6209 | D_fake_loss: 0.6324 | G_loss:
0.7965
Iteration [ 430/1300] | D_real_loss: 0.7349 | D_fake_loss: 0.6337 | G_loss:
0.9316
Iteration [ 440/1300] | D_real_loss: 0.6383 | D_fake_loss: 0.6932 | G_loss:
0.7909
Iteration [ 450/1300] | D_real_loss: 0.6106 | D_fake_loss: 0.7087 | G_loss:
0.8567
Iteration [ 460/1300] | D_real_loss: 0.6718 | D_fake_loss: 0.6049 | G_loss:
0.7786
Iteration [ 470/1300] | D_real_loss: 0.6738 | D_fake_loss: 0.6880 | G_loss:
0.8174
Iteration [ 480/1300] | D_real_loss: 0.7419 | D_fake_loss: 0.6626 | G_loss:
0.8104
Iteration [ 490/1300] | D_real_loss: 0.7214 | D_fake_loss: 0.6542 | G_loss:
0.7857
Iteration [ 500/1300] | D_real_loss: 0.7249 | D_fake_loss: 0.6264 | G_loss:
0.8200
Iteration [ 510/1300] | D_real_loss: 0.6940 | D_fake_loss: 0.6559 | G_loss:
0.7781
Iteration [ 520/1300] | D_real_loss: 0.7040 | D_fake_loss: 0.7257 | G_loss:
0.7409
Iteration [ 530/1300] | D_real_loss: 0.6720 | D_fake_loss: 0.6289 | G_loss:
0.7615
Iteration [ 540/1300] | D_real_loss: 0.5934 | D_fake_loss: 0.7089 | G_loss:
0.7076
Iteration [ 550/1300] | D_real_loss: 0.6582 | D_fake_loss: 0.7226 | G_loss:
0.7482
Iteration [ 560/1300] | D_real_loss: 0.6229 | D_fake_loss: 0.6692 | G_loss:
0.7901
Iteration [ 570/1300] | D_real_loss: 0.7151 | D_fake_loss: 0.6038 | G_loss:
```

```
0.8265
Iteration [ 580/1300] | D_real_loss: 0.6674 | D_fake_loss: 0.6851 | G_loss:
0.7460
Iteration [ 590/1300] | D_real_loss: 0.6905 | D_fake_loss: 0.6562 | G_loss:
0.7693
Iteration [ 600/1300] | D_real_loss: 0.6324 | D_fake_loss: 0.6939 | G_loss:
0.7982
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000600.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000600.png
Iteration [ 610/1300] | D_real_loss: 0.6283 | D_fake_loss: 0.6857 | G_loss:
0.7418
Iteration [ 620/1300] | D_real_loss: 0.7039 | D_fake_loss: 0.6603 | G_loss:
0.7497
Iteration [ 630/1300] | D_real_loss: 0.6515 | D_fake_loss: 0.6722 | G_loss:
0.7772
Iteration [ 640/1300] | D_real_loss: 0.7148 | D_fake_loss: 0.6557 | G_loss:
0.7932
Iteration [ 650/1300] | D_real_loss: 0.6826 | D_fake_loss: 0.6561 | G_loss:
0.7732
Iteration [ 660/1300] | D_real_loss: 0.6969 | D_fake_loss: 0.6583 | G_loss:
0.7778
Iteration [ 670/1300] | D_real_loss: 0.6485 | D_fake_loss: 0.6999 | G_loss:
0.7190
Iteration [ 680/1300] | D_real_loss: 0.6767 | D_fake_loss: 0.6705 | G_loss:
0.7509
Iteration [ 690/1300] | D_real_loss: 0.6604 | D_fake_loss: 0.6965 | G_loss:
0.7336
Iteration [ 700/1300] | D_real_loss: 0.6732 | D_fake_loss: 0.6392 | G_loss:
0.7836
Iteration [ 710/1300] | D_real_loss: 0.6775 | D_fake_loss: 0.7133 | G_loss:
0.7151
Iteration [ 720/1300] | D_real_loss: 0.6418 | D_fake_loss: 0.6631 | G_loss:
0.7708
Iteration [ 730/1300] | D_real_loss: 0.6601 | D_fake_loss: 0.7017 | G_loss:
0.7401
Iteration [ 740/1300] | D_real_loss: 0.7210 | D_fake_loss: 0.6251 | G_loss:
0.8006
Iteration [ 750/1300] | D_real_loss: 0.6619 | D_fake_loss: 0.6948 | G_loss:
0.7658
Iteration [ 760/1300] | D_real_loss: 0.5669 | D_fake_loss: 0.7481 | G_loss:
0.7076
Iteration [ 770/1300] | D_real_loss: 0.6805 | D_fake_loss: 0.7217 | G_loss:
0.7409
Iteration [ 780/1300] | D_real_loss: 0.7222 | D_fake_loss: 0.6480 | G_loss:
0.7985
Iteration [ 790/1300] | D_real_loss: 0.7094 | D_fake_loss: 0.6230 | G_loss:
0.8059
Iteration [ 800/1300] | D_real_loss: 0.6500 | D_fake_loss: 0.6729 | G_loss:
```

```
0.7564
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000800.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000800.png
Iteration [ 810/1300] | D_real_loss: 0.6832 | D_fake_loss: 0.6614 | G_loss:
0.7462
Iteration [ 820/1300] | D_real_loss: 0.6681 | D_fake_loss: 0.6714 | G_loss:
0.7489
Iteration [ 830/1300] | D_real_loss: 0.7004 | D_fake_loss: 0.6755 | G_loss:
0.7295
Iteration [ 840/1300] | D_real_loss: 0.6868 | D_fake_loss: 0.6635 | G_loss:
0.8329
Iteration [ 850/1300] | D_real_loss: 0.6399 | D_fake_loss: 0.6747 | G_loss:
0.7613
Iteration [ 860/1300] | D_real_loss: 0.6409 | D_fake_loss: 0.6963 | G_loss:
0.7778
Iteration [ 870/1300] | D_real_loss: 0.6527 | D_fake_loss: 0.6924 | G_loss:
0.7613
Iteration [ 880/1300] | D_real_loss: 0.6977 | D_fake_loss: 0.6707 | G_loss:
0.7706
Iteration [ 890/1300] | D_real_loss: 0.6751 | D_fake_loss: 0.6641 | G_loss:
0.7731
Iteration [ 900/1300] | D_real_loss: 0.6902 | D_fake_loss: 0.6611 | G_loss:
0.7379
Iteration [ 910/1300] | D_real_loss: 0.6683 | D_fake_loss: 0.6636 | G_loss:
0.8229
Iteration [ 920/1300] | D_real_loss: 0.6484 | D_fake_loss: 0.7391 | G_loss:
0.7344
Iteration [ 930/1300] | D_real_loss: 0.7164 | D_fake_loss: 0.6408 | G_loss:
0.7948
Iteration [ 940/1300] | D_real_loss: 0.6957 | D_fake_loss: 0.6637 | G_loss:
0.7455
Iteration [ 950/1300] | D_real_loss: 0.6673 | D_fake_loss: 0.6648 | G_loss:
0.7576
Iteration [ 960/1300] | D_real_loss: 0.6907 | D_fake_loss: 0.6528 | G_loss:
0.7636
Iteration [ 970/1300] | D_real_loss: 0.6444 | D_fake_loss: 0.6727 | G_loss:
0.7208
Iteration [ 980/1300] | D_real_loss: 0.6735 | D_fake_loss: 0.6547 | G_loss:
0.7849
Iteration [ 990/1300] | D_real_loss: 0.6800 | D_fake_loss: 0.6982 | G_loss:
0.7726
Iteration [1000/1300] | D_real_loss: 0.6413 | D_fake_loss: 0.6870 | G_loss:
0.7552
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-001000.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-001000.png
Iteration [1010/1300] | D_real_loss: 0.7567 | D_fake_loss: 0.5987 | G_loss:
0.8225
Iteration [1020/1300] | D_real_loss: 0.6467 | D_fake_loss: 0.6800 | G_loss:
```

```
0.7801
Iteration [1030/1300] | D_real_loss: 0.6472 | D_fake_loss: 0.7130 | G_loss:
0.7658
Iteration [1040/1300] | D_real_loss: 0.6661 | D_fake_loss: 0.6763 | G_loss:
0.7950
Iteration [1050/1300] | D_real_loss: 0.7084 | D_fake_loss: 0.6860 | G_loss:
0.7460
Iteration [1060/1300] | D_real_loss: 0.6644 | D_fake_loss: 0.6767 | G_loss:
0.7474
Iteration [1070/1300] | D_real_loss: 0.6245 | D_fake_loss: 0.7047 | G_loss:
0.7266
Iteration [1080/1300] | D_real_loss: 0.6756 | D_fake_loss: 0.6548 | G_loss:
0.7885
Iteration [1090/1300] | D_real_loss: 0.7132 | D_fake_loss: 0.6741 | G_loss:
0.7870
Iteration [1100/1300] | D_real_loss: 0.6345 | D_fake_loss: 0.6880 | G_loss:
0.7423
Iteration [1110/1300] | D_real_loss: 0.6813 | D_fake_loss: 0.6694 | G_loss:
0.7458
Iteration [1120/1300] | D_real_loss: 0.6820 | D_fake_loss: 0.6678 | G_loss:
0.7760
Iteration [1130/1300] | D_real_loss: 0.6544 | D_fake_loss: 0.6448 | G_loss:
0.7543
Iteration [1140/1300] | D_real_loss: 0.6575 | D_fake_loss: 0.6643 | G_loss:
0.7685
Iteration [1150/1300] | D_real_loss: 0.6669 | D_fake_loss: 0.6918 | G_loss:
0.7588
Iteration [1160/1300] | D_real_loss: 0.7278 | D_fake_loss: 0.6173 | G_loss:
0.7919
Iteration [1170/1300] | D_real_loss: 0.6707 | D_fake_loss: 0.6825 | G_loss:
0.7289
Iteration [1180/1300] | D_real_loss: 0.6403 | D_fake_loss: 0.6737 | G_loss:
0.7620
Iteration [1190/1300] | D_real_loss: 0.6769 | D_fake_loss: 0.6723 | G_loss:
0.7601
Iteration [1200/1300] | D_real_loss: 0.7513 | D_fake_loss: 0.6694 | G_loss:
0.8829
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-001200.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-001200.png
Iteration [1210/1300] | D_real_loss: 0.6822 | D_fake_loss: 0.6871 | G_loss:
0.7733
Iteration [1220/1300] | D_real_loss: 0.6469 | D_fake_loss: 0.6713 | G_loss:
0.7590
Iteration [1230/1300] | D_real_loss: 0.6481 | D_fake_loss: 0.6802 | G_loss:
0.7612
Iteration [1240/1300] | D_real_loss: 0.6213 | D_fake_loss: 0.6889 | G_loss:
0.7453
Iteration [1250/1300] | D_real_loss: 0.6690 | D_fake_loss: 0.6901 | G_loss:
```

```
0.7484
Iteration [1260/1300] | D_real_loss: 0.6924 | D_fake_loss: 0.6529 | G_loss:
0.7544
Iteration [1270/1300] | D_real_loss: 0.6670 | D_fake_loss: 0.6639 | G_loss:
0.7559
Iteration [1280/1300] | D_real_loss: 0.6490 | D_fake_loss: 0.7078 | G_loss:
0.7389
Iteration [1290/1300] | D_real_loss: 0.6926 | D_fake_loss: 0.6488 | G_loss:
0.7875
Iteration [1300/1300] | D_real_loss: 0.6655 | D_fake_loss: 0.7227 | G_loss:
0.7622
```

Basic Execution

[15]:
```
!python vanilla_gan.py --data_aug=basic  --num_epochs=100
```

```
Namespace(image_size=64, conv_dim=32, noise_size=100, num_epochs=100,
batch_size=16, num_workers=0, lr=0.0003, beta1=0.5, beta2=0.999,
data='cat/grumpifyBprocessed', data_aug='basic', ext='*.png',
checkpoint_dir='./checkpoints_vanilla',
sample_dir='output/./vanilla\\grumpifyBprocessed_basic', log_step=10,
sample_every=200, checkpoint_every=400)
data/cat/grumpifyBprocessed\*.png
204
                    G
---------------------------------------
DCGenerator(
  (deconv1): Sequential(
    (0): ConvTranspose2d(100, 256, kernel_size=(4, 4), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv2): Sequential(
    (0): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2),
padding=(1, 1), bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv3): Sequential(
    (0): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv4): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
```

```
track_running_stats=True)
  )
  (deconv5): Sequential(
    (0): ConvTranspose2d(32, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1))
    (1): Tanh()
  )
)
---------------------------------------
                    D
---------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
---------------------------------------
Iteration [  10/1300] | D_real_loss: 0.6041 | D_fake_loss: 0.7462 | G_loss:
0.6567
Iteration [  20/1300] | D_real_loss: 0.5838 | D_fake_loss: 0.6873 | G_loss:
0.7164
Iteration [  30/1300] | D_real_loss: 0.5513 | D_fake_loss: 0.6330 | G_loss:
0.7803
Iteration [  40/1300] | D_real_loss: 0.5418 | D_fake_loss: 0.5875 | G_loss:
0.8382
```

```
Iteration [  50/1300] | D_real_loss: 0.5337 | D_fake_loss: 0.5550 | G_loss:
0.8731
Iteration [  60/1300] | D_real_loss: 0.5213 | D_fake_loss: 0.5384 | G_loss:
0.9003
Iteration [  70/1300] | D_real_loss: 0.4992 | D_fake_loss: 0.5252 | G_loss:
0.9241
Iteration [  80/1300] | D_real_loss: 0.4914 | D_fake_loss: 0.5112 | G_loss:
0.9453
Iteration [  90/1300] | D_real_loss: 0.4874 | D_fake_loss: 0.4971 | G_loss:
0.9638
Iteration [ 100/1300] | D_real_loss: 0.4943 | D_fake_loss: 0.4874 | G_loss:
0.9835
Iteration [ 110/1300] | D_real_loss: 0.4520 | D_fake_loss: 0.4971 | G_loss:
0.9847
Iteration [ 120/1300] | D_real_loss: 0.4595 | D_fake_loss: 0.5451 | G_loss:
1.0497
Iteration [ 130/1300] | D_real_loss: 0.4374 | D_fake_loss: 0.4402 | G_loss:
1.0816
Iteration [ 140/1300] | D_real_loss: 0.4315 | D_fake_loss: 0.4444 | G_loss:
1.0870
Iteration [ 150/1300] | D_real_loss: 0.4131 | D_fake_loss: 0.4374 | G_loss:
1.0901
Iteration [ 160/1300] | D_real_loss: 0.4329 | D_fake_loss: 0.4448 | G_loss:
1.0992
Iteration [ 170/1300] | D_real_loss: 0.4062 | D_fake_loss: 0.4186 | G_loss:
1.1299
Iteration [ 180/1300] | D_real_loss: 0.3876 | D_fake_loss: 0.4189 | G_loss:
1.1360
Iteration [ 190/1300] | D_real_loss: 0.3684 | D_fake_loss: 0.4412 | G_loss:
1.1699
Iteration [ 200/1300] | D_real_loss: 0.3676 | D_fake_loss: 0.4654 | G_loss:
1.2078
Saved output/./vanilla\grumpifyBprocessed_basic\sample-000200.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-000200.png
Iteration [ 210/1300] | D_real_loss: 0.3713 | D_fake_loss: 0.3740 | G_loss:
1.2325
Iteration [ 220/1300] | D_real_loss: 0.4286 | D_fake_loss: 0.3788 | G_loss:
1.2523
Iteration [ 230/1300] | D_real_loss: 0.3503 | D_fake_loss: 0.3891 | G_loss:
1.2381
Iteration [ 240/1300] | D_real_loss: 0.3180 | D_fake_loss: 0.3788 | G_loss:
1.2560
Iteration [ 250/1300] | D_real_loss: 0.3246 | D_fake_loss: 0.3573 | G_loss:
1.2735
Iteration [ 260/1300] | D_real_loss: 0.3295 | D_fake_loss: 0.3751 | G_loss:
1.2429
Iteration [ 270/1300] | D_real_loss: 0.5174 | D_fake_loss: 0.4407 | G_loss:
1.2551
```

```
Iteration [ 280/1300] | D_real_loss: 0.4830 | D_fake_loss: 0.5191 | G_loss:
1.1150
Iteration [ 290/1300] | D_real_loss: 0.5271 | D_fake_loss: 0.5288 | G_loss:
0.9770
Iteration [ 300/1300] | D_real_loss: 0.4810 | D_fake_loss: 0.6131 | G_loss:
1.0458
Iteration [ 310/1300] | D_real_loss: 0.6896 | D_fake_loss: 0.4863 | G_loss:
1.0474
Iteration [ 320/1300] | D_real_loss: 0.6540 | D_fake_loss: 0.4578 | G_loss:
0.8292
Iteration [ 330/1300] | D_real_loss: 0.5185 | D_fake_loss: 0.8220 | G_loss:
0.8256
Iteration [ 340/1300] | D_real_loss: 0.6885 | D_fake_loss: 0.5590 | G_loss:
1.0097
Iteration [ 350/1300] | D_real_loss: 0.7085 | D_fake_loss: 0.5771 | G_loss:
0.9611
Iteration [ 360/1300] | D_real_loss: 0.5774 | D_fake_loss: 0.7589 | G_loss:
0.9231
Iteration [ 370/1300] | D_real_loss: 0.6888 | D_fake_loss: 0.6725 | G_loss:
0.9311
Iteration [ 380/1300] | D_real_loss: 0.6640 | D_fake_loss: 0.5923 | G_loss:
0.9852
Iteration [ 390/1300] | D_real_loss: 0.6662 | D_fake_loss: 0.6626 | G_loss:
0.9525
Iteration [ 400/1300] | D_real_loss: 0.5960 | D_fake_loss: 0.6347 | G_loss:
0.9822
Saved output/./vanilla\grumpifyBprocessed_basic\sample-000400.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-000400.png
Iteration [ 410/1300] | D_real_loss: 0.6347 | D_fake_loss: 0.6148 | G_loss:
0.9818
Iteration [ 420/1300] | D_real_loss: 0.6500 | D_fake_loss: 0.6203 | G_loss:
1.0027
Iteration [ 430/1300] | D_real_loss: 0.5928 | D_fake_loss: 0.6687 | G_loss:
0.9824
Iteration [ 440/1300] | D_real_loss: 0.6726 | D_fake_loss: 0.5969 | G_loss:
1.0524
Iteration [ 450/1300] | D_real_loss: 0.6906 | D_fake_loss: 0.5315 | G_loss:
1.0012
Iteration [ 460/1300] | D_real_loss: 0.6217 | D_fake_loss: 0.6266 | G_loss:
0.9712
Iteration [ 470/1300] | D_real_loss: 0.5636 | D_fake_loss: 0.5632 | G_loss:
1.0469
Iteration [ 480/1300] | D_real_loss: 0.6834 | D_fake_loss: 0.5775 | G_loss:
0.9767
Iteration [ 490/1300] | D_real_loss: 0.6297 | D_fake_loss: 0.6364 | G_loss:
0.9065
Iteration [ 500/1300] | D_real_loss: 0.6041 | D_fake_loss: 0.5820 | G_loss:
0.9258
```

```
Iteration [ 510/1300] | D_real_loss: 0.6142 | D_fake_loss: 0.6061 | G_loss:
0.9974
Iteration [ 520/1300] | D_real_loss: 0.6430 | D_fake_loss: 0.6305 | G_loss:
0.9322
Iteration [ 530/1300] | D_real_loss: 0.6289 | D_fake_loss: 0.5630 | G_loss:
0.9481
Iteration [ 540/1300] | D_real_loss: 0.6309 | D_fake_loss: 0.7447 | G_loss:
0.7985
Iteration [ 550/1300] | D_real_loss: 0.6081 | D_fake_loss: 0.6796 | G_loss:
0.8506
Iteration [ 560/1300] | D_real_loss: 0.6798 | D_fake_loss: 0.5958 | G_loss:
0.8812
Iteration [ 570/1300] | D_real_loss: 0.5898 | D_fake_loss: 0.6859 | G_loss:
0.8552
Iteration [ 580/1300] | D_real_loss: 0.6298 | D_fake_loss: 0.5940 | G_loss:
0.9407
Iteration [ 590/1300] | D_real_loss: 0.6382 | D_fake_loss: 0.6427 | G_loss:
0.9366
Iteration [ 600/1300] | D_real_loss: 0.5048 | D_fake_loss: 0.7400 | G_loss:
0.8659
Saved output/./vanilla\grumpifyBprocessed_basic\sample-000600.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-000600.png
Iteration [ 610/1300] | D_real_loss: 0.7615 | D_fake_loss: 0.5498 | G_loss:
0.9830
Iteration [ 620/1300] | D_real_loss: 0.5195 | D_fake_loss: 0.6005 | G_loss:
0.9230
Iteration [ 630/1300] | D_real_loss: 0.5937 | D_fake_loss: 0.6595 | G_loss:
0.8133
Iteration [ 640/1300] | D_real_loss: 0.6385 | D_fake_loss: 0.6600 | G_loss:
0.8613
Iteration [ 650/1300] | D_real_loss: 0.6518 | D_fake_loss: 0.6044 | G_loss:
0.9752
Iteration [ 660/1300] | D_real_loss: 0.5895 | D_fake_loss: 0.6680 | G_loss:
1.0031
Iteration [ 670/1300] | D_real_loss: 0.6929 | D_fake_loss: 0.6069 | G_loss:
0.8651
Iteration [ 680/1300] | D_real_loss: 0.5808 | D_fake_loss: 0.6119 | G_loss:
0.8691
Iteration [ 690/1300] | D_real_loss: 0.6718 | D_fake_loss: 0.5341 | G_loss:
0.9923
Iteration [ 700/1300] | D_real_loss: 0.5873 | D_fake_loss: 0.6372 | G_loss:
0.9082
Iteration [ 710/1300] | D_real_loss: 0.6366 | D_fake_loss: 0.5945 | G_loss:
0.8541
Iteration [ 720/1300] | D_real_loss: 0.6437 | D_fake_loss: 0.6133 | G_loss:
0.8965
Iteration [ 730/1300] | D_real_loss: 0.5955 | D_fake_loss: 0.6461 | G_loss:
0.8945
```

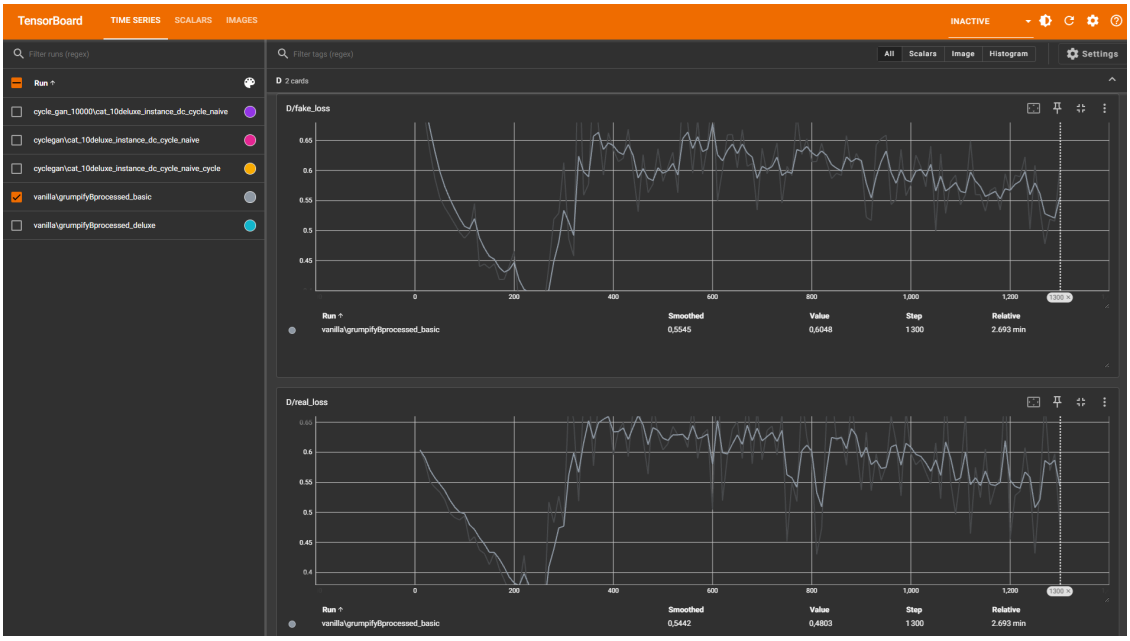
```
Iteration [ 740/1300] | D_real_loss: 0.6640 | D_fake_loss: 0.5458 | G_loss:
0.9287
Iteration [ 750/1300] | D_real_loss: 0.4517 | D_fake_loss: 0.6072 | G_loss:
0.9138
Iteration [ 760/1300] | D_real_loss: 0.5500 | D_fake_loss: 0.5916 | G_loss:
0.8836
Iteration [ 770/1300] | D_real_loss: 0.5180 | D_fake_loss: 0.6942 | G_loss:
0.8717
Iteration [ 780/1300] | D_real_loss: 0.6938 | D_fake_loss: 0.6266 | G_loss:
0.8988
Iteration [ 790/1300] | D_real_loss: 0.6259 | D_fake_loss: 0.6529 | G_loss:
0.8962
Iteration [ 800/1300] | D_real_loss: 0.5875 | D_fake_loss: 0.6146 | G_loss:
0.9957
Saved output/./vanilla\grumpifyBprocessed_basic\sample-000800.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-000800.png
Iteration [ 810/1300] | D_real_loss: 0.4300 | D_fake_loss: 0.6151 | G_loss:
0.9979
Iteration [ 820/1300] | D_real_loss: 0.4734 | D_fake_loss: 0.6455 | G_loss:
0.8749
Iteration [ 830/1300] | D_real_loss: 0.6650 | D_fake_loss: 0.6143 | G_loss:
0.8300
Iteration [ 840/1300] | D_real_loss: 0.7046 | D_fake_loss: 0.5865 | G_loss:
0.9329
Iteration [ 850/1300] | D_real_loss: 0.6194 | D_fake_loss: 0.5952 | G_loss:
0.8704
Iteration [ 860/1300] | D_real_loss: 0.6270 | D_fake_loss: 0.5918 | G_loss:
0.9777
Iteration [ 870/1300] | D_real_loss: 0.5783 | D_fake_loss: 0.6581 | G_loss:
0.8010
Iteration [ 880/1300] | D_real_loss: 0.6900 | D_fake_loss: 0.6026 | G_loss:
0.9816
Iteration [ 890/1300] | D_real_loss: 0.6101 | D_fake_loss: 0.6285 | G_loss:
0.8695
Iteration [ 900/1300] | D_real_loss: 0.5372 | D_fake_loss: 0.6116 | G_loss:
0.8865
Iteration [ 910/1300] | D_real_loss: 0.6336 | D_fake_loss: 0.5223 | G_loss:
0.9759
Iteration [ 920/1300] | D_real_loss: 0.5360 | D_fake_loss: 0.5169 | G_loss:
1.0237
Iteration [ 930/1300] | D_real_loss: 0.5977 | D_fake_loss: 0.6429 | G_loss:
0.8635
Iteration [ 940/1300] | D_real_loss: 0.5531 | D_fake_loss: 0.6531 | G_loss:
0.8696
Iteration [ 950/1300] | D_real_loss: 0.5772 | D_fake_loss: 0.6589 | G_loss:
0.8816
Iteration [ 960/1300] | D_real_loss: 0.6603 | D_fake_loss: 0.5433 | G_loss:
0.9518
```

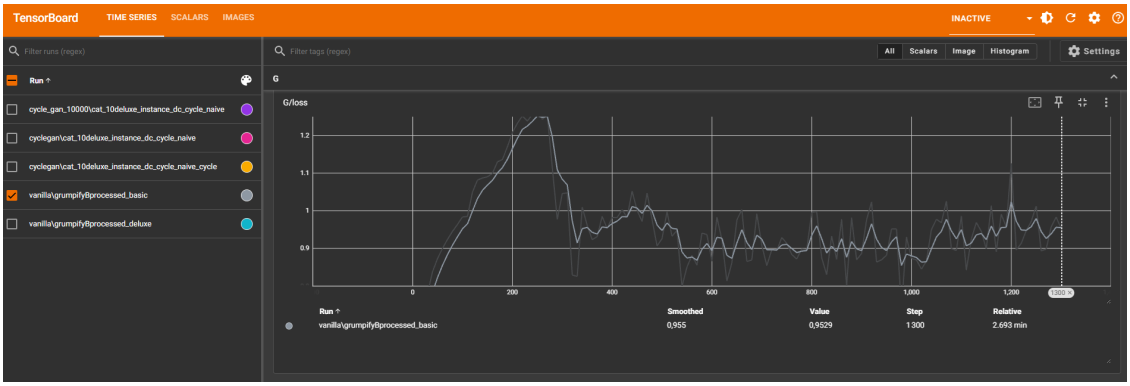
```
Iteration [ 970/1300] | D_real_loss: 0.6178 | D_fake_loss: 0.5517 | G_loss:
0.9516
Iteration [ 980/1300] | D_real_loss: 0.5275 | D_fake_loss: 0.6354 | G_loss:
0.7397
Iteration [ 990/1300] | D_real_loss: 0.6699 | D_fake_loss: 0.5581 | G_loss:
0.9303
Iteration [1000/1300] | D_real_loss: 0.5983 | D_fake_loss: 0.5765 | G_loss:
0.8731
Saved output/./vanilla\grumpifyBprocessed_basic\sample-001000.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-001000.png
Iteration [1010/1300] | D_real_loss: 0.5791 | D_fake_loss: 0.6258 | G_loss:
0.8694
Iteration [1020/1300] | D_real_loss: 0.5877 | D_fake_loss: 0.6063 | G_loss:
0.8447
Iteration [1030/1300] | D_real_loss: 0.5668 | D_fake_loss: 0.5728 | G_loss:
0.8666
Iteration [1040/1300] | D_real_loss: 0.5471 | D_fake_loss: 0.6406 | G_loss:
0.9500
Iteration [1050/1300] | D_real_loss: 0.6159 | D_fake_loss: 0.4992 | G_loss:
0.9774
Iteration [1060/1300] | D_real_loss: 0.5226 | D_fake_loss: 0.6278 | G_loss:
0.9683
Iteration [1070/1300] | D_real_loss: 0.7009 | D_fake_loss: 0.5286 | G_loss:
1.0256
Iteration [1080/1300] | D_real_loss: 0.5508 | D_fake_loss: 0.5814 | G_loss:
0.8996
Iteration [1090/1300] | D_real_loss: 0.4967 | D_fake_loss: 0.5925 | G_loss:
0.8943
Iteration [1100/1300] | D_real_loss: 0.5638 | D_fake_loss: 0.5429 | G_loss:
0.9862
Iteration [1110/1300] | D_real_loss: 0.6649 | D_fake_loss: 0.5597 | G_loss:
0.8423
Iteration [1120/1300] | D_real_loss: 0.4646 | D_fake_loss: 0.6497 | G_loss:
0.9251
Iteration [1130/1300] | D_real_loss: 0.5752 | D_fake_loss: 0.5599 | G_loss:
0.9688
Iteration [1140/1300] | D_real_loss: 0.5244 | D_fake_loss: 0.5596 | G_loss:
0.9453
Iteration [1150/1300] | D_real_loss: 0.6056 | D_fake_loss: 0.5322 | G_loss:
0.8983
Iteration [1160/1300] | D_real_loss: 0.5130 | D_fake_loss: 0.5667 | G_loss:
1.0137
Iteration [1170/1300] | D_real_loss: 0.5423 | D_fake_loss: 0.5721 | G_loss:
0.8907
Iteration [1180/1300] | D_real_loss: 0.5577 | D_fake_loss: 0.5341 | G_loss:
0.9912
Iteration [1190/1300] | D_real_loss: 0.7232 | D_fake_loss: 0.5946 | G_loss:
0.9565
```

```
Iteration [1200/1300] | D_real_loss: 0.4549 | D_fake_loss: 0.5636 | G_loss:
1.1260
Saved output/./vanilla\grumpifyBprocessed_basic\sample-001200.png
Saved output/./vanilla\grumpifyBprocessed_basic\real-001200.png
Iteration [1210/1300] | D_real_loss: 0.5279 | D_fake_loss: 0.5921 | G_loss:
0.8958
Iteration [1220/1300] | D_real_loss: 0.5354 | D_fake_loss: 0.5897 | G_loss:
0.9132
Iteration [1230/1300] | D_real_loss: 0.6079 | D_fake_loss: 0.6228 | G_loss:
0.9469
Iteration [1240/1300] | D_real_loss: 0.5456 | D_fake_loss: 0.5020 | G_loss:
0.9713
Iteration [1250/1300] | D_real_loss: 0.4316 | D_fake_loss: 0.6086 | G_loss:
1.0125
Iteration [1260/1300] | D_real_loss: 0.5406 | D_fake_loss: 0.5351 | G_loss:
0.8934
Iteration [1270/1300] | D_real_loss: 0.6847 | D_fake_loss: 0.4777 | G_loss:
0.8979
Iteration [1280/1300] | D_real_loss: 0.5695 | D_fake_loss: 0.5188 | G_loss:
0.9570
Iteration [1290/1300] | D_real_loss: 0.5977 | D_fake_loss: 0.5158 | G_loss:
0.9832
Iteration [1300/1300] | D_real_loss: 0.4803 | D_fake_loss: 0.6048 | G_loss:
0.9529
```

Basic Loss Curves On the Generator losses we can see an increase trend over the training steps. This indicates that the discriminator is getting better at separating false images from real ones, making it more difficult for the generator to "fool" it. This increase in loss may suggest that the generator requires additional effort to achieve good images, and that the discriminator is indeed getting better, outperforming the generator. However, it may also indicate that the generator is not performing as well as the training progresses. Looking at the generated images is important to understand which scenario is happening.

The discriminator losses show a decreasing trend in both false images and real images. This indicates that, as training progresses, the discriminator fails to differentiate between false and real images. Now, it is possible to expect that D/false_loss is reduced as the generator gets closer to producing better images, in the same way that a reduction in D/real_loss can be seen. Both reductions would be indications that the generator is managing to "fool" the discriminator, however, it could also indicate that there is a learning problem and that therefore the discriminator is losing the ability to effectively differentiate between the two groups of images.





Deluxe Execution

[17]:
```
!python vanilla_gan.py --data_aug=deluxe  --num_epochs=100
```

```
Namespace(image_size=64, conv_dim=32, noise_size=100, num_epochs=100,
batch_size=16, num_workers=0, lr=0.0003, beta1=0.5, beta2=0.999,
data='cat/grumpifyBprocessed', data_aug='deluxe', ext='*.png',
checkpoint_dir='./checkpoints_vanilla',
sample_dir='output/./vanilla\\grumpifyBprocessed_deluxe', log_step=10,
sample_every=200, checkpoint_every=400)
data/cat/grumpifyBprocessed\*.png
204
                    G
---------------------------------------
DCGenerator(
  (deconv1): Sequential(
    (0): ConvTranspose2d(100, 256, kernel_size=(4, 4), stride=(1, 1),
bias=False)
```

```
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv2): Sequential(
    (0): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2),
padding=(1, 1), bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv3): Sequential(
    (0): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv4): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (deconv5): Sequential(
    (0): ConvTranspose2d(32, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1))
    (1): Tanh()
  )
)
---------------------------------------
                    D
---------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv4): Sequential(
```

```
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
---------------------------------------
Iteration [  10/1300] | D_real_loss: 0.6178 | D_fake_loss: 0.7502 | G_loss:
0.6525
Iteration [  20/1300] | D_real_loss: 0.6142 | D_fake_loss: 0.6889 | G_loss:
0.7106
Iteration [  30/1300] | D_real_loss: 0.6136 | D_fake_loss: 0.6380 | G_loss:
0.7735
Iteration [  40/1300] | D_real_loss: 0.6197 | D_fake_loss: 0.5993 | G_loss:
0.8131
Iteration [  50/1300] | D_real_loss: 0.5597 | D_fake_loss: 0.5798 | G_loss:
0.8414
Iteration [  60/1300] | D_real_loss: 0.5511 | D_fake_loss: 0.5561 | G_loss:
0.8760
Iteration [  70/1300] | D_real_loss: 0.5666 | D_fake_loss: 0.5429 | G_loss:
0.8949
Iteration [  80/1300] | D_real_loss: 0.5230 | D_fake_loss: 0.5236 | G_loss:
0.9362
Iteration [  90/1300] | D_real_loss: 0.4994 | D_fake_loss: 0.5078 | G_loss:
0.9562
Iteration [ 100/1300] | D_real_loss: 0.4876 | D_fake_loss: 0.5042 | G_loss:
0.9720
Iteration [ 110/1300] | D_real_loss: 0.5292 | D_fake_loss: 0.5037 | G_loss:
0.9872
Iteration [ 120/1300] | D_real_loss: 0.4859 | D_fake_loss: 0.4822 | G_loss:
1.0049
Iteration [ 130/1300] | D_real_loss: 0.4613 | D_fake_loss: 0.4800 | G_loss:
1.0250
Iteration [ 140/1300] | D_real_loss: 0.4790 | D_fake_loss: 0.4701 | G_loss:
1.0304
Iteration [ 150/1300] | D_real_loss: 0.5439 | D_fake_loss: 0.4753 | G_loss:
1.0636
Iteration [ 160/1300] | D_real_loss: 0.5240 | D_fake_loss: 0.7516 | G_loss:
0.8212
Iteration [ 170/1300] | D_real_loss: 0.7106 | D_fake_loss: 0.6148 | G_loss:
0.9410
Iteration [ 180/1300] | D_real_loss: 0.6782 | D_fake_loss: 0.6707 | G_loss:
0.8691
Iteration [ 190/1300] | D_real_loss: 0.6560 | D_fake_loss: 0.6227 | G_loss:
```

```
0.9067
Iteration [ 200/1300] | D_real_loss: 0.6389 | D_fake_loss: 0.5957 | G_loss:
0.9406
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000200.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000200.png
Iteration [ 210/1300] | D_real_loss: 0.6120 | D_fake_loss: 0.6410 | G_loss:
0.7780
Iteration [ 220/1300] | D_real_loss: 0.6320 | D_fake_loss: 0.6819 | G_loss:
0.8670
Iteration [ 230/1300] | D_real_loss: 0.6197 | D_fake_loss: 0.6466 | G_loss:
0.8606
Iteration [ 240/1300] | D_real_loss: 0.6379 | D_fake_loss: 0.6505 | G_loss:
0.8493
Iteration [ 250/1300] | D_real_loss: 0.5951 | D_fake_loss: 0.6552 | G_loss:
0.8564
Iteration [ 260/1300] | D_real_loss: 0.6986 | D_fake_loss: 0.6543 | G_loss:
0.8097
Iteration [ 270/1300] | D_real_loss: 0.6528 | D_fake_loss: 0.6498 | G_loss:
0.8544
Iteration [ 280/1300] | D_real_loss: 0.6566 | D_fake_loss: 0.6575 | G_loss:
0.7644
Iteration [ 290/1300] | D_real_loss: 0.6537 | D_fake_loss: 0.6390 | G_loss:
0.8233
Iteration [ 300/1300] | D_real_loss: 0.6749 | D_fake_loss: 0.6065 | G_loss:
0.7766
Iteration [ 310/1300] | D_real_loss: 0.6958 | D_fake_loss: 0.6174 | G_loss:
0.8497
Iteration [ 320/1300] | D_real_loss: 0.6245 | D_fake_loss: 0.7284 | G_loss:
0.8159
Iteration [ 330/1300] | D_real_loss: 0.5676 | D_fake_loss: 0.6776 | G_loss:
0.8000
Iteration [ 340/1300] | D_real_loss: 0.6663 | D_fake_loss: 0.6813 | G_loss:
0.8646
Iteration [ 350/1300] | D_real_loss: 0.6642 | D_fake_loss: 0.6103 | G_loss:
0.8398
Iteration [ 360/1300] | D_real_loss: 0.6668 | D_fake_loss: 0.5932 | G_loss:
0.7523
Iteration [ 370/1300] | D_real_loss: 0.7290 | D_fake_loss: 0.6219 | G_loss:
0.7837
Iteration [ 380/1300] | D_real_loss: 0.6581 | D_fake_loss: 0.6134 | G_loss:
0.8027
Iteration [ 390/1300] | D_real_loss: 0.6969 | D_fake_loss: 0.6002 | G_loss:
0.7968
Iteration [ 400/1300] | D_real_loss: 0.6334 | D_fake_loss: 0.6706 | G_loss:
0.7645
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000400.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000400.png
Iteration [ 410/1300] | D_real_loss: 0.6471 | D_fake_loss: 0.6601 | G_loss:
```

```
0.7701
Iteration [ 420/1300] | D_real_loss: 0.6209 | D_fake_loss: 0.6324 | G_loss:
0.7965
Iteration [ 430/1300] | D_real_loss: 0.7349 | D_fake_loss: 0.6337 | G_loss:
0.9316
Iteration [ 440/1300] | D_real_loss: 0.6383 | D_fake_loss: 0.6932 | G_loss:
0.7909
Iteration [ 450/1300] | D_real_loss: 0.6106 | D_fake_loss: 0.7087 | G_loss:
0.8567
Iteration [ 460/1300] | D_real_loss: 0.6718 | D_fake_loss: 0.6049 | G_loss:
0.7786
Iteration [ 470/1300] | D_real_loss: 0.6738 | D_fake_loss: 0.6880 | G_loss:
0.8174
Iteration [ 480/1300] | D_real_loss: 0.7419 | D_fake_loss: 0.6626 | G_loss:
0.8104
Iteration [ 490/1300] | D_real_loss: 0.7214 | D_fake_loss: 0.6542 | G_loss:
0.7857
Iteration [ 500/1300] | D_real_loss: 0.7249 | D_fake_loss: 0.6264 | G_loss:
0.8200
Iteration [ 510/1300] | D_real_loss: 0.6940 | D_fake_loss: 0.6559 | G_loss:
0.7781
Iteration [ 520/1300] | D_real_loss: 0.7040 | D_fake_loss: 0.7257 | G_loss:
0.7409
Iteration [ 530/1300] | D_real_loss: 0.6720 | D_fake_loss: 0.6289 | G_loss:
0.7615
Iteration [ 540/1300] | D_real_loss: 0.5934 | D_fake_loss: 0.7089 | G_loss:
0.7076
Iteration [ 550/1300] | D_real_loss: 0.6582 | D_fake_loss: 0.7226 | G_loss:
0.7482
Iteration [ 560/1300] | D_real_loss: 0.6229 | D_fake_loss: 0.6692 | G_loss:
0.7901
Iteration [ 570/1300] | D_real_loss: 0.7151 | D_fake_loss: 0.6038 | G_loss:
0.8265
Iteration [ 580/1300] | D_real_loss: 0.6674 | D_fake_loss: 0.6851 | G_loss:
0.7460
Iteration [ 590/1300] | D_real_loss: 0.6905 | D_fake_loss: 0.6562 | G_loss:
0.7693
Iteration [ 600/1300] | D_real_loss: 0.6324 | D_fake_loss: 0.6939 | G_loss:
0.7982
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000600.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000600.png
Iteration [ 610/1300] | D_real_loss: 0.6283 | D_fake_loss: 0.6857 | G_loss:
0.7418
Iteration [ 620/1300] | D_real_loss: 0.7039 | D_fake_loss: 0.6603 | G_loss:
0.7497
Iteration [ 630/1300] | D_real_loss: 0.6515 | D_fake_loss: 0.6722 | G_loss:
0.7772
Iteration [ 640/1300] | D_real_loss: 0.7148 | D_fake_loss: 0.6557 | G_loss:
```

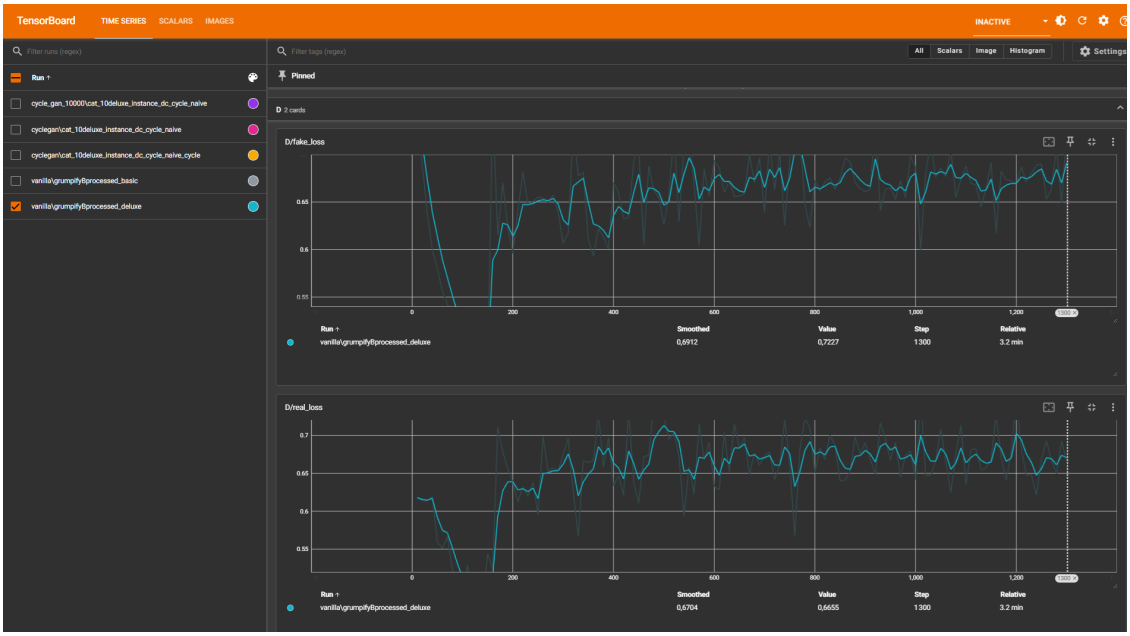
```
0.7932
Iteration [ 650/1300] | D_real_loss: 0.6826 | D_fake_loss: 0.6561 | G_loss:
0.7732
Iteration [ 660/1300] | D_real_loss: 0.6969 | D_fake_loss: 0.6583 | G_loss:
0.7778
Iteration [ 670/1300] | D_real_loss: 0.6485 | D_fake_loss: 0.6999 | G_loss:
0.7190
Iteration [ 680/1300] | D_real_loss: 0.6767 | D_fake_loss: 0.6705 | G_loss:
0.7509
Iteration [ 690/1300] | D_real_loss: 0.6604 | D_fake_loss: 0.6965 | G_loss:
0.7336
Iteration [ 700/1300] | D_real_loss: 0.6732 | D_fake_loss: 0.6392 | G_loss:
0.7836
Iteration [ 710/1300] | D_real_loss: 0.6775 | D_fake_loss: 0.7133 | G_loss:
0.7151
Iteration [ 720/1300] | D_real_loss: 0.6418 | D_fake_loss: 0.6631 | G_loss:
0.7708
Iteration [ 730/1300] | D_real_loss: 0.6601 | D_fake_loss: 0.7017 | G_loss:
0.7401
Iteration [ 740/1300] | D_real_loss: 0.7210 | D_fake_loss: 0.6251 | G_loss:
0.8006
Iteration [ 750/1300] | D_real_loss: 0.6619 | D_fake_loss: 0.6948 | G_loss:
0.7658
Iteration [ 760/1300] | D_real_loss: 0.5669 | D_fake_loss: 0.7481 | G_loss:
0.7076
Iteration [ 770/1300] | D_real_loss: 0.6805 | D_fake_loss: 0.7217 | G_loss:
0.7409
Iteration [ 780/1300] | D_real_loss: 0.7222 | D_fake_loss: 0.6480 | G_loss:
0.7985
Iteration [ 790/1300] | D_real_loss: 0.7094 | D_fake_loss: 0.6230 | G_loss:
0.8059
Iteration [ 800/1300] | D_real_loss: 0.6500 | D_fake_loss: 0.6729 | G_loss:
0.7564
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-000800.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-000800.png
Iteration [ 810/1300] | D_real_loss: 0.6832 | D_fake_loss: 0.6614 | G_loss:
0.7462
Iteration [ 820/1300] | D_real_loss: 0.6681 | D_fake_loss: 0.6714 | G_loss:
0.7489
Iteration [ 830/1300] | D_real_loss: 0.7004 | D_fake_loss: 0.6755 | G_loss:
0.7295
Iteration [ 840/1300] | D_real_loss: 0.6868 | D_fake_loss: 0.6635 | G_loss:
0.8329
Iteration [ 850/1300] | D_real_loss: 0.6399 | D_fake_loss: 0.6747 | G_loss:
0.7613
Iteration [ 860/1300] | D_real_loss: 0.6409 | D_fake_loss: 0.6963 | G_loss:
0.7778
Iteration [ 870/1300] | D_real_loss: 0.6527 | D_fake_loss: 0.6924 | G_loss:
```

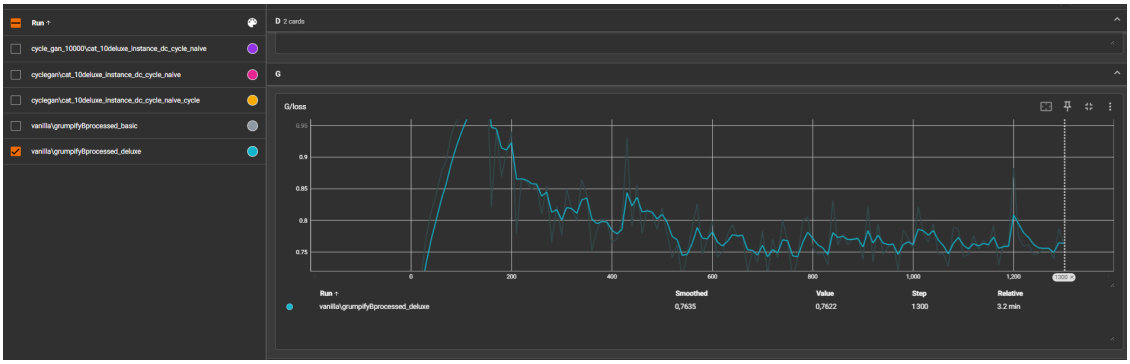
```
0.7613
Iteration [ 880/1300] | D_real_loss: 0.6977 | D_fake_loss: 0.6707 | G_loss:
0.7706
Iteration [ 890/1300] | D_real_loss: 0.6751 | D_fake_loss: 0.6641 | G_loss:
0.7731
Iteration [ 900/1300] | D_real_loss: 0.6902 | D_fake_loss: 0.6611 | G_loss:
0.7379
Iteration [ 910/1300] | D_real_loss: 0.6683 | D_fake_loss: 0.6636 | G_loss:
0.8229
Iteration [ 920/1300] | D_real_loss: 0.6484 | D_fake_loss: 0.7391 | G_loss:
0.7344
Iteration [ 930/1300] | D_real_loss: 0.7164 | D_fake_loss: 0.6408 | G_loss:
0.7948
Iteration [ 940/1300] | D_real_loss: 0.6957 | D_fake_loss: 0.6637 | G_loss:
0.7455
Iteration [ 950/1300] | D_real_loss: 0.6673 | D_fake_loss: 0.6648 | G_loss:
0.7576
Iteration [ 960/1300] | D_real_loss: 0.6907 | D_fake_loss: 0.6528 | G_loss:
0.7636
Iteration [ 970/1300] | D_real_loss: 0.6444 | D_fake_loss: 0.6727 | G_loss:
0.7208
Iteration [ 980/1300] | D_real_loss: 0.6735 | D_fake_loss: 0.6547 | G_loss:
0.7849
Iteration [ 990/1300] | D_real_loss: 0.6800 | D_fake_loss: 0.6982 | G_loss:
0.7726
Iteration [1000/1300] | D_real_loss: 0.6413 | D_fake_loss: 0.6870 | G_loss:
0.7552
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-001000.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-001000.png
Iteration [1010/1300] | D_real_loss: 0.7567 | D_fake_loss: 0.5987 | G_loss:
0.8225
Iteration [1020/1300] | D_real_loss: 0.6467 | D_fake_loss: 0.6800 | G_loss:
0.7801
Iteration [1030/1300] | D_real_loss: 0.6472 | D_fake_loss: 0.7130 | G_loss:
0.7658
Iteration [1040/1300] | D_real_loss: 0.6661 | D_fake_loss: 0.6763 | G_loss:
0.7950
Iteration [1050/1300] | D_real_loss: 0.7084 | D_fake_loss: 0.6860 | G_loss:
0.7460
Iteration [1060/1300] | D_real_loss: 0.6644 | D_fake_loss: 0.6767 | G_loss:
0.7474
Iteration [1070/1300] | D_real_loss: 0.6245 | D_fake_loss: 0.7047 | G_loss:
0.7266
Iteration [1080/1300] | D_real_loss: 0.6756 | D_fake_loss: 0.6548 | G_loss:
0.7885
Iteration [1090/1300] | D_real_loss: 0.7132 | D_fake_loss: 0.6741 | G_loss:
0.7870
Iteration [1100/1300] | D_real_loss: 0.6345 | D_fake_loss: 0.6880 | G_loss:
```

```
0.7423
Iteration [1110/1300] | D_real_loss: 0.6813 | D_fake_loss: 0.6694 | G_loss:
0.7458
Iteration [1120/1300] | D_real_loss: 0.6820 | D_fake_loss: 0.6678 | G_loss:
0.7760
Iteration [1130/1300] | D_real_loss: 0.6544 | D_fake_loss: 0.6448 | G_loss:
0.7543
Iteration [1140/1300] | D_real_loss: 0.6575 | D_fake_loss: 0.6643 | G_loss:
0.7685
Iteration [1150/1300] | D_real_loss: 0.6669 | D_fake_loss: 0.6918 | G_loss:
0.7588
Iteration [1160/1300] | D_real_loss: 0.7278 | D_fake_loss: 0.6173 | G_loss:
0.7919
Iteration [1170/1300] | D_real_loss: 0.6707 | D_fake_loss: 0.6825 | G_loss:
0.7289
Iteration [1180/1300] | D_real_loss: 0.6403 | D_fake_loss: 0.6737 | G_loss:
0.7620
Iteration [1190/1300] | D_real_loss: 0.6769 | D_fake_loss: 0.6723 | G_loss:
0.7601
Iteration [1200/1300] | D_real_loss: 0.7513 | D_fake_loss: 0.6694 | G_loss:
0.8829
Saved output/./vanilla\grumpifyBprocessed_deluxe\sample-001200.png
Saved output/./vanilla\grumpifyBprocessed_deluxe\real-001200.png
Iteration [1210/1300] | D_real_loss: 0.6822 | D_fake_loss: 0.6871 | G_loss:
0.7733
Iteration [1220/1300] | D_real_loss: 0.6469 | D_fake_loss: 0.6713 | G_loss:
0.7590
Iteration [1230/1300] | D_real_loss: 0.6481 | D_fake_loss: 0.6802 | G_loss:
0.7612
Iteration [1240/1300] | D_real_loss: 0.6213 | D_fake_loss: 0.6889 | G_loss:
0.7453
Iteration [1250/1300] | D_real_loss: 0.6690 | D_fake_loss: 0.6901 | G_loss:
0.7484
Iteration [1260/1300] | D_real_loss: 0.6924 | D_fake_loss: 0.6529 | G_loss:
0.7544
Iteration [1270/1300] | D_real_loss: 0.6670 | D_fake_loss: 0.6639 | G_loss:
0.7559
Iteration [1280/1300] | D_real_loss: 0.6490 | D_fake_loss: 0.7078 | G_loss:
0.7389
Iteration [1290/1300] | D_real_loss: 0.6926 | D_fake_loss: 0.6488 | G_loss:
0.7875
Iteration [1300/1300] | D_real_loss: 0.6655 | D_fake_loss: 0.7227 | G_loss:
0.7622
```

Deluxe Loss Curves The curve D/false_loss starts high and decreases as the training progresses, showing some fluctuations during the process. Again, at the beginning, the discriminator manages to identify the generated images as false, but its accuracy decreases as the training advances,

suggesting that the generator improves its performance. Now, the fluctuations that we see could suggest moments when the discriminator adapts to the strategies of the generator to produce more realistic false images. The same way D/real_loss shows a dropping tendency from a high starting point, this suggests that the performance of the discriminator in trying to classify images decreases over the training process. The presence of the augmentations is an important component for this to happen, as the discriminator learns to identify real images even under various transformations, but the decreasing loss indicates that the generator also improves. The fluctuations could represent the Discriminator continued attempts to learn features and correctly identify images.





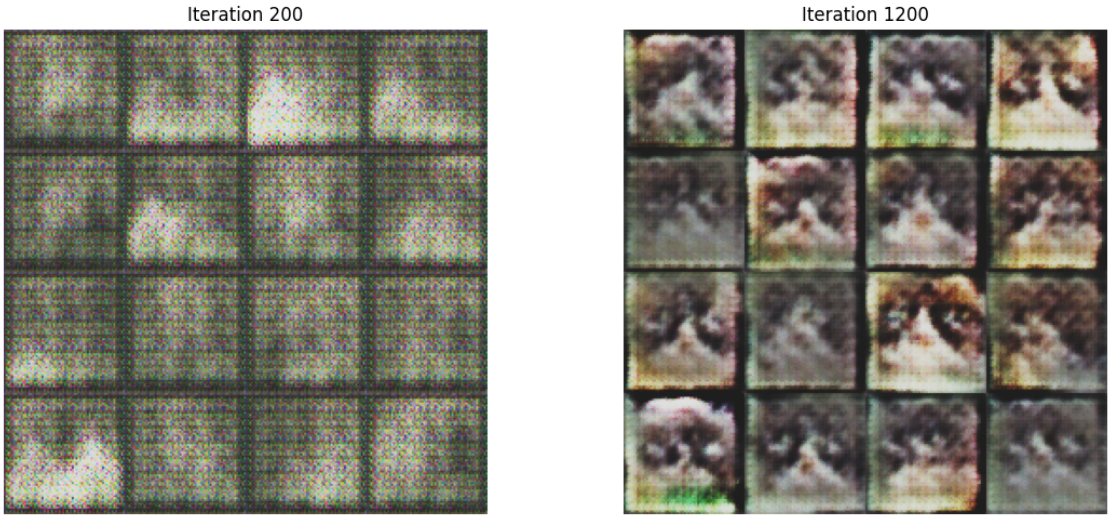
[13]:
```
# Load images from specific iterations
img_early = mpimg.imread("output/vanilla/grumpifyBprocessed_deluxe/
  ↪sample-000200.png")
img_late = mpimg.imread("output/vanilla/grumpifyBprocessed_deluxe/sample-001200.
  ↪png")

# Plot the images for comparison
plt.figure(figsize=(12, 5))
```

```
# Plot early stage
plt.subplot(1, 2, 1)
plt.imshow(img_early)
plt.title("Iteration 200")
plt.axis('off')

# Plot late stage
plt.subplot(1, 2, 2)
plt.imshow(img_late)
plt.title("Iteration 1200")
plt.axis('off')

plt.tight_layout()
plt.show()
```



In this comparison of images that Vanilla GAN generated, we can see that at the beginning, around step 200, the images are just messy noise and do not look like anything specific, not even a cat. This is normal because the network that creates the images is just starting to learn and being as basic as Vanilla_Gan is, it may not be as fast in generating good results from the beginning. However, if we look at step 1200, the images improve a bit. However, they are still blurry and although you can see some shapes and colors that maybe look a bit like cats they are not of the best quality. It is as if the network is slowly realizing what a grumpy cat looks like, but it is still not very clear or real, it is clear at this stage that the network requires many more steps to learn how to create cat images.

2 PART 2: CycleGAN

2.1 Generator [20 points]

```python
class CycleGenerator(nn.Module):
    """Defines the architecture of the generator network.
       Note: Both generators G_XtoY and G_YtoX have the same architecture in this assignment.
    """
    def __init__(self, conv_dim=64, init_zero_weights=False, norm='batch'):
        super(CycleGenerator, self).__init__()

        ##########################################
        ##   FILL THIS IN: CREATE ARCHITECTURE   ##
        ##########################################

        # 1. Define the encoder part of the generator (that extracts features from the input image)
        self.pad = nn.ReflectionPad2d(3)
        self.conv1 = nn.Sequential(self.pad, conv(3, conv_dim, kernel_size=7, stride=1, padding=0, norm=norm))
        self.conv2 = conv(conv_dim, conv_dim * 2, kernel_size=3, stride=2, padding=1, norm=norm)

        # 2. Define the transformation part of the generator
        self.resnet_block = nn.Sequential(
            *[ResnetBlock(conv_dim * 2, norm) for _ in range(6)]
        )

        # 3. Define the decoder part of the generator (that builds up the output image from features)
        self.deconv1 = deconv(conv_dim * 2, conv_dim, kernel_size=4, stride=2, padding=1, norm=norm)
        self.deconv2 = nn.Sequential(
            nn.ReflectionPad2d(3),
            nn.Conv2d(conv_dim, 3, kernel_size=7, stride=1, padding=0, bias=False),
            nn.Tanh()
        )

    def forward(self, x):
        """Generates an image conditioned on an input image.

            Input
            -----
                x: BS x 3 x 32 x 32

            Output
            ------
                out: BS x 3 x 32 x 32
        """

        out = F.relu(self.conv1(x))
        out = F.relu(self.conv2(out))

        out = F.relu(self.resnet_block(out))

        out = F.relu(self.deconv1(out))
        out = F.tanh(self.deconv2(out))

        return out
```

2.2 CycleGAN Training Loop [20 points]

2.2.1 Discriminator

```
# =========================================
#            TRAIN THE DISCRIMINATORS
# =========================================

########################################
##             FILL THIS IN           ##
########################################

# Train with real images
d_optimizer.zero_grad()

# 1. Compute the discriminator losses on real images
D_X_loss = F.binary_cross_entropy(D_X(images_X), torch.ones_like(D_X(images_X)))  # Real image loss for D_X
D_Y_loss = F.binary_cross_entropy(D_Y(images_Y), torch.ones_like(D_Y(images_Y)))  # Real image loss for D_Y


d_real_loss = D_X_loss + D_Y_loss
d_real_loss.backward()
d_optimizer.step()
logger.add_scalar('D/XY/real', D_X_loss, iteration)
logger.add_scalar('D/YX/real', D_Y_loss, iteration)
# Train with fake images
d_optimizer.zero_grad()

# 2. Generate fake images that look like domain X based on real images in domain Y
fake_X = G_YtoX(images_Y)

# 3. Compute the loss for D_X
D_X_fake = D_X(fake_X)
D_X_loss = F.binary_cross_entropy(D_X_fake, torch.zeros_like(D_X_fake))

# 4. Generate fake images that look like domain Y based on real images in domain X
fake_Y = G_XtoY(images_X)

# 5. Compute the loss for D_Y
D_Y_loss = F.binary_cross_entropy(D_Y(fake_Y), torch.zeros_like(D_Y(fake_Y)))  # Fake image loss for D_Y


d_fake_loss = D_X_loss + D_Y_loss
if iteration % 2 == 0:
    d_fake_loss.backward()
    d_optimizer.step()
logger.add_scalar('D/XY/fake', D_X_loss, iteration)
logger.add_scalar('D/YX/fake', D_Y_loss, iteration)
```

2.2.2 Generator Y–X–>Y CYCLE

We are going to use L1 loss, as suggested in the original paper.

```
# =========================================
#            TRAIN THE GENERATORS
# =========================================


########################################
##    FILL THIS IN: Y--X-->Y CYCLE     ##
########################################
g_optimizer.zero_grad()

# 1. Generate fake images that look like domain X based on real images in domain Y
fake_X = G_YtoX(images_Y)

# 2. Compute the generator loss based on domain X
g_loss = F.binary_cross_entropy(D_X(fake_X), torch.ones_like(D_X(fake_X)))
logger.add_scalar('G/XY/fake', g_loss, iteration)

if opts.use_cycle_consistency_loss:
    reconstructed_Y = G_XtoY(fake_X)
    # 3. Compute the cycle consistency loss (the reconstruction loss)
    cycle_consistency_loss = torch.mean(torch.abs(images_Y - reconstructed_Y))
    g_loss += opts.lambda_cycle * cycle_consistency_loss
    logger.add_scalar('G/XY/cycle', opts.lambda_cycle * cycle_consistency_loss, iteration)

g_loss.backward()
g_optimizer.step()
```

2.2.3 Generator X–Y–>X CYCLE

```
########################################
##    FILL THIS IN: X--Y-->X CYCLE    ##
########################################

g_optimizer.zero_grad()

# 1. Generate fake images that look like domain Y based on real images in domain X
fake_Y = G_XtoY(images_X)

# 2. Compute the generator loss based on domain Y
g_loss = F.binary_cross_entropy(D_Y(fake_Y), torch.ones_like(D_Y(fake_Y)))
logger.add_scalar('G/YX/fake', g_loss, iteration)

if opts.use_cycle_consistency_loss:
    reconstructed_X = G_YtoX(fake_Y)
    # 3. Compute the cycle consistency loss (the reconstruction loss)
    cycle_consistency_loss = torch.mean(torch.abs(images_X - reconstructed_X))
    g_loss += opts.lambda_cycle * cycle_consistency_loss
    logger.add_scalar('G/YX/cycle', cycle_consistency_loss, iteration)

g_loss.backward()
g_optimizer.step()

# Print the log info
if iteration % opts.log_step == 0:
    print('Iteration [{:5d}/{:5d}] | d_real_loss: {:6.4f} | d_Y_loss: {:6.4f} | d_X_loss: {:6.4f} | '
          'd_fake_loss: {:6.4f} | g_loss: {:6.4f}'.format(
            iteration, opts.train_iters, d_real_loss.item(), D_Y_loss.item(),
            D_X_loss.item(), d_fake_loss.item(), g_loss.item()))

# Save the generated samples
if iteration % opts.sample_every == 0:
    save_samples(iteration, fixed_Y, fixed_X, G_YtoX, G_XtoY, opts)

if iteration in [400, 600]:
    save_samples(iteration, fixed_Y, fixed_X, G_YtoX, G_XtoY, opts)

# Save the model parameters
if iteration % opts.checkpoint_every == 0:
    checkpoint(iteration, G_XtoY, G_YtoX, D_X, D_Y, opts)
```

2.3 CycleGAN Experiments [15 points]

Basic Execution

[19]:
```
!python cycle_gan.py
```

```
================================================================================
                                 Opts
--------------------------------------------------------------------------------
                image_size: 64
                      disc: dc
                       gen: cycle
               g_conv_dim: 32
               d_conv_dim: 32
                      norm: instance
                 init_type: naive
              train_iters: 1000
               batch_size: 16
                        lr: 0.0003
                     beta1: 0.5
                     beta2: 0.999
```

```
                          lambda_cycle: 10
                                     X: cat/grumpifyAprocessed
                                     Y: cat/grumpifyBprocessed
                                   ext: *.png
                              data_aug: deluxe
                         checkpoint_dir: checkpoints_cyclegan
                    sample_dir:
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive
                                log_step: 10
                            sample_every: 100
                       checkpoint_every: 800
                                   gpu: 0
================================================================================
data/cat/grumpifyAprocessed\*.png
75
data/cat/grumpifyBprocessed\*.png
204
                 G_XtoY
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
```

```
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
```

```
)
---------------------------------------
                G_YtoX
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
```

```
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
----------------------------------------
                D_X
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
```

```
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
----------------------------------------
                D_Y
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
```

```
---------------------------------------
Iteration [   10/ 1000] | d_real_loss: 1.0821 | d_Y_loss: 0.7438 | d_X_loss:
0.9189 | d_fake_loss: 1.6627 | g_loss: 0.6854
Iteration [   20/ 1000] | d_real_loss: 0.9386 | d_Y_loss: 0.7841 | d_X_loss:
0.9434 | d_fake_loss: 1.7275 | g_loss: 0.6445
Iteration [   30/ 1000] | d_real_loss: 0.8331 | d_Y_loss: 0.8127 | d_X_loss:
0.9950 | d_fake_loss: 1.8077 | g_loss: 0.6254
Iteration [   40/ 1000] | d_real_loss: 0.8104 | d_Y_loss: 0.9460 | d_X_loss:
1.0215 | d_fake_loss: 1.9675 | g_loss: 0.5360
Iteration [   50/ 1000] | d_real_loss: 0.8040 | d_Y_loss: 0.8903 | d_X_loss:
1.0238 | d_fake_loss: 1.9141 | g_loss: 0.5629
Iteration [   60/ 1000] | d_real_loss: 0.7763 | d_Y_loss: 0.9402 | d_X_loss:
1.0022 | d_fake_loss: 1.9424 | g_loss: 0.5325
Iteration [   70/ 1000] | d_real_loss: 0.7765 | d_Y_loss: 0.9922 | d_X_loss:
1.0567 | d_fake_loss: 2.0488 | g_loss: 0.5135
Iteration [   80/ 1000] | d_real_loss: 0.7889 | d_Y_loss: 1.0651 | d_X_loss:
1.0511 | d_fake_loss: 2.1161 | g_loss: 0.4713
Iteration [   90/ 1000] | d_real_loss: 0.7915 | d_Y_loss: 1.0598 | d_X_loss:
1.0201 | d_fake_loss: 2.0799 | g_loss: 0.4700
Iteration [  100/ 1000] | d_real_loss: 0.8009 | d_Y_loss: 0.9943 | d_X_loss:
1.0292 | d_fake_loss: 2.0235 | g_loss: 0.5069
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000100-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000100-Y-X.png
Iteration [  110/ 1000] | d_real_loss: 0.7990 | d_Y_loss: 1.0290 | d_X_loss:
0.9832 | d_fake_loss: 2.0122 | g_loss: 0.4802
Iteration [  120/ 1000] | d_real_loss: 0.7755 | d_Y_loss: 1.0504 | d_X_loss:
1.0989 | d_fake_loss: 2.1494 | g_loss: 0.4838
Iteration [  130/ 1000] | d_real_loss: 0.8146 | d_Y_loss: 1.0599 | d_X_loss:
1.0317 | d_fake_loss: 2.0915 | g_loss: 0.4767
Iteration [  140/ 1000] | d_real_loss: 0.8109 | d_Y_loss: 1.0400 | d_X_loss:
1.0516 | d_fake_loss: 2.0916 | g_loss: 0.4816
Iteration [  150/ 1000] | d_real_loss: 0.7909 | d_Y_loss: 1.0168 | d_X_loss:
1.0973 | d_fake_loss: 2.1141 | g_loss: 0.4880
Iteration [  160/ 1000] | d_real_loss: 0.7780 | d_Y_loss: 1.0556 | d_X_loss:
1.0352 | d_fake_loss: 2.0909 | g_loss: 0.4712
Iteration [  170/ 1000] | d_real_loss: 0.8038 | d_Y_loss: 1.0513 | d_X_loss:
1.0263 | d_fake_loss: 2.0777 | g_loss: 0.4941
Iteration [  180/ 1000] | d_real_loss: 0.8099 | d_Y_loss: 1.0655 | d_X_loss:
0.9909 | d_fake_loss: 2.0564 | g_loss: 0.4737
Iteration [  190/ 1000] | d_real_loss: 0.7707 | d_Y_loss: 1.0773 | d_X_loss:
1.0015 | d_fake_loss: 2.0788 | g_loss: 0.4611
Iteration [  200/ 1000] | d_real_loss: 0.8212 | d_Y_loss: 1.0737 | d_X_loss:
0.9869 | d_fake_loss: 2.0606 | g_loss: 0.4633
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000200-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000200-Y-X.png
Iteration [  210/ 1000] | d_real_loss: 0.7983 | d_Y_loss: 1.0704 | d_X_loss:
1.0192 | d_fake_loss: 2.0896 | g_loss: 0.4609
Iteration [  220/ 1000] | d_real_loss: 0.8080 | d_Y_loss: 1.0679 | d_X_loss:
```

```
1.0523 | d_fake_loss: 2.1202 | g_loss: 0.4635
Iteration [  230/ 1000] | d_real_loss: 0.8150 | d_Y_loss: 1.0348 | d_X_loss:
1.0929 | d_fake_loss: 2.1277 | g_loss: 0.4759
Iteration [  240/ 1000] | d_real_loss: 0.7792 | d_Y_loss: 1.0606 | d_X_loss:
1.0405 | d_fake_loss: 2.1011 | g_loss: 0.4679
Iteration [  250/ 1000] | d_real_loss: 0.7982 | d_Y_loss: 1.0947 | d_X_loss:
1.0444 | d_fake_loss: 2.1390 | g_loss: 0.4500
Iteration [  260/ 1000] | d_real_loss: 0.7895 | d_Y_loss: 1.0609 | d_X_loss:
1.0195 | d_fake_loss: 2.0805 | g_loss: 0.4665
Iteration [  270/ 1000] | d_real_loss: 0.7827 | d_Y_loss: 1.0563 | d_X_loss:
1.0833 | d_fake_loss: 2.1396 | g_loss: 0.4650
Iteration [  280/ 1000] | d_real_loss: 0.7828 | d_Y_loss: 1.0460 | d_X_loss:
1.0667 | d_fake_loss: 2.1128 | g_loss: 0.4643
Iteration [  290/ 1000] | d_real_loss: 0.7902 | d_Y_loss: 1.0529 | d_X_loss:
0.9837 | d_fake_loss: 2.0366 | g_loss: 0.4640
Iteration [  300/ 1000] | d_real_loss: 0.7914 | d_Y_loss: 1.0709 | d_X_loss:
0.9403 | d_fake_loss: 2.0111 | g_loss: 0.4553
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000300-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000300-Y-X.png
Iteration [  310/ 1000] | d_real_loss: 0.7558 | d_Y_loss: 1.0525 | d_X_loss:
0.9453 | d_fake_loss: 1.9977 | g_loss: 0.4646
Iteration [  320/ 1000] | d_real_loss: 0.7490 | d_Y_loss: 1.0662 | d_X_loss:
1.0233 | d_fake_loss: 2.0896 | g_loss: 0.4546
Iteration [  330/ 1000] | d_real_loss: 0.7503 | d_Y_loss: 1.1154 | d_X_loss:
1.0483 | d_fake_loss: 2.1637 | g_loss: 0.4416
Iteration [  340/ 1000] | d_real_loss: 0.7616 | d_Y_loss: 1.0806 | d_X_loss:
0.9275 | d_fake_loss: 2.0081 | g_loss: 0.4488
Iteration [  350/ 1000] | d_real_loss: 0.8151 | d_Y_loss: 1.0814 | d_X_loss:
0.8961 | d_fake_loss: 1.9775 | g_loss: 0.4469
Iteration [  360/ 1000] | d_real_loss: 0.7570 | d_Y_loss: 1.0660 | d_X_loss:
1.0082 | d_fake_loss: 2.0742 | g_loss: 0.4602
Iteration [  370/ 1000] | d_real_loss: 0.7865 | d_Y_loss: 1.0631 | d_X_loss:
0.9953 | d_fake_loss: 2.0584 | g_loss: 0.4553
Iteration [  380/ 1000] | d_real_loss: 0.7620 | d_Y_loss: 1.0828 | d_X_loss:
0.9949 | d_fake_loss: 2.0777 | g_loss: 0.4498
Iteration [  390/ 1000] | d_real_loss: 0.7737 | d_Y_loss: 1.0426 | d_X_loss:
1.1330 | d_fake_loss: 2.1756 | g_loss: 0.4695
Iteration [  400/ 1000] | d_real_loss: 0.7508 | d_Y_loss: 1.0733 | d_X_loss:
0.9440 | d_fake_loss: 2.0173 | g_loss: 0.4557
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000400-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000400-Y-X.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000400-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000400-Y-X.png
Iteration [  410/ 1000] | d_real_loss: 0.7443 | d_Y_loss: 1.0712 | d_X_loss:
0.9016 | d_fake_loss: 1.9728 | g_loss: 0.4567
Iteration [  420/ 1000] | d_real_loss: 0.7054 | d_Y_loss: 1.0681 | d_X_loss:
0.9659 | d_fake_loss: 2.0340 | g_loss: 0.4565
Iteration [  430/ 1000] | d_real_loss: 0.7405 | d_Y_loss: 1.0769 | d_X_loss:
```

```
0.8880 | d_fake_loss: 1.9649 | g_loss: 0.4556
Iteration [  440/ 1000] | d_real_loss: 0.7055 | d_Y_loss: 1.0711 | d_X_loss:
0.9890 | d_fake_loss: 2.0600 | g_loss: 0.4545
Iteration [  450/ 1000] | d_real_loss: 0.7461 | d_Y_loss: 1.0677 | d_X_loss:
0.8890 | d_fake_loss: 1.9567 | g_loss: 0.4622
Iteration [  460/ 1000] | d_real_loss: 0.7693 | d_Y_loss: 1.0941 | d_X_loss:
0.9135 | d_fake_loss: 2.0075 | g_loss: 0.4585
Iteration [  470/ 1000] | d_real_loss: 0.7714 | d_Y_loss: 1.0809 | d_X_loss:
0.9061 | d_fake_loss: 1.9870 | g_loss: 0.4492
Iteration [  480/ 1000] | d_real_loss: 0.6985 | d_Y_loss: 1.1042 | d_X_loss:
0.8068 | d_fake_loss: 1.9110 | g_loss: 0.4358
Iteration [  490/ 1000] | d_real_loss: 0.7162 | d_Y_loss: 1.0759 | d_X_loss:
0.8082 | d_fake_loss: 1.8841 | g_loss: 0.4447
Iteration [  500/ 1000] | d_real_loss: 0.6891 | d_Y_loss: 1.0910 | d_X_loss:
0.7579 | d_fake_loss: 1.8489 | g_loss: 0.4434
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000500-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000500-Y-X.png
Iteration [  510/ 1000] | d_real_loss: 0.6896 | d_Y_loss: 1.0650 | d_X_loss:
0.7470 | d_fake_loss: 1.8121 | g_loss: 0.4599
Iteration [  520/ 1000] | d_real_loss: 0.6857 | d_Y_loss: 1.1201 | d_X_loss:
0.7654 | d_fake_loss: 1.8856 | g_loss: 0.4393
Iteration [  530/ 1000] | d_real_loss: 0.7149 | d_Y_loss: 1.0686 | d_X_loss:
0.7786 | d_fake_loss: 1.8472 | g_loss: 0.4526
Iteration [  540/ 1000] | d_real_loss: 0.7317 | d_Y_loss: 1.0764 | d_X_loss:
0.7402 | d_fake_loss: 1.8166 | g_loss: 0.4639
Iteration [  550/ 1000] | d_real_loss: 0.7341 | d_Y_loss: 1.0661 | d_X_loss:
0.7369 | d_fake_loss: 1.8029 | g_loss: 0.4560
Iteration [  560/ 1000] | d_real_loss: 0.6964 | d_Y_loss: 1.0647 | d_X_loss:
0.7260 | d_fake_loss: 1.7907 | g_loss: 0.4481
Iteration [  570/ 1000] | d_real_loss: 0.6719 | d_Y_loss: 1.0754 | d_X_loss:
0.7164 | d_fake_loss: 1.7918 | g_loss: 0.4467
Iteration [  580/ 1000] | d_real_loss: 0.6954 | d_Y_loss: 1.0456 | d_X_loss:
0.6918 | d_fake_loss: 1.7373 | g_loss: 0.4616
Iteration [  590/ 1000] | d_real_loss: 0.6623 | d_Y_loss: 1.0687 | d_X_loss:
0.6423 | d_fake_loss: 1.7109 | g_loss: 0.4545
Iteration [  600/ 1000] | d_real_loss: 0.6471 | d_Y_loss: 1.0692 | d_X_loss:
0.6149 | d_fake_loss: 1.6841 | g_loss: 0.4504
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000600-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000600-Y-X.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000600-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000600-Y-X.png
Iteration [  610/ 1000] | d_real_loss: 0.6510 | d_Y_loss: 1.0839 | d_X_loss:
0.6580 | d_fake_loss: 1.7418 | g_loss: 0.4488
Iteration [  620/ 1000] | d_real_loss: 0.7036 | d_Y_loss: 1.0704 | d_X_loss:
0.6858 | d_fake_loss: 1.7561 | g_loss: 0.4548
Iteration [  630/ 1000] | d_real_loss: 0.6707 | d_Y_loss: 1.0757 | d_X_loss:
0.6634 | d_fake_loss: 1.7392 | g_loss: 0.4503
Iteration [  640/ 1000] | d_real_loss: 0.6725 | d_Y_loss: 1.0434 | d_X_loss:
```

```
0.6721 | d_fake_loss: 1.7155 | g_loss: 0.4663
Iteration [  650/ 1000] | d_real_loss: 0.6451 | d_Y_loss: 1.0656 | d_X_loss:
0.5757 | d_fake_loss: 1.6413 | g_loss: 0.4589
Iteration [  660/ 1000] | d_real_loss: 0.6205 | d_Y_loss: 1.1162 | d_X_loss:
0.5813 | d_fake_loss: 1.6975 | g_loss: 0.4386
Iteration [  670/ 1000] | d_real_loss: 0.6720 | d_Y_loss: 1.0396 | d_X_loss:
0.9941 | d_fake_loss: 2.0337 | g_loss: 0.4723
Iteration [  680/ 1000] | d_real_loss: 0.7087 | d_Y_loss: 1.0811 | d_X_loss:
0.6208 | d_fake_loss: 1.7019 | g_loss: 0.4504
Iteration [  690/ 1000] | d_real_loss: 0.6694 | d_Y_loss: 1.0723 | d_X_loss:
0.7064 | d_fake_loss: 1.7787 | g_loss: 0.4556
Iteration [  700/ 1000] | d_real_loss: 0.6531 | d_Y_loss: 1.0555 | d_X_loss:
0.8078 | d_fake_loss: 1.8633 | g_loss: 0.4625
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000700-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000700-Y-X.png
Iteration [  710/ 1000] | d_real_loss: 0.6673 | d_Y_loss: 1.1201 | d_X_loss:
0.5670 | d_fake_loss: 1.6871 | g_loss: 0.4363
Iteration [  720/ 1000] | d_real_loss: 0.7628 | d_Y_loss: 1.0819 | d_X_loss:
0.6062 | d_fake_loss: 1.6881 | g_loss: 0.4510
Iteration [  730/ 1000] | d_real_loss: 0.6403 | d_Y_loss: 1.0712 | d_X_loss:
0.6345 | d_fake_loss: 1.7057 | g_loss: 0.4517
Iteration [  740/ 1000] | d_real_loss: 0.6816 | d_Y_loss: 1.0631 | d_X_loss:
0.5423 | d_fake_loss: 1.6054 | g_loss: 0.4562
Iteration [  750/ 1000] | d_real_loss: 0.6483 | d_Y_loss: 1.0559 | d_X_loss:
0.7390 | d_fake_loss: 1.7948 | g_loss: 0.4650
Iteration [  760/ 1000] | d_real_loss: 0.7136 | d_Y_loss: 1.0532 | d_X_loss:
0.5988 | d_fake_loss: 1.6519 | g_loss: 0.4667
Iteration [  770/ 1000] | d_real_loss: 0.6276 | d_Y_loss: 1.0576 | d_X_loss:
0.5082 | d_fake_loss: 1.5658 | g_loss: 0.4571
Iteration [  780/ 1000] | d_real_loss: 0.6063 | d_Y_loss: 1.0615 | d_X_loss:
0.4989 | d_fake_loss: 1.5605 | g_loss: 0.4617
Iteration [  790/ 1000] | d_real_loss: 0.7413 | d_Y_loss: 1.0499 | d_X_loss:
0.5653 | d_fake_loss: 1.6153 | g_loss: 0.4619
Iteration [  800/ 1000] | d_real_loss: 0.6034 | d_Y_loss: 1.0623 | d_X_loss:
0.7959 | d_fake_loss: 1.8582 | g_loss: 0.4783
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000800-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000800-Y-X.png
Iteration [  810/ 1000] | d_real_loss: 0.5865 | d_Y_loss: 1.1120 | d_X_loss:
0.6170 | d_fake_loss: 1.7290 | g_loss: 0.4420
Iteration [  820/ 1000] | d_real_loss: 0.6392 | d_Y_loss: 1.0394 | d_X_loss:
0.4966 | d_fake_loss: 1.5359 | g_loss: 0.4782
Iteration [  830/ 1000] | d_real_loss: 0.6089 | d_Y_loss: 1.0923 | d_X_loss:
0.4816 | d_fake_loss: 1.5739 | g_loss: 0.4431
Iteration [  840/ 1000] | d_real_loss: 0.6246 | d_Y_loss: 1.0652 | d_X_loss:
0.4509 | d_fake_loss: 1.5161 | g_loss: 0.4709
Iteration [  850/ 1000] | d_real_loss: 0.5837 | d_Y_loss: 1.0883 | d_X_loss:
0.4141 | d_fake_loss: 1.5024 | g_loss: 0.4573
Iteration [  860/ 1000] | d_real_loss: 0.6028 | d_Y_loss: 1.0642 | d_X_loss:
```

```
0.3995 | d_fake_loss: 1.4637 | g_loss: 0.4535
Iteration [  870/ 1000] | d_real_loss: 0.5689 | d_Y_loss: 1.0715 | d_X_loss:
0.3839 | d_fake_loss: 1.4554 | g_loss: 0.4569
Iteration [  880/ 1000] | d_real_loss: 0.5564 | d_Y_loss: 1.0732 | d_X_loss:
0.4138 | d_fake_loss: 1.4870 | g_loss: 0.4628
Iteration [  890/ 1000] | d_real_loss: 0.6618 | d_Y_loss: 1.0507 | d_X_loss:
0.4527 | d_fake_loss: 1.5034 | g_loss: 0.4741
Iteration [  900/ 1000] | d_real_loss: 0.6254 | d_Y_loss: 1.0880 | d_X_loss:
0.4229 | d_fake_loss: 1.5109 | g_loss: 0.4506
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000900-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-000900-Y-X.png
Iteration [  910/ 1000] | d_real_loss: 0.5688 | d_Y_loss: 1.0609 | d_X_loss:
0.3625 | d_fake_loss: 1.4234 | g_loss: 0.4581
Iteration [  920/ 1000] | d_real_loss: 0.5722 | d_Y_loss: 1.0616 | d_X_loss:
0.5199 | d_fake_loss: 1.5815 | g_loss: 0.4770
Iteration [  930/ 1000] | d_real_loss: 0.6247 | d_Y_loss: 1.0724 | d_X_loss:
0.4270 | d_fake_loss: 1.4994 | g_loss: 0.4507
Iteration [  940/ 1000] | d_real_loss: 0.5723 | d_Y_loss: 1.0771 | d_X_loss:
0.3703 | d_fake_loss: 1.4473 | g_loss: 0.4523
Iteration [  950/ 1000] | d_real_loss: 0.5695 | d_Y_loss: 1.0783 | d_X_loss:
0.3611 | d_fake_loss: 1.4394 | g_loss: 0.4458
Iteration [  960/ 1000] | d_real_loss: 0.5604 | d_Y_loss: 1.0337 | d_X_loss:
0.3115 | d_fake_loss: 1.3451 | g_loss: 0.4782
Iteration [  970/ 1000] | d_real_loss: 0.5696 | d_Y_loss: 1.0555 | d_X_loss:
0.3861 | d_fake_loss: 1.4415 | g_loss: 0.4618
Iteration [  980/ 1000] | d_real_loss: 0.5630 | d_Y_loss: 1.0809 | d_X_loss:
0.6572 | d_fake_loss: 1.7380 | g_loss: 0.4536
Iteration [  990/ 1000] | d_real_loss: 0.5897 | d_Y_loss: 1.0391 | d_X_loss:
0.4771 | d_fake_loss: 1.5162 | g_loss: 0.4706
Iteration [ 1000/ 1000] | d_real_loss: 0.5801 | d_Y_loss: 1.0567 | d_X_loss:
0.5179 | d_fake_loss: 1.5745 | g_loss: 0.4659
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-001000-X-Y.png
Saved output/cyclegan\cat_10deluxe_instance_dc_cycle_naive\sample-001000-Y-X.png
```

Cycle Consistency Loss

[20]:
```
!python cycle_gan.py --use_cycle_consistency_loss
```

```
================================================================================
                                    Opts
--------------------------------------------------------------------------------
                           image_size: 64
                                 disc: dc
                                  gen: cycle
                           g_conv_dim: 32
                           d_conv_dim: 32
                                 norm: instance
          use_cycle_consistency_loss: 1
                            init_type: naive
```

```
                            train_iters: 1000
                             batch_size: 16
                                     lr: 0.0003
                                  beta1: 0.5
                                  beta2: 0.999
                             lambda_cycle: 10
                                       X: cat/grumpifyAprocessed
                                       Y: cat/grumpifyBprocessed
                                     ext: *.png
                                data_aug: deluxe
                           checkpoint_dir: checkpoints_cyclegan
                       sample_dir:
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle
                                log_step: 10
                             sample_every: 100
                          checkpoint_every: 800
                                      gpu: 0
================================================================================
data/cat/grumpifyAprocessed\*.png
75
data/cat/grumpifyBprocessed\*.png
204
                 G_XtoY
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
```

```
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
```

```
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
---------------------------------------
                G_YtoX
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
```

```
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
---------------------------------------
                D_X
---------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
```

```
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
----------------------------------------
                D_Y
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
```

```
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
---------------------------------------
Models moved to GPU.
Iteration [   10/ 1000] | d_real_loss: 1.0926 | d_Y_loss: 0.7399 | d_X_loss:
0.9026 | d_fake_loss: 1.6424 | g_loss: 3.7225
Iteration [   20/ 1000] | d_real_loss: 0.9553 | d_Y_loss: 0.7651 | d_X_loss:
0.8975 | d_fake_loss: 1.6626 | g_loss: 3.2417
Iteration [   30/ 1000] | d_real_loss: 0.8446 | d_Y_loss: 0.7666 | d_X_loss:
0.8501 | d_fake_loss: 1.6166 | g_loss: 3.2244
Iteration [   40/ 1000] | d_real_loss: 0.7699 | d_Y_loss: 0.7542 | d_X_loss:
0.8044 | d_fake_loss: 1.5586 | g_loss: 3.0294
Iteration [   50/ 1000] | d_real_loss: 0.7292 | d_Y_loss: 0.7593 | d_X_loss:
0.7789 | d_fake_loss: 1.5382 | g_loss: 3.6035
Iteration [   60/ 1000] | d_real_loss: 0.6857 | d_Y_loss: 0.7418 | d_X_loss:
0.8003 | d_fake_loss: 1.5421 | g_loss: 2.8928
Iteration [   70/ 1000] | d_real_loss: 0.6633 | d_Y_loss: 0.7229 | d_X_loss:
0.7727 | d_fake_loss: 1.4956 | g_loss: 3.3540
Iteration [   80/ 1000] | d_real_loss: 0.6187 | d_Y_loss: 0.6984 | d_X_loss:
0.7362 | d_fake_loss: 1.4346 | g_loss: 3.4542
Iteration [   90/ 1000] | d_real_loss: 0.6209 | d_Y_loss: 0.6994 | d_X_loss:
0.7913 | d_fake_loss: 1.4907 | g_loss: 2.7776
Iteration [  100/ 1000] | d_real_loss: 0.6063 | d_Y_loss: 0.6842 | d_X_loss:
0.7713 | d_fake_loss: 1.4555 | g_loss: 3.2380
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000100-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000100-Y-X.png
Iteration [  110/ 1000] | d_real_loss: 0.6313 | d_Y_loss: 0.7297 | d_X_loss:
0.8010 | d_fake_loss: 1.5307 | g_loss: 3.0432
Iteration [  120/ 1000] | d_real_loss: 0.5906 | d_Y_loss: 0.7177 | d_X_loss:
0.7674 | d_fake_loss: 1.4851 | g_loss: 2.7397
Iteration [  130/ 1000] | d_real_loss: 0.5469 | d_Y_loss: 0.5947 | d_X_loss:
0.7067 | d_fake_loss: 1.3014 | g_loss: 3.2032
Iteration [  140/ 1000] | d_real_loss: 0.5823 | d_Y_loss: 0.7052 | d_X_loss:
0.6907 | d_fake_loss: 1.3959 | g_loss: 2.9367
Iteration [  150/ 1000] | d_real_loss: 0.5454 | d_Y_loss: 0.6557 | d_X_loss:
0.6925 | d_fake_loss: 1.3482 | g_loss: 2.6213
Iteration [  160/ 1000] | d_real_loss: 0.5236 | d_Y_loss: 0.6245 | d_X_loss:
0.6385 | d_fake_loss: 1.2630 | g_loss: 3.1502
Iteration [  170/ 1000] | d_real_loss: 0.5329 | d_Y_loss: 0.6798 | d_X_loss:
0.6886 | d_fake_loss: 1.3684 | g_loss: 3.2788
Iteration [  180/ 1000] | d_real_loss: 0.5386 | d_Y_loss: 0.6201 | d_X_loss:
0.6982 | d_fake_loss: 1.3184 | g_loss: 3.3668
Iteration [  190/ 1000] | d_real_loss: 0.4770 | d_Y_loss: 0.6465 | d_X_loss:
```

```
0.6065 | d_fake_loss: 1.2530 | g_loss: 3.3691
Iteration [  200/ 1000] | d_real_loss: 0.4607 | d_Y_loss: 0.5923 | d_X_loss:
0.5973 | d_fake_loss: 1.1896 | g_loss: 3.0049
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000200-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000200-Y-X.png
Iteration [  210/ 1000] | d_real_loss: 0.4898 | d_Y_loss: 0.5851 | d_X_loss:
0.6468 | d_fake_loss: 1.2319 | g_loss: 3.0182
Iteration [  220/ 1000] | d_real_loss: 0.5205 | d_Y_loss: 0.5755 | d_X_loss:
0.7440 | d_fake_loss: 1.3195 | g_loss: 2.9692
Iteration [  230/ 1000] | d_real_loss: 0.4863 | d_Y_loss: 0.5856 | d_X_loss:
0.6528 | d_fake_loss: 1.2384 | g_loss: 3.4565
Iteration [  240/ 1000] | d_real_loss: 0.4901 | d_Y_loss: 0.5427 | d_X_loss:
0.5886 | d_fake_loss: 1.1313 | g_loss: 3.0343
Iteration [  250/ 1000] | d_real_loss: 0.4759 | d_Y_loss: 0.5621 | d_X_loss:
0.6105 | d_fake_loss: 1.1725 | g_loss: 3.1231
Iteration [  260/ 1000] | d_real_loss: 0.4594 | d_Y_loss: 0.5305 | d_X_loss:
0.5874 | d_fake_loss: 1.1179 | g_loss: 2.9861
Iteration [  270/ 1000] | d_real_loss: 0.4858 | d_Y_loss: 0.5633 | d_X_loss:
0.6124 | d_fake_loss: 1.1757 | g_loss: 3.2480
Iteration [  280/ 1000] | d_real_loss: 0.4560 | d_Y_loss: 0.5341 | d_X_loss:
0.6992 | d_fake_loss: 1.2333 | g_loss: 3.3455
Iteration [  290/ 1000] | d_real_loss: 0.4551 | d_Y_loss: 0.5094 | d_X_loss:
0.6527 | d_fake_loss: 1.1621 | g_loss: 2.8491
Iteration [  300/ 1000] | d_real_loss: 0.4326 | d_Y_loss: 0.5041 | d_X_loss:
0.5810 | d_fake_loss: 1.0851 | g_loss: 3.0398
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000300-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000300-Y-X.png
Iteration [  310/ 1000] | d_real_loss: 0.4230 | d_Y_loss: 0.4891 | d_X_loss:
0.6210 | d_fake_loss: 1.1100 | g_loss: 3.4734
Iteration [  320/ 1000] | d_real_loss: 0.4247 | d_Y_loss: 0.4692 | d_X_loss:
0.6370 | d_fake_loss: 1.1061 | g_loss: 3.3676
Iteration [  330/ 1000] | d_real_loss: 0.4265 | d_Y_loss: 0.4886 | d_X_loss:
0.5593 | d_fake_loss: 1.0479 | g_loss: 3.7398
Iteration [  340/ 1000] | d_real_loss: 0.4297 | d_Y_loss: 0.4732 | d_X_loss:
0.5476 | d_fake_loss: 1.0208 | g_loss: 3.3441
Iteration [  350/ 1000] | d_real_loss: 0.4841 | d_Y_loss: 0.4834 | d_X_loss:
0.5918 | d_fake_loss: 1.0753 | g_loss: 3.1970
Iteration [  360/ 1000] | d_real_loss: 0.4259 | d_Y_loss: 0.4576 | d_X_loss:
0.5879 | d_fake_loss: 1.0455 | g_loss: 3.5018
Iteration [  370/ 1000] | d_real_loss: 0.4072 | d_Y_loss: 0.4523 | d_X_loss:
0.4965 | d_fake_loss: 0.9488 | g_loss: 3.4922
Iteration [  380/ 1000] | d_real_loss: 0.4501 | d_Y_loss: 0.4790 | d_X_loss:
0.5589 | d_fake_loss: 1.0379 | g_loss: 3.4729
Iteration [  390/ 1000] | d_real_loss: 0.3939 | d_Y_loss: 0.4637 | d_X_loss:
```

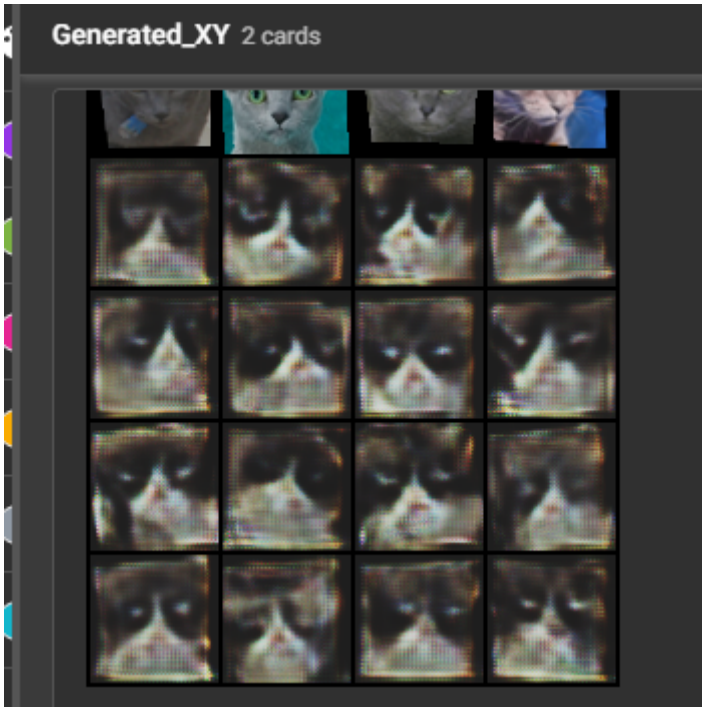
```
0.5363 | d_fake_loss: 1.0000 | g_loss: 3.5035
Iteration [  400/ 1000] | d_real_loss: 0.3856 | d_Y_loss: 0.4407 | d_X_loss:
0.4912 | d_fake_loss: 0.9319 | g_loss: 3.3607
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000400-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000400-Y-X.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000400-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000400-Y-X.png
Iteration [  410/ 1000] | d_real_loss: 0.3884 | d_Y_loss: 0.4387 | d_X_loss:
0.6153 | d_fake_loss: 1.0540 | g_loss: 3.5661
Iteration [  420/ 1000] | d_real_loss: 0.3744 | d_Y_loss: 0.4019 | d_X_loss:
0.5514 | d_fake_loss: 0.9533 | g_loss: 3.4213
Iteration [  430/ 1000] | d_real_loss: 0.3565 | d_Y_loss: 0.3822 | d_X_loss:
0.6029 | d_fake_loss: 0.9852 | g_loss: 3.7084
Iteration [  440/ 1000] | d_real_loss: 0.3657 | d_Y_loss: 0.3746 | d_X_loss:
0.5196 | d_fake_loss: 0.8942 | g_loss: 3.6820
Iteration [  450/ 1000] | d_real_loss: 0.3823 | d_Y_loss: 0.4308 | d_X_loss:
0.6456 | d_fake_loss: 1.0764 | g_loss: 2.9605
Iteration [  460/ 1000] | d_real_loss: 0.3587 | d_Y_loss: 0.3578 | d_X_loss:
0.4930 | d_fake_loss: 0.8508 | g_loss: 3.2716
Iteration [  470/ 1000] | d_real_loss: 0.3753 | d_Y_loss: 0.3544 | d_X_loss:
0.4885 | d_fake_loss: 0.8429 | g_loss: 3.2771
Iteration [  480/ 1000] | d_real_loss: 0.3659 | d_Y_loss: 0.4048 | d_X_loss:
0.5202 | d_fake_loss: 0.9250 | g_loss: 3.1675
Iteration [  490/ 1000] | d_real_loss: 0.3949 | d_Y_loss: 0.3861 | d_X_loss:
0.5231 | d_fake_loss: 0.9092 | g_loss: 3.4897
Iteration [  500/ 1000] | d_real_loss: 0.3558 | d_Y_loss: 0.4402 | d_X_loss:
0.4291 | d_fake_loss: 0.8693 | g_loss: 3.6057
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000500-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000500-Y-X.png
Iteration [  510/ 1000] | d_real_loss: 0.3645 | d_Y_loss: 0.3663 | d_X_loss:
0.4961 | d_fake_loss: 0.8625 | g_loss: 3.7603
Iteration [  520/ 1000] | d_real_loss: 0.3395 | d_Y_loss: 0.3406 | d_X_loss:
0.4686 | d_fake_loss: 0.8093 | g_loss: 3.8906
Iteration [  530/ 1000] | d_real_loss: 0.3534 | d_Y_loss: 0.3876 | d_X_loss:
0.5730 | d_fake_loss: 0.9606 | g_loss: 3.8032
Iteration [  540/ 1000] | d_real_loss: 0.3646 | d_Y_loss: 0.3508 | d_X_loss:
0.4805 | d_fake_loss: 0.8314 | g_loss: 3.3705
Iteration [  550/ 1000] | d_real_loss: 0.3613 | d_Y_loss: 0.3994 | d_X_loss:
0.5193 | d_fake_loss: 0.9188 | g_loss: 3.4954
Iteration [  560/ 1000] | d_real_loss: 0.3917 | d_Y_loss: 0.3927 | d_X_loss:
0.4484 | d_fake_loss: 0.8411 | g_loss: 3.3703
Iteration [  570/ 1000] | d_real_loss: 0.3266 | d_Y_loss: 0.3174 | d_X_loss:
```

```
0.4784 | d_fake_loss: 0.7957 | g_loss: 3.7567
Iteration [  580/ 1000] | d_real_loss: 0.3423 | d_Y_loss: 0.3317 | d_X_loss:
0.4774 | d_fake_loss: 0.8091 | g_loss: 3.7006
Iteration [  590/ 1000] | d_real_loss: 0.3599 | d_Y_loss: 0.3307 | d_X_loss:
0.4384 | d_fake_loss: 0.7691 | g_loss: 3.4155
Iteration [  600/ 1000] | d_real_loss: 0.3055 | d_Y_loss: 0.3069 | d_X_loss:
0.5293 | d_fake_loss: 0.8361 | g_loss: 3.7300
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000600-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000600-Y-X.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000600-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000600-Y-X.png
Iteration [  610/ 1000] | d_real_loss: 0.3289 | d_Y_loss: 0.3671 | d_X_loss:
0.4180 | d_fake_loss: 0.7851 | g_loss: 3.8708
Iteration [  620/ 1000] | d_real_loss: 0.3381 | d_Y_loss: 0.3022 | d_X_loss:
0.4732 | d_fake_loss: 0.7754 | g_loss: 3.5837
Iteration [  630/ 1000] | d_real_loss: 0.2963 | d_Y_loss: 0.2801 | d_X_loss:
0.4139 | d_fake_loss: 0.6940 | g_loss: 3.6907
Iteration [  640/ 1000] | d_real_loss: 0.3076 | d_Y_loss: 0.3101 | d_X_loss:
0.3842 | d_fake_loss: 0.6943 | g_loss: 3.6921
Iteration [  650/ 1000] | d_real_loss: 0.3026 | d_Y_loss: 0.3167 | d_X_loss:
0.4724 | d_fake_loss: 0.7891 | g_loss: 3.5752
Iteration [  660/ 1000] | d_real_loss: 0.3066 | d_Y_loss: 0.2715 | d_X_loss:
0.3906 | d_fake_loss: 0.6621 | g_loss: 3.8924
Iteration [  670/ 1000] | d_real_loss: 0.3566 | d_Y_loss: 0.2523 | d_X_loss:
0.3932 | d_fake_loss: 0.6454 | g_loss: 4.0757
Iteration [  680/ 1000] | d_real_loss: 0.2947 | d_Y_loss: 0.3240 | d_X_loss:
0.4620 | d_fake_loss: 0.7860 | g_loss: 3.8965
Iteration [  690/ 1000] | d_real_loss: 0.2948 | d_Y_loss: 0.2604 | d_X_loss:
0.4676 | d_fake_loss: 0.7280 | g_loss: 3.7343
Iteration [  700/ 1000] | d_real_loss: 0.3805 | d_Y_loss: 0.2405 | d_X_loss:
0.4236 | d_fake_loss: 0.6641 | g_loss: 4.0489
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000700-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000700-Y-X.png
Iteration [  710/ 1000] | d_real_loss: 0.3718 | d_Y_loss: 0.2632 | d_X_loss:
0.3706 | d_fake_loss: 0.6338 | g_loss: 3.7823
Iteration [  720/ 1000] | d_real_loss: 0.2525 | d_Y_loss: 0.2553 | d_X_loss:
0.3931 | d_fake_loss: 0.6485 | g_loss: 3.7051
Iteration [  730/ 1000] | d_real_loss: 0.2950 | d_Y_loss: 0.2877 | d_X_loss:
0.4273 | d_fake_loss: 0.7150 | g_loss: 4.2817
Iteration [  740/ 1000] | d_real_loss: 0.2653 | d_Y_loss: 0.2597 | d_X_loss:
0.3500 | d_fake_loss: 0.6097 | g_loss: 3.7555
Iteration [  750/ 1000] | d_real_loss: 0.3082 | d_Y_loss: 0.2378 | d_X_loss:
```

```
0.4929 | d_fake_loss: 0.7307 | g_loss: 4.0472
Iteration [  760/ 1000] | d_real_loss: 0.2445 | d_Y_loss: 0.2362 | d_X_loss:
0.3913 | d_fake_loss: 0.6275 | g_loss: 3.7147
Iteration [  770/ 1000] | d_real_loss: 0.2598 | d_Y_loss: 0.2169 | d_X_loss:
0.4527 | d_fake_loss: 0.6696 | g_loss: 3.9453
Iteration [  780/ 1000] | d_real_loss: 0.2542 | d_Y_loss: 0.3758 | d_X_loss:
0.3852 | d_fake_loss: 0.7610 | g_loss: 3.9480
Iteration [  790/ 1000] | d_real_loss: 0.2944 | d_Y_loss: 0.2955 | d_X_loss:
0.3419 | d_fake_loss: 0.6374 | g_loss: 3.6647
Iteration [  800/ 1000] | d_real_loss: 0.2500 | d_Y_loss: 0.2436 | d_X_loss:
0.3392 | d_fake_loss: 0.5828 | g_loss: 3.8089
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000800-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000800-Y-X.png
Iteration [  810/ 1000] | d_real_loss: 0.2981 | d_Y_loss: 0.2396 | d_X_loss:
0.5094 | d_fake_loss: 0.7490 | g_loss: 3.8000
Iteration [  820/ 1000] | d_real_loss: 0.2519 | d_Y_loss: 0.2253 | d_X_loss:
0.3377 | d_fake_loss: 0.5630 | g_loss: 3.8316
Iteration [  830/ 1000] | d_real_loss: 0.3651 | d_Y_loss: 0.2294 | d_X_loss:
0.3378 | d_fake_loss: 0.5672 | g_loss: 3.6448
Iteration [  840/ 1000] | d_real_loss: 0.2340 | d_Y_loss: 0.2443 | d_X_loss:
0.4760 | d_fake_loss: 0.7203 | g_loss: 4.2214
Iteration [  850/ 1000] | d_real_loss: 0.2164 | d_Y_loss: 0.1966 | d_X_loss:
0.3335 | d_fake_loss: 0.5301 | g_loss: 3.7413
Iteration [  860/ 1000] | d_real_loss: 0.2917 | d_Y_loss: 0.2082 | d_X_loss:
0.3038 | d_fake_loss: 0.5120 | g_loss: 4.3014
Iteration [  870/ 1000] | d_real_loss: 0.2233 | d_Y_loss: 0.2452 | d_X_loss:
0.4677 | d_fake_loss: 0.7129 | g_loss: 4.6673
Iteration [  880/ 1000] | d_real_loss: 0.2214 | d_Y_loss: 0.2773 | d_X_loss:
0.3681 | d_fake_loss: 0.6454 | g_loss: 4.9843
Iteration [  890/ 1000] | d_real_loss: 0.2178 | d_Y_loss: 0.1954 | d_X_loss:
0.2986 | d_fake_loss: 0.4941 | g_loss: 5.0520
Iteration [  900/ 1000] | d_real_loss: 0.2273 | d_Y_loss: 0.2011 | d_X_loss:
0.2846 | d_fake_loss: 0.4857 | g_loss: 3.9441
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000900-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-000900-Y-X.png
Iteration [  910/ 1000] | d_real_loss: 0.2148 | d_Y_loss: 0.2506 | d_X_loss:
0.3055 | d_fake_loss: 0.5561 | g_loss: 4.2227
Iteration [  920/ 1000] | d_real_loss: 0.2321 | d_Y_loss: 0.1961 | d_X_loss:
0.2758 | d_fake_loss: 0.4719 | g_loss: 4.0503
Iteration [  930/ 1000] | d_real_loss: 0.2611 | d_Y_loss: 0.1960 | d_X_loss:
0.3510 | d_fake_loss: 0.5470 | g_loss: 3.8536
Iteration [  940/ 1000] | d_real_loss: 0.2248 | d_Y_loss: 0.1817 | d_X_loss:
0.8134 | d_fake_loss: 0.9950 | g_loss: 4.2456
Iteration [  950/ 1000] | d_real_loss: 0.2135 | d_Y_loss: 0.2653 | d_X_loss:
```

```
0.3421 | d_fake_loss: 0.6074 | g_loss: 3.5620
Iteration [  960/ 1000] | d_real_loss: 0.2181 | d_Y_loss: 0.2504 | d_X_loss:
0.2984 | d_fake_loss: 0.5488 | g_loss: 5.0266
Iteration [  970/ 1000] | d_real_loss: 0.2301 | d_Y_loss: 0.2267 | d_X_loss:
0.3758 | d_fake_loss: 0.6024 | g_loss: 4.0287
Iteration [  980/ 1000] | d_real_loss: 0.2156 | d_Y_loss: 0.1836 | d_X_loss:
0.2746 | d_fake_loss: 0.4582 | g_loss: 4.0677
Iteration [  990/ 1000] | d_real_loss: 0.2342 | d_Y_loss: 0.2200 | d_X_loss:
0.2743 | d_fake_loss: 0.4943 | g_loss: 4.1718
Iteration [ 1000/ 1000] | d_real_loss: 0.1909 | d_Y_loss: 0.4740 | d_X_loss:
0.3387 | d_fake_loss: 0.8127 | g_loss: 4.1811
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-001000-X-Y.png
Saved
output/cyclegan\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-001000-Y-X.png
```

Images 400 Iteraction



700 Iteraction

Looking at the images, we can see that the ones from step 400 may be somewhat similar to the ones from step 700, although they change and look somewhat better. This is probably because when the training has just started the generator has not learned very well how to change the first type of image to look like the second type. Now, in this case, both images may not be very well created and we could at a glance identify which are fake and which are real, since if we look at step 700, even though the network tries to get better at making the images look like the other type, it may not be enough to have better textures, match colors and look more real. So we may still see some problems or things that don't look quite right because the model is still learning and trying to improve in the next steps. Something that could be improved if we revisit images in later steps in the training.

Execution 10000 Iteractions

[2]:
```
!python cycle_gan.py --train_iters=10000 --sample_dir=cycle_gan_10000
```

```
================================================================================
                                 Opts
--------------------------------------------------------------------------------
                          image_size: 64
                                disc: dc
                                 gen: cycle
                          g_conv_dim: 32
                          d_conv_dim: 32
                                norm: instance
                           init_type: naive
                         train_iters: 10000
                          batch_size: 16
```

```
                                    lr: 0.0003
                                 beta1: 0.5
                                 beta2: 0.999
                           lambda_cycle: 10
                                     X: cat/grumpifyAprocessed
                                     Y: cat/grumpifyBprocessed
                                   ext: *.png
                              data_aug: deluxe
                         checkpoint_dir: checkpoints_cyclegan
                     sample_dir:
output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive
                                log_step: 10
                             sample_every: 100
                          checkpoint_every: 800
                                    gpu: 0
================================================================================
data/cat/grumpifyAprocessed\*.png
75
data/cat/grumpifyBprocessed\*.png
204
                 G_XtoY
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
```

```
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
```

```
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
---------------------------------------
                G_YtoX
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
```

```
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
----------------------------------------
                D_X
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
```

```
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
----------------------------------------
                D_Y
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
```

```
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
---------------------------------------
Models moved to GPU.
Iteration [   10/10000] | d_real_loss: 1.0837 | d_Y_loss: 0.7461 | d_X_loss:
0.9174 | d_fake_loss: 1.6634 | g_loss: 0.6849
Iteration [   20/10000] | d_real_loss: 0.9459 | d_Y_loss: 0.7960 | d_X_loss:
0.9447 | d_fake_loss: 1.7407 | g_loss: 0.6315
Iteration [   30/10000] | d_real_loss: 0.8566 | d_Y_loss: 0.8612 | d_X_loss:
1.0213 | d_fake_loss: 1.8825 | g_loss: 0.5856
Iteration [   40/10000] | d_real_loss: 0.8131 | d_Y_loss: 0.8937 | d_X_loss:
0.9952 | d_fake_loss: 1.8889 | g_loss: 0.5684
Iteration [   50/10000] | d_real_loss: 0.7938 | d_Y_loss: 0.9183 | d_X_loss:
1.0456 | d_fake_loss: 1.9640 | g_loss: 0.5417
Iteration [   60/10000] | d_real_loss: 0.7906 | d_Y_loss: 0.9175 | d_X_loss:
1.0205 | d_fake_loss: 1.9380 | g_loss: 0.5607
Iteration [   70/10000] | d_real_loss: 0.8056 | d_Y_loss: 1.0157 | d_X_loss:
1.0309 | d_fake_loss: 2.0466 | g_loss: 0.4988
Iteration [   80/10000] | d_real_loss: 0.8075 | d_Y_loss: 1.0541 | d_X_loss:
1.0684 | d_fake_loss: 2.1225 | g_loss: 0.4789
Iteration [   90/10000] | d_real_loss: 0.8035 | d_Y_loss: 1.0401 | d_X_loss:
1.0494 | d_fake_loss: 2.0895 | g_loss: 0.4873
Iteration [  100/10000] | d_real_loss: 0.8229 | d_Y_loss: 1.0346 | d_X_loss:
1.0517 | d_fake_loss: 2.0862 | g_loss: 0.4906
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000100-
Y-X.png
Iteration [  110/10000] | d_real_loss: 0.8490 | d_Y_loss: 1.0063 | d_X_loss:
1.0297 | d_fake_loss: 2.0360 | g_loss: 0.4911
Iteration [  120/10000] | d_real_loss: 0.8142 | d_Y_loss: 1.0447 | d_X_loss:
1.0833 | d_fake_loss: 2.1280 | g_loss: 0.4805
Iteration [  130/10000] | d_real_loss: 0.8428 | d_Y_loss: 1.0636 | d_X_loss:
1.0450 | d_fake_loss: 2.1086 | g_loss: 0.4727
Iteration [  140/10000] | d_real_loss: 0.8311 | d_Y_loss: 1.0625 | d_X_loss:
1.0968 | d_fake_loss: 2.1593 | g_loss: 0.4659
Iteration [  150/10000] | d_real_loss: 0.8226 | d_Y_loss: 1.0197 | d_X_loss:
1.0352 | d_fake_loss: 2.0549 | g_loss: 0.4820
Iteration [  160/10000] | d_real_loss: 0.7989 | d_Y_loss: 1.0560 | d_X_loss:
1.0851 | d_fake_loss: 2.1411 | g_loss: 0.4681
Iteration [  170/10000] | d_real_loss: 0.8236 | d_Y_loss: 1.0311 | d_X_loss:
1.0732 | d_fake_loss: 2.1043 | g_loss: 0.4878
Iteration [  180/10000] | d_real_loss: 0.8289 | d_Y_loss: 1.0614 | d_X_loss:
1.0438 | d_fake_loss: 2.1052 | g_loss: 0.4667
Iteration [  190/10000] | d_real_loss: 0.8108 | d_Y_loss: 1.0416 | d_X_loss:
1.0751 | d_fake_loss: 2.1167 | g_loss: 0.4788
Iteration [  200/10000] | d_real_loss: 0.7925 | d_Y_loss: 1.0961 | d_X_loss:
```

```
1.0812 | d_fake_loss: 2.1773 | g_loss: 0.4445
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000200-
Y-X.png
Iteration [  210/10000] | d_real_loss: 0.7986 | d_Y_loss: 1.0728 | d_X_loss:
1.0523 | d_fake_loss: 2.1251 | g_loss: 0.4601
Iteration [  220/10000] | d_real_loss: 0.8224 | d_Y_loss: 1.0496 | d_X_loss:
1.0621 | d_fake_loss: 2.1116 | g_loss: 0.4786
Iteration [  230/10000] | d_real_loss: 0.8094 | d_Y_loss: 1.0498 | d_X_loss:
1.0444 | d_fake_loss: 2.0942 | g_loss: 0.4608
Iteration [  240/10000] | d_real_loss: 0.8093 | d_Y_loss: 1.0916 | d_X_loss:
1.0767 | d_fake_loss: 2.1684 | g_loss: 0.4517
Iteration [  250/10000] | d_real_loss: 0.8260 | d_Y_loss: 1.0797 | d_X_loss:
1.0436 | d_fake_loss: 2.1232 | g_loss: 0.4486
Iteration [  260/10000] | d_real_loss: 0.8096 | d_Y_loss: 1.0620 | d_X_loss:
1.0869 | d_fake_loss: 2.1489 | g_loss: 0.4603
Iteration [  270/10000] | d_real_loss: 0.8012 | d_Y_loss: 1.0792 | d_X_loss:
1.0593 | d_fake_loss: 2.1385 | g_loss: 0.4490
Iteration [  280/10000] | d_real_loss: 0.7932 | d_Y_loss: 1.0628 | d_X_loss:
1.0734 | d_fake_loss: 2.1362 | g_loss: 0.4608
Iteration [  290/10000] | d_real_loss: 0.7977 | d_Y_loss: 1.0592 | d_X_loss:
1.0769 | d_fake_loss: 2.1361 | g_loss: 0.4629
Iteration [  300/10000] | d_real_loss: 0.7926 | d_Y_loss: 1.0804 | d_X_loss:
0.9985 | d_fake_loss: 2.0789 | g_loss: 0.4511
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000300-
Y-X.png
Iteration [  310/10000] | d_real_loss: 0.8002 | d_Y_loss: 1.0767 | d_X_loss:
1.0451 | d_fake_loss: 2.1217 | g_loss: 0.4507
Iteration [  320/10000] | d_real_loss: 0.8072 | d_Y_loss: 1.0714 | d_X_loss:
1.0717 | d_fake_loss: 2.1432 | g_loss: 0.4577
Iteration [  330/10000] | d_real_loss: 0.8080 | d_Y_loss: 1.1004 | d_X_loss:
1.0293 | d_fake_loss: 2.1297 | g_loss: 0.4380
Iteration [  340/10000] | d_real_loss: 0.7891 | d_Y_loss: 1.0684 | d_X_loss:
1.0580 | d_fake_loss: 2.1264 | g_loss: 0.4521
Iteration [  350/10000] | d_real_loss: 0.7826 | d_Y_loss: 1.0822 | d_X_loss:
1.1068 | d_fake_loss: 2.1890 | g_loss: 0.4500
Iteration [  360/10000] | d_real_loss: 0.8074 | d_Y_loss: 1.0681 | d_X_loss:
1.0999 | d_fake_loss: 2.1680 | g_loss: 0.4533
Iteration [  370/10000] | d_real_loss: 0.7886 | d_Y_loss: 1.0847 | d_X_loss:
0.9715 | d_fake_loss: 2.0562 | g_loss: 0.4443
Iteration [  380/10000] | d_real_loss: 0.7619 | d_Y_loss: 1.0765 | d_X_loss:
0.9650 | d_fake_loss: 2.0415 | g_loss: 0.4468
Iteration [  390/10000] | d_real_loss: 0.7763 | d_Y_loss: 1.0355 | d_X_loss:
1.0409 | d_fake_loss: 2.0764 | g_loss: 0.4693
Iteration [  400/10000] | d_real_loss: 0.7558 | d_Y_loss: 1.0755 | d_X_loss:
```

```
1.0693 | d_fake_loss: 2.1448 | g_loss: 0.4499
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000400-
Y-X.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000400-
Y-X.png
Iteration [  410/10000] | d_real_loss: 0.7692 | d_Y_loss: 1.0577 | d_X_loss:
1.0244 | d_fake_loss: 2.0820 | g_loss: 0.4636
Iteration [  420/10000] | d_real_loss: 0.7825 | d_Y_loss: 1.1147 | d_X_loss:
1.0834 | d_fake_loss: 2.1981 | g_loss: 0.4397
Iteration [  430/10000] | d_real_loss: 0.7744 | d_Y_loss: 1.0932 | d_X_loss:
1.0495 | d_fake_loss: 2.1427 | g_loss: 0.4443
Iteration [  440/10000] | d_real_loss: 0.7841 | d_Y_loss: 1.0649 | d_X_loss:
1.1294 | d_fake_loss: 2.1943 | g_loss: 0.4537
Iteration [  450/10000] | d_real_loss: 0.7924 | d_Y_loss: 1.0769 | d_X_loss:
1.0516 | d_fake_loss: 2.1285 | g_loss: 0.4497
Iteration [  460/10000] | d_real_loss: 0.7964 | d_Y_loss: 1.0983 | d_X_loss:
1.0182 | d_fake_loss: 2.1164 | g_loss: 0.4470
Iteration [  470/10000] | d_real_loss: 0.7902 | d_Y_loss: 1.0790 | d_X_loss:
1.0772 | d_fake_loss: 2.1562 | g_loss: 0.4492
Iteration [  480/10000] | d_real_loss: 0.8079 | d_Y_loss: 1.0614 | d_X_loss:
1.0252 | d_fake_loss: 2.0867 | g_loss: 0.4561
Iteration [  490/10000] | d_real_loss: 0.7983 | d_Y_loss: 1.0659 | d_X_loss:
1.0223 | d_fake_loss: 2.0881 | g_loss: 0.4555
Iteration [  500/10000] | d_real_loss: 0.7948 | d_Y_loss: 1.1152 | d_X_loss:
1.0860 | d_fake_loss: 2.2012 | g_loss: 0.4333
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000500-
Y-X.png
Iteration [  510/10000] | d_real_loss: 0.7726 | d_Y_loss: 1.0739 | d_X_loss:
1.0019 | d_fake_loss: 2.0759 | g_loss: 0.4472
Iteration [  520/10000] | d_real_loss: 0.7571 | d_Y_loss: 1.1083 | d_X_loss:
1.0627 | d_fake_loss: 2.1711 | g_loss: 0.4340
Iteration [  530/10000] | d_real_loss: 0.8028 | d_Y_loss: 1.0780 | d_X_loss:
1.0748 | d_fake_loss: 2.1528 | g_loss: 0.4491
Iteration [  540/10000] | d_real_loss: 0.8185 | d_Y_loss: 1.0740 | d_X_loss:
1.0110 | d_fake_loss: 2.0850 | g_loss: 0.4571
Iteration [  550/10000] | d_real_loss: 0.7840 | d_Y_loss: 1.0397 | d_X_loss:
1.0089 | d_fake_loss: 2.0486 | g_loss: 0.4683
Iteration [  560/10000] | d_real_loss: 0.8128 | d_Y_loss: 1.0724 | d_X_loss:
1.0589 | d_fake_loss: 2.1313 | g_loss: 0.4446
Iteration [  570/10000] | d_real_loss: 0.7737 | d_Y_loss: 1.0777 | d_X_loss:
1.0514 | d_fake_loss: 2.1291 | g_loss: 0.4432
Iteration [  580/10000] | d_real_loss: 0.7891 | d_Y_loss: 1.0810 | d_X_loss:
```

```
1.0923 | d_fake_loss: 2.1734 | g_loss: 0.4389
Iteration [  590/10000] | d_real_loss: 0.7746 | d_Y_loss: 1.0880 | d_X_loss:
0.9943 | d_fake_loss: 2.0823 | g_loss: 0.4479
Iteration [  600/10000] | d_real_loss: 0.7878 | d_Y_loss: 1.0733 | d_X_loss:
1.0665 | d_fake_loss: 2.1398 | g_loss: 0.4461
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000600-
Y-X.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000600-
Y-X.png
Iteration [  610/10000] | d_real_loss: 0.7804 | d_Y_loss: 1.0805 | d_X_loss:
1.0441 | d_fake_loss: 2.1247 | g_loss: 0.4434
Iteration [  620/10000] | d_real_loss: 0.7915 | d_Y_loss: 1.0815 | d_X_loss:
1.0547 | d_fake_loss: 2.1362 | g_loss: 0.4459
Iteration [  630/10000] | d_real_loss: 0.7638 | d_Y_loss: 1.0718 | d_X_loss:
1.0323 | d_fake_loss: 2.1042 | g_loss: 0.4489
Iteration [  640/10000] | d_real_loss: 0.7727 | d_Y_loss: 1.0645 | d_X_loss:
1.0544 | d_fake_loss: 2.1189 | g_loss: 0.4573
Iteration [  650/10000] | d_real_loss: 0.7934 | d_Y_loss: 1.0600 | d_X_loss:
1.0609 | d_fake_loss: 2.1209 | g_loss: 0.4653
Iteration [  660/10000] | d_real_loss: 0.7545 | d_Y_loss: 1.1188 | d_X_loss:
0.9947 | d_fake_loss: 2.1135 | g_loss: 0.4357
Iteration [  670/10000] | d_real_loss: 0.7820 | d_Y_loss: 1.0832 | d_X_loss:
1.0090 | d_fake_loss: 2.0922 | g_loss: 0.4454
Iteration [  680/10000] | d_real_loss: 0.8093 | d_Y_loss: 1.0829 | d_X_loss:
1.0314 | d_fake_loss: 2.1144 | g_loss: 0.4461
Iteration [  690/10000] | d_real_loss: 0.7580 | d_Y_loss: 1.0603 | d_X_loss:
1.0712 | d_fake_loss: 2.1315 | g_loss: 0.4581
Iteration [  700/10000] | d_real_loss: 0.7640 | d_Y_loss: 1.0718 | d_X_loss:
0.9273 | d_fake_loss: 1.9991 | g_loss: 0.4561
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000700-
Y-X.png
Iteration [  710/10000] | d_real_loss: 0.7716 | d_Y_loss: 1.1041 | d_X_loss:
0.9989 | d_fake_loss: 2.1030 | g_loss: 0.4392
Iteration [  720/10000] | d_real_loss: 0.7970 | d_Y_loss: 1.0424 | d_X_loss:
0.9302 | d_fake_loss: 1.9726 | g_loss: 0.4774
Iteration [  730/10000] | d_real_loss: 0.7288 | d_Y_loss: 1.0776 | d_X_loss:
0.9779 | d_fake_loss: 2.0555 | g_loss: 0.4458
Iteration [  740/10000] | d_real_loss: 0.7638 | d_Y_loss: 1.0956 | d_X_loss:
1.0208 | d_fake_loss: 2.1164 | g_loss: 0.4368
Iteration [  750/10000] | d_real_loss: 0.7571 | d_Y_loss: 1.0776 | d_X_loss:
0.9318 | d_fake_loss: 2.0094 | g_loss: 0.4546
Iteration [  760/10000] | d_real_loss: 0.7549 | d_Y_loss: 1.0852 | d_X_loss:
```

```
1.0103 | d_fake_loss: 2.0955 | g_loss: 0.4466
Iteration [  770/10000] | d_real_loss: 0.7337 | d_Y_loss: 1.0651 | d_X_loss:
1.0011 | d_fake_loss: 2.0661 | g_loss: 0.4562
Iteration [  780/10000] | d_real_loss: 0.7645 | d_Y_loss: 1.0858 | d_X_loss:
1.0789 | d_fake_loss: 2.1647 | g_loss: 0.4483
Iteration [  790/10000] | d_real_loss: 0.7764 | d_Y_loss: 1.0610 | d_X_loss:
0.9371 | d_fake_loss: 1.9981 | g_loss: 0.4538
Iteration [  800/10000] | d_real_loss: 0.7791 | d_Y_loss: 1.0906 | d_X_loss:
0.9901 | d_fake_loss: 2.0807 | g_loss: 0.4538
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000800-
Y-X.png
Iteration [  810/10000] | d_real_loss: 0.7651 | d_Y_loss: 1.0607 | d_X_loss:
1.0326 | d_fake_loss: 2.0933 | g_loss: 0.4544
Iteration [  820/10000] | d_real_loss: 0.7879 | d_Y_loss: 1.0804 | d_X_loss:
0.9535 | d_fake_loss: 2.0339 | g_loss: 0.4483
Iteration [  830/10000] | d_real_loss: 0.7744 | d_Y_loss: 1.0610 | d_X_loss:
1.0111 | d_fake_loss: 2.0722 | g_loss: 0.4548
Iteration [  840/10000] | d_real_loss: 0.7883 | d_Y_loss: 1.0716 | d_X_loss:
0.9792 | d_fake_loss: 2.0507 | g_loss: 0.4473
Iteration [  850/10000] | d_real_loss: 0.7423 | d_Y_loss: 1.0421 | d_X_loss:
1.0117 | d_fake_loss: 2.0538 | g_loss: 0.4733
Iteration [  860/10000] | d_real_loss: 0.7528 | d_Y_loss: 1.0712 | d_X_loss:
0.9889 | d_fake_loss: 2.0601 | g_loss: 0.4551
Iteration [  870/10000] | d_real_loss: 0.7760 | d_Y_loss: 1.0993 | d_X_loss:
0.9299 | d_fake_loss: 2.0292 | g_loss: 0.4409
Iteration [  880/10000] | d_real_loss: 0.7615 | d_Y_loss: 1.0551 | d_X_loss:
1.0003 | d_fake_loss: 2.0553 | g_loss: 0.4635
Iteration [  890/10000] | d_real_loss: 0.7578 | d_Y_loss: 1.1138 | d_X_loss:
0.8440 | d_fake_loss: 1.9579 | g_loss: 0.4402
Iteration [  900/10000] | d_real_loss: 0.7510 | d_Y_loss: 1.0465 | d_X_loss:
0.8662 | d_fake_loss: 1.9127 | g_loss: 0.4684
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-000900-
Y-X.png
Iteration [  910/10000] | d_real_loss: 0.7191 | d_Y_loss: 1.0796 | d_X_loss:
1.0319 | d_fake_loss: 2.1115 | g_loss: 0.4480
Iteration [  920/10000] | d_real_loss: 0.7504 | d_Y_loss: 1.1051 | d_X_loss:
0.9900 | d_fake_loss: 2.0951 | g_loss: 0.4427
Iteration [  930/10000] | d_real_loss: 0.7037 | d_Y_loss: 1.0304 | d_X_loss:
0.8553 | d_fake_loss: 1.8857 | g_loss: 0.4741
Iteration [  940/10000] | d_real_loss: 0.7136 | d_Y_loss: 1.1352 | d_X_loss:
0.9351 | d_fake_loss: 2.0704 | g_loss: 0.4279
Iteration [  950/10000] | d_real_loss: 0.7509 | d_Y_loss: 1.0814 | d_X_loss:
0.9927 | d_fake_loss: 2.0740 | g_loss: 0.4418
Iteration [  960/10000] | d_real_loss: 0.7385 | d_Y_loss: 1.0594 | d_X_loss:
```

```
0.9392 | d_fake_loss: 1.9986 | g_loss: 0.4623
Iteration [  970/10000] | d_real_loss: 0.7607 | d_Y_loss: 1.0653 | d_X_loss:
0.8253 | d_fake_loss: 1.8906 | g_loss: 0.4547
Iteration [  980/10000] | d_real_loss: 0.7261 | d_Y_loss: 1.0760 | d_X_loss:
0.9418 | d_fake_loss: 2.0178 | g_loss: 0.4447
Iteration [  990/10000] | d_real_loss: 0.7953 | d_Y_loss: 1.0881 | d_X_loss:
0.9775 | d_fake_loss: 2.0656 | g_loss: 0.4464
Iteration [ 1000/10000] | d_real_loss: 0.7661 | d_Y_loss: 1.0443 | d_X_loss:
0.9781 | d_fake_loss: 2.0224 | g_loss: 0.4650
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001000-
Y-X.png
Iteration [ 1010/10000] | d_real_loss: 0.7469 | d_Y_loss: 1.0916 | d_X_loss:
0.8829 | d_fake_loss: 1.9745 | g_loss: 0.4365
Iteration [ 1020/10000] | d_real_loss: 0.7559 | d_Y_loss: 1.0933 | d_X_loss:
0.9305 | d_fake_loss: 2.0238 | g_loss: 0.4431
Iteration [ 1030/10000] | d_real_loss: 0.7545 | d_Y_loss: 1.0476 | d_X_loss:
0.9415 | d_fake_loss: 1.9891 | g_loss: 0.4630
Iteration [ 1040/10000] | d_real_loss: 0.7205 | d_Y_loss: 1.1233 | d_X_loss:
0.9871 | d_fake_loss: 2.1104 | g_loss: 0.4390
Iteration [ 1050/10000] | d_real_loss: 0.6945 | d_Y_loss: 1.0538 | d_X_loss:
0.7827 | d_fake_loss: 1.8365 | g_loss: 0.4581
Iteration [ 1060/10000] | d_real_loss: 0.7352 | d_Y_loss: 1.1000 | d_X_loss:
0.8937 | d_fake_loss: 1.9936 | g_loss: 0.4336
Iteration [ 1070/10000] | d_real_loss: 0.7168 | d_Y_loss: 1.0898 | d_X_loss:
1.0598 | d_fake_loss: 2.1496 | g_loss: 0.4454
Iteration [ 1080/10000] | d_real_loss: 0.7206 | d_Y_loss: 1.0411 | d_X_loss:
0.8101 | d_fake_loss: 1.8511 | g_loss: 0.4766
Iteration [ 1090/10000] | d_real_loss: 0.7464 | d_Y_loss: 1.1262 | d_X_loss:
0.9204 | d_fake_loss: 2.0465 | g_loss: 0.4365
Iteration [ 1100/10000] | d_real_loss: 0.8274 | d_Y_loss: 1.0753 | d_X_loss:
0.7861 | d_fake_loss: 1.8614 | g_loss: 0.4498
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001100-
Y-X.png
Iteration [ 1110/10000] | d_real_loss: 0.7082 | d_Y_loss: 1.0728 | d_X_loss:
0.8640 | d_fake_loss: 1.9368 | g_loss: 0.4440
Iteration [ 1120/10000] | d_real_loss: 0.7593 | d_Y_loss: 1.0616 | d_X_loss:
0.9384 | d_fake_loss: 2.0000 | g_loss: 0.4617
Iteration [ 1130/10000] | d_real_loss: 0.6985 | d_Y_loss: 1.0386 | d_X_loss:
0.8735 | d_fake_loss: 1.9122 | g_loss: 0.4693
Iteration [ 1140/10000] | d_real_loss: 0.6951 | d_Y_loss: 1.1189 | d_X_loss:
0.8213 | d_fake_loss: 1.9402 | g_loss: 0.4414
Iteration [ 1150/10000] | d_real_loss: 0.6864 | d_Y_loss: 1.1187 | d_X_loss:
0.8613 | d_fake_loss: 1.9801 | g_loss: 0.4375
Iteration [ 1160/10000] | d_real_loss: 0.6996 | d_Y_loss: 1.0627 | d_X_loss:
```

```
0.7455 | d_fake_loss: 1.8082 | g_loss: 0.4609
Iteration [ 1170/10000] | d_real_loss: 0.6477 | d_Y_loss: 1.0905 | d_X_loss:
0.6896 | d_fake_loss: 1.7801 | g_loss: 0.4386
Iteration [ 1180/10000] | d_real_loss: 0.6451 | d_Y_loss: 1.0711 | d_X_loss:
0.6523 | d_fake_loss: 1.7234 | g_loss: 0.4616
Iteration [ 1190/10000] | d_real_loss: 0.6532 | d_Y_loss: 1.0720 | d_X_loss:
0.8110 | d_fake_loss: 1.8830 | g_loss: 0.4561
Iteration [ 1200/10000] | d_real_loss: 0.6511 | d_Y_loss: 1.0482 | d_X_loss:
0.6247 | d_fake_loss: 1.6729 | g_loss: 0.4650
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001200-
Y-X.png
Iteration [ 1210/10000] | d_real_loss: 0.6309 | d_Y_loss: 1.0706 | d_X_loss:
0.5830 | d_fake_loss: 1.6535 | g_loss: 0.4575
Iteration [ 1220/10000] | d_real_loss: 0.7302 | d_Y_loss: 1.0817 | d_X_loss:
0.6609 | d_fake_loss: 1.7427 | g_loss: 0.4507
Iteration [ 1230/10000] | d_real_loss: 0.6611 | d_Y_loss: 1.0563 | d_X_loss:
0.6229 | d_fake_loss: 1.6793 | g_loss: 0.4559
Iteration [ 1240/10000] | d_real_loss: 0.6912 | d_Y_loss: 1.0689 | d_X_loss:
0.6080 | d_fake_loss: 1.6769 | g_loss: 0.4535
Iteration [ 1250/10000] | d_real_loss: 0.6285 | d_Y_loss: 1.0480 | d_X_loss:
0.5418 | d_fake_loss: 1.5898 | g_loss: 0.4621
Iteration [ 1260/10000] | d_real_loss: 0.6167 | d_Y_loss: 1.0501 | d_X_loss:
0.5584 | d_fake_loss: 1.6084 | g_loss: 0.4668
Iteration [ 1270/10000] | d_real_loss: 0.6292 | d_Y_loss: 1.0663 | d_X_loss:
0.5082 | d_fake_loss: 1.5745 | g_loss: 0.4564
Iteration [ 1280/10000] | d_real_loss: 0.5936 | d_Y_loss: 1.0430 | d_X_loss:
0.5107 | d_fake_loss: 1.5537 | g_loss: 0.4662
Iteration [ 1290/10000] | d_real_loss: 0.6053 | d_Y_loss: 1.0643 | d_X_loss:
0.4718 | d_fake_loss: 1.5360 | g_loss: 0.4547
Iteration [ 1300/10000] | d_real_loss: 0.5952 | d_Y_loss: 1.0926 | d_X_loss:
0.4490 | d_fake_loss: 1.5416 | g_loss: 0.4368
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001300-
Y-X.png
Iteration [ 1310/10000] | d_real_loss: 0.6448 | d_Y_loss: 1.0718 | d_X_loss:
0.5502 | d_fake_loss: 1.6220 | g_loss: 0.4649
Iteration [ 1320/10000] | d_real_loss: 0.6277 | d_Y_loss: 1.0367 | d_X_loss:
0.6071 | d_fake_loss: 1.6438 | g_loss: 0.4683
Iteration [ 1330/10000] | d_real_loss: 0.6219 | d_Y_loss: 1.0678 | d_X_loss:
0.5070 | d_fake_loss: 1.5748 | g_loss: 0.4510
Iteration [ 1340/10000] | d_real_loss: 0.6760 | d_Y_loss: 1.0966 | d_X_loss:
0.4944 | d_fake_loss: 1.5910 | g_loss: 0.4417
Iteration [ 1350/10000] | d_real_loss: 0.6020 | d_Y_loss: 1.0597 | d_X_loss:
0.5281 | d_fake_loss: 1.5879 | g_loss: 0.4571
Iteration [ 1360/10000] | d_real_loss: 0.6078 | d_Y_loss: 1.0686 | d_X_loss:
```

```
0.4481 | d_fake_loss: 1.5167 | g_loss: 0.4678
Iteration [ 1370/10000] | d_real_loss: 0.5914 | d_Y_loss: 1.0869 | d_X_loss:
0.4629 | d_fake_loss: 1.5499 | g_loss: 0.4471
Iteration [ 1380/10000] | d_real_loss: 0.5776 | d_Y_loss: 1.0831 | d_X_loss:
0.4095 | d_fake_loss: 1.4927 | g_loss: 0.4557
Iteration [ 1390/10000] | d_real_loss: 0.5796 | d_Y_loss: 1.0584 | d_X_loss:
0.4860 | d_fake_loss: 1.5443 | g_loss: 0.4575
Iteration [ 1400/10000] | d_real_loss: 0.5972 | d_Y_loss: 1.0822 | d_X_loss:
0.4153 | d_fake_loss: 1.4975 | g_loss: 0.4473
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001400-
Y-X.png
Iteration [ 1410/10000] | d_real_loss: 0.6076 | d_Y_loss: 1.0694 | d_X_loss:
0.4584 | d_fake_loss: 1.5278 | g_loss: 0.4510
Iteration [ 1420/10000] | d_real_loss: 0.5851 | d_Y_loss: 1.0722 | d_X_loss:
0.3666 | d_fake_loss: 1.4388 | g_loss: 0.4672
Iteration [ 1430/10000] | d_real_loss: 0.5581 | d_Y_loss: 1.0589 | d_X_loss:
0.4102 | d_fake_loss: 1.4691 | g_loss: 0.4567
Iteration [ 1440/10000] | d_real_loss: 0.5510 | d_Y_loss: 1.0608 | d_X_loss:
0.4677 | d_fake_loss: 1.5285 | g_loss: 0.4511
Iteration [ 1450/10000] | d_real_loss: 0.5749 | d_Y_loss: 1.0114 | d_X_loss:
0.3429 | d_fake_loss: 1.3543 | g_loss: 0.4848
Iteration [ 1460/10000] | d_real_loss: 0.5384 | d_Y_loss: 1.0913 | d_X_loss:
0.3458 | d_fake_loss: 1.4371 | g_loss: 0.4570
Iteration [ 1470/10000] | d_real_loss: 0.5803 | d_Y_loss: 1.0733 | d_X_loss:
0.3814 | d_fake_loss: 1.4547 | g_loss: 0.4606
Iteration [ 1480/10000] | d_real_loss: 0.5494 | d_Y_loss: 1.0003 | d_X_loss:
0.4080 | d_fake_loss: 1.4083 | g_loss: 0.4905
Iteration [ 1490/10000] | d_real_loss: 0.5370 | d_Y_loss: 1.1579 | d_X_loss:
0.3202 | d_fake_loss: 1.4782 | g_loss: 0.4235
Iteration [ 1500/10000] | d_real_loss: 0.5486 | d_Y_loss: 1.0816 | d_X_loss:
0.2919 | d_fake_loss: 1.3735 | g_loss: 0.4455
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001500-
Y-X.png
Iteration [ 1510/10000] | d_real_loss: 0.5550 | d_Y_loss: 1.0850 | d_X_loss:
0.3320 | d_fake_loss: 1.4169 | g_loss: 0.4544
Iteration [ 1520/10000] | d_real_loss: 0.5534 | d_Y_loss: 1.0683 | d_X_loss:
0.4032 | d_fake_loss: 1.4715 | g_loss: 0.4599
Iteration [ 1530/10000] | d_real_loss: 0.5458 | d_Y_loss: 1.0377 | d_X_loss:
0.2816 | d_fake_loss: 1.3193 | g_loss: 0.4665
Iteration [ 1540/10000] | d_real_loss: 0.5152 | d_Y_loss: 1.1170 | d_X_loss:
0.2812 | d_fake_loss: 1.3982 | g_loss: 0.4423
Iteration [ 1550/10000] | d_real_loss: 0.6371 | d_Y_loss: 1.0492 | d_X_loss:
0.3391 | d_fake_loss: 1.3883 | g_loss: 0.4741
Iteration [ 1560/10000] | d_real_loss: 0.5533 | d_Y_loss: 1.1233 | d_X_loss:
```

```
0.6586 | d_fake_loss: 1.7819 | g_loss: 0.4373
Iteration [ 1570/10000] | d_real_loss: 0.5497 | d_Y_loss: 1.0622 | d_X_loss:
0.2796 | d_fake_loss: 1.3418 | g_loss: 0.4588
Iteration [ 1580/10000] | d_real_loss: 0.5308 | d_Y_loss: 1.0564 | d_X_loss:
0.4774 | d_fake_loss: 1.5337 | g_loss: 0.4610
Iteration [ 1590/10000] | d_real_loss: 0.5421 | d_Y_loss: 1.0761 | d_X_loss:
0.3177 | d_fake_loss: 1.3938 | g_loss: 0.4543
Iteration [ 1600/10000] | d_real_loss: 0.5723 | d_Y_loss: 1.0255 | d_X_loss:
0.3786 | d_fake_loss: 1.4041 | g_loss: 0.4716
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001600-
Y-X.png
Iteration [ 1610/10000] | d_real_loss: 0.5665 | d_Y_loss: 1.0922 | d_X_loss:
0.3052 | d_fake_loss: 1.3974 | g_loss: 0.4645
Iteration [ 1620/10000] | d_real_loss: 0.5419 | d_Y_loss: 1.0455 | d_X_loss:
0.3097 | d_fake_loss: 1.3553 | g_loss: 0.4734
Iteration [ 1630/10000] | d_real_loss: 0.5434 | d_Y_loss: 1.0649 | d_X_loss:
0.2811 | d_fake_loss: 1.3461 | g_loss: 0.4572
Iteration [ 1640/10000] | d_real_loss: 0.5927 | d_Y_loss: 1.0198 | d_X_loss:
0.4666 | d_fake_loss: 1.4864 | g_loss: 0.4877
Iteration [ 1650/10000] | d_real_loss: 0.5464 | d_Y_loss: 1.1057 | d_X_loss:
0.2859 | d_fake_loss: 1.3916 | g_loss: 0.4351
Iteration [ 1660/10000] | d_real_loss: 0.5181 | d_Y_loss: 1.0668 | d_X_loss:
0.2674 | d_fake_loss: 1.3342 | g_loss: 0.4605
Iteration [ 1670/10000] | d_real_loss: 0.5688 | d_Y_loss: 1.0367 | d_X_loss:
0.3008 | d_fake_loss: 1.3375 | g_loss: 0.4757
Iteration [ 1680/10000] | d_real_loss: 0.5391 | d_Y_loss: 1.0946 | d_X_loss:
0.3595 | d_fake_loss: 1.4541 | g_loss: 0.4407
Iteration [ 1690/10000] | d_real_loss: 0.5345 | d_Y_loss: 1.0342 | d_X_loss:
0.2831 | d_fake_loss: 1.3173 | g_loss: 0.4737
Iteration [ 1700/10000] | d_real_loss: 0.5233 | d_Y_loss: 1.0906 | d_X_loss:
0.2791 | d_fake_loss: 1.3698 | g_loss: 0.4606
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001700-
Y-X.png
Iteration [ 1710/10000] | d_real_loss: 0.5273 | d_Y_loss: 1.0708 | d_X_loss:
0.3219 | d_fake_loss: 1.3927 | g_loss: 0.4570
Iteration [ 1720/10000] | d_real_loss: 0.5528 | d_Y_loss: 1.0730 | d_X_loss:
0.2484 | d_fake_loss: 1.3214 | g_loss: 0.4506
Iteration [ 1730/10000] | d_real_loss: 0.6226 | d_Y_loss: 1.0559 | d_X_loss:
0.3321 | d_fake_loss: 1.3880 | g_loss: 0.4633
Iteration [ 1740/10000] | d_real_loss: 0.5340 | d_Y_loss: 1.0368 | d_X_loss:
0.3627 | d_fake_loss: 1.3995 | g_loss: 0.4787
Iteration [ 1750/10000] | d_real_loss: 0.4996 | d_Y_loss: 1.0883 | d_X_loss:
0.2927 | d_fake_loss: 1.3810 | g_loss: 0.4536
Iteration [ 1760/10000] | d_real_loss: 0.5030 | d_Y_loss: 1.0652 | d_X_loss:
```

```
0.3606 | d_fake_loss: 1.4258 | g_loss: 0.4593
Iteration [ 1770/10000] | d_real_loss: 0.5010 | d_Y_loss: 1.0290 | d_X_loss:
0.2268 | d_fake_loss: 1.2557 | g_loss: 0.4786
Iteration [ 1780/10000] | d_real_loss: 0.4992 | d_Y_loss: 1.1109 | d_X_loss:
0.2165 | d_fake_loss: 1.3274 | g_loss: 0.4513
Iteration [ 1790/10000] | d_real_loss: 0.5022 | d_Y_loss: 1.0189 | d_X_loss:
0.2092 | d_fake_loss: 1.2281 | g_loss: 0.4781
Iteration [ 1800/10000] | d_real_loss: 0.5182 | d_Y_loss: 1.0831 | d_X_loss:
0.2608 | d_fake_loss: 1.3439 | g_loss: 0.4577
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001800-
Y-X.png
Iteration [ 1810/10000] | d_real_loss: 0.5397 | d_Y_loss: 1.0530 | d_X_loss:
0.2501 | d_fake_loss: 1.3031 | g_loss: 0.4711
Iteration [ 1820/10000] | d_real_loss: 0.5042 | d_Y_loss: 1.0569 | d_X_loss:
0.6656 | d_fake_loss: 1.7225 | g_loss: 0.4637
Iteration [ 1830/10000] | d_real_loss: 0.5249 | d_Y_loss: 1.0641 | d_X_loss:
0.2436 | d_fake_loss: 1.3077 | g_loss: 0.4612
Iteration [ 1840/10000] | d_real_loss: 0.5113 | d_Y_loss: 1.1011 | d_X_loss:
0.2060 | d_fake_loss: 1.3071 | g_loss: 0.4538
Iteration [ 1850/10000] | d_real_loss: 0.4908 | d_Y_loss: 1.0529 | d_X_loss:
0.2027 | d_fake_loss: 1.2557 | g_loss: 0.4657
Iteration [ 1860/10000] | d_real_loss: 0.4913 | d_Y_loss: 1.0502 | d_X_loss:
0.1995 | d_fake_loss: 1.2498 | g_loss: 0.4637
Iteration [ 1870/10000] | d_real_loss: 0.5052 | d_Y_loss: 1.0527 | d_X_loss:
0.2411 | d_fake_loss: 1.2938 | g_loss: 0.4643
Iteration [ 1880/10000] | d_real_loss: 0.4851 | d_Y_loss: 1.1350 | d_X_loss:
0.2792 | d_fake_loss: 1.4142 | g_loss: 0.4453
Iteration [ 1890/10000] | d_real_loss: 0.4889 | d_Y_loss: 1.0357 | d_X_loss:
0.1882 | d_fake_loss: 1.2239 | g_loss: 0.4833
Iteration [ 1900/10000] | d_real_loss: 0.4915 | d_Y_loss: 1.0716 | d_X_loss:
0.1844 | d_fake_loss: 1.2560 | g_loss: 0.4644
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-001900-
Y-X.png
Iteration [ 1910/10000] | d_real_loss: 0.5001 | d_Y_loss: 1.0187 | d_X_loss:
0.2272 | d_fake_loss: 1.2458 | g_loss: 0.4759
Iteration [ 1920/10000] | d_real_loss: 0.5023 | d_Y_loss: 1.0599 | d_X_loss:
0.2488 | d_fake_loss: 1.3087 | g_loss: 0.4649
Iteration [ 1930/10000] | d_real_loss: 0.5788 | d_Y_loss: 1.0748 | d_X_loss:
0.2658 | d_fake_loss: 1.3406 | g_loss: 0.4589
Iteration [ 1940/10000] | d_real_loss: 0.5001 | d_Y_loss: 1.0849 | d_X_loss:
0.1965 | d_fake_loss: 1.2813 | g_loss: 0.4571
Iteration [ 1950/10000] | d_real_loss: 0.4951 | d_Y_loss: 1.0355 | d_X_loss:
0.2555 | d_fake_loss: 1.2909 | g_loss: 0.4703
Iteration [ 1960/10000] | d_real_loss: 0.4771 | d_Y_loss: 1.0819 | d_X_loss:
```

```
0.2282 | d_fake_loss: 1.3100 | g_loss: 0.4547
Iteration [ 1970/10000] | d_real_loss: 0.5091 | d_Y_loss: 1.0716 | d_X_loss:
0.2015 | d_fake_loss: 1.2731 | g_loss: 0.4578
Iteration [ 1980/10000] | d_real_loss: 0.5074 | d_Y_loss: 1.0443 | d_X_loss:
0.2028 | d_fake_loss: 1.2471 | g_loss: 0.4715
Iteration [ 1990/10000] | d_real_loss: 0.4865 | d_Y_loss: 1.0756 | d_X_loss:
0.2877 | d_fake_loss: 1.3632 | g_loss: 0.4623
Iteration [ 2000/10000] | d_real_loss: 0.4883 | d_Y_loss: 1.0685 | d_X_loss:
0.2669 | d_fake_loss: 1.3354 | g_loss: 0.4618
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002000-
Y-X.png
Iteration [ 2010/10000] | d_real_loss: 0.4829 | d_Y_loss: 1.0167 | d_X_loss:
0.2787 | d_fake_loss: 1.2954 | g_loss: 0.4788
Iteration [ 2020/10000] | d_real_loss: 0.5184 | d_Y_loss: 1.1350 | d_X_loss:
0.2901 | d_fake_loss: 1.4251 | g_loss: 0.4487
Iteration [ 2030/10000] | d_real_loss: 0.4853 | d_Y_loss: 1.0692 | d_X_loss:
0.1635 | d_fake_loss: 1.2328 | g_loss: 0.4637
Iteration [ 2040/10000] | d_real_loss: 0.4711 | d_Y_loss: 1.0799 | d_X_loss:
0.1655 | d_fake_loss: 1.2454 | g_loss: 0.4570
Iteration [ 2050/10000] | d_real_loss: 0.4605 | d_Y_loss: 1.0510 | d_X_loss:
0.2934 | d_fake_loss: 1.3445 | g_loss: 0.4644
Iteration [ 2060/10000] | d_real_loss: 0.4806 | d_Y_loss: 1.0321 | d_X_loss:
0.1961 | d_fake_loss: 1.2281 | g_loss: 0.4775
Iteration [ 2070/10000] | d_real_loss: 0.4679 | d_Y_loss: 1.0202 | d_X_loss:
0.1611 | d_fake_loss: 1.1813 | g_loss: 0.4887
Iteration [ 2080/10000] | d_real_loss: 0.4776 | d_Y_loss: 1.0409 | d_X_loss:
0.2018 | d_fake_loss: 1.2427 | g_loss: 0.4736
Iteration [ 2090/10000] | d_real_loss: 0.4803 | d_Y_loss: 1.0579 | d_X_loss:
0.8045 | d_fake_loss: 1.8624 | g_loss: 0.4599
Iteration [ 2100/10000] | d_real_loss: 0.5199 | d_Y_loss: 1.0812 | d_X_loss:
0.4727 | d_fake_loss: 1.5539 | g_loss: 0.4594
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002100-
Y-X.png
Iteration [ 2110/10000] | d_real_loss: 0.5092 | d_Y_loss: 1.0161 | d_X_loss:
0.2455 | d_fake_loss: 1.2615 | g_loss: 0.4807
Iteration [ 2120/10000] | d_real_loss: 0.4998 | d_Y_loss: 1.0561 | d_X_loss:
0.3887 | d_fake_loss: 1.4448 | g_loss: 0.4700
Iteration [ 2130/10000] | d_real_loss: 0.6667 | d_Y_loss: 1.0633 | d_X_loss:
0.4980 | d_fake_loss: 1.5614 | g_loss: 0.4824
Iteration [ 2140/10000] | d_real_loss: 0.5236 | d_Y_loss: 1.0528 | d_X_loss:
0.3100 | d_fake_loss: 1.3628 | g_loss: 0.4810
Iteration [ 2150/10000] | d_real_loss: 0.6128 | d_Y_loss: 1.0851 | d_X_loss:
0.2984 | d_fake_loss: 1.3835 | g_loss: 0.4706
Iteration [ 2160/10000] | d_real_loss: 0.5463 | d_Y_loss: 1.0980 | d_X_loss:
```

```
0.2864 | d_fake_loss: 1.3844 | g_loss: 0.4427
Iteration [ 2170/10000] | d_real_loss: 0.5626 | d_Y_loss: 1.0871 | d_X_loss:
0.1894 | d_fake_loss: 1.2765 | g_loss: 0.4583
Iteration [ 2180/10000] | d_real_loss: 0.4880 | d_Y_loss: 1.0670 | d_X_loss:
0.1624 | d_fake_loss: 1.2294 | g_loss: 0.4614
Iteration [ 2190/10000] | d_real_loss: 0.5363 | d_Y_loss: 1.0344 | d_X_loss:
0.1659 | d_fake_loss: 1.2003 | g_loss: 0.4793
Iteration [ 2200/10000] | d_real_loss: 0.4627 | d_Y_loss: 1.0676 | d_X_loss:
0.2161 | d_fake_loss: 1.2837 | g_loss: 0.4560
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002200-
Y-X.png
Iteration [ 2210/10000] | d_real_loss: 0.7714 | d_Y_loss: 1.0841 | d_X_loss:
0.2551 | d_fake_loss: 1.3392 | g_loss: 0.4529
Iteration [ 2220/10000] | d_real_loss: 0.4959 | d_Y_loss: 1.0265 | d_X_loss:
0.2162 | d_fake_loss: 1.2427 | g_loss: 0.4738
Iteration [ 2230/10000] | d_real_loss: 0.4739 | d_Y_loss: 1.0531 | d_X_loss:
0.1719 | d_fake_loss: 1.2250 | g_loss: 0.4693
Iteration [ 2240/10000] | d_real_loss: 0.5482 | d_Y_loss: 1.0491 | d_X_loss:
0.2197 | d_fake_loss: 1.2688 | g_loss: 0.4797
Iteration [ 2250/10000] | d_real_loss: 0.4845 | d_Y_loss: 1.0442 | d_X_loss:
0.2663 | d_fake_loss: 1.3105 | g_loss: 0.4728
Iteration [ 2260/10000] | d_real_loss: 0.4942 | d_Y_loss: 1.1431 | d_X_loss:
0.2763 | d_fake_loss: 1.4194 | g_loss: 0.4204
Iteration [ 2270/10000] | d_real_loss: 0.5034 | d_Y_loss: 1.0615 | d_X_loss:
0.2222 | d_fake_loss: 1.2837 | g_loss: 0.4659
Iteration [ 2280/10000] | d_real_loss: 0.4994 | d_Y_loss: 1.0567 | d_X_loss:
0.2147 | d_fake_loss: 1.2714 | g_loss: 0.4737
Iteration [ 2290/10000] | d_real_loss: 0.5016 | d_Y_loss: 1.0553 | d_X_loss:
0.4255 | d_fake_loss: 1.4808 | g_loss: 0.4658
Iteration [ 2300/10000] | d_real_loss: 0.4909 | d_Y_loss: 1.0779 | d_X_loss:
0.3630 | d_fake_loss: 1.4410 | g_loss: 0.4580
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002300-
Y-X.png
Iteration [ 2310/10000] | d_real_loss: 0.4753 | d_Y_loss: 1.0587 | d_X_loss:
0.1933 | d_fake_loss: 1.2521 | g_loss: 0.4634
Iteration [ 2320/10000] | d_real_loss: 0.5269 | d_Y_loss: 1.0526 | d_X_loss:
0.2366 | d_fake_loss: 1.2891 | g_loss: 0.4680
Iteration [ 2330/10000] | d_real_loss: 0.4829 | d_Y_loss: 1.0487 | d_X_loss:
0.2670 | d_fake_loss: 1.3157 | g_loss: 0.4693
Iteration [ 2340/10000] | d_real_loss: 0.4922 | d_Y_loss: 1.0514 | d_X_loss:
0.1745 | d_fake_loss: 1.2259 | g_loss: 0.4809
Iteration [ 2350/10000] | d_real_loss: 0.4750 | d_Y_loss: 1.0552 | d_X_loss:
0.3333 | d_fake_loss: 1.3884 | g_loss: 0.4675
Iteration [ 2360/10000] | d_real_loss: 0.5160 | d_Y_loss: 1.0371 | d_X_loss:
```

```
0.2055 | d_fake_loss: 1.2426 | g_loss: 0.4755
Iteration [ 2370/10000] | d_real_loss: 0.4713 | d_Y_loss: 1.1222 | d_X_loss:
0.2180 | d_fake_loss: 1.3402 | g_loss: 0.4601
Iteration [ 2380/10000] | d_real_loss: 0.4951 | d_Y_loss: 1.0339 | d_X_loss:
0.5956 | d_fake_loss: 1.6295 | g_loss: 0.4720
Iteration [ 2390/10000] | d_real_loss: 0.4773 | d_Y_loss: 1.1131 | d_X_loss:
0.1745 | d_fake_loss: 1.2876 | g_loss: 0.4659
Iteration [ 2400/10000] | d_real_loss: 0.4565 | d_Y_loss: 1.0843 | d_X_loss:
0.1758 | d_fake_loss: 1.2601 | g_loss: 0.4685
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002400-
Y-X.png
Iteration [ 2410/10000] | d_real_loss: 0.4804 | d_Y_loss: 1.0156 | d_X_loss:
0.2360 | d_fake_loss: 1.2516 | g_loss: 0.4838
Iteration [ 2420/10000] | d_real_loss: 0.6491 | d_Y_loss: 1.0196 | d_X_loss:
0.1607 | d_fake_loss: 1.1803 | g_loss: 0.4969
Iteration [ 2430/10000] | d_real_loss: 0.4693 | d_Y_loss: 0.9886 | d_X_loss:
0.4146 | d_fake_loss: 1.4032 | g_loss: 0.5025
Iteration [ 2440/10000] | d_real_loss: 0.4717 | d_Y_loss: 1.0267 | d_X_loss:
0.2443 | d_fake_loss: 1.2710 | g_loss: 0.4760
Iteration [ 2450/10000] | d_real_loss: 0.4895 | d_Y_loss: 1.0374 | d_X_loss:
0.2169 | d_fake_loss: 1.2543 | g_loss: 0.4803
Iteration [ 2460/10000] | d_real_loss: 0.4624 | d_Y_loss: 1.0571 | d_X_loss:
0.1878 | d_fake_loss: 1.2449 | g_loss: 0.4699
Iteration [ 2470/10000] | d_real_loss: 0.5582 | d_Y_loss: 1.0781 | d_X_loss:
0.2039 | d_fake_loss: 1.2820 | g_loss: 0.4610
Iteration [ 2480/10000] | d_real_loss: 0.4403 | d_Y_loss: 1.0231 | d_X_loss:
0.2283 | d_fake_loss: 1.2514 | g_loss: 0.4862
Iteration [ 2490/10000] | d_real_loss: 0.4587 | d_Y_loss: 1.0125 | d_X_loss:
0.1641 | d_fake_loss: 1.1766 | g_loss: 0.4807
Iteration [ 2500/10000] | d_real_loss: 0.4840 | d_Y_loss: 1.0381 | d_X_loss:
0.1695 | d_fake_loss: 1.2076 | g_loss: 0.4710
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002500-
Y-X.png
Iteration [ 2510/10000] | d_real_loss: 0.5068 | d_Y_loss: 1.0866 | d_X_loss:
0.2087 | d_fake_loss: 1.2952 | g_loss: 0.4645
Iteration [ 2520/10000] | d_real_loss: 0.4845 | d_Y_loss: 1.0532 | d_X_loss:
0.2652 | d_fake_loss: 1.3184 | g_loss: 0.4686
Iteration [ 2530/10000] | d_real_loss: 0.5305 | d_Y_loss: 1.0636 | d_X_loss:
0.2185 | d_fake_loss: 1.2821 | g_loss: 0.4651
Iteration [ 2540/10000] | d_real_loss: 0.7231 | d_Y_loss: 1.0562 | d_X_loss:
0.2153 | d_fake_loss: 1.2715 | g_loss: 0.4778
Iteration [ 2550/10000] | d_real_loss: 0.4598 | d_Y_loss: 1.0408 | d_X_loss:
0.1751 | d_fake_loss: 1.2159 | g_loss: 0.4769
Iteration [ 2560/10000] | d_real_loss: 0.4337 | d_Y_loss: 1.0718 | d_X_loss:
```

```
0.1331 | d_fake_loss: 1.2049 | g_loss: 0.4712
Iteration [ 2570/10000] | d_real_loss: 0.5458 | d_Y_loss: 1.0271 | d_X_loss:
0.2050 | d_fake_loss: 1.2321 | g_loss: 0.4922
Iteration [ 2580/10000] | d_real_loss: 0.5431 | d_Y_loss: 1.0760 | d_X_loss:
0.1751 | d_fake_loss: 1.2511 | g_loss: 0.4621
Iteration [ 2590/10000] | d_real_loss: 0.5759 | d_Y_loss: 1.0065 | d_X_loss:
0.1427 | d_fake_loss: 1.1492 | g_loss: 0.4924
Iteration [ 2600/10000] | d_real_loss: 0.4984 | d_Y_loss: 1.0033 | d_X_loss:
0.6055 | d_fake_loss: 1.6088 | g_loss: 0.4945
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002600-
Y-X.png
Iteration [ 2610/10000] | d_real_loss: 0.5391 | d_Y_loss: 0.9678 | d_X_loss:
0.2357 | d_fake_loss: 1.2035 | g_loss: 0.5065
Iteration [ 2620/10000] | d_real_loss: 0.4374 | d_Y_loss: 1.1595 | d_X_loss:
0.3515 | d_fake_loss: 1.5110 | g_loss: 0.4388
Iteration [ 2630/10000] | d_real_loss: 0.5345 | d_Y_loss: 1.0723 | d_X_loss:
0.3739 | d_fake_loss: 1.4462 | g_loss: 0.4612
Iteration [ 2640/10000] | d_real_loss: 0.5821 | d_Y_loss: 1.0301 | d_X_loss:
0.1700 | d_fake_loss: 1.2001 | g_loss: 0.4796
Iteration [ 2650/10000] | d_real_loss: 0.4935 | d_Y_loss: 1.0207 | d_X_loss:
0.3895 | d_fake_loss: 1.4103 | g_loss: 0.4864
Iteration [ 2660/10000] | d_real_loss: 0.4690 | d_Y_loss: 1.0506 | d_X_loss:
0.4601 | d_fake_loss: 1.5107 | g_loss: 0.4778
Iteration [ 2670/10000] | d_real_loss: 0.6327 | d_Y_loss: 0.9991 | d_X_loss:
0.4055 | d_fake_loss: 1.4046 | g_loss: 0.4940
Iteration [ 2680/10000] | d_real_loss: 0.4427 | d_Y_loss: 1.0653 | d_X_loss:
0.8167 | d_fake_loss: 1.8819 | g_loss: 0.4824
Iteration [ 2690/10000] | d_real_loss: 0.6370 | d_Y_loss: 1.0341 | d_X_loss:
0.2807 | d_fake_loss: 1.3148 | g_loss: 0.4836
Iteration [ 2700/10000] | d_real_loss: 0.6452 | d_Y_loss: 1.1039 | d_X_loss:
0.2699 | d_fake_loss: 1.3738 | g_loss: 0.4495
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002700-
Y-X.png
Iteration [ 2710/10000] | d_real_loss: 0.4823 | d_Y_loss: 1.0383 | d_X_loss:
0.3884 | d_fake_loss: 1.4267 | g_loss: 0.4715
Iteration [ 2720/10000] | d_real_loss: 0.5013 | d_Y_loss: 0.9994 | d_X_loss:
0.5982 | d_fake_loss: 1.5977 | g_loss: 0.4953
Iteration [ 2730/10000] | d_real_loss: 0.4949 | d_Y_loss: 1.0362 | d_X_loss:
0.4509 | d_fake_loss: 1.4871 | g_loss: 0.4860
Iteration [ 2740/10000] | d_real_loss: 0.4902 | d_Y_loss: 1.0344 | d_X_loss:
0.2582 | d_fake_loss: 1.2926 | g_loss: 0.4821
Iteration [ 2750/10000] | d_real_loss: 0.5691 | d_Y_loss: 1.0265 | d_X_loss:
0.1864 | d_fake_loss: 1.2129 | g_loss: 0.4849
Iteration [ 2760/10000] | d_real_loss: 0.5009 | d_Y_loss: 1.0468 | d_X_loss:
```

```
0.3172 | d_fake_loss: 1.3640 | g_loss: 0.4852
Iteration [ 2770/10000] | d_real_loss: 0.5315 | d_Y_loss: 0.9981 | d_X_loss:
0.7056 | d_fake_loss: 1.7037 | g_loss: 0.4946
Iteration [ 2780/10000] | d_real_loss: 0.6505 | d_Y_loss: 0.9983 | d_X_loss:
0.3062 | d_fake_loss: 1.3045 | g_loss: 0.5198
Iteration [ 2790/10000] | d_real_loss: 0.6360 | d_Y_loss: 0.9769 | d_X_loss:
0.3295 | d_fake_loss: 1.3065 | g_loss: 0.4992
Iteration [ 2800/10000] | d_real_loss: 0.4488 | d_Y_loss: 1.1300 | d_X_loss:
0.1921 | d_fake_loss: 1.3221 | g_loss: 0.4439
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002800-
Y-X.png
Iteration [ 2810/10000] | d_real_loss: 0.5768 | d_Y_loss: 1.1045 | d_X_loss:
0.2569 | d_fake_loss: 1.3614 | g_loss: 0.4421
Iteration [ 2820/10000] | d_real_loss: 0.5081 | d_Y_loss: 1.0845 | d_X_loss:
0.3610 | d_fake_loss: 1.4455 | g_loss: 0.5053
Iteration [ 2830/10000] | d_real_loss: 0.6959 | d_Y_loss: 1.0948 | d_X_loss:
0.4320 | d_fake_loss: 1.5268 | g_loss: 0.4566
Iteration [ 2840/10000] | d_real_loss: 0.7228 | d_Y_loss: 1.0574 | d_X_loss:
0.7307 | d_fake_loss: 1.7881 | g_loss: 0.4719
Iteration [ 2850/10000] | d_real_loss: 0.6637 | d_Y_loss: 1.0400 | d_X_loss:
0.4735 | d_fake_loss: 1.5135 | g_loss: 0.4772
Iteration [ 2860/10000] | d_real_loss: 0.7189 | d_Y_loss: 1.0402 | d_X_loss:
0.5790 | d_fake_loss: 1.6192 | g_loss: 0.4954
Iteration [ 2870/10000] | d_real_loss: 0.5887 | d_Y_loss: 1.0040 | d_X_loss:
0.4601 | d_fake_loss: 1.4641 | g_loss: 0.4937
Iteration [ 2880/10000] | d_real_loss: 0.4861 | d_Y_loss: 1.0658 | d_X_loss:
0.7345 | d_fake_loss: 1.8002 | g_loss: 0.4790
Iteration [ 2890/10000] | d_real_loss: 0.8498 | d_Y_loss: 1.0581 | d_X_loss:
0.3106 | d_fake_loss: 1.3688 | g_loss: 0.4654
Iteration [ 2900/10000] | d_real_loss: 0.5195 | d_Y_loss: 1.0412 | d_X_loss:
0.7256 | d_fake_loss: 1.7668 | g_loss: 0.4781
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-002900-
Y-X.png
Iteration [ 2910/10000] | d_real_loss: 0.5084 | d_Y_loss: 1.0277 | d_X_loss:
0.2932 | d_fake_loss: 1.3208 | g_loss: 0.4888
Iteration [ 2920/10000] | d_real_loss: 0.5038 | d_Y_loss: 1.1095 | d_X_loss:
0.4253 | d_fake_loss: 1.5347 | g_loss: 0.4833
Iteration [ 2930/10000] | d_real_loss: 0.6098 | d_Y_loss: 1.0057 | d_X_loss:
0.3692 | d_fake_loss: 1.3749 | g_loss: 0.5055
Iteration [ 2940/10000] | d_real_loss: 0.4992 | d_Y_loss: 1.1711 | d_X_loss:
0.2726 | d_fake_loss: 1.4437 | g_loss: 0.4264
Iteration [ 2950/10000] | d_real_loss: 0.5099 | d_Y_loss: 1.0337 | d_X_loss:
0.2320 | d_fake_loss: 1.2656 | g_loss: 0.4804
Iteration [ 2960/10000] | d_real_loss: 0.5231 | d_Y_loss: 1.0009 | d_X_loss:
```

```
0.1901 | d_fake_loss: 1.1910 | g_loss: 0.4964
Iteration [ 2970/10000] | d_real_loss: 0.4979 | d_Y_loss: 1.0489 | d_X_loss:
0.3092 | d_fake_loss: 1.3581 | g_loss: 0.4697
Iteration [ 2980/10000] | d_real_loss: 0.5250 | d_Y_loss: 0.9969 | d_X_loss:
0.5737 | d_fake_loss: 1.5706 | g_loss: 0.4985
Iteration [ 2990/10000] | d_real_loss: 0.4794 | d_Y_loss: 1.0532 | d_X_loss:
0.2415 | d_fake_loss: 1.2947 | g_loss: 0.4955
Iteration [ 3000/10000] | d_real_loss: 0.5151 | d_Y_loss: 1.0584 | d_X_loss:
0.5055 | d_fake_loss: 1.5640 | g_loss: 0.4709
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003000-
Y-X.png
Iteration [ 3010/10000] | d_real_loss: 0.5251 | d_Y_loss: 1.0241 | d_X_loss:
0.2367 | d_fake_loss: 1.2608 | g_loss: 0.4884
Iteration [ 3020/10000] | d_real_loss: 0.6291 | d_Y_loss: 1.0178 | d_X_loss:
0.2810 | d_fake_loss: 1.2988 | g_loss: 0.4977
Iteration [ 3030/10000] | d_real_loss: 0.7417 | d_Y_loss: 1.0220 | d_X_loss:
0.2352 | d_fake_loss: 1.2572 | g_loss: 0.4899
Iteration [ 3040/10000] | d_real_loss: 0.5126 | d_Y_loss: 1.0183 | d_X_loss:
0.3174 | d_fake_loss: 1.3357 | g_loss: 0.4842
Iteration [ 3050/10000] | d_real_loss: 0.6013 | d_Y_loss: 1.0177 | d_X_loss:
1.1370 | d_fake_loss: 2.1547 | g_loss: 0.4995
Iteration [ 3060/10000] | d_real_loss: 0.5493 | d_Y_loss: 1.0664 | d_X_loss:
0.4957 | d_fake_loss: 1.5621 | g_loss: 0.4700
Iteration [ 3070/10000] | d_real_loss: 0.5656 | d_Y_loss: 1.0481 | d_X_loss:
0.6500 | d_fake_loss: 1.6982 | g_loss: 0.4838
Iteration [ 3080/10000] | d_real_loss: 0.5841 | d_Y_loss: 0.9933 | d_X_loss:
0.6421 | d_fake_loss: 1.6354 | g_loss: 0.5055
Iteration [ 3090/10000] | d_real_loss: 0.5683 | d_Y_loss: 1.0648 | d_X_loss:
0.3748 | d_fake_loss: 1.4396 | g_loss: 0.4841
Iteration [ 3100/10000] | d_real_loss: 0.6127 | d_Y_loss: 1.0588 | d_X_loss:
0.3367 | d_fake_loss: 1.3955 | g_loss: 0.4839
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003100-
Y-X.png
Iteration [ 3110/10000] | d_real_loss: 0.5202 | d_Y_loss: 1.0152 | d_X_loss:
0.4317 | d_fake_loss: 1.4469 | g_loss: 0.4916
Iteration [ 3120/10000] | d_real_loss: 0.5406 | d_Y_loss: 1.1114 | d_X_loss:
0.3636 | d_fake_loss: 1.4749 | g_loss: 0.4847
Iteration [ 3130/10000] | d_real_loss: 0.9243 | d_Y_loss: 1.0906 | d_X_loss:
0.3028 | d_fake_loss: 1.3934 | g_loss: 0.4591
Iteration [ 3140/10000] | d_real_loss: 0.5310 | d_Y_loss: 1.0090 | d_X_loss:
0.3472 | d_fake_loss: 1.3562 | g_loss: 0.4952
Iteration [ 3150/10000] | d_real_loss: 0.5520 | d_Y_loss: 1.0023 | d_X_loss:
0.4454 | d_fake_loss: 1.4477 | g_loss: 0.5072
Iteration [ 3160/10000] | d_real_loss: 0.5566 | d_Y_loss: 1.0394 | d_X_loss:
```

```
0.4585 | d_fake_loss: 1.4979 | g_loss: 0.4822
Iteration [ 3170/10000] | d_real_loss: 0.5167 | d_Y_loss: 1.0144 | d_X_loss:
0.3381 | d_fake_loss: 1.3525 | g_loss: 0.5022
Iteration [ 3180/10000] | d_real_loss: 0.6373 | d_Y_loss: 1.0132 | d_X_loss:
0.5706 | d_fake_loss: 1.5837 | g_loss: 0.4903
Iteration [ 3190/10000] | d_real_loss: 0.5191 | d_Y_loss: 1.0138 | d_X_loss:
0.4846 | d_fake_loss: 1.4984 | g_loss: 0.5005
Iteration [ 3200/10000] | d_real_loss: 0.4553 | d_Y_loss: 1.0984 | d_X_loss:
0.6719 | d_fake_loss: 1.7703 | g_loss: 0.4910
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003200-
Y-X.png
Iteration [ 3210/10000] | d_real_loss: 0.5203 | d_Y_loss: 1.0855 | d_X_loss:
0.4555 | d_fake_loss: 1.5411 | g_loss: 0.4704
Iteration [ 3220/10000] | d_real_loss: 0.5710 | d_Y_loss: 1.0053 | d_X_loss:
0.4320 | d_fake_loss: 1.4372 | g_loss: 0.5091
Iteration [ 3230/10000] | d_real_loss: 0.5149 | d_Y_loss: 1.0320 | d_X_loss:
0.2497 | d_fake_loss: 1.2818 | g_loss: 0.4940
Iteration [ 3240/10000] | d_real_loss: 0.5055 | d_Y_loss: 1.1029 | d_X_loss:
0.4042 | d_fake_loss: 1.5071 | g_loss: 0.4939
Iteration [ 3250/10000] | d_real_loss: 0.5190 | d_Y_loss: 1.0047 | d_X_loss:
0.4631 | d_fake_loss: 1.4679 | g_loss: 0.5008
Iteration [ 3260/10000] | d_real_loss: 0.5212 | d_Y_loss: 1.2426 | d_X_loss:
0.5397 | d_fake_loss: 1.7823 | g_loss: 0.4078
Iteration [ 3270/10000] | d_real_loss: 0.5855 | d_Y_loss: 1.0301 | d_X_loss:
0.5380 | d_fake_loss: 1.5682 | g_loss: 0.4882
Iteration [ 3280/10000] | d_real_loss: 0.5451 | d_Y_loss: 1.0458 | d_X_loss:
0.3207 | d_fake_loss: 1.3665 | g_loss: 0.4769
Iteration [ 3290/10000] | d_real_loss: 0.7370 | d_Y_loss: 0.9781 | d_X_loss:
0.3715 | d_fake_loss: 1.3496 | g_loss: 0.5043
Iteration [ 3300/10000] | d_real_loss: 0.5385 | d_Y_loss: 1.0866 | d_X_loss:
0.8536 | d_fake_loss: 1.9402 | g_loss: 0.4628
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003300-
Y-X.png
Iteration [ 3310/10000] | d_real_loss: 0.4998 | d_Y_loss: 1.1424 | d_X_loss:
0.4335 | d_fake_loss: 1.5758 | g_loss: 0.4687
Iteration [ 3320/10000] | d_real_loss: 0.5914 | d_Y_loss: 0.9874 | d_X_loss:
0.6465 | d_fake_loss: 1.6339 | g_loss: 0.5203
Iteration [ 3330/10000] | d_real_loss: 0.5728 | d_Y_loss: 1.1276 | d_X_loss:
0.2792 | d_fake_loss: 1.4068 | g_loss: 0.4666
Iteration [ 3340/10000] | d_real_loss: 0.7102 | d_Y_loss: 1.0675 | d_X_loss:
0.4178 | d_fake_loss: 1.4853 | g_loss: 0.4598
Iteration [ 3350/10000] | d_real_loss: 0.6159 | d_Y_loss: 1.0581 | d_X_loss:
0.5228 | d_fake_loss: 1.5809 | g_loss: 0.4847
Iteration [ 3360/10000] | d_real_loss: 0.6585 | d_Y_loss: 0.9728 | d_X_loss:
```

```
0.5940 | d_fake_loss: 1.5668 | g_loss: 0.5133
Iteration [ 3370/10000] | d_real_loss: 0.7481 | d_Y_loss: 1.0276 | d_X_loss:
0.2934 | d_fake_loss: 1.3210 | g_loss: 0.4853
Iteration [ 3380/10000] | d_real_loss: 0.7547 | d_Y_loss: 1.0815 | d_X_loss:
0.3832 | d_fake_loss: 1.4646 | g_loss: 0.4586
Iteration [ 3390/10000] | d_real_loss: 0.5119 | d_Y_loss: 1.0345 | d_X_loss:
0.9327 | d_fake_loss: 1.9672 | g_loss: 0.5021
Iteration [ 3400/10000] | d_real_loss: 0.5761 | d_Y_loss: 1.0302 | d_X_loss:
0.7435 | d_fake_loss: 1.7737 | g_loss: 0.4968
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003400-
Y-X.png
Iteration [ 3410/10000] | d_real_loss: 0.5710 | d_Y_loss: 0.9971 | d_X_loss:
0.3943 | d_fake_loss: 1.3914 | g_loss: 0.5094
Iteration [ 3420/10000] | d_real_loss: 0.9151 | d_Y_loss: 1.0362 | d_X_loss:
0.5975 | d_fake_loss: 1.6336 | g_loss: 0.4785
Iteration [ 3430/10000] | d_real_loss: 0.6116 | d_Y_loss: 1.0306 | d_X_loss:
0.5385 | d_fake_loss: 1.5691 | g_loss: 0.4854
Iteration [ 3440/10000] | d_real_loss: 0.6644 | d_Y_loss: 0.9566 | d_X_loss:
0.4831 | d_fake_loss: 1.4396 | g_loss: 0.5189
Iteration [ 3450/10000] | d_real_loss: 0.8112 | d_Y_loss: 1.0885 | d_X_loss:
0.5029 | d_fake_loss: 1.5915 | g_loss: 0.4756
Iteration [ 3460/10000] | d_real_loss: 0.5267 | d_Y_loss: 1.0187 | d_X_loss:
1.0702 | d_fake_loss: 2.0889 | g_loss: 0.5034
Iteration [ 3470/10000] | d_real_loss: 0.7329 | d_Y_loss: 1.0468 | d_X_loss:
0.6885 | d_fake_loss: 1.7353 | g_loss: 0.4890
Iteration [ 3480/10000] | d_real_loss: 0.5652 | d_Y_loss: 1.1205 | d_X_loss:
0.4110 | d_fake_loss: 1.5315 | g_loss: 0.4641
Iteration [ 3490/10000] | d_real_loss: 0.6197 | d_Y_loss: 1.0158 | d_X_loss:
0.4533 | d_fake_loss: 1.4691 | g_loss: 0.4921
Iteration [ 3500/10000] | d_real_loss: 0.6309 | d_Y_loss: 1.0225 | d_X_loss:
0.4238 | d_fake_loss: 1.4463 | g_loss: 0.4849
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003500-
Y-X.png
Iteration [ 3510/10000] | d_real_loss: 0.6425 | d_Y_loss: 1.0187 | d_X_loss:
0.4096 | d_fake_loss: 1.4283 | g_loss: 0.5097
Iteration [ 3520/10000] | d_real_loss: 0.5452 | d_Y_loss: 1.0657 | d_X_loss:
0.5494 | d_fake_loss: 1.6151 | g_loss: 0.4732
Iteration [ 3530/10000] | d_real_loss: 0.5919 | d_Y_loss: 1.0485 | d_X_loss:
0.2942 | d_fake_loss: 1.3427 | g_loss: 0.4680
Iteration [ 3540/10000] | d_real_loss: 0.6352 | d_Y_loss: 1.0270 | d_X_loss:
0.4305 | d_fake_loss: 1.4575 | g_loss: 0.4918
Iteration [ 3550/10000] | d_real_loss: 0.6705 | d_Y_loss: 1.0335 | d_X_loss:
0.7526 | d_fake_loss: 1.7861 | g_loss: 0.4828
Iteration [ 3560/10000] | d_real_loss: 0.6167 | d_Y_loss: 1.0303 | d_X_loss:
```

```
0.3924 | d_fake_loss: 1.4227 | g_loss: 0.4819
Iteration [ 3570/10000] | d_real_loss: 0.5923 | d_Y_loss: 1.0391 | d_X_loss:
0.4667 | d_fake_loss: 1.5058 | g_loss: 0.4748
Iteration [ 3580/10000] | d_real_loss: 0.6575 | d_Y_loss: 1.0292 | d_X_loss:
0.2859 | d_fake_loss: 1.3150 | g_loss: 0.4948
Iteration [ 3590/10000] | d_real_loss: 0.4971 | d_Y_loss: 1.0795 | d_X_loss:
0.3453 | d_fake_loss: 1.4247 | g_loss: 0.4854
Iteration [ 3600/10000] | d_real_loss: 0.6821 | d_Y_loss: 0.9826 | d_X_loss:
0.4786 | d_fake_loss: 1.4611 | g_loss: 0.5093
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003600-
Y-X.png
Iteration [ 3610/10000] | d_real_loss: 0.4628 | d_Y_loss: 1.0750 | d_X_loss:
0.7782 | d_fake_loss: 1.8532 | g_loss: 0.4869
Iteration [ 3620/10000] | d_real_loss: 0.5439 | d_Y_loss: 1.0891 | d_X_loss:
0.5997 | d_fake_loss: 1.6888 | g_loss: 0.4635
Iteration [ 3630/10000] | d_real_loss: 0.6364 | d_Y_loss: 1.0643 | d_X_loss:
0.3895 | d_fake_loss: 1.4538 | g_loss: 0.4639
Iteration [ 3640/10000] | d_real_loss: 0.6573 | d_Y_loss: 1.0425 | d_X_loss:
0.5702 | d_fake_loss: 1.6127 | g_loss: 0.4725
Iteration [ 3650/10000] | d_real_loss: 0.5961 | d_Y_loss: 1.0180 | d_X_loss:
0.4291 | d_fake_loss: 1.4471 | g_loss: 0.4960
Iteration [ 3660/10000] | d_real_loss: 0.6416 | d_Y_loss: 1.0025 | d_X_loss:
0.3995 | d_fake_loss: 1.4019 | g_loss: 0.5133
Iteration [ 3670/10000] | d_real_loss: 0.5932 | d_Y_loss: 1.0003 | d_X_loss:
0.4634 | d_fake_loss: 1.4637 | g_loss: 0.4950
Iteration [ 3680/10000] | d_real_loss: 0.5602 | d_Y_loss: 1.0479 | d_X_loss:
0.5473 | d_fake_loss: 1.5952 | g_loss: 0.4821
Iteration [ 3690/10000] | d_real_loss: 0.6025 | d_Y_loss: 1.0416 | d_X_loss:
0.4202 | d_fake_loss: 1.4618 | g_loss: 0.4848
Iteration [ 3700/10000] | d_real_loss: 0.5429 | d_Y_loss: 1.0442 | d_X_loss:
0.6735 | d_fake_loss: 1.7177 | g_loss: 0.4916
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003700-
Y-X.png
Iteration [ 3710/10000] | d_real_loss: 0.5709 | d_Y_loss: 1.0625 | d_X_loss:
0.4454 | d_fake_loss: 1.5079 | g_loss: 0.4834
Iteration [ 3720/10000] | d_real_loss: 0.6788 | d_Y_loss: 1.0174 | d_X_loss:
0.6310 | d_fake_loss: 1.6484 | g_loss: 0.4967
Iteration [ 3730/10000] | d_real_loss: 0.5873 | d_Y_loss: 1.0610 | d_X_loss:
0.3676 | d_fake_loss: 1.4287 | g_loss: 0.4776
Iteration [ 3740/10000] | d_real_loss: 0.7547 | d_Y_loss: 1.0180 | d_X_loss:
0.3962 | d_fake_loss: 1.4142 | g_loss: 0.4963
Iteration [ 3750/10000] | d_real_loss: 0.5171 | d_Y_loss: 0.9970 | d_X_loss:
1.1539 | d_fake_loss: 2.1509 | g_loss: 0.5172
Iteration [ 3760/10000] | d_real_loss: 0.6273 | d_Y_loss: 1.0462 | d_X_loss:
```

```
0.6780 | d_fake_loss: 1.7242 | g_loss: 0.4925
Iteration [ 3770/10000] | d_real_loss: 0.5705 | d_Y_loss: 1.0451 | d_X_loss:
0.4273 | d_fake_loss: 1.4724 | g_loss: 0.4938
Iteration [ 3780/10000] | d_real_loss: 0.5740 | d_Y_loss: 1.0391 | d_X_loss:
0.4454 | d_fake_loss: 1.4845 | g_loss: 0.4781
Iteration [ 3790/10000] | d_real_loss: 0.7557 | d_Y_loss: 1.1380 | d_X_loss:
0.5014 | d_fake_loss: 1.6394 | g_loss: 0.4500
Iteration [ 3800/10000] | d_real_loss: 0.5359 | d_Y_loss: 1.0072 | d_X_loss:
0.7285 | d_fake_loss: 1.7357 | g_loss: 0.5033
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003800-
Y-X.png
Iteration [ 3810/10000] | d_real_loss: 0.6597 | d_Y_loss: 1.0243 | d_X_loss:
0.3955 | d_fake_loss: 1.4198 | g_loss: 0.4856
Iteration [ 3820/10000] | d_real_loss: 0.4985 | d_Y_loss: 1.0265 | d_X_loss:
0.6190 | d_fake_loss: 1.6455 | g_loss: 0.4876
Iteration [ 3830/10000] | d_real_loss: 0.6478 | d_Y_loss: 0.9796 | d_X_loss:
0.9912 | d_fake_loss: 1.9708 | g_loss: 0.5025
Iteration [ 3840/10000] | d_real_loss: 0.6766 | d_Y_loss: 1.0549 | d_X_loss:
0.5561 | d_fake_loss: 1.6110 | g_loss: 0.5007
Iteration [ 3850/10000] | d_real_loss: 0.5543 | d_Y_loss: 1.0753 | d_X_loss:
0.5868 | d_fake_loss: 1.6620 | g_loss: 0.4705
Iteration [ 3860/10000] | d_real_loss: 0.6181 | d_Y_loss: 0.9717 | d_X_loss:
0.4829 | d_fake_loss: 1.4547 | g_loss: 0.5298
Iteration [ 3870/10000] | d_real_loss: 0.5803 | d_Y_loss: 1.1276 | d_X_loss:
0.5484 | d_fake_loss: 1.6759 | g_loss: 0.4428
Iteration [ 3880/10000] | d_real_loss: 0.8027 | d_Y_loss: 1.0429 | d_X_loss:
0.3380 | d_fake_loss: 1.3810 | g_loss: 0.4785
Iteration [ 3890/10000] | d_real_loss: 0.5442 | d_Y_loss: 0.9993 | d_X_loss:
0.2910 | d_fake_loss: 1.2903 | g_loss: 0.5113
Iteration [ 3900/10000] | d_real_loss: 0.6677 | d_Y_loss: 0.9687 | d_X_loss:
0.7409 | d_fake_loss: 1.7096 | g_loss: 0.5080
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-003900-
Y-X.png
Iteration [ 3910/10000] | d_real_loss: 0.5946 | d_Y_loss: 1.0699 | d_X_loss:
0.4636 | d_fake_loss: 1.5335 | g_loss: 0.4859
Iteration [ 3920/10000] | d_real_loss: 0.7411 | d_Y_loss: 1.0101 | d_X_loss:
0.7231 | d_fake_loss: 1.7332 | g_loss: 0.5033
Iteration [ 3930/10000] | d_real_loss: 0.6048 | d_Y_loss: 1.1530 | d_X_loss:
0.5739 | d_fake_loss: 1.7269 | g_loss: 0.4602
Iteration [ 3940/10000] | d_real_loss: 0.6309 | d_Y_loss: 1.0675 | d_X_loss:
0.7371 | d_fake_loss: 1.8046 | g_loss: 0.4795
Iteration [ 3950/10000] | d_real_loss: 0.6447 | d_Y_loss: 0.9994 | d_X_loss:
0.5635 | d_fake_loss: 1.5629 | g_loss: 0.5151
Iteration [ 3960/10000] | d_real_loss: 0.7632 | d_Y_loss: 1.0296 | d_X_loss:
```

```
0.4077 | d_fake_loss: 1.4373 | g_loss: 0.5088
Iteration [ 3970/10000] | d_real_loss: 0.8379 | d_Y_loss: 0.9999 | d_X_loss:
0.6975 | d_fake_loss: 1.6974 | g_loss: 0.5018
Iteration [ 3980/10000] | d_real_loss: 0.6924 | d_Y_loss: 1.0464 | d_X_loss:
0.7416 | d_fake_loss: 1.7880 | g_loss: 0.4782
Iteration [ 3990/10000] | d_real_loss: 0.6479 | d_Y_loss: 1.0056 | d_X_loss:
0.5996 | d_fake_loss: 1.6052 | g_loss: 0.5057
Iteration [ 4000/10000] | d_real_loss: 0.6151 | d_Y_loss: 1.0482 | d_X_loss:
0.8555 | d_fake_loss: 1.9037 | g_loss: 0.4732
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004000-
Y-X.png
Iteration [ 4010/10000] | d_real_loss: 0.6086 | d_Y_loss: 0.9992 | d_X_loss:
0.8133 | d_fake_loss: 1.8125 | g_loss: 0.4993
Iteration [ 4020/10000] | d_real_loss: 0.8031 | d_Y_loss: 1.0843 | d_X_loss:
0.3790 | d_fake_loss: 1.4633 | g_loss: 0.4575
Iteration [ 4030/10000] | d_real_loss: 0.5623 | d_Y_loss: 1.0802 | d_X_loss:
0.5106 | d_fake_loss: 1.5908 | g_loss: 0.4804
Iteration [ 4040/10000] | d_real_loss: 0.6346 | d_Y_loss: 1.0105 | d_X_loss:
0.5680 | d_fake_loss: 1.5786 | g_loss: 0.5058
Iteration [ 4050/10000] | d_real_loss: 0.4797 | d_Y_loss: 1.0616 | d_X_loss:
0.4682 | d_fake_loss: 1.5298 | g_loss: 0.4842
Iteration [ 4060/10000] | d_real_loss: 0.5834 | d_Y_loss: 1.0079 | d_X_loss:
0.8223 | d_fake_loss: 1.8302 | g_loss: 0.4920
Iteration [ 4070/10000] | d_real_loss: 0.5440 | d_Y_loss: 1.0104 | d_X_loss:
0.4378 | d_fake_loss: 1.4482 | g_loss: 0.5101
Iteration [ 4080/10000] | d_real_loss: 0.6038 | d_Y_loss: 1.0976 | d_X_loss:
0.7183 | d_fake_loss: 1.8158 | g_loss: 0.4518
Iteration [ 4090/10000] | d_real_loss: 0.8074 | d_Y_loss: 1.1213 | d_X_loss:
0.5313 | d_fake_loss: 1.6526 | g_loss: 0.4489
Iteration [ 4100/10000] | d_real_loss: 0.6119 | d_Y_loss: 0.9998 | d_X_loss:
0.4803 | d_fake_loss: 1.4802 | g_loss: 0.5088
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004100-
Y-X.png
Iteration [ 4110/10000] | d_real_loss: 0.6244 | d_Y_loss: 1.0534 | d_X_loss:
0.8415 | d_fake_loss: 1.8949 | g_loss: 0.5190
Iteration [ 4120/10000] | d_real_loss: 0.5818 | d_Y_loss: 1.0101 | d_X_loss:
0.4899 | d_fake_loss: 1.5000 | g_loss: 0.4993
Iteration [ 4130/10000] | d_real_loss: 0.5705 | d_Y_loss: 1.0065 | d_X_loss:
0.4605 | d_fake_loss: 1.4670 | g_loss: 0.5170
Iteration [ 4140/10000] | d_real_loss: 0.7515 | d_Y_loss: 1.0396 | d_X_loss:
0.4888 | d_fake_loss: 1.5284 | g_loss: 0.4790
Iteration [ 4150/10000] | d_real_loss: 0.5599 | d_Y_loss: 1.1794 | d_X_loss:
0.5809 | d_fake_loss: 1.7603 | g_loss: 0.4322
Iteration [ 4160/10000] | d_real_loss: 0.5795 | d_Y_loss: 1.0178 | d_X_loss:
```

```
0.7634 | d_fake_loss: 1.7812 | g_loss: 0.4969
Iteration [ 4170/10000] | d_real_loss: 0.5579 | d_Y_loss: 1.0606 | d_X_loss:
0.3488 | d_fake_loss: 1.4093 | g_loss: 0.4791
Iteration [ 4180/10000] | d_real_loss: 0.5297 | d_Y_loss: 1.0000 | d_X_loss:
0.3483 | d_fake_loss: 1.3483 | g_loss: 0.4954
Iteration [ 4190/10000] | d_real_loss: 0.7014 | d_Y_loss: 1.0366 | d_X_loss:
0.7252 | d_fake_loss: 1.7618 | g_loss: 0.4809
Iteration [ 4200/10000] | d_real_loss: 0.6482 | d_Y_loss: 1.0489 | d_X_loss:
0.8705 | d_fake_loss: 1.9194 | g_loss: 0.4798
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004200-
Y-X.png
Iteration [ 4210/10000] | d_real_loss: 0.7208 | d_Y_loss: 1.0038 | d_X_loss:
0.6798 | d_fake_loss: 1.6836 | g_loss: 0.4950
Iteration [ 4220/10000] | d_real_loss: 0.7414 | d_Y_loss: 1.0530 | d_X_loss:
0.5480 | d_fake_loss: 1.6010 | g_loss: 0.4902
Iteration [ 4230/10000] | d_real_loss: 0.6602 | d_Y_loss: 0.9621 | d_X_loss:
0.6871 | d_fake_loss: 1.6492 | g_loss: 0.5146
Iteration [ 4240/10000] | d_real_loss: 0.6232 | d_Y_loss: 1.0354 | d_X_loss:
0.5893 | d_fake_loss: 1.6247 | g_loss: 0.4853
Iteration [ 4250/10000] | d_real_loss: 0.6073 | d_Y_loss: 1.0414 | d_X_loss:
0.5664 | d_fake_loss: 1.6078 | g_loss: 0.4801
Iteration [ 4260/10000] | d_real_loss: 0.8402 | d_Y_loss: 1.0424 | d_X_loss:
0.4995 | d_fake_loss: 1.5420 | g_loss: 0.4878
Iteration [ 4270/10000] | d_real_loss: 0.6835 | d_Y_loss: 1.0019 | d_X_loss:
0.4104 | d_fake_loss: 1.4123 | g_loss: 0.5056
Iteration [ 4280/10000] | d_real_loss: 0.6606 | d_Y_loss: 1.0183 | d_X_loss:
0.4272 | d_fake_loss: 1.4456 | g_loss: 0.4966
Iteration [ 4290/10000] | d_real_loss: 0.7362 | d_Y_loss: 1.1260 | d_X_loss:
0.5992 | d_fake_loss: 1.7252 | g_loss: 0.4605
Iteration [ 4300/10000] | d_real_loss: 0.6381 | d_Y_loss: 0.9834 | d_X_loss:
0.3777 | d_fake_loss: 1.3611 | g_loss: 0.5067
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004300-
Y-X.png
Iteration [ 4310/10000] | d_real_loss: 0.6115 | d_Y_loss: 1.0798 | d_X_loss:
0.5117 | d_fake_loss: 1.5915 | g_loss: 0.4910
Iteration [ 4320/10000] | d_real_loss: 0.6230 | d_Y_loss: 1.0269 | d_X_loss:
0.4825 | d_fake_loss: 1.5093 | g_loss: 0.4803
Iteration [ 4330/10000] | d_real_loss: 0.5186 | d_Y_loss: 1.0314 | d_X_loss:
1.1203 | d_fake_loss: 2.1517 | g_loss: 0.4945
Iteration [ 4340/10000] | d_real_loss: 0.5198 | d_Y_loss: 1.0183 | d_X_loss:
0.3560 | d_fake_loss: 1.3744 | g_loss: 0.4995
Iteration [ 4350/10000] | d_real_loss: 0.5011 | d_Y_loss: 1.0605 | d_X_loss:
0.4445 | d_fake_loss: 1.5050 | g_loss: 0.4788
Iteration [ 4360/10000] | d_real_loss: 0.6805 | d_Y_loss: 1.0043 | d_X_loss:
```

```
0.3612 | d_fake_loss: 1.3655 | g_loss: 0.4990
Iteration [ 4370/10000] | d_real_loss: 0.5736 | d_Y_loss: 1.0561 | d_X_loss:
0.5852 | d_fake_loss: 1.6413 | g_loss: 0.4712
Iteration [ 4380/10000] | d_real_loss: 0.6355 | d_Y_loss: 1.0024 | d_X_loss:
0.6422 | d_fake_loss: 1.6446 | g_loss: 0.5122
Iteration [ 4390/10000] | d_real_loss: 0.6880 | d_Y_loss: 1.1155 | d_X_loss:
0.5204 | d_fake_loss: 1.6358 | g_loss: 0.4404
Iteration [ 4400/10000] | d_real_loss: 0.5744 | d_Y_loss: 1.0363 | d_X_loss:
0.5445 | d_fake_loss: 1.5807 | g_loss: 0.4759
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004400-
Y-X.png
Iteration [ 4410/10000] | d_real_loss: 0.6312 | d_Y_loss: 1.0130 | d_X_loss:
0.5377 | d_fake_loss: 1.5506 | g_loss: 0.5072
Iteration [ 4420/10000] | d_real_loss: 0.6086 | d_Y_loss: 1.0428 | d_X_loss:
0.6068 | d_fake_loss: 1.6497 | g_loss: 0.4760
Iteration [ 4430/10000] | d_real_loss: 0.6814 | d_Y_loss: 1.0187 | d_X_loss:
0.9246 | d_fake_loss: 1.9433 | g_loss: 0.4996
Iteration [ 4440/10000] | d_real_loss: 0.5830 | d_Y_loss: 1.0305 | d_X_loss:
0.7666 | d_fake_loss: 1.7971 | g_loss: 0.5090
Iteration [ 4450/10000] | d_real_loss: 0.6012 | d_Y_loss: 1.1264 | d_X_loss:
0.6125 | d_fake_loss: 1.7388 | g_loss: 0.4447
Iteration [ 4460/10000] | d_real_loss: 0.5837 | d_Y_loss: 1.0539 | d_X_loss:
0.8919 | d_fake_loss: 1.9458 | g_loss: 0.4748
Iteration [ 4470/10000] | d_real_loss: 0.6523 | d_Y_loss: 1.0333 | d_X_loss:
0.4243 | d_fake_loss: 1.4576 | g_loss: 0.4914
Iteration [ 4480/10000] | d_real_loss: 0.5788 | d_Y_loss: 1.0856 | d_X_loss:
0.7266 | d_fake_loss: 1.8122 | g_loss: 0.4604
Iteration [ 4490/10000] | d_real_loss: 0.7167 | d_Y_loss: 1.0818 | d_X_loss:
0.5066 | d_fake_loss: 1.5885 | g_loss: 0.4605
Iteration [ 4500/10000] | d_real_loss: 0.6937 | d_Y_loss: 0.9779 | d_X_loss:
0.6634 | d_fake_loss: 1.6413 | g_loss: 0.5054
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004500-
Y-X.png
Iteration [ 4510/10000] | d_real_loss: 0.6183 | d_Y_loss: 1.1177 | d_X_loss:
0.4231 | d_fake_loss: 1.5409 | g_loss: 0.4781
Iteration [ 4520/10000] | d_real_loss: 0.6775 | d_Y_loss: 0.9812 | d_X_loss:
0.9958 | d_fake_loss: 1.9771 | g_loss: 0.5245
Iteration [ 4530/10000] | d_real_loss: 0.5882 | d_Y_loss: 1.0517 | d_X_loss:
0.5152 | d_fake_loss: 1.5669 | g_loss: 0.4965
Iteration [ 4540/10000] | d_real_loss: 0.5487 | d_Y_loss: 0.9800 | d_X_loss:
0.5773 | d_fake_loss: 1.5572 | g_loss: 0.5059
Iteration [ 4550/10000] | d_real_loss: 0.5376 | d_Y_loss: 1.0519 | d_X_loss:
0.4407 | d_fake_loss: 1.4927 | g_loss: 0.4843
Iteration [ 4560/10000] | d_real_loss: 0.6542 | d_Y_loss: 0.9825 | d_X_loss:
```

```
0.7846 | d_fake_loss: 1.7671 | g_loss: 0.5190
Iteration [ 4570/10000] | d_real_loss: 0.6904 | d_Y_loss: 1.0290 | d_X_loss:
0.5917 | d_fake_loss: 1.6207 | g_loss: 0.4963
Iteration [ 4580/10000] | d_real_loss: 0.6895 | d_Y_loss: 1.0251 | d_X_loss:
0.5732 | d_fake_loss: 1.5983 | g_loss: 0.4946
Iteration [ 4590/10000] | d_real_loss: 0.6514 | d_Y_loss: 0.9542 | d_X_loss:
0.7821 | d_fake_loss: 1.7364 | g_loss: 0.5224
Iteration [ 4600/10000] | d_real_loss: 0.4937 | d_Y_loss: 1.0901 | d_X_loss:
0.6241 | d_fake_loss: 1.7142 | g_loss: 0.4880
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004600-
Y-X.png
Iteration [ 4610/10000] | d_real_loss: 0.6681 | d_Y_loss: 0.9706 | d_X_loss:
0.5381 | d_fake_loss: 1.5088 | g_loss: 0.5143
Iteration [ 4620/10000] | d_real_loss: 0.6275 | d_Y_loss: 1.0683 | d_X_loss:
0.5638 | d_fake_loss: 1.6321 | g_loss: 0.4780
Iteration [ 4630/10000] | d_real_loss: 0.6886 | d_Y_loss: 1.0462 | d_X_loss:
0.5471 | d_fake_loss: 1.5932 | g_loss: 0.4785
Iteration [ 4640/10000] | d_real_loss: 0.5700 | d_Y_loss: 1.0229 | d_X_loss:
0.7688 | d_fake_loss: 1.7916 | g_loss: 0.4984
Iteration [ 4650/10000] | d_real_loss: 0.6128 | d_Y_loss: 1.0622 | d_X_loss:
0.8136 | d_fake_loss: 1.8757 | g_loss: 0.4715
Iteration [ 4660/10000] | d_real_loss: 0.6464 | d_Y_loss: 1.0486 | d_X_loss:
0.6467 | d_fake_loss: 1.6953 | g_loss: 0.4734
Iteration [ 4670/10000] | d_real_loss: 0.6524 | d_Y_loss: 1.0123 | d_X_loss:
0.7615 | d_fake_loss: 1.7738 | g_loss: 0.4993
Iteration [ 4680/10000] | d_real_loss: 0.6344 | d_Y_loss: 0.9953 | d_X_loss:
0.6143 | d_fake_loss: 1.6096 | g_loss: 0.5132
Iteration [ 4690/10000] | d_real_loss: 0.7877 | d_Y_loss: 1.0346 | d_X_loss:
0.5262 | d_fake_loss: 1.5608 | g_loss: 0.4881
Iteration [ 4700/10000] | d_real_loss: 0.7027 | d_Y_loss: 0.9882 | d_X_loss:
0.8429 | d_fake_loss: 1.8310 | g_loss: 0.4989
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004700-
Y-X.png
Iteration [ 4710/10000] | d_real_loss: 0.5407 | d_Y_loss: 1.0680 | d_X_loss:
0.6355 | d_fake_loss: 1.7035 | g_loss: 0.4730
Iteration [ 4720/10000] | d_real_loss: 0.6188 | d_Y_loss: 1.0192 | d_X_loss:
0.4141 | d_fake_loss: 1.4333 | g_loss: 0.5008
Iteration [ 4730/10000] | d_real_loss: 0.5601 | d_Y_loss: 1.0598 | d_X_loss:
0.8368 | d_fake_loss: 1.8967 | g_loss: 0.5121
Iteration [ 4740/10000] | d_real_loss: 0.6354 | d_Y_loss: 1.0210 | d_X_loss:
0.6161 | d_fake_loss: 1.6371 | g_loss: 0.4961
Iteration [ 4750/10000] | d_real_loss: 0.7328 | d_Y_loss: 1.0868 | d_X_loss:
0.6979 | d_fake_loss: 1.7847 | g_loss: 0.4698
Iteration [ 4760/10000] | d_real_loss: 0.6098 | d_Y_loss: 0.9970 | d_X_loss:
```

```
0.4869 | d_fake_loss: 1.4838 | g_loss: 0.5132
Iteration [ 4770/10000] | d_real_loss: 0.6638 | d_Y_loss: 1.0350 | d_X_loss:
0.8506 | d_fake_loss: 1.8857 | g_loss: 0.4908
Iteration [ 4780/10000] | d_real_loss: 0.5930 | d_Y_loss: 1.0336 | d_X_loss:
0.5461 | d_fake_loss: 1.5797 | g_loss: 0.4887
Iteration [ 4790/10000] | d_real_loss: 0.7399 | d_Y_loss: 1.0364 | d_X_loss:
0.6992 | d_fake_loss: 1.7356 | g_loss: 0.4936
Iteration [ 4800/10000] | d_real_loss: 0.5874 | d_Y_loss: 1.0876 | d_X_loss:
0.6345 | d_fake_loss: 1.7221 | g_loss: 0.4735
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004800-
Y-X.png
Iteration [ 4810/10000] | d_real_loss: 0.6230 | d_Y_loss: 1.0246 | d_X_loss:
0.9047 | d_fake_loss: 1.9294 | g_loss: 0.4995
Iteration [ 4820/10000] | d_real_loss: 0.6085 | d_Y_loss: 1.1553 | d_X_loss:
0.7878 | d_fake_loss: 1.9431 | g_loss: 0.4862
Iteration [ 4830/10000] | d_real_loss: 0.6420 | d_Y_loss: 1.0220 | d_X_loss:
0.6423 | d_fake_loss: 1.6643 | g_loss: 0.4943
Iteration [ 4840/10000] | d_real_loss: 0.5994 | d_Y_loss: 1.0518 | d_X_loss:
0.7812 | d_fake_loss: 1.8330 | g_loss: 0.4725
Iteration [ 4850/10000] | d_real_loss: 0.5178 | d_Y_loss: 1.0463 | d_X_loss:
0.4931 | d_fake_loss: 1.5394 | g_loss: 0.4883
Iteration [ 4860/10000] | d_real_loss: 0.6732 | d_Y_loss: 0.9891 | d_X_loss:
0.4926 | d_fake_loss: 1.4817 | g_loss: 0.5075
Iteration [ 4870/10000] | d_real_loss: 0.5198 | d_Y_loss: 1.0906 | d_X_loss:
0.6736 | d_fake_loss: 1.7642 | g_loss: 0.4585
Iteration [ 4880/10000] | d_real_loss: 0.8069 | d_Y_loss: 0.9768 | d_X_loss:
0.7528 | d_fake_loss: 1.7296 | g_loss: 0.5235
Iteration [ 4890/10000] | d_real_loss: 0.7199 | d_Y_loss: 1.0060 | d_X_loss:
0.6373 | d_fake_loss: 1.6433 | g_loss: 0.5055
Iteration [ 4900/10000] | d_real_loss: 0.5635 | d_Y_loss: 1.0170 | d_X_loss:
0.8592 | d_fake_loss: 1.8762 | g_loss: 0.5067
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-004900-
Y-X.png
Iteration [ 4910/10000] | d_real_loss: 0.6480 | d_Y_loss: 0.9838 | d_X_loss:
0.7521 | d_fake_loss: 1.7359 | g_loss: 0.5124
Iteration [ 4920/10000] | d_real_loss: 0.6532 | d_Y_loss: 1.0480 | d_X_loss:
0.7908 | d_fake_loss: 1.8389 | g_loss: 0.4885
Iteration [ 4930/10000] | d_real_loss: 0.6923 | d_Y_loss: 0.9741 | d_X_loss:
0.6264 | d_fake_loss: 1.6005 | g_loss: 0.5165
Iteration [ 4940/10000] | d_real_loss: 0.6018 | d_Y_loss: 1.0990 | d_X_loss:
0.4190 | d_fake_loss: 1.5180 | g_loss: 0.5076
Iteration [ 4950/10000] | d_real_loss: 0.5501 | d_Y_loss: 1.0726 | d_X_loss:
0.5423 | d_fake_loss: 1.6148 | g_loss: 0.4863
Iteration [ 4960/10000] | d_real_loss: 0.5891 | d_Y_loss: 1.0341 | d_X_loss:
```

```
0.5608 | d_fake_loss: 1.5950 | g_loss: 0.4814
Iteration [ 4970/10000] | d_real_loss: 0.6425 | d_Y_loss: 1.0001 | d_X_loss:
0.7302 | d_fake_loss: 1.7302 | g_loss: 0.5083
Iteration [ 4980/10000] | d_real_loss: 0.5865 | d_Y_loss: 1.0207 | d_X_loss:
0.6205 | d_fake_loss: 1.6412 | g_loss: 0.4922
Iteration [ 4990/10000] | d_real_loss: 0.5615 | d_Y_loss: 0.9955 | d_X_loss:
0.3965 | d_fake_loss: 1.3920 | g_loss: 0.5023
Iteration [ 5000/10000] | d_real_loss: 0.5093 | d_Y_loss: 1.0144 | d_X_loss:
0.5665 | d_fake_loss: 1.5809 | g_loss: 0.5129
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005000-
Y-X.png
Iteration [ 5010/10000] | d_real_loss: 0.4425 | d_Y_loss: 1.0383 | d_X_loss:
0.4060 | d_fake_loss: 1.4443 | g_loss: 0.4874
Iteration [ 5020/10000] | d_real_loss: 0.5430 | d_Y_loss: 1.0563 | d_X_loss:
0.5498 | d_fake_loss: 1.6062 | g_loss: 0.4853
Iteration [ 5030/10000] | d_real_loss: 0.6314 | d_Y_loss: 0.9521 | d_X_loss:
1.1542 | d_fake_loss: 2.1063 | g_loss: 0.5423
Iteration [ 5040/10000] | d_real_loss: 0.7159 | d_Y_loss: 1.0915 | d_X_loss:
0.7770 | d_fake_loss: 1.8685 | g_loss: 0.4732
Iteration [ 5050/10000] | d_real_loss: 0.6032 | d_Y_loss: 0.9647 | d_X_loss:
0.6247 | d_fake_loss: 1.5893 | g_loss: 0.5281
Iteration [ 5060/10000] | d_real_loss: 0.8022 | d_Y_loss: 1.0161 | d_X_loss:
0.8133 | d_fake_loss: 1.8294 | g_loss: 0.5228
Iteration [ 5070/10000] | d_real_loss: 0.5566 | d_Y_loss: 1.0892 | d_X_loss:
0.6100 | d_fake_loss: 1.6992 | g_loss: 0.4693
Iteration [ 5080/10000] | d_real_loss: 0.6192 | d_Y_loss: 1.0109 | d_X_loss:
0.7759 | d_fake_loss: 1.7868 | g_loss: 0.5078
Iteration [ 5090/10000] | d_real_loss: 0.4769 | d_Y_loss: 1.1807 | d_X_loss:
0.6951 | d_fake_loss: 1.8758 | g_loss: 0.4787
Iteration [ 5100/10000] | d_real_loss: 0.5872 | d_Y_loss: 1.0631 | d_X_loss:
0.8775 | d_fake_loss: 1.9406 | g_loss: 0.4755
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005100-
Y-X.png
Iteration [ 5110/10000] | d_real_loss: 0.7512 | d_Y_loss: 1.0254 | d_X_loss:
0.5509 | d_fake_loss: 1.5763 | g_loss: 0.4986
Iteration [ 5120/10000] | d_real_loss: 0.9166 | d_Y_loss: 1.0260 | d_X_loss:
0.4741 | d_fake_loss: 1.5001 | g_loss: 0.4891
Iteration [ 5130/10000] | d_real_loss: 0.8560 | d_Y_loss: 1.0062 | d_X_loss:
0.7188 | d_fake_loss: 1.7250 | g_loss: 0.5063
Iteration [ 5140/10000] | d_real_loss: 0.7074 | d_Y_loss: 1.0094 | d_X_loss:
0.5215 | d_fake_loss: 1.5309 | g_loss: 0.5013
Iteration [ 5150/10000] | d_real_loss: 0.7247 | d_Y_loss: 1.0075 | d_X_loss:
0.6766 | d_fake_loss: 1.6841 | g_loss: 0.5018
Iteration [ 5160/10000] | d_real_loss: 0.7751 | d_Y_loss: 1.0409 | d_X_loss:
```

```
0.9152 | d_fake_loss: 1.9561 | g_loss: 0.4917
Iteration [ 5170/10000] | d_real_loss: 0.6627 | d_Y_loss: 1.0372 | d_X_loss:
0.5018 | d_fake_loss: 1.5390 | g_loss: 0.4834
Iteration [ 5180/10000] | d_real_loss: 0.5214 | d_Y_loss: 1.0186 | d_X_loss:
0.4244 | d_fake_loss: 1.4430 | g_loss: 0.5056
Iteration [ 5190/10000] | d_real_loss: 0.6815 | d_Y_loss: 1.0114 | d_X_loss:
0.6371 | d_fake_loss: 1.6485 | g_loss: 0.5058
Iteration [ 5200/10000] | d_real_loss: 0.5949 | d_Y_loss: 1.0171 | d_X_loss:
0.5641 | d_fake_loss: 1.5812 | g_loss: 0.4972
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005200-
Y-X.png
Iteration [ 5210/10000] | d_real_loss: 0.8222 | d_Y_loss: 1.0306 | d_X_loss:
0.7560 | d_fake_loss: 1.7866 | g_loss: 0.4872
Iteration [ 5220/10000] | d_real_loss: 0.8365 | d_Y_loss: 1.0088 | d_X_loss:
0.7091 | d_fake_loss: 1.7179 | g_loss: 0.5067
Iteration [ 5230/10000] | d_real_loss: 0.7113 | d_Y_loss: 0.9645 | d_X_loss:
0.8464 | d_fake_loss: 1.8109 | g_loss: 0.5179
Iteration [ 5240/10000] | d_real_loss: 0.6283 | d_Y_loss: 1.0541 | d_X_loss:
0.7696 | d_fake_loss: 1.8237 | g_loss: 0.4763
Iteration [ 5250/10000] | d_real_loss: 0.6751 | d_Y_loss: 1.1032 | d_X_loss:
0.6530 | d_fake_loss: 1.7563 | g_loss: 0.4729
Iteration [ 5260/10000] | d_real_loss: 0.5034 | d_Y_loss: 1.0935 | d_X_loss:
0.4956 | d_fake_loss: 1.5891 | g_loss: 0.4701
Iteration [ 5270/10000] | d_real_loss: 0.5358 | d_Y_loss: 1.0488 | d_X_loss:
0.6038 | d_fake_loss: 1.6526 | g_loss: 0.4862
Iteration [ 5280/10000] | d_real_loss: 0.7015 | d_Y_loss: 0.9889 | d_X_loss:
0.5801 | d_fake_loss: 1.5690 | g_loss: 0.5031
Iteration [ 5290/10000] | d_real_loss: 0.7344 | d_Y_loss: 1.1465 | d_X_loss:
0.5452 | d_fake_loss: 1.6917 | g_loss: 0.4671
Iteration [ 5300/10000] | d_real_loss: 0.7932 | d_Y_loss: 0.9707 | d_X_loss:
0.6918 | d_fake_loss: 1.6625 | g_loss: 0.5321
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005300-
Y-X.png
Iteration [ 5310/10000] | d_real_loss: 0.5772 | d_Y_loss: 1.0378 | d_X_loss:
0.7464 | d_fake_loss: 1.7842 | g_loss: 0.4972
Iteration [ 5320/10000] | d_real_loss: 0.6831 | d_Y_loss: 1.0205 | d_X_loss:
0.6523 | d_fake_loss: 1.6728 | g_loss: 0.5137
Iteration [ 5330/10000] | d_real_loss: 0.6228 | d_Y_loss: 1.0513 | d_X_loss:
0.3845 | d_fake_loss: 1.4357 | g_loss: 0.4968
Iteration [ 5340/10000] | d_real_loss: 0.8063 | d_Y_loss: 1.0550 | d_X_loss:
0.7778 | d_fake_loss: 1.8328 | g_loss: 0.4822
Iteration [ 5350/10000] | d_real_loss: 0.9005 | d_Y_loss: 1.0423 | d_X_loss:
0.9436 | d_fake_loss: 1.9859 | g_loss: 0.4778
Iteration [ 5360/10000] | d_real_loss: 0.6027 | d_Y_loss: 1.0200 | d_X_loss:
```

```
0.6422 | d_fake_loss: 1.6622 | g_loss: 0.5018
Iteration [ 5370/10000] | d_real_loss: 0.7239 | d_Y_loss: 1.0134 | d_X_loss:
0.4670 | d_fake_loss: 1.4804 | g_loss: 0.5033
Iteration [ 5380/10000] | d_real_loss: 0.6208 | d_Y_loss: 1.0014 | d_X_loss:
0.7553 | d_fake_loss: 1.7567 | g_loss: 0.5072
Iteration [ 5390/10000] | d_real_loss: 0.5973 | d_Y_loss: 0.9982 | d_X_loss:
0.5359 | d_fake_loss: 1.5341 | g_loss: 0.5021
Iteration [ 5400/10000] | d_real_loss: 0.6724 | d_Y_loss: 1.0426 | d_X_loss:
0.6388 | d_fake_loss: 1.6814 | g_loss: 0.4959
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005400-
Y-X.png
Iteration [ 5410/10000] | d_real_loss: 0.6398 | d_Y_loss: 0.9557 | d_X_loss:
0.5740 | d_fake_loss: 1.5297 | g_loss: 0.5352
Iteration [ 5420/10000] | d_real_loss: 0.7208 | d_Y_loss: 1.1285 | d_X_loss:
0.7300 | d_fake_loss: 1.8585 | g_loss: 0.4702
Iteration [ 5430/10000] | d_real_loss: 0.6870 | d_Y_loss: 1.0550 | d_X_loss:
0.4804 | d_fake_loss: 1.5354 | g_loss: 0.4737
Iteration [ 5440/10000] | d_real_loss: 0.7181 | d_Y_loss: 0.9788 | d_X_loss:
1.0842 | d_fake_loss: 2.0630 | g_loss: 0.5269
Iteration [ 5450/10000] | d_real_loss: 0.8300 | d_Y_loss: 1.0179 | d_X_loss:
0.8922 | d_fake_loss: 1.9101 | g_loss: 0.5035
Iteration [ 5460/10000] | d_real_loss: 0.5971 | d_Y_loss: 0.9935 | d_X_loss:
0.4265 | d_fake_loss: 1.4199 | g_loss: 0.5170
Iteration [ 5470/10000] | d_real_loss: 0.7547 | d_Y_loss: 1.0256 | d_X_loss:
0.6183 | d_fake_loss: 1.6440 | g_loss: 0.5111
Iteration [ 5480/10000] | d_real_loss: 0.7053 | d_Y_loss: 0.9768 | d_X_loss:
0.6512 | d_fake_loss: 1.6280 | g_loss: 0.5354
Iteration [ 5490/10000] | d_real_loss: 0.7532 | d_Y_loss: 0.9885 | d_X_loss:
0.7377 | d_fake_loss: 1.7263 | g_loss: 0.5082
Iteration [ 5500/10000] | d_real_loss: 0.6309 | d_Y_loss: 1.0624 | d_X_loss:
0.4115 | d_fake_loss: 1.4739 | g_loss: 0.4978
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005500-
Y-X.png
Iteration [ 5510/10000] | d_real_loss: 0.7894 | d_Y_loss: 1.0674 | d_X_loss:
0.7028 | d_fake_loss: 1.7702 | g_loss: 0.4898
Iteration [ 5520/10000] | d_real_loss: 0.7149 | d_Y_loss: 1.0443 | d_X_loss:
0.7919 | d_fake_loss: 1.8362 | g_loss: 0.4959
Iteration [ 5530/10000] | d_real_loss: 0.6822 | d_Y_loss: 0.9998 | d_X_loss:
0.5993 | d_fake_loss: 1.5991 | g_loss: 0.5172
Iteration [ 5540/10000] | d_real_loss: 0.6046 | d_Y_loss: 0.9715 | d_X_loss:
0.8779 | d_fake_loss: 1.8494 | g_loss: 0.5350
Iteration [ 5550/10000] | d_real_loss: 0.5306 | d_Y_loss: 1.1088 | d_X_loss:
0.6894 | d_fake_loss: 1.7982 | g_loss: 0.4597
Iteration [ 5560/10000] | d_real_loss: 0.6102 | d_Y_loss: 1.0294 | d_X_loss:
```

```
0.5580 | d_fake_loss: 1.5874 | g_loss: 0.5075
Iteration [ 5570/10000] | d_real_loss: 0.5773 | d_Y_loss: 1.0077 | d_X_loss:
0.7320 | d_fake_loss: 1.7397 | g_loss: 0.5026
Iteration [ 5580/10000] | d_real_loss: 0.6174 | d_Y_loss: 1.0126 | d_X_loss:
0.8096 | d_fake_loss: 1.8222 | g_loss: 0.5072
Iteration [ 5590/10000] | d_real_loss: 0.5905 | d_Y_loss: 1.0257 | d_X_loss:
0.8710 | d_fake_loss: 1.8967 | g_loss: 0.5025
Iteration [ 5600/10000] | d_real_loss: 0.6528 | d_Y_loss: 1.0523 | d_X_loss:
0.6499 | d_fake_loss: 1.7021 | g_loss: 0.5046
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005600-
Y-X.png
Iteration [ 5610/10000] | d_real_loss: 0.5320 | d_Y_loss: 1.0279 | d_X_loss:
0.4388 | d_fake_loss: 1.4667 | g_loss: 0.5106
Iteration [ 5620/10000] | d_real_loss: 0.7276 | d_Y_loss: 1.0352 | d_X_loss:
0.5066 | d_fake_loss: 1.5418 | g_loss: 0.5007
Iteration [ 5630/10000] | d_real_loss: 0.6462 | d_Y_loss: 1.0184 | d_X_loss:
0.6143 | d_fake_loss: 1.6327 | g_loss: 0.5015
Iteration [ 5640/10000] | d_real_loss: 0.6666 | d_Y_loss: 0.9859 | d_X_loss:
0.4917 | d_fake_loss: 1.4776 | g_loss: 0.5101
Iteration [ 5650/10000] | d_real_loss: 0.5496 | d_Y_loss: 1.0451 | d_X_loss:
0.6108 | d_fake_loss: 1.6559 | g_loss: 0.4892
Iteration [ 5660/10000] | d_real_loss: 0.7381 | d_Y_loss: 1.0329 | d_X_loss:
0.5823 | d_fake_loss: 1.6153 | g_loss: 0.4844
Iteration [ 5670/10000] | d_real_loss: 0.6181 | d_Y_loss: 1.0451 | d_X_loss:
0.9126 | d_fake_loss: 1.9576 | g_loss: 0.4908
Iteration [ 5680/10000] | d_real_loss: 0.6656 | d_Y_loss: 1.0617 | d_X_loss:
0.7007 | d_fake_loss: 1.7624 | g_loss: 0.4967
Iteration [ 5690/10000] | d_real_loss: 0.6798 | d_Y_loss: 1.0034 | d_X_loss:
0.8071 | d_fake_loss: 1.8106 | g_loss: 0.5041
Iteration [ 5700/10000] | d_real_loss: 0.5960 | d_Y_loss: 1.0499 | d_X_loss:
0.7722 | d_fake_loss: 1.8221 | g_loss: 0.4907
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005700-
Y-X.png
Iteration [ 5710/10000] | d_real_loss: 0.7713 | d_Y_loss: 1.0360 | d_X_loss:
0.7829 | d_fake_loss: 1.8189 | g_loss: 0.4938
Iteration [ 5720/10000] | d_real_loss: 0.6322 | d_Y_loss: 1.0490 | d_X_loss:
0.5344 | d_fake_loss: 1.5833 | g_loss: 0.4964
Iteration [ 5730/10000] | d_real_loss: 0.6918 | d_Y_loss: 1.0703 | d_X_loss:
0.5893 | d_fake_loss: 1.6596 | g_loss: 0.4764
Iteration [ 5740/10000] | d_real_loss: 0.5453 | d_Y_loss: 1.0201 | d_X_loss:
0.5217 | d_fake_loss: 1.5418 | g_loss: 0.5040
Iteration [ 5750/10000] | d_real_loss: 0.6472 | d_Y_loss: 0.9723 | d_X_loss:
0.7724 | d_fake_loss: 1.7447 | g_loss: 0.5390
Iteration [ 5760/10000] | d_real_loss: 0.6215 | d_Y_loss: 1.0434 | d_X_loss:
```

```
0.4759 | d_fake_loss: 1.5192 | g_loss: 0.4806
Iteration [ 5770/10000] | d_real_loss: 0.6906 | d_Y_loss: 1.0360 | d_X_loss:
0.6893 | d_fake_loss: 1.7253 | g_loss: 0.5127
Iteration [ 5780/10000] | d_real_loss: 0.7022 | d_Y_loss: 1.0982 | d_X_loss:
0.5885 | d_fake_loss: 1.6867 | g_loss: 0.4628
Iteration [ 5790/10000] | d_real_loss: 0.6510 | d_Y_loss: 0.9941 | d_X_loss:
0.9665 | d_fake_loss: 1.9605 | g_loss: 0.5099
Iteration [ 5800/10000] | d_real_loss: 0.5451 | d_Y_loss: 0.9843 | d_X_loss:
0.5387 | d_fake_loss: 1.5230 | g_loss: 0.5110
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005800-
Y-X.png
Iteration [ 5810/10000] | d_real_loss: 0.5951 | d_Y_loss: 1.0454 | d_X_loss:
0.8427 | d_fake_loss: 1.8882 | g_loss: 0.4835
Iteration [ 5820/10000] | d_real_loss: 0.6243 | d_Y_loss: 0.9792 | d_X_loss:
0.4515 | d_fake_loss: 1.4307 | g_loss: 0.5300
Iteration [ 5830/10000] | d_real_loss: 0.7398 | d_Y_loss: 1.0403 | d_X_loss:
0.5928 | d_fake_loss: 1.6332 | g_loss: 0.5017
Iteration [ 5840/10000] | d_real_loss: 0.6036 | d_Y_loss: 0.9983 | d_X_loss:
0.7763 | d_fake_loss: 1.7746 | g_loss: 0.5083
Iteration [ 5850/10000] | d_real_loss: 0.6840 | d_Y_loss: 0.9751 | d_X_loss:
0.6342 | d_fake_loss: 1.6093 | g_loss: 0.5270
Iteration [ 5860/10000] | d_real_loss: 0.6004 | d_Y_loss: 1.0526 | d_X_loss:
0.4951 | d_fake_loss: 1.5477 | g_loss: 0.4855
Iteration [ 5870/10000] | d_real_loss: 0.5574 | d_Y_loss: 0.9455 | d_X_loss:
0.5745 | d_fake_loss: 1.5200 | g_loss: 0.5358
Iteration [ 5880/10000] | d_real_loss: 0.5623 | d_Y_loss: 1.0762 | d_X_loss:
0.4745 | d_fake_loss: 1.5507 | g_loss: 0.4815
Iteration [ 5890/10000] | d_real_loss: 0.7167 | d_Y_loss: 0.9839 | d_X_loss:
0.6585 | d_fake_loss: 1.6425 | g_loss: 0.5190
Iteration [ 5900/10000] | d_real_loss: 0.6200 | d_Y_loss: 1.0458 | d_X_loss:
0.5098 | d_fake_loss: 1.5556 | g_loss: 0.4982
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-005900-
Y-X.png
Iteration [ 5910/10000] | d_real_loss: 0.6030 | d_Y_loss: 1.0312 | d_X_loss:
0.5720 | d_fake_loss: 1.6032 | g_loss: 0.5009
Iteration [ 5920/10000] | d_real_loss: 0.6494 | d_Y_loss: 1.0218 | d_X_loss:
0.5965 | d_fake_loss: 1.6182 | g_loss: 0.5405
Iteration [ 5930/10000] | d_real_loss: 0.6259 | d_Y_loss: 0.9837 | d_X_loss:
0.5320 | d_fake_loss: 1.5157 | g_loss: 0.5180
Iteration [ 5940/10000] | d_real_loss: 0.5712 | d_Y_loss: 0.9951 | d_X_loss:
0.7719 | d_fake_loss: 1.7670 | g_loss: 0.5166
Iteration [ 5950/10000] | d_real_loss: 0.6295 | d_Y_loss: 0.9669 | d_X_loss:
0.6522 | d_fake_loss: 1.6191 | g_loss: 0.5415
Iteration [ 5960/10000] | d_real_loss: 0.6512 | d_Y_loss: 0.9416 | d_X_loss:
```

```
0.6740 | d_fake_loss: 1.6157 | g_loss: 0.5422
Iteration [ 5970/10000] | d_real_loss: 0.7619 | d_Y_loss: 1.1461 | d_X_loss:
0.5178 | d_fake_loss: 1.6639 | g_loss: 0.4836
Iteration [ 5980/10000] | d_real_loss: 0.7652 | d_Y_loss: 1.0489 | d_X_loss:
0.5862 | d_fake_loss: 1.6351 | g_loss: 0.4886
Iteration [ 5990/10000] | d_real_loss: 0.6758 | d_Y_loss: 1.0098 | d_X_loss:
0.6112 | d_fake_loss: 1.6209 | g_loss: 0.5150
Iteration [ 6000/10000] | d_real_loss: 0.6980 | d_Y_loss: 1.0443 | d_X_loss:
0.5684 | d_fake_loss: 1.6127 | g_loss: 0.4852
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006000-
Y-X.png
Iteration [ 6010/10000] | d_real_loss: 0.4863 | d_Y_loss: 1.0358 | d_X_loss:
0.3029 | d_fake_loss: 1.3387 | g_loss: 0.5058
Iteration [ 6020/10000] | d_real_loss: 0.5414 | d_Y_loss: 1.0324 | d_X_loss:
0.5189 | d_fake_loss: 1.5513 | g_loss: 0.5036
Iteration [ 6030/10000] | d_real_loss: 0.7198 | d_Y_loss: 1.0804 | d_X_loss:
0.7499 | d_fake_loss: 1.8303 | g_loss: 0.4737
Iteration [ 6040/10000] | d_real_loss: 0.6846 | d_Y_loss: 0.9872 | d_X_loss:
0.6067 | d_fake_loss: 1.5939 | g_loss: 0.5207
Iteration [ 6050/10000] | d_real_loss: 0.5752 | d_Y_loss: 1.0704 | d_X_loss:
0.9378 | d_fake_loss: 2.0082 | g_loss: 0.4948
Iteration [ 6060/10000] | d_real_loss: 0.6158 | d_Y_loss: 1.0510 | d_X_loss:
0.6455 | d_fake_loss: 1.6965 | g_loss: 0.4914
Iteration [ 6070/10000] | d_real_loss: 0.6902 | d_Y_loss: 0.9502 | d_X_loss:
0.6105 | d_fake_loss: 1.5607 | g_loss: 0.5447
Iteration [ 6080/10000] | d_real_loss: 0.5691 | d_Y_loss: 1.0001 | d_X_loss:
0.4430 | d_fake_loss: 1.4431 | g_loss: 0.5049
Iteration [ 6090/10000] | d_real_loss: 0.4826 | d_Y_loss: 1.0368 | d_X_loss:
0.6133 | d_fake_loss: 1.6501 | g_loss: 0.4874
Iteration [ 6100/10000] | d_real_loss: 0.6306 | d_Y_loss: 1.0292 | d_X_loss:
0.6508 | d_fake_loss: 1.6799 | g_loss: 0.5116
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006100-
Y-X.png
Iteration [ 6110/10000] | d_real_loss: 0.8558 | d_Y_loss: 0.9624 | d_X_loss:
0.7466 | d_fake_loss: 1.7090 | g_loss: 0.5341
Iteration [ 6120/10000] | d_real_loss: 0.6654 | d_Y_loss: 1.0686 | d_X_loss:
0.4782 | d_fake_loss: 1.5468 | g_loss: 0.4826
Iteration [ 6130/10000] | d_real_loss: 0.6468 | d_Y_loss: 1.0334 | d_X_loss:
0.8253 | d_fake_loss: 1.8587 | g_loss: 0.4887
Iteration [ 6140/10000] | d_real_loss: 0.6768 | d_Y_loss: 1.0378 | d_X_loss:
0.4015 | d_fake_loss: 1.4393 | g_loss: 0.5219
Iteration [ 6150/10000] | d_real_loss: 0.5802 | d_Y_loss: 1.0048 | d_X_loss:
0.5162 | d_fake_loss: 1.5210 | g_loss: 0.5162
Iteration [ 6160/10000] | d_real_loss: 0.7009 | d_Y_loss: 0.9756 | d_X_loss:
```

```
0.9522 | d_fake_loss: 1.9278 | g_loss: 0.5274
Iteration [ 6170/10000] | d_real_loss: 0.7064 | d_Y_loss: 1.0254 | d_X_loss:
0.4378 | d_fake_loss: 1.4632 | g_loss: 0.5053
Iteration [ 6180/10000] | d_real_loss: 0.6274 | d_Y_loss: 1.0270 | d_X_loss:
0.4879 | d_fake_loss: 1.5149 | g_loss: 0.4946
Iteration [ 6190/10000] | d_real_loss: 0.6809 | d_Y_loss: 1.0392 | d_X_loss:
0.7359 | d_fake_loss: 1.7752 | g_loss: 0.4973
Iteration [ 6200/10000] | d_real_loss: 0.7028 | d_Y_loss: 0.9776 | d_X_loss:
0.7659 | d_fake_loss: 1.7435 | g_loss: 0.5346
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006200-
Y-X.png
Iteration [ 6210/10000] | d_real_loss: 0.6999 | d_Y_loss: 1.0745 | d_X_loss:
0.5154 | d_fake_loss: 1.5899 | g_loss: 0.4753
Iteration [ 6220/10000] | d_real_loss: 0.5381 | d_Y_loss: 1.0000 | d_X_loss:
0.8657 | d_fake_loss: 1.8657 | g_loss: 0.5087
Iteration [ 6230/10000] | d_real_loss: 0.7926 | d_Y_loss: 1.0640 | d_X_loss:
0.5967 | d_fake_loss: 1.6606 | g_loss: 0.5076
Iteration [ 6240/10000] | d_real_loss: 0.6331 | d_Y_loss: 0.9680 | d_X_loss:
0.5314 | d_fake_loss: 1.4995 | g_loss: 0.5361
Iteration [ 6250/10000] | d_real_loss: 0.7741 | d_Y_loss: 1.0488 | d_X_loss:
0.6692 | d_fake_loss: 1.7180 | g_loss: 0.4831
Iteration [ 6260/10000] | d_real_loss: 0.6704 | d_Y_loss: 0.9928 | d_X_loss:
0.5545 | d_fake_loss: 1.5474 | g_loss: 0.5182
Iteration [ 6270/10000] | d_real_loss: 0.6167 | d_Y_loss: 0.9587 | d_X_loss:
0.5297 | d_fake_loss: 1.4884 | g_loss: 0.5321
Iteration [ 6280/10000] | d_real_loss: 0.6099 | d_Y_loss: 1.0674 | d_X_loss:
0.4843 | d_fake_loss: 1.5517 | g_loss: 0.4990
Iteration [ 6290/10000] | d_real_loss: 0.5241 | d_Y_loss: 0.9886 | d_X_loss:
0.4793 | d_fake_loss: 1.4679 | g_loss: 0.5249
Iteration [ 6300/10000] | d_real_loss: 0.7805 | d_Y_loss: 1.0496 | d_X_loss:
0.5256 | d_fake_loss: 1.5752 | g_loss: 0.4968
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006300-
Y-X.png
Iteration [ 6310/10000] | d_real_loss: 0.7162 | d_Y_loss: 1.0035 | d_X_loss:
0.8768 | d_fake_loss: 1.8803 | g_loss: 0.5374
Iteration [ 6320/10000] | d_real_loss: 0.5728 | d_Y_loss: 1.0331 | d_X_loss:
0.4909 | d_fake_loss: 1.5241 | g_loss: 0.5081
Iteration [ 6330/10000] | d_real_loss: 0.6683 | d_Y_loss: 1.0496 | d_X_loss:
0.5167 | d_fake_loss: 1.5663 | g_loss: 0.4900
Iteration [ 6340/10000] | d_real_loss: 0.5080 | d_Y_loss: 1.0881 | d_X_loss:
0.9255 | d_fake_loss: 2.0136 | g_loss: 0.4818
Iteration [ 6350/10000] | d_real_loss: 0.8035 | d_Y_loss: 0.9501 | d_X_loss:
0.4201 | d_fake_loss: 1.3702 | g_loss: 0.5379
Iteration [ 6360/10000] | d_real_loss: 0.7366 | d_Y_loss: 1.0352 | d_X_loss:
```

```
0.6650 | d_fake_loss: 1.7002 | g_loss: 0.5070
Iteration [ 6370/10000] | d_real_loss: 0.7281 | d_Y_loss: 1.0144 | d_X_loss:
0.4258 | d_fake_loss: 1.4402 | g_loss: 0.5084
Iteration [ 6380/10000] | d_real_loss: 0.5638 | d_Y_loss: 1.0241 | d_X_loss:
0.4535 | d_fake_loss: 1.4777 | g_loss: 0.5126
Iteration [ 6390/10000] | d_real_loss: 0.5171 | d_Y_loss: 1.0342 | d_X_loss:
0.8435 | d_fake_loss: 1.8777 | g_loss: 0.5096
Iteration [ 6400/10000] | d_real_loss: 0.6932 | d_Y_loss: 1.0429 | d_X_loss:
0.6175 | d_fake_loss: 1.6604 | g_loss: 0.4940
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006400-
Y-X.png
Iteration [ 6410/10000] | d_real_loss: 0.7900 | d_Y_loss: 1.0457 | d_X_loss:
0.6545 | d_fake_loss: 1.7002 | g_loss: 0.5128
Iteration [ 6420/10000] | d_real_loss: 0.5889 | d_Y_loss: 1.0509 | d_X_loss:
0.6611 | d_fake_loss: 1.7119 | g_loss: 0.4883
Iteration [ 6430/10000] | d_real_loss: 0.5909 | d_Y_loss: 0.9426 | d_X_loss:
0.8394 | d_fake_loss: 1.7820 | g_loss: 0.5652
Iteration [ 6440/10000] | d_real_loss: 0.6153 | d_Y_loss: 1.1340 | d_X_loss:
0.6067 | d_fake_loss: 1.7406 | g_loss: 0.4468
Iteration [ 6450/10000] | d_real_loss: 0.5354 | d_Y_loss: 1.0257 | d_X_loss:
0.6016 | d_fake_loss: 1.6273 | g_loss: 0.5036
Iteration [ 6460/10000] | d_real_loss: 0.6631 | d_Y_loss: 0.9712 | d_X_loss:
0.7287 | d_fake_loss: 1.6999 | g_loss: 0.5355
Iteration [ 6470/10000] | d_real_loss: 0.7855 | d_Y_loss: 1.0170 | d_X_loss:
0.4843 | d_fake_loss: 1.5013 | g_loss: 0.5008
Iteration [ 6480/10000] | d_real_loss: 0.5484 | d_Y_loss: 1.0249 | d_X_loss:
0.5624 | d_fake_loss: 1.5873 | g_loss: 0.5120
Iteration [ 6490/10000] | d_real_loss: 0.7229 | d_Y_loss: 0.9762 | d_X_loss:
0.4561 | d_fake_loss: 1.4324 | g_loss: 0.5196
Iteration [ 6500/10000] | d_real_loss: 0.5544 | d_Y_loss: 1.0534 | d_X_loss:
0.7160 | d_fake_loss: 1.7694 | g_loss: 0.4834
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006500-
Y-X.png
Iteration [ 6510/10000] | d_real_loss: 0.5506 | d_Y_loss: 1.0671 | d_X_loss:
0.7251 | d_fake_loss: 1.7922 | g_loss: 0.4919
Iteration [ 6520/10000] | d_real_loss: 0.7648 | d_Y_loss: 1.0040 | d_X_loss:
0.3600 | d_fake_loss: 1.3640 | g_loss: 0.5160
Iteration [ 6530/10000] | d_real_loss: 0.5644 | d_Y_loss: 0.9961 | d_X_loss:
0.4128 | d_fake_loss: 1.4089 | g_loss: 0.5237
Iteration [ 6540/10000] | d_real_loss: 0.6154 | d_Y_loss: 0.9328 | d_X_loss:
0.5496 | d_fake_loss: 1.4824 | g_loss: 0.5467
Iteration [ 6550/10000] | d_real_loss: 0.5999 | d_Y_loss: 1.0584 | d_X_loss:
0.4930 | d_fake_loss: 1.5514 | g_loss: 0.4875
Iteration [ 6560/10000] | d_real_loss: 0.7262 | d_Y_loss: 1.0411 | d_X_loss:
```

```
0.4985 | d_fake_loss: 1.5396 | g_loss: 0.4834
Iteration [ 6570/10000] | d_real_loss: 0.5391 | d_Y_loss: 1.0416 | d_X_loss:
0.7607 | d_fake_loss: 1.8023 | g_loss: 0.5034
Iteration [ 6580/10000] | d_real_loss: 0.5969 | d_Y_loss: 1.0150 | d_X_loss:
0.6102 | d_fake_loss: 1.6253 | g_loss: 0.4964
Iteration [ 6590/10000] | d_real_loss: 0.7087 | d_Y_loss: 1.0228 | d_X_loss:
0.5504 | d_fake_loss: 1.5732 | g_loss: 0.5058
Iteration [ 6600/10000] | d_real_loss: 0.6288 | d_Y_loss: 1.0540 | d_X_loss:
0.6954 | d_fake_loss: 1.7495 | g_loss: 0.5003
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006600-
Y-X.png
Iteration [ 6610/10000] | d_real_loss: 0.6826 | d_Y_loss: 0.9723 | d_X_loss:
0.9247 | d_fake_loss: 1.8970 | g_loss: 0.5188
Iteration [ 6620/10000] | d_real_loss: 0.5955 | d_Y_loss: 0.9147 | d_X_loss:
0.6859 | d_fake_loss: 1.6006 | g_loss: 0.5591
Iteration [ 6630/10000] | d_real_loss: 0.6995 | d_Y_loss: 1.0761 | d_X_loss:
0.6109 | d_fake_loss: 1.6870 | g_loss: 0.4821
Iteration [ 6640/10000] | d_real_loss: 0.6056 | d_Y_loss: 1.0084 | d_X_loss:
0.7918 | d_fake_loss: 1.8002 | g_loss: 0.5014
Iteration [ 6650/10000] | d_real_loss: 0.6137 | d_Y_loss: 0.9538 | d_X_loss:
0.7083 | d_fake_loss: 1.6621 | g_loss: 0.5455
Iteration [ 6660/10000] | d_real_loss: 0.6706 | d_Y_loss: 1.0155 | d_X_loss:
0.7140 | d_fake_loss: 1.7295 | g_loss: 0.5092
Iteration [ 6670/10000] | d_real_loss: 0.6437 | d_Y_loss: 1.0801 | d_X_loss:
0.5191 | d_fake_loss: 1.5992 | g_loss: 0.4718
Iteration [ 6680/10000] | d_real_loss: 0.5137 | d_Y_loss: 1.0266 | d_X_loss:
0.5768 | d_fake_loss: 1.6035 | g_loss: 0.5196
Iteration [ 6690/10000] | d_real_loss: 0.6327 | d_Y_loss: 1.0666 | d_X_loss:
0.8360 | d_fake_loss: 1.9026 | g_loss: 0.4899
Iteration [ 6700/10000] | d_real_loss: 0.8854 | d_Y_loss: 0.9731 | d_X_loss:
0.5202 | d_fake_loss: 1.4933 | g_loss: 0.5322
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006700-
Y-X.png
Iteration [ 6710/10000] | d_real_loss: 0.6400 | d_Y_loss: 0.9922 | d_X_loss:
0.6335 | d_fake_loss: 1.6257 | g_loss: 0.5169
Iteration [ 6720/10000] | d_real_loss: 0.6214 | d_Y_loss: 1.0268 | d_X_loss:
0.5832 | d_fake_loss: 1.6100 | g_loss: 0.5030
Iteration [ 6730/10000] | d_real_loss: 0.5920 | d_Y_loss: 1.0830 | d_X_loss:
0.5915 | d_fake_loss: 1.6745 | g_loss: 0.4736
Iteration [ 6740/10000] | d_real_loss: 0.6482 | d_Y_loss: 0.9577 | d_X_loss:
0.6665 | d_fake_loss: 1.6242 | g_loss: 0.5305
Iteration [ 6750/10000] | d_real_loss: 0.6198 | d_Y_loss: 1.0776 | d_X_loss:
0.7508 | d_fake_loss: 1.8284 | g_loss: 0.4928
Iteration [ 6760/10000] | d_real_loss: 0.4894 | d_Y_loss: 0.9964 | d_X_loss:
```

```
0.4295 | d_fake_loss: 1.4258 | g_loss: 0.5094
Iteration [ 6770/10000] | d_real_loss: 0.6273 | d_Y_loss: 1.0102 | d_X_loss:
0.6168 | d_fake_loss: 1.6270 | g_loss: 0.5157
Iteration [ 6780/10000] | d_real_loss: 0.6376 | d_Y_loss: 1.0952 | d_X_loss:
0.5283 | d_fake_loss: 1.6234 | g_loss: 0.4813
Iteration [ 6790/10000] | d_real_loss: 0.7015 | d_Y_loss: 0.9747 | d_X_loss:
0.5976 | d_fake_loss: 1.5723 | g_loss: 0.5316
Iteration [ 6800/10000] | d_real_loss: 0.6228 | d_Y_loss: 0.9455 | d_X_loss:
0.6114 | d_fake_loss: 1.5569 | g_loss: 0.5426
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006800-
Y-X.png
Iteration [ 6810/10000] | d_real_loss: 0.6028 | d_Y_loss: 1.0110 | d_X_loss:
0.6296 | d_fake_loss: 1.6406 | g_loss: 0.5056
Iteration [ 6820/10000] | d_real_loss: 0.7053 | d_Y_loss: 1.0577 | d_X_loss:
0.6418 | d_fake_loss: 1.6995 | g_loss: 0.4836
Iteration [ 6830/10000] | d_real_loss: 0.5022 | d_Y_loss: 0.9955 | d_X_loss:
0.7333 | d_fake_loss: 1.7289 | g_loss: 0.5321
Iteration [ 6840/10000] | d_real_loss: 0.6561 | d_Y_loss: 1.1013 | d_X_loss:
0.5161 | d_fake_loss: 1.6174 | g_loss: 0.5005
Iteration [ 6850/10000] | d_real_loss: 0.7233 | d_Y_loss: 0.9847 | d_X_loss:
0.6018 | d_fake_loss: 1.5866 | g_loss: 0.5243
Iteration [ 6860/10000] | d_real_loss: 0.6679 | d_Y_loss: 0.9889 | d_X_loss:
0.6845 | d_fake_loss: 1.6734 | g_loss: 0.5409
Iteration [ 6870/10000] | d_real_loss: 0.6177 | d_Y_loss: 1.1518 | d_X_loss:
0.5491 | d_fake_loss: 1.7009 | g_loss: 0.4506
Iteration [ 6880/10000] | d_real_loss: 0.6269 | d_Y_loss: 0.9815 | d_X_loss:
0.7175 | d_fake_loss: 1.6990 | g_loss: 0.5365
Iteration [ 6890/10000] | d_real_loss: 0.6074 | d_Y_loss: 1.0133 | d_X_loss:
0.4076 | d_fake_loss: 1.4209 | g_loss: 0.5007
Iteration [ 6900/10000] | d_real_loss: 0.5310 | d_Y_loss: 1.0251 | d_X_loss:
0.5197 | d_fake_loss: 1.5448 | g_loss: 0.5104
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-006900-
Y-X.png
Iteration [ 6910/10000] | d_real_loss: 0.7022 | d_Y_loss: 1.0184 | d_X_loss:
0.4847 | d_fake_loss: 1.5032 | g_loss: 0.5029
Iteration [ 6920/10000] | d_real_loss: 0.6295 | d_Y_loss: 1.0123 | d_X_loss:
0.6512 | d_fake_loss: 1.6635 | g_loss: 0.5152
Iteration [ 6930/10000] | d_real_loss: 0.6030 | d_Y_loss: 1.0051 | d_X_loss:
0.7782 | d_fake_loss: 1.7834 | g_loss: 0.5229
Iteration [ 6940/10000] | d_real_loss: 0.5877 | d_Y_loss: 1.1080 | d_X_loss:
0.6351 | d_fake_loss: 1.7431 | g_loss: 0.4896
Iteration [ 6950/10000] | d_real_loss: 0.7268 | d_Y_loss: 0.9365 | d_X_loss:
0.7393 | d_fake_loss: 1.6759 | g_loss: 0.5468
Iteration [ 6960/10000] | d_real_loss: 0.6181 | d_Y_loss: 1.0942 | d_X_loss:
```

```
0.5165 | d_fake_loss: 1.6107 | g_loss: 0.4710
Iteration [ 6970/10000] | d_real_loss: 0.6859 | d_Y_loss: 1.1023 | d_X_loss:
0.5375 | d_fake_loss: 1.6398 | g_loss: 0.4907
Iteration [ 6980/10000] | d_real_loss: 0.6204 | d_Y_loss: 0.9829 | d_X_loss:
0.5374 | d_fake_loss: 1.5203 | g_loss: 0.5227
Iteration [ 6990/10000] | d_real_loss: 0.6894 | d_Y_loss: 1.0057 | d_X_loss:
0.8903 | d_fake_loss: 1.8960 | g_loss: 0.5146
Iteration [ 7000/10000] | d_real_loss: 0.5077 | d_Y_loss: 0.9386 | d_X_loss:
0.3926 | d_fake_loss: 1.3312 | g_loss: 0.5452
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007000-
Y-X.png
Iteration [ 7010/10000] | d_real_loss: 0.5098 | d_Y_loss: 1.0657 | d_X_loss:
0.3785 | d_fake_loss: 1.4443 | g_loss: 0.4970
Iteration [ 7020/10000] | d_real_loss: 0.6186 | d_Y_loss: 0.9871 | d_X_loss:
0.5378 | d_fake_loss: 1.5249 | g_loss: 0.5349
Iteration [ 7030/10000] | d_real_loss: 0.6439 | d_Y_loss: 0.9902 | d_X_loss:
0.4393 | d_fake_loss: 1.4295 | g_loss: 0.5146
Iteration [ 7040/10000] | d_real_loss: 0.5760 | d_Y_loss: 1.0250 | d_X_loss:
0.8435 | d_fake_loss: 1.8685 | g_loss: 0.5241
Iteration [ 7050/10000] | d_real_loss: 0.7480 | d_Y_loss: 0.9547 | d_X_loss:
0.6793 | d_fake_loss: 1.6340 | g_loss: 0.5324
Iteration [ 7060/10000] | d_real_loss: 0.5864 | d_Y_loss: 1.0529 | d_X_loss:
0.5032 | d_fake_loss: 1.5561 | g_loss: 0.4826
Iteration [ 7070/10000] | d_real_loss: 0.6483 | d_Y_loss: 1.0264 | d_X_loss:
0.5638 | d_fake_loss: 1.5902 | g_loss: 0.5007
Iteration [ 7080/10000] | d_real_loss: 0.4956 | d_Y_loss: 0.9713 | d_X_loss:
0.6818 | d_fake_loss: 1.6531 | g_loss: 0.5398
Iteration [ 7090/10000] | d_real_loss: 0.5275 | d_Y_loss: 0.9583 | d_X_loss:
0.5955 | d_fake_loss: 1.5538 | g_loss: 0.5526
Iteration [ 7100/10000] | d_real_loss: 0.7179 | d_Y_loss: 0.9422 | d_X_loss:
0.7345 | d_fake_loss: 1.6767 | g_loss: 0.5475
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007100-
Y-X.png
Iteration [ 7110/10000] | d_real_loss: 0.5645 | d_Y_loss: 1.0558 | d_X_loss:
0.6919 | d_fake_loss: 1.7477 | g_loss: 0.4881
Iteration [ 7120/10000] | d_real_loss: 0.6740 | d_Y_loss: 1.0244 | d_X_loss:
0.8280 | d_fake_loss: 1.8524 | g_loss: 0.4911
Iteration [ 7130/10000] | d_real_loss: 0.6479 | d_Y_loss: 1.0237 | d_X_loss:
0.9125 | d_fake_loss: 1.9362 | g_loss: 0.5127
Iteration [ 7140/10000] | d_real_loss: 0.5531 | d_Y_loss: 0.9961 | d_X_loss:
0.6947 | d_fake_loss: 1.6908 | g_loss: 0.5236
Iteration [ 7150/10000] | d_real_loss: 0.6380 | d_Y_loss: 0.9939 | d_X_loss:
0.5066 | d_fake_loss: 1.5004 | g_loss: 0.5595
Iteration [ 7160/10000] | d_real_loss: 0.8483 | d_Y_loss: 1.0156 | d_X_loss:
```

```
0.5682 | d_fake_loss: 1.5838 | g_loss: 0.5119
Iteration [ 7170/10000] | d_real_loss: 0.6382 | d_Y_loss: 1.0316 | d_X_loss:
0.6203 | d_fake_loss: 1.6519 | g_loss: 0.5051
Iteration [ 7180/10000] | d_real_loss: 0.6120 | d_Y_loss: 1.0624 | d_X_loss:
0.7420 | d_fake_loss: 1.8044 | g_loss: 0.4811
Iteration [ 7190/10000] | d_real_loss: 0.5986 | d_Y_loss: 0.9771 | d_X_loss:
0.5608 | d_fake_loss: 1.5379 | g_loss: 0.5227
Iteration [ 7200/10000] | d_real_loss: 0.7187 | d_Y_loss: 1.0315 | d_X_loss:
0.6551 | d_fake_loss: 1.6866 | g_loss: 0.4997
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007200-
Y-X.png
Iteration [ 7210/10000] | d_real_loss: 0.6501 | d_Y_loss: 1.0010 | d_X_loss:
0.6025 | d_fake_loss: 1.6035 | g_loss: 0.5155
Iteration [ 7220/10000] | d_real_loss: 0.5757 | d_Y_loss: 0.9922 | d_X_loss:
0.5285 | d_fake_loss: 1.5207 | g_loss: 0.5164
Iteration [ 7230/10000] | d_real_loss: 0.6657 | d_Y_loss: 0.9814 | d_X_loss:
0.5076 | d_fake_loss: 1.4890 | g_loss: 0.5309
Iteration [ 7240/10000] | d_real_loss: 0.5605 | d_Y_loss: 0.9984 | d_X_loss:
0.8260 | d_fake_loss: 1.8244 | g_loss: 0.5251
Iteration [ 7250/10000] | d_real_loss: 0.7273 | d_Y_loss: 1.0514 | d_X_loss:
0.6647 | d_fake_loss: 1.7160 | g_loss: 0.5070
Iteration [ 7260/10000] | d_real_loss: 0.6852 | d_Y_loss: 0.9609 | d_X_loss:
0.6572 | d_fake_loss: 1.6181 | g_loss: 0.5336
Iteration [ 7270/10000] | d_real_loss: 0.5837 | d_Y_loss: 1.0391 | d_X_loss:
0.6318 | d_fake_loss: 1.6709 | g_loss: 0.4905
Iteration [ 7280/10000] | d_real_loss: 0.6918 | d_Y_loss: 1.0450 | d_X_loss:
0.4455 | d_fake_loss: 1.4904 | g_loss: 0.5086
Iteration [ 7290/10000] | d_real_loss: 0.5245 | d_Y_loss: 0.9405 | d_X_loss:
0.6243 | d_fake_loss: 1.5648 | g_loss: 0.5343
Iteration [ 7300/10000] | d_real_loss: 0.5179 | d_Y_loss: 0.9874 | d_X_loss:
0.6331 | d_fake_loss: 1.6205 | g_loss: 0.5210
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007300-
Y-X.png
Iteration [ 7310/10000] | d_real_loss: 0.6054 | d_Y_loss: 1.0218 | d_X_loss:
1.0557 | d_fake_loss: 2.0775 | g_loss: 0.5366
Iteration [ 7320/10000] | d_real_loss: 0.5082 | d_Y_loss: 1.1047 | d_X_loss:
0.4103 | d_fake_loss: 1.5150 | g_loss: 0.4727
Iteration [ 7330/10000] | d_real_loss: 0.5267 | d_Y_loss: 1.0183 | d_X_loss:
0.6019 | d_fake_loss: 1.6202 | g_loss: 0.5197
Iteration [ 7340/10000] | d_real_loss: 0.6214 | d_Y_loss: 0.9245 | d_X_loss:
0.5340 | d_fake_loss: 1.4585 | g_loss: 0.5723
Iteration [ 7350/10000] | d_real_loss: 0.5944 | d_Y_loss: 1.0940 | d_X_loss:
0.8294 | d_fake_loss: 1.9234 | g_loss: 0.4693
Iteration [ 7360/10000] | d_real_loss: 0.6469 | d_Y_loss: 1.0509 | d_X_loss:
```

```
0.6644 | d_fake_loss: 1.7153 | g_loss: 0.4750
Iteration [ 7370/10000] | d_real_loss: 0.5673 | d_Y_loss: 1.0592 | d_X_loss:
0.6467 | d_fake_loss: 1.7059 | g_loss: 0.4994
Iteration [ 7380/10000] | d_real_loss: 0.5751 | d_Y_loss: 0.9322 | d_X_loss:
0.4427 | d_fake_loss: 1.3749 | g_loss: 0.5481
Iteration [ 7390/10000] | d_real_loss: 0.5815 | d_Y_loss: 1.0426 | d_X_loss:
0.7428 | d_fake_loss: 1.7854 | g_loss: 0.4997
Iteration [ 7400/10000] | d_real_loss: 0.6840 | d_Y_loss: 0.9275 | d_X_loss:
0.9013 | d_fake_loss: 1.8288 | g_loss: 0.5440
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007400-
Y-X.png
Iteration [ 7410/10000] | d_real_loss: 0.6524 | d_Y_loss: 0.9873 | d_X_loss:
0.7276 | d_fake_loss: 1.7149 | g_loss: 0.5329
Iteration [ 7420/10000] | d_real_loss: 0.5754 | d_Y_loss: 1.1022 | d_X_loss:
0.5886 | d_fake_loss: 1.6908 | g_loss: 0.4769
Iteration [ 7430/10000] | d_real_loss: 0.7325 | d_Y_loss: 1.0296 | d_X_loss:
0.5498 | d_fake_loss: 1.5794 | g_loss: 0.4938
Iteration [ 7440/10000] | d_real_loss: 0.7288 | d_Y_loss: 1.0125 | d_X_loss:
0.5626 | d_fake_loss: 1.5750 | g_loss: 0.5200
Iteration [ 7450/10000] | d_real_loss: 0.6727 | d_Y_loss: 0.9857 | d_X_loss:
0.7094 | d_fake_loss: 1.6952 | g_loss: 0.5445
Iteration [ 7460/10000] | d_real_loss: 0.5940 | d_Y_loss: 0.9884 | d_X_loss:
0.5716 | d_fake_loss: 1.5600 | g_loss: 0.5207
Iteration [ 7470/10000] | d_real_loss: 0.7300 | d_Y_loss: 1.0584 | d_X_loss:
0.7202 | d_fake_loss: 1.7786 | g_loss: 0.4978
Iteration [ 7480/10000] | d_real_loss: 0.6782 | d_Y_loss: 1.0366 | d_X_loss:
0.5309 | d_fake_loss: 1.5675 | g_loss: 0.5079
Iteration [ 7490/10000] | d_real_loss: 0.6190 | d_Y_loss: 0.9981 | d_X_loss:
0.6713 | d_fake_loss: 1.6693 | g_loss: 0.5368
Iteration [ 7500/10000] | d_real_loss: 0.6359 | d_Y_loss: 1.0625 | d_X_loss:
0.7533 | d_fake_loss: 1.8158 | g_loss: 0.5104
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007500-
Y-X.png
Iteration [ 7510/10000] | d_real_loss: 0.7656 | d_Y_loss: 1.0501 | d_X_loss:
0.6231 | d_fake_loss: 1.6732 | g_loss: 0.4924
Iteration [ 7520/10000] | d_real_loss: 0.5889 | d_Y_loss: 1.0640 | d_X_loss:
0.4844 | d_fake_loss: 1.5484 | g_loss: 0.4876
Iteration [ 7530/10000] | d_real_loss: 0.6000 | d_Y_loss: 0.9563 | d_X_loss:
0.5610 | d_fake_loss: 1.5173 | g_loss: 0.5378
Iteration [ 7540/10000] | d_real_loss: 0.6278 | d_Y_loss: 0.9874 | d_X_loss:
0.8370 | d_fake_loss: 1.8244 | g_loss: 0.5263
Iteration [ 7550/10000] | d_real_loss: 0.5591 | d_Y_loss: 0.9407 | d_X_loss:
0.5872 | d_fake_loss: 1.5279 | g_loss: 0.5431
Iteration [ 7560/10000] | d_real_loss: 0.5610 | d_Y_loss: 1.0126 | d_X_loss:
```

```
0.3776 | d_fake_loss: 1.3901 | g_loss: 0.5263
Iteration [ 7570/10000] | d_real_loss: 0.5832 | d_Y_loss: 1.0922 | d_X_loss:
0.7322 | d_fake_loss: 1.8244 | g_loss: 0.5049
Iteration [ 7580/10000] | d_real_loss: 0.7823 | d_Y_loss: 0.9419 | d_X_loss:
0.5254 | d_fake_loss: 1.4673 | g_loss: 0.5590
Iteration [ 7590/10000] | d_real_loss: 0.6496 | d_Y_loss: 1.0485 | d_X_loss:
0.7120 | d_fake_loss: 1.7605 | g_loss: 0.4909
Iteration [ 7600/10000] | d_real_loss: 0.5820 | d_Y_loss: 1.0088 | d_X_loss:
1.0000 | d_fake_loss: 2.0088 | g_loss: 0.5046
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007600-
Y-X.png
Iteration [ 7610/10000] | d_real_loss: 0.6949 | d_Y_loss: 0.9979 | d_X_loss:
0.6489 | d_fake_loss: 1.6468 | g_loss: 0.5577
Iteration [ 7620/10000] | d_real_loss: 0.6231 | d_Y_loss: 0.9989 | d_X_loss:
0.6780 | d_fake_loss: 1.6770 | g_loss: 0.5309
Iteration [ 7630/10000] | d_real_loss: 0.6009 | d_Y_loss: 0.9887 | d_X_loss:
0.5736 | d_fake_loss: 1.5623 | g_loss: 0.5302
Iteration [ 7640/10000] | d_real_loss: 0.7022 | d_Y_loss: 0.9949 | d_X_loss:
0.7195 | d_fake_loss: 1.7144 | g_loss: 0.5399
Iteration [ 7650/10000] | d_real_loss: 0.5735 | d_Y_loss: 0.9509 | d_X_loss:
0.6909 | d_fake_loss: 1.6419 | g_loss: 0.5509
Iteration [ 7660/10000] | d_real_loss: 0.5801 | d_Y_loss: 0.9747 | d_X_loss:
0.6440 | d_fake_loss: 1.6188 | g_loss: 0.5301
Iteration [ 7670/10000] | d_real_loss: 0.5998 | d_Y_loss: 0.9595 | d_X_loss:
0.4734 | d_fake_loss: 1.4329 | g_loss: 0.5487
Iteration [ 7680/10000] | d_real_loss: 0.6793 | d_Y_loss: 1.0504 | d_X_loss:
0.7223 | d_fake_loss: 1.7727 | g_loss: 0.5380
Iteration [ 7690/10000] | d_real_loss: 0.5885 | d_Y_loss: 0.9619 | d_X_loss:
0.4316 | d_fake_loss: 1.3934 | g_loss: 0.5372
Iteration [ 7700/10000] | d_real_loss: 0.5603 | d_Y_loss: 1.0426 | d_X_loss:
0.5898 | d_fake_loss: 1.6324 | g_loss: 0.4929
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007700-
Y-X.png
Iteration [ 7710/10000] | d_real_loss: 0.7330 | d_Y_loss: 0.9676 | d_X_loss:
0.8664 | d_fake_loss: 1.8341 | g_loss: 0.5404
Iteration [ 7720/10000] | d_real_loss: 0.5964 | d_Y_loss: 0.9957 | d_X_loss:
0.6321 | d_fake_loss: 1.6278 | g_loss: 0.5243
Iteration [ 7730/10000] | d_real_loss: 0.5732 | d_Y_loss: 1.0289 | d_X_loss:
0.4821 | d_fake_loss: 1.5110 | g_loss: 0.5310
Iteration [ 7740/10000] | d_real_loss: 0.7204 | d_Y_loss: 0.9936 | d_X_loss:
0.5213 | d_fake_loss: 1.5150 | g_loss: 0.5332
Iteration [ 7750/10000] | d_real_loss: 0.6279 | d_Y_loss: 0.9622 | d_X_loss:
0.4685 | d_fake_loss: 1.4307 | g_loss: 0.5494
Iteration [ 7760/10000] | d_real_loss: 0.5975 | d_Y_loss: 1.0023 | d_X_loss:
```

```
0.5198 | d_fake_loss: 1.5222 | g_loss: 0.5327
Iteration [ 7770/10000] | d_real_loss: 0.4822 | d_Y_loss: 1.0424 | d_X_loss:
0.3559 | d_fake_loss: 1.3984 | g_loss: 0.5047
Iteration [ 7780/10000] | d_real_loss: 0.6084 | d_Y_loss: 0.9301 | d_X_loss:
0.4819 | d_fake_loss: 1.4121 | g_loss: 0.5697
Iteration [ 7790/10000] | d_real_loss: 0.6539 | d_Y_loss: 1.0166 | d_X_loss:
0.5593 | d_fake_loss: 1.5760 | g_loss: 0.5289
Iteration [ 7800/10000] | d_real_loss: 0.6669 | d_Y_loss: 0.9879 | d_X_loss:
0.7667 | d_fake_loss: 1.7546 | g_loss: 0.5227
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007800-
Y-X.png
Iteration [ 7810/10000] | d_real_loss: 0.6058 | d_Y_loss: 0.9904 | d_X_loss:
0.8883 | d_fake_loss: 1.8786 | g_loss: 0.5365
Iteration [ 7820/10000] | d_real_loss: 0.6694 | d_Y_loss: 0.9867 | d_X_loss:
0.5538 | d_fake_loss: 1.5405 | g_loss: 0.5426
Iteration [ 7830/10000] | d_real_loss: 0.5673 | d_Y_loss: 1.0119 | d_X_loss:
0.4496 | d_fake_loss: 1.4615 | g_loss: 0.5118
Iteration [ 7840/10000] | d_real_loss: 0.6860 | d_Y_loss: 0.9824 | d_X_loss:
0.6704 | d_fake_loss: 1.6528 | g_loss: 0.5299
Iteration [ 7850/10000] | d_real_loss: 0.8299 | d_Y_loss: 0.9396 | d_X_loss:
0.5680 | d_fake_loss: 1.5076 | g_loss: 0.5519
Iteration [ 7860/10000] | d_real_loss: 0.6782 | d_Y_loss: 1.0203 | d_X_loss:
0.7221 | d_fake_loss: 1.7424 | g_loss: 0.5382
Iteration [ 7870/10000] | d_real_loss: 0.5847 | d_Y_loss: 0.9929 | d_X_loss:
0.3302 | d_fake_loss: 1.3232 | g_loss: 0.5262
Iteration [ 7880/10000] | d_real_loss: 0.8456 | d_Y_loss: 0.9926 | d_X_loss:
0.8449 | d_fake_loss: 1.8375 | g_loss: 0.5159
Iteration [ 7890/10000] | d_real_loss: 0.5696 | d_Y_loss: 0.9811 | d_X_loss:
0.5618 | d_fake_loss: 1.5429 | g_loss: 0.5406
Iteration [ 7900/10000] | d_real_loss: 0.6683 | d_Y_loss: 1.3185 | d_X_loss:
0.5554 | d_fake_loss: 1.8740 | g_loss: 0.4640
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-007900-
Y-X.png
Iteration [ 7910/10000] | d_real_loss: 0.7436 | d_Y_loss: 0.9328 | d_X_loss:
0.4682 | d_fake_loss: 1.4010 | g_loss: 0.5568
Iteration [ 7920/10000] | d_real_loss: 0.6038 | d_Y_loss: 1.0397 | d_X_loss:
0.6399 | d_fake_loss: 1.6796 | g_loss: 0.5275
Iteration [ 7930/10000] | d_real_loss: 0.5486 | d_Y_loss: 1.0052 | d_X_loss:
0.7424 | d_fake_loss: 1.7475 | g_loss: 0.5213
Iteration [ 7940/10000] | d_real_loss: 0.6451 | d_Y_loss: 1.0208 | d_X_loss:
0.6053 | d_fake_loss: 1.6261 | g_loss: 0.5156
Iteration [ 7950/10000] | d_real_loss: 0.6162 | d_Y_loss: 1.0397 | d_X_loss:
0.6138 | d_fake_loss: 1.6535 | g_loss: 0.5290
Iteration [ 7960/10000] | d_real_loss: 0.6318 | d_Y_loss: 1.0118 | d_X_loss:
```

```
0.3754 | d_fake_loss: 1.3872 | g_loss: 0.5156
Iteration [ 7970/10000] | d_real_loss: 0.8000 | d_Y_loss: 1.0055 | d_X_loss:
0.6922 | d_fake_loss: 1.6977 | g_loss: 0.5403
Iteration [ 7980/10000] | d_real_loss: 0.5876 | d_Y_loss: 1.0365 | d_X_loss:
0.6611 | d_fake_loss: 1.6977 | g_loss: 0.5103
Iteration [ 7990/10000] | d_real_loss: 0.6188 | d_Y_loss: 1.0689 | d_X_loss:
0.4711 | d_fake_loss: 1.5400 | g_loss: 0.4929
Iteration [ 8000/10000] | d_real_loss: 0.7479 | d_Y_loss: 0.9611 | d_X_loss:
0.4931 | d_fake_loss: 1.4541 | g_loss: 0.5359
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008000-
Y-X.png
Iteration [ 8010/10000] | d_real_loss: 0.5995 | d_Y_loss: 1.0259 | d_X_loss:
0.4048 | d_fake_loss: 1.4306 | g_loss: 0.5194
Iteration [ 8020/10000] | d_real_loss: 0.5406 | d_Y_loss: 0.8999 | d_X_loss:
0.8310 | d_fake_loss: 1.7309 | g_loss: 0.5659
Iteration [ 8030/10000] | d_real_loss: 0.6022 | d_Y_loss: 0.9937 | d_X_loss:
0.7134 | d_fake_loss: 1.7071 | g_loss: 0.5227
Iteration [ 8040/10000] | d_real_loss: 0.7177 | d_Y_loss: 1.0799 | d_X_loss:
0.6475 | d_fake_loss: 1.7274 | g_loss: 0.5132
Iteration [ 8050/10000] | d_real_loss: 0.7437 | d_Y_loss: 1.0272 | d_X_loss:
0.5645 | d_fake_loss: 1.5916 | g_loss: 0.5015
Iteration [ 8060/10000] | d_real_loss: 0.5943 | d_Y_loss: 1.0516 | d_X_loss:
0.4794 | d_fake_loss: 1.5310 | g_loss: 0.4882
Iteration [ 8070/10000] | d_real_loss: 0.5671 | d_Y_loss: 1.0197 | d_X_loss:
0.5813 | d_fake_loss: 1.6009 | g_loss: 0.5170
Iteration [ 8080/10000] | d_real_loss: 0.6888 | d_Y_loss: 0.9421 | d_X_loss:
0.6243 | d_fake_loss: 1.5663 | g_loss: 0.5551
Iteration [ 8090/10000] | d_real_loss: 0.7031 | d_Y_loss: 0.9849 | d_X_loss:
0.5462 | d_fake_loss: 1.5311 | g_loss: 0.5811
Iteration [ 8100/10000] | d_real_loss: 0.5925 | d_Y_loss: 1.0949 | d_X_loss:
0.7435 | d_fake_loss: 1.8384 | g_loss: 0.4787
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008100-
Y-X.png
Iteration [ 8110/10000] | d_real_loss: 0.6720 | d_Y_loss: 1.0870 | d_X_loss:
0.5809 | d_fake_loss: 1.6679 | g_loss: 0.4982
Iteration [ 8120/10000] | d_real_loss: 0.6497 | d_Y_loss: 0.9547 | d_X_loss:
0.6413 | d_fake_loss: 1.5961 | g_loss: 0.5469
Iteration [ 8130/10000] | d_real_loss: 0.7032 | d_Y_loss: 1.0626 | d_X_loss:
0.5507 | d_fake_loss: 1.6132 | g_loss: 0.5018
Iteration [ 8140/10000] | d_real_loss: 0.6199 | d_Y_loss: 0.9864 | d_X_loss:
0.6104 | d_fake_loss: 1.5969 | g_loss: 0.5205
Iteration [ 8150/10000] | d_real_loss: 0.7031 | d_Y_loss: 1.0382 | d_X_loss:
0.8375 | d_fake_loss: 1.8756 | g_loss: 0.5323
Iteration [ 8160/10000] | d_real_loss: 0.6825 | d_Y_loss: 1.0187 | d_X_loss:
```

```
0.9092 | d_fake_loss: 1.9279 | g_loss: 0.5071
Iteration [ 8170/10000] | d_real_loss: 0.6740 | d_Y_loss: 0.9807 | d_X_loss:
0.7027 | d_fake_loss: 1.6834 | g_loss: 0.5442
Iteration [ 8180/10000] | d_real_loss: 0.5612 | d_Y_loss: 1.0392 | d_X_loss:
0.6451 | d_fake_loss: 1.6844 | g_loss: 0.5229
Iteration [ 8190/10000] | d_real_loss: 0.7279 | d_Y_loss: 1.0884 | d_X_loss:
0.5513 | d_fake_loss: 1.6398 | g_loss: 0.5005
Iteration [ 8200/10000] | d_real_loss: 0.6303 | d_Y_loss: 0.9677 | d_X_loss:
0.4501 | d_fake_loss: 1.4178 | g_loss: 0.5329
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008200-
Y-X.png
Iteration [ 8210/10000] | d_real_loss: 0.6019 | d_Y_loss: 0.9806 | d_X_loss:
0.6442 | d_fake_loss: 1.6248 | g_loss: 0.5208
Iteration [ 8220/10000] | d_real_loss: 0.5947 | d_Y_loss: 1.0312 | d_X_loss:
0.3819 | d_fake_loss: 1.4131 | g_loss: 0.5352
Iteration [ 8230/10000] | d_real_loss: 0.6412 | d_Y_loss: 1.0449 | d_X_loss:
0.6497 | d_fake_loss: 1.6945 | g_loss: 0.5203
Iteration [ 8240/10000] | d_real_loss: 0.6322 | d_Y_loss: 0.9991 | d_X_loss:
0.4240 | d_fake_loss: 1.4231 | g_loss: 0.5371
Iteration [ 8250/10000] | d_real_loss: 0.8063 | d_Y_loss: 1.0124 | d_X_loss:
0.5497 | d_fake_loss: 1.5621 | g_loss: 0.5164
Iteration [ 8260/10000] | d_real_loss: 0.6152 | d_Y_loss: 0.9580 | d_X_loss:
0.6430 | d_fake_loss: 1.6010 | g_loss: 0.5393
Iteration [ 8270/10000] | d_real_loss: 0.6612 | d_Y_loss: 0.9582 | d_X_loss:
0.8119 | d_fake_loss: 1.7701 | g_loss: 0.5453
Iteration [ 8280/10000] | d_real_loss: 0.6315 | d_Y_loss: 0.9505 | d_X_loss:
0.4199 | d_fake_loss: 1.3704 | g_loss: 0.5820
Iteration [ 8290/10000] | d_real_loss: 0.6289 | d_Y_loss: 1.0638 | d_X_loss:
0.5563 | d_fake_loss: 1.6201 | g_loss: 0.4839
Iteration [ 8300/10000] | d_real_loss: 0.8199 | d_Y_loss: 1.0108 | d_X_loss:
0.6391 | d_fake_loss: 1.6499 | g_loss: 0.5248
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008300-
Y-X.png
Iteration [ 8310/10000] | d_real_loss: 0.6186 | d_Y_loss: 1.0136 | d_X_loss:
0.5764 | d_fake_loss: 1.5900 | g_loss: 0.5054
Iteration [ 8320/10000] | d_real_loss: 0.6151 | d_Y_loss: 1.0434 | d_X_loss:
0.6385 | d_fake_loss: 1.6819 | g_loss: 0.5195
Iteration [ 8330/10000] | d_real_loss: 0.6202 | d_Y_loss: 0.9790 | d_X_loss:
0.5022 | d_fake_loss: 1.4812 | g_loss: 0.5269
Iteration [ 8340/10000] | d_real_loss: 0.5936 | d_Y_loss: 1.0952 | d_X_loss:
0.4594 | d_fake_loss: 1.5546 | g_loss: 0.4920
Iteration [ 8350/10000] | d_real_loss: 0.6038 | d_Y_loss: 1.0022 | d_X_loss:
0.5179 | d_fake_loss: 1.5201 | g_loss: 0.5284
Iteration [ 8360/10000] | d_real_loss: 0.6399 | d_Y_loss: 1.1608 | d_X_loss:
```

```
0.4490 | d_fake_loss: 1.6098 | g_loss: 0.4825
Iteration [ 8370/10000] | d_real_loss: 0.5810 | d_Y_loss: 1.0473 | d_X_loss:
0.5169 | d_fake_loss: 1.5642 | g_loss: 0.5055
Iteration [ 8380/10000] | d_real_loss: 0.6112 | d_Y_loss: 0.9972 | d_X_loss:
0.5359 | d_fake_loss: 1.5331 | g_loss: 0.5120
Iteration [ 8390/10000] | d_real_loss: 0.6374 | d_Y_loss: 1.0070 | d_X_loss:
0.6654 | d_fake_loss: 1.6724 | g_loss: 0.5110
Iteration [ 8400/10000] | d_real_loss: 0.5598 | d_Y_loss: 1.0507 | d_X_loss:
0.5461 | d_fake_loss: 1.5968 | g_loss: 0.4921
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008400-
Y-X.png
Iteration [ 8410/10000] | d_real_loss: 0.7006 | d_Y_loss: 1.0154 | d_X_loss:
0.6422 | d_fake_loss: 1.6576 | g_loss: 0.5233
Iteration [ 8420/10000] | d_real_loss: 0.6392 | d_Y_loss: 0.9135 | d_X_loss:
0.4491 | d_fake_loss: 1.3626 | g_loss: 0.5717
Iteration [ 8430/10000] | d_real_loss: 0.7135 | d_Y_loss: 1.0176 | d_X_loss:
0.6314 | d_fake_loss: 1.6490 | g_loss: 0.5361
Iteration [ 8440/10000] | d_real_loss: 0.7148 | d_Y_loss: 1.0044 | d_X_loss:
0.5612 | d_fake_loss: 1.5657 | g_loss: 0.5129
Iteration [ 8450/10000] | d_real_loss: 0.6401 | d_Y_loss: 1.0379 | d_X_loss:
0.6617 | d_fake_loss: 1.6996 | g_loss: 0.5176
Iteration [ 8460/10000] | d_real_loss: 0.5460 | d_Y_loss: 1.0312 | d_X_loss:
0.5505 | d_fake_loss: 1.5816 | g_loss: 0.5084
Iteration [ 8470/10000] | d_real_loss: 0.6894 | d_Y_loss: 0.9745 | d_X_loss:
0.4228 | d_fake_loss: 1.3974 | g_loss: 0.5316
Iteration [ 8480/10000] | d_real_loss: 0.6102 | d_Y_loss: 0.9603 | d_X_loss:
0.8119 | d_fake_loss: 1.7722 | g_loss: 0.5363
Iteration [ 8490/10000] | d_real_loss: 0.7896 | d_Y_loss: 1.1105 | d_X_loss:
0.4980 | d_fake_loss: 1.6085 | g_loss: 0.5216
Iteration [ 8500/10000] | d_real_loss: 0.6353 | d_Y_loss: 0.9761 | d_X_loss:
0.6506 | d_fake_loss: 1.6267 | g_loss: 0.5483
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008500-
Y-X.png
Iteration [ 8510/10000] | d_real_loss: 0.5919 | d_Y_loss: 1.0243 | d_X_loss:
0.6490 | d_fake_loss: 1.6733 | g_loss: 0.5058
Iteration [ 8520/10000] | d_real_loss: 0.6035 | d_Y_loss: 1.0483 | d_X_loss:
0.3527 | d_fake_loss: 1.4010 | g_loss: 0.5043
Iteration [ 8530/10000] | d_real_loss: 0.6314 | d_Y_loss: 1.0347 | d_X_loss:
0.8373 | d_fake_loss: 1.8720 | g_loss: 0.5176
Iteration [ 8540/10000] | d_real_loss: 0.6627 | d_Y_loss: 0.9594 | d_X_loss:
0.8822 | d_fake_loss: 1.8415 | g_loss: 0.5612
Iteration [ 8550/10000] | d_real_loss: 0.7123 | d_Y_loss: 0.9609 | d_X_loss:
0.3648 | d_fake_loss: 1.3258 | g_loss: 0.5519
Iteration [ 8560/10000] | d_real_loss: 0.6533 | d_Y_loss: 0.9860 | d_X_loss:
```

```
0.8976 | d_fake_loss: 1.8836 | g_loss: 0.5194
Iteration [ 8570/10000] | d_real_loss: 0.6435 | d_Y_loss: 1.1036 | d_X_loss:
0.7720 | d_fake_loss: 1.8756 | g_loss: 0.5105
Iteration [ 8580/10000] | d_real_loss: 0.5501 | d_Y_loss: 1.0803 | d_X_loss:
0.5122 | d_fake_loss: 1.5925 | g_loss: 0.4947
Iteration [ 8590/10000] | d_real_loss: 0.6345 | d_Y_loss: 0.9707 | d_X_loss:
0.5949 | d_fake_loss: 1.5656 | g_loss: 0.5419
Iteration [ 8600/10000] | d_real_loss: 0.6114 | d_Y_loss: 1.0398 | d_X_loss:
0.4891 | d_fake_loss: 1.5289 | g_loss: 0.5234
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008600-
Y-X.png
Iteration [ 8610/10000] | d_real_loss: 0.6858 | d_Y_loss: 1.0346 | d_X_loss:
0.6698 | d_fake_loss: 1.7044 | g_loss: 0.5307
Iteration [ 8620/10000] | d_real_loss: 0.6625 | d_Y_loss: 1.0313 | d_X_loss:
0.8967 | d_fake_loss: 1.9279 | g_loss: 0.5232
Iteration [ 8630/10000] | d_real_loss: 0.6270 | d_Y_loss: 0.9555 | d_X_loss:
0.6715 | d_fake_loss: 1.6270 | g_loss: 0.5525
Iteration [ 8640/10000] | d_real_loss: 0.5640 | d_Y_loss: 0.9204 | d_X_loss:
0.4905 | d_fake_loss: 1.4109 | g_loss: 0.5791
Iteration [ 8650/10000] | d_real_loss: 0.5869 | d_Y_loss: 0.9901 | d_X_loss:
0.7200 | d_fake_loss: 1.7101 | g_loss: 0.5464
Iteration [ 8660/10000] | d_real_loss: 0.6752 | d_Y_loss: 0.9712 | d_X_loss:
0.5071 | d_fake_loss: 1.4783 | g_loss: 0.5266
Iteration [ 8670/10000] | d_real_loss: 0.6408 | d_Y_loss: 1.1895 | d_X_loss:
0.9450 | d_fake_loss: 2.1345 | g_loss: 0.5055
Iteration [ 8680/10000] | d_real_loss: 0.5795 | d_Y_loss: 1.0171 | d_X_loss:
0.5271 | d_fake_loss: 1.5442 | g_loss: 0.5249
Iteration [ 8690/10000] | d_real_loss: 0.6008 | d_Y_loss: 1.0037 | d_X_loss:
0.4446 | d_fake_loss: 1.4483 | g_loss: 0.5155
Iteration [ 8700/10000] | d_real_loss: 0.5583 | d_Y_loss: 1.0900 | d_X_loss:
0.9101 | d_fake_loss: 2.0002 | g_loss: 0.4995
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008700-
Y-X.png
Iteration [ 8710/10000] | d_real_loss: 0.6486 | d_Y_loss: 0.9153 | d_X_loss:
0.5496 | d_fake_loss: 1.4648 | g_loss: 0.5735
Iteration [ 8720/10000] | d_real_loss: 0.7401 | d_Y_loss: 1.0827 | d_X_loss:
0.8250 | d_fake_loss: 1.9077 | g_loss: 0.4882
Iteration [ 8730/10000] | d_real_loss: 0.5921 | d_Y_loss: 1.0166 | d_X_loss:
0.8268 | d_fake_loss: 1.8434 | g_loss: 0.5119
Iteration [ 8740/10000] | d_real_loss: 0.6659 | d_Y_loss: 0.9666 | d_X_loss:
0.5060 | d_fake_loss: 1.4726 | g_loss: 0.5373
Iteration [ 8750/10000] | d_real_loss: 0.6268 | d_Y_loss: 1.0649 | d_X_loss:
0.5605 | d_fake_loss: 1.6254 | g_loss: 0.4828
Iteration [ 8760/10000] | d_real_loss: 0.6159 | d_Y_loss: 1.0136 | d_X_loss:
```

```
0.5808 | d_fake_loss: 1.5944 | g_loss: 0.5208
Iteration [ 8770/10000] | d_real_loss: 0.5929 | d_Y_loss: 0.9821 | d_X_loss:
0.5814 | d_fake_loss: 1.5635 | g_loss: 0.5378
Iteration [ 8780/10000] | d_real_loss: 0.6426 | d_Y_loss: 0.9637 | d_X_loss:
0.5778 | d_fake_loss: 1.5415 | g_loss: 0.5376
Iteration [ 8790/10000] | d_real_loss: 0.6658 | d_Y_loss: 0.9689 | d_X_loss:
0.5064 | d_fake_loss: 1.4753 | g_loss: 0.5531
Iteration [ 8800/10000] | d_real_loss: 0.7135 | d_Y_loss: 1.0041 | d_X_loss:
0.6768 | d_fake_loss: 1.6809 | g_loss: 0.5562
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008800-
Y-X.png
Iteration [ 8810/10000] | d_real_loss: 0.5089 | d_Y_loss: 1.0189 | d_X_loss:
0.7174 | d_fake_loss: 1.7364 | g_loss: 0.5120
Iteration [ 8820/10000] | d_real_loss: 0.5615 | d_Y_loss: 0.9925 | d_X_loss:
0.8090 | d_fake_loss: 1.8015 | g_loss: 0.5284
Iteration [ 8830/10000] | d_real_loss: 0.6581 | d_Y_loss: 0.9853 | d_X_loss:
0.6433 | d_fake_loss: 1.6286 | g_loss: 0.5442
Iteration [ 8840/10000] | d_real_loss: 0.6073 | d_Y_loss: 0.9384 | d_X_loss:
0.4828 | d_fake_loss: 1.4212 | g_loss: 0.5549
Iteration [ 8850/10000] | d_real_loss: 0.5380 | d_Y_loss: 1.0355 | d_X_loss:
0.6884 | d_fake_loss: 1.7238 | g_loss: 0.5101
Iteration [ 8860/10000] | d_real_loss: 0.7514 | d_Y_loss: 0.9550 | d_X_loss:
0.7761 | d_fake_loss: 1.7311 | g_loss: 0.5483
Iteration [ 8870/10000] | d_real_loss: 0.6218 | d_Y_loss: 0.9781 | d_X_loss:
0.5701 | d_fake_loss: 1.5482 | g_loss: 0.5474
Iteration [ 8880/10000] | d_real_loss: 0.7805 | d_Y_loss: 1.0043 | d_X_loss:
0.5899 | d_fake_loss: 1.5942 | g_loss: 0.5254
Iteration [ 8890/10000] | d_real_loss: 0.6653 | d_Y_loss: 0.9974 | d_X_loss:
0.6818 | d_fake_loss: 1.6792 | g_loss: 0.5383
Iteration [ 8900/10000] | d_real_loss: 0.7374 | d_Y_loss: 1.0074 | d_X_loss:
0.6539 | d_fake_loss: 1.6613 | g_loss: 0.5180
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-008900-
Y-X.png
Iteration [ 8910/10000] | d_real_loss: 0.5808 | d_Y_loss: 0.9622 | d_X_loss:
0.6475 | d_fake_loss: 1.6097 | g_loss: 0.5596
Iteration [ 8920/10000] | d_real_loss: 0.6060 | d_Y_loss: 0.9655 | d_X_loss:
0.6468 | d_fake_loss: 1.6123 | g_loss: 0.5581
Iteration [ 8930/10000] | d_real_loss: 0.6059 | d_Y_loss: 0.9303 | d_X_loss:
0.3783 | d_fake_loss: 1.3085 | g_loss: 0.5594
Iteration [ 8940/10000] | d_real_loss: 0.5767 | d_Y_loss: 1.0599 | d_X_loss:
0.6099 | d_fake_loss: 1.6698 | g_loss: 0.5212
Iteration [ 8950/10000] | d_real_loss: 0.7143 | d_Y_loss: 1.0191 | d_X_loss:
0.8907 | d_fake_loss: 1.9098 | g_loss: 0.5113
Iteration [ 8960/10000] | d_real_loss: 0.5707 | d_Y_loss: 0.9973 | d_X_loss:
```

```
0.6363 | d_fake_loss: 1.6336 | g_loss: 0.5314
Iteration [ 8970/10000] | d_real_loss: 0.6330 | d_Y_loss: 1.0024 | d_X_loss:
0.6242 | d_fake_loss: 1.6266 | g_loss: 0.5622
Iteration [ 8980/10000] | d_real_loss: 0.5230 | d_Y_loss: 1.0114 | d_X_loss:
0.4252 | d_fake_loss: 1.4365 | g_loss: 0.5256
Iteration [ 8990/10000] | d_real_loss: 0.7360 | d_Y_loss: 0.9728 | d_X_loss:
0.6480 | d_fake_loss: 1.6208 | g_loss: 0.5367
Iteration [ 9000/10000] | d_real_loss: 0.6791 | d_Y_loss: 1.0222 | d_X_loss:
0.6788 | d_fake_loss: 1.7009 | g_loss: 0.5102
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009000-
Y-X.png
Iteration [ 9010/10000] | d_real_loss: 0.5567 | d_Y_loss: 0.9971 | d_X_loss:
0.5452 | d_fake_loss: 1.5423 | g_loss: 0.5348
Iteration [ 9020/10000] | d_real_loss: 0.6815 | d_Y_loss: 1.0071 | d_X_loss:
0.5362 | d_fake_loss: 1.5433 | g_loss: 0.5109
Iteration [ 9030/10000] | d_real_loss: 0.6215 | d_Y_loss: 1.0433 | d_X_loss:
0.4344 | d_fake_loss: 1.4776 | g_loss: 0.4999
Iteration [ 9040/10000] | d_real_loss: 0.6320 | d_Y_loss: 1.0089 | d_X_loss:
0.5089 | d_fake_loss: 1.5178 | g_loss: 0.5184
Iteration [ 9050/10000] | d_real_loss: 0.9210 | d_Y_loss: 0.9189 | d_X_loss:
1.0320 | d_fake_loss: 1.9509 | g_loss: 0.5894
Iteration [ 9060/10000] | d_real_loss: 0.6183 | d_Y_loss: 0.9377 | d_X_loss:
0.5633 | d_fake_loss: 1.5009 | g_loss: 0.5519
Iteration [ 9070/10000] | d_real_loss: 0.6755 | d_Y_loss: 1.0244 | d_X_loss:
0.6240 | d_fake_loss: 1.6484 | g_loss: 0.5190
Iteration [ 9080/10000] | d_real_loss: 0.5888 | d_Y_loss: 1.0341 | d_X_loss:
0.6682 | d_fake_loss: 1.7023 | g_loss: 0.5676
Iteration [ 9090/10000] | d_real_loss: 0.7061 | d_Y_loss: 1.0985 | d_X_loss:
0.5756 | d_fake_loss: 1.6741 | g_loss: 0.4778
Iteration [ 9100/10000] | d_real_loss: 0.6448 | d_Y_loss: 0.9251 | d_X_loss:
0.8949 | d_fake_loss: 1.8200 | g_loss: 0.5572
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009100-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009100-
Y-X.png
Iteration [ 9110/10000] | d_real_loss: 0.6088 | d_Y_loss: 1.0170 | d_X_loss:
0.4797 | d_fake_loss: 1.4967 | g_loss: 0.5216
Iteration [ 9120/10000] | d_real_loss: 0.5257 | d_Y_loss: 1.0308 | d_X_loss:
0.5604 | d_fake_loss: 1.5913 | g_loss: 0.5189
Iteration [ 9130/10000] | d_real_loss: 0.5313 | d_Y_loss: 0.9697 | d_X_loss:
0.6944 | d_fake_loss: 1.6641 | g_loss: 0.5457
Iteration [ 9140/10000] | d_real_loss: 0.4908 | d_Y_loss: 1.0155 | d_X_loss:
0.5760 | d_fake_loss: 1.5915 | g_loss: 0.5236
Iteration [ 9150/10000] | d_real_loss: 0.5211 | d_Y_loss: 0.9958 | d_X_loss:
0.4059 | d_fake_loss: 1.4017 | g_loss: 0.5273
Iteration [ 9160/10000] | d_real_loss: 0.6269 | d_Y_loss: 1.0339 | d_X_loss:
```

```
0.5634 | d_fake_loss: 1.5974 | g_loss: 0.5211
Iteration [ 9170/10000] | d_real_loss: 0.6986 | d_Y_loss: 1.0204 | d_X_loss:
0.6869 | d_fake_loss: 1.7073 | g_loss: 0.5317
Iteration [ 9180/10000] | d_real_loss: 0.6344 | d_Y_loss: 0.9374 | d_X_loss:
0.5111 | d_fake_loss: 1.4485 | g_loss: 0.5723
Iteration [ 9190/10000] | d_real_loss: 0.5619 | d_Y_loss: 0.9919 | d_X_loss:
0.4516 | d_fake_loss: 1.4435 | g_loss: 0.5198
Iteration [ 9200/10000] | d_real_loss: 0.5742 | d_Y_loss: 1.0230 | d_X_loss:
0.3830 | d_fake_loss: 1.4059 | g_loss: 0.5130
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009200-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009200-
Y-X.png
Iteration [ 9210/10000] | d_real_loss: 0.5190 | d_Y_loss: 1.0408 | d_X_loss:
0.7377 | d_fake_loss: 1.7784 | g_loss: 0.5120
Iteration [ 9220/10000] | d_real_loss: 0.8296 | d_Y_loss: 0.9904 | d_X_loss:
0.4643 | d_fake_loss: 1.4547 | g_loss: 0.5342
Iteration [ 9230/10000] | d_real_loss: 0.7592 | d_Y_loss: 0.9520 | d_X_loss:
0.6202 | d_fake_loss: 1.5721 | g_loss: 0.5430
Iteration [ 9240/10000] | d_real_loss: 0.6210 | d_Y_loss: 1.0064 | d_X_loss:
0.6264 | d_fake_loss: 1.6328 | g_loss: 0.5161
Iteration [ 9250/10000] | d_real_loss: 0.5387 | d_Y_loss: 1.0014 | d_X_loss:
0.4829 | d_fake_loss: 1.4843 | g_loss: 0.5297
Iteration [ 9260/10000] | d_real_loss: 0.5616 | d_Y_loss: 1.0017 | d_X_loss:
0.7425 | d_fake_loss: 1.7441 | g_loss: 0.5502
Iteration [ 9270/10000] | d_real_loss: 0.5687 | d_Y_loss: 0.9639 | d_X_loss:
0.3214 | d_fake_loss: 1.2853 | g_loss: 0.5376
Iteration [ 9280/10000] | d_real_loss: 0.5384 | d_Y_loss: 1.0011 | d_X_loss:
0.5592 | d_fake_loss: 1.5602 | g_loss: 0.5397
Iteration [ 9290/10000] | d_real_loss: 0.5677 | d_Y_loss: 1.0571 | d_X_loss:
0.4856 | d_fake_loss: 1.5427 | g_loss: 0.5070
Iteration [ 9300/10000] | d_real_loss: 0.5027 | d_Y_loss: 0.9763 | d_X_loss:
0.5767 | d_fake_loss: 1.5531 | g_loss: 0.5467
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009300-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009300-
Y-X.png
Iteration [ 9310/10000] | d_real_loss: 0.7006 | d_Y_loss: 0.9596 | d_X_loss:
0.9454 | d_fake_loss: 1.9050 | g_loss: 0.5579
Iteration [ 9320/10000] | d_real_loss: 0.5606 | d_Y_loss: 1.0224 | d_X_loss:
0.5008 | d_fake_loss: 1.5232 | g_loss: 0.5206
Iteration [ 9330/10000] | d_real_loss: 0.6395 | d_Y_loss: 0.9476 | d_X_loss:
0.5941 | d_fake_loss: 1.5418 | g_loss: 0.5511
Iteration [ 9340/10000] | d_real_loss: 0.4823 | d_Y_loss: 1.0314 | d_X_loss:
0.6230 | d_fake_loss: 1.6543 | g_loss: 0.5308
Iteration [ 9350/10000] | d_real_loss: 0.6902 | d_Y_loss: 1.0476 | d_X_loss:
0.5045 | d_fake_loss: 1.5520 | g_loss: 0.5247
Iteration [ 9360/10000] | d_real_loss: 0.6306 | d_Y_loss: 1.0047 | d_X_loss:
```

```
0.5683 | d_fake_loss: 1.5731 | g_loss: 0.5329
Iteration [ 9370/10000] | d_real_loss: 0.5831 | d_Y_loss: 0.9953 | d_X_loss:
0.8898 | d_fake_loss: 1.8851 | g_loss: 0.5465
Iteration [ 9380/10000] | d_real_loss: 0.5715 | d_Y_loss: 0.9642 | d_X_loss:
0.4639 | d_fake_loss: 1.4281 | g_loss: 0.5549
Iteration [ 9390/10000] | d_real_loss: 0.7431 | d_Y_loss: 1.0475 | d_X_loss:
0.7388 | d_fake_loss: 1.7862 | g_loss: 0.5104
Iteration [ 9400/10000] | d_real_loss: 0.5490 | d_Y_loss: 1.0318 | d_X_loss:
0.6036 | d_fake_loss: 1.6354 | g_loss: 0.4984
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009400-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009400-
Y-X.png
Iteration [ 9410/10000] | d_real_loss: 0.5756 | d_Y_loss: 0.9158 | d_X_loss:
0.6800 | d_fake_loss: 1.5958 | g_loss: 0.5807
Iteration [ 9420/10000] | d_real_loss: 0.6311 | d_Y_loss: 1.0164 | d_X_loss:
0.6445 | d_fake_loss: 1.6609 | g_loss: 0.5276
Iteration [ 9430/10000] | d_real_loss: 0.6424 | d_Y_loss: 1.0561 | d_X_loss:
0.5691 | d_fake_loss: 1.6252 | g_loss: 0.5145
Iteration [ 9440/10000] | d_real_loss: 0.6377 | d_Y_loss: 1.0421 | d_X_loss:
1.0432 | d_fake_loss: 2.0854 | g_loss: 0.4999
Iteration [ 9450/10000] | d_real_loss: 0.7688 | d_Y_loss: 1.0091 | d_X_loss:
0.6047 | d_fake_loss: 1.6138 | g_loss: 0.5179
Iteration [ 9460/10000] | d_real_loss: 0.6818 | d_Y_loss: 1.0235 | d_X_loss:
0.6040 | d_fake_loss: 1.6275 | g_loss: 0.5237
Iteration [ 9470/10000] | d_real_loss: 0.5440 | d_Y_loss: 1.0624 | d_X_loss:
0.6877 | d_fake_loss: 1.7501 | g_loss: 0.5107
Iteration [ 9480/10000] | d_real_loss: 0.6280 | d_Y_loss: 0.9827 | d_X_loss:
0.5429 | d_fake_loss: 1.5256 | g_loss: 0.5476
Iteration [ 9490/10000] | d_real_loss: 0.4993 | d_Y_loss: 0.9282 | d_X_loss:
0.8274 | d_fake_loss: 1.7555 | g_loss: 0.5630
Iteration [ 9500/10000] | d_real_loss: 0.7445 | d_Y_loss: 1.1888 | d_X_loss:
0.6244 | d_fake_loss: 1.8133 | g_loss: 0.4586
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009500-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009500-
Y-X.png
Iteration [ 9510/10000] | d_real_loss: 0.7209 | d_Y_loss: 1.0173 | d_X_loss:
0.5816 | d_fake_loss: 1.5989 | g_loss: 0.5189
Iteration [ 9520/10000] | d_real_loss: 0.6715 | d_Y_loss: 0.9723 | d_X_loss:
0.4525 | d_fake_loss: 1.4248 | g_loss: 0.5669
Iteration [ 9530/10000] | d_real_loss: 0.5906 | d_Y_loss: 0.9811 | d_X_loss:
0.6133 | d_fake_loss: 1.5944 | g_loss: 0.5374
Iteration [ 9540/10000] | d_real_loss: 0.6245 | d_Y_loss: 1.0407 | d_X_loss:
0.5538 | d_fake_loss: 1.5945 | g_loss: 0.5262
Iteration [ 9550/10000] | d_real_loss: 0.5910 | d_Y_loss: 0.9760 | d_X_loss:
0.6536 | d_fake_loss: 1.6296 | g_loss: 0.5581
Iteration [ 9560/10000] | d_real_loss: 0.7588 | d_Y_loss: 1.0249 | d_X_loss:
```

```
0.5024 | d_fake_loss: 1.5273 | g_loss: 0.5131
Iteration [ 9570/10000] | d_real_loss: 0.6385 | d_Y_loss: 1.0122 | d_X_loss:
0.9176 | d_fake_loss: 1.9298 | g_loss: 0.5142
Iteration [ 9580/10000] | d_real_loss: 0.6929 | d_Y_loss: 0.9658 | d_X_loss:
0.5452 | d_fake_loss: 1.5110 | g_loss: 0.5552
Iteration [ 9590/10000] | d_real_loss: 0.5372 | d_Y_loss: 0.9264 | d_X_loss:
0.5943 | d_fake_loss: 1.5207 | g_loss: 0.5862
Iteration [ 9600/10000] | d_real_loss: 0.6186 | d_Y_loss: 0.9541 | d_X_loss:
0.4867 | d_fake_loss: 1.4408 | g_loss: 0.5675
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009600-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009600-
Y-X.png
Iteration [ 9610/10000] | d_real_loss: 0.7084 | d_Y_loss: 0.9724 | d_X_loss:
0.6218 | d_fake_loss: 1.5942 | g_loss: 0.5464
Iteration [ 9620/10000] | d_real_loss: 0.5079 | d_Y_loss: 0.9490 | d_X_loss:
0.5039 | d_fake_loss: 1.4529 | g_loss: 0.5744
Iteration [ 9630/10000] | d_real_loss: 0.6976 | d_Y_loss: 1.0706 | d_X_loss:
0.4934 | d_fake_loss: 1.5639 | g_loss: 0.5161
Iteration [ 9640/10000] | d_real_loss: 0.5838 | d_Y_loss: 1.0498 | d_X_loss:
0.6614 | d_fake_loss: 1.7112 | g_loss: 0.5051
Iteration [ 9650/10000] | d_real_loss: 0.5807 | d_Y_loss: 0.9801 | d_X_loss:
0.4034 | d_fake_loss: 1.3835 | g_loss: 0.5354
Iteration [ 9660/10000] | d_real_loss: 0.6518 | d_Y_loss: 1.0150 | d_X_loss:
0.4514 | d_fake_loss: 1.4664 | g_loss: 0.5445
Iteration [ 9670/10000] | d_real_loss: 0.5629 | d_Y_loss: 0.9247 | d_X_loss:
0.5912 | d_fake_loss: 1.5159 | g_loss: 0.5802
Iteration [ 9680/10000] | d_real_loss: 0.5670 | d_Y_loss: 0.9866 | d_X_loss:
0.5229 | d_fake_loss: 1.5095 | g_loss: 0.5421
Iteration [ 9690/10000] | d_real_loss: 0.5066 | d_Y_loss: 1.0518 | d_X_loss:
0.3319 | d_fake_loss: 1.3837 | g_loss: 0.5340
Iteration [ 9700/10000] | d_real_loss: 0.4480 | d_Y_loss: 0.9888 | d_X_loss:
0.3878 | d_fake_loss: 1.3766 | g_loss: 0.5281
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009700-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009700-
Y-X.png
Iteration [ 9710/10000] | d_real_loss: 0.5553 | d_Y_loss: 1.0101 | d_X_loss:
0.4907 | d_fake_loss: 1.5008 | g_loss: 0.5384
Iteration [ 9720/10000] | d_real_loss: 0.5681 | d_Y_loss: 1.0051 | d_X_loss:
0.5054 | d_fake_loss: 1.5105 | g_loss: 0.5506
Iteration [ 9730/10000] | d_real_loss: 0.5382 | d_Y_loss: 0.9487 | d_X_loss:
0.8902 | d_fake_loss: 1.8389 | g_loss: 0.5545
Iteration [ 9740/10000] | d_real_loss: 0.5042 | d_Y_loss: 0.9547 | d_X_loss:
0.4445 | d_fake_loss: 1.3993 | g_loss: 0.5551
Iteration [ 9750/10000] | d_real_loss: 0.5538 | d_Y_loss: 1.0083 | d_X_loss:
0.6139 | d_fake_loss: 1.6222 | g_loss: 0.5430
Iteration [ 9760/10000] | d_real_loss: 0.6988 | d_Y_loss: 1.0412 | d_X_loss:
```

```
0.6261 | d_fake_loss: 1.6673 | g_loss: 0.5362
Iteration [ 9770/10000] | d_real_loss: 0.4919 | d_Y_loss: 1.1154 | d_X_loss:
0.6548 | d_fake_loss: 1.7702 | g_loss: 0.4893
Iteration [ 9780/10000] | d_real_loss: 0.6485 | d_Y_loss: 0.9329 | d_X_loss:
0.6881 | d_fake_loss: 1.6210 | g_loss: 0.5692
Iteration [ 9790/10000] | d_real_loss: 0.5668 | d_Y_loss: 1.0266 | d_X_loss:
0.6536 | d_fake_loss: 1.6802 | g_loss: 0.5446
Iteration [ 9800/10000] | d_real_loss: 0.6573 | d_Y_loss: 1.0696 | d_X_loss:
0.5374 | d_fake_loss: 1.6069 | g_loss: 0.5813
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009800-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009800-
Y-X.png
Iteration [ 9810/10000] | d_real_loss: 0.6468 | d_Y_loss: 0.9400 | d_X_loss:
0.5083 | d_fake_loss: 1.4483 | g_loss: 0.5803
Iteration [ 9820/10000] | d_real_loss: 0.6988 | d_Y_loss: 0.8775 | d_X_loss:
0.4237 | d_fake_loss: 1.3012 | g_loss: 0.6063
Iteration [ 9830/10000] | d_real_loss: 0.5654 | d_Y_loss: 0.9665 | d_X_loss:
0.5177 | d_fake_loss: 1.4842 | g_loss: 0.5537
Iteration [ 9840/10000] | d_real_loss: 0.5852 | d_Y_loss: 0.9242 | d_X_loss:
0.8558 | d_fake_loss: 1.7800 | g_loss: 0.5761
Iteration [ 9850/10000] | d_real_loss: 0.6899 | d_Y_loss: 1.0198 | d_X_loss:
0.5085 | d_fake_loss: 1.5283 | g_loss: 0.5342
Iteration [ 9860/10000] | d_real_loss: 0.5891 | d_Y_loss: 1.0062 | d_X_loss:
0.4346 | d_fake_loss: 1.4408 | g_loss: 0.5489
Iteration [ 9870/10000] | d_real_loss: 0.6127 | d_Y_loss: 1.0251 | d_X_loss:
0.4826 | d_fake_loss: 1.5078 | g_loss: 0.5486
Iteration [ 9880/10000] | d_real_loss: 0.5698 | d_Y_loss: 0.9512 | d_X_loss:
0.7432 | d_fake_loss: 1.6944 | g_loss: 0.5662
Iteration [ 9890/10000] | d_real_loss: 0.5886 | d_Y_loss: 0.9797 | d_X_loss:
0.7757 | d_fake_loss: 1.7554 | g_loss: 0.5701
Iteration [ 9900/10000] | d_real_loss: 0.6105 | d_Y_loss: 1.0006 | d_X_loss:
0.5793 | d_fake_loss: 1.5799 | g_loss: 0.5480
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009900-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-009900-
Y-X.png
Iteration [ 9910/10000] | d_real_loss: 0.6298 | d_Y_loss: 0.9204 | d_X_loss:
0.6161 | d_fake_loss: 1.5365 | g_loss: 0.5928
Iteration [ 9920/10000] | d_real_loss: 0.5212 | d_Y_loss: 1.0056 | d_X_loss:
0.5267 | d_fake_loss: 1.5323 | g_loss: 0.5315
Iteration [ 9930/10000] | d_real_loss: 0.6402 | d_Y_loss: 1.0676 | d_X_loss:
0.4989 | d_fake_loss: 1.5666 | g_loss: 0.5034
Iteration [ 9940/10000] | d_real_loss: 0.6424 | d_Y_loss: 0.9714 | d_X_loss:
0.7288 | d_fake_loss: 1.7002 | g_loss: 0.5474
Iteration [ 9950/10000] | d_real_loss: 0.7268 | d_Y_loss: 0.9664 | d_X_loss:
0.3777 | d_fake_loss: 1.3442 | g_loss: 0.5429
Iteration [ 9960/10000] | d_real_loss: 0.6292 | d_Y_loss: 0.9848 | d_X_loss:
```

```
0.5922 | d_fake_loss: 1.5770 | g_loss: 0.5918
Iteration [ 9970/10000] | d_real_loss: 0.5677 | d_Y_loss: 1.0154 | d_X_loss:
0.7070 | d_fake_loss: 1.7224 | g_loss: 0.5195
Iteration [ 9980/10000] | d_real_loss: 0.6798 | d_Y_loss: 1.0291 | d_X_loss:
0.5895 | d_fake_loss: 1.6186 | g_loss: 0.5068
Iteration [ 9990/10000] | d_real_loss: 0.5452 | d_Y_loss: 1.0102 | d_X_loss:
0.6512 | d_fake_loss: 1.6614 | g_loss: 0.5641
Iteration [10000/10000] | d_real_loss: 0.7287 | d_Y_loss: 1.0384 | d_X_loss:
0.3791 | d_fake_loss: 1.4175 | g_loss: 0.5118
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-010000-
X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive\sample-010000-
Y-X.png
```

[3]:
```
!python cycle_gan.py --train_iters=10000 --sample_dir=cycle_gan_10000␣
↪--use_cycle_consistency_loss
```

```
================================================================================
                                     Opts
--------------------------------------------------------------------------------
                           image_size: 64
                                 disc: dc
                                  gen: cycle
                           g_conv_dim: 32
                           d_conv_dim: 32
                                 norm: instance
          use_cycle_consistency_loss: 1
                            init_type: naive
                         train_iters: 10000
                           batch_size: 16
                                   lr: 0.0003
                                beta1: 0.5
                                beta2: 0.999
                         lambda_cycle: 10
                                    X: cat/grumpifyAprocessed
                                    Y: cat/grumpifyBprocessed
                                  ext: *.png
                             data_aug: deluxe
                       checkpoint_dir: checkpoints_cyclegan
                   sample_dir:
output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle
                           log_step: 10
                        sample_every: 100
                     checkpoint_every: 800
                                  gpu: 0
================================================================================
data/cat/grumpifyAprocessed\*.png
75
```

```
data/cat/grumpifyBprocessed\*.png
204
                 G_XtoY
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
```

```
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
---------------------------------------
                G_YtoX
---------------------------------------
CycleGenerator(
  (pad): ReflectionPad2d((3, 3, 3, 3))
  (conv1): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Sequential(
      (0): Conv2d(3, 32, kernel_size=(7, 7), stride=(1, 1), bias=False)
      (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
    )
  )
  (conv2): Sequential(
```

```
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (resnet_block): Sequential(
    (0): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (1): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (2): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (3): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (4): ResnetBlock(
      (conv_layer): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
    (5): ResnetBlock(
      (conv_layer): Sequential(
```

```
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
      )
    )
  )
  (deconv1): Sequential(
    (0): ConvTranspose2d(64, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1,
1), bias=False)
    (1): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (deconv2): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(32, 3, kernel_size=(7, 7), stride=(1, 1), bias=False)
    (2): Tanh()
  )
)
----------------------------------------
                D_X
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

```
  )
)
----------------------------------------
                  D_Y
----------------------------------------
DCDiscriminator(
  (conv1): Sequential(
    (0): Conv2d(3, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
  )
  (conv2): Sequential(
    (0): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv3): Sequential(
    (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv4): Sequential(
    (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
  )
  (conv5): Sequential(
    (0): AdaptiveAvgPool2d(output_size=1)
    (1): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)
----------------------------------------
Models moved to GPU.
Iteration [   10/10000] | d_real_loss: 1.0904 | d_Y_loss: 0.7427 | d_X_loss:
0.9016 | d_fake_loss: 1.6443 | g_loss: 3.7038
Iteration [   20/10000] | d_real_loss: 0.9562 | d_Y_loss: 0.7654 | d_X_loss:
0.8931 | d_fake_loss: 1.6585 | g_loss: 3.2409
Iteration [   30/10000] | d_real_loss: 0.8490 | d_Y_loss: 0.7691 | d_X_loss:
0.8501 | d_fake_loss: 1.6192 | g_loss: 3.2216
Iteration [   40/10000] | d_real_loss: 0.7736 | d_Y_loss: 0.7559 | d_X_loss:
0.7985 | d_fake_loss: 1.5544 | g_loss: 3.0341
Iteration [   50/10000] | d_real_loss: 0.7342 | d_Y_loss: 0.7628 | d_X_loss:
0.7803 | d_fake_loss: 1.5430 | g_loss: 3.5896
Iteration [   60/10000] | d_real_loss: 0.6889 | d_Y_loss: 0.7442 | d_X_loss:
0.7736 | d_fake_loss: 1.5178 | g_loss: 2.8941
Iteration [   70/10000] | d_real_loss: 0.6697 | d_Y_loss: 0.7272 | d_X_loss:
```

```
0.7744 | d_fake_loss: 1.5015 | g_loss: 3.3504
Iteration [   80/10000] | d_real_loss: 0.6233 | d_Y_loss: 0.7090 | d_X_loss:
0.7458 | d_fake_loss: 1.4548 | g_loss: 3.4476
Iteration [   90/10000] | d_real_loss: 0.6266 | d_Y_loss: 0.6959 | d_X_loss:
0.7982 | d_fake_loss: 1.4941 | g_loss: 2.7771
Iteration [  100/10000] | d_real_loss: 0.6078 | d_Y_loss: 0.6876 | d_X_loss:
0.7643 | d_fake_loss: 1.4519 | g_loss: 3.2223
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000100-Y-X.png
Iteration [  110/10000] | d_real_loss: 0.6307 | d_Y_loss: 0.7322 | d_X_loss:
0.7946 | d_fake_loss: 1.5268 | g_loss: 3.0436
Iteration [  120/10000] | d_real_loss: 0.5902 | d_Y_loss: 0.7166 | d_X_loss:
0.7819 | d_fake_loss: 1.4985 | g_loss: 2.7443
Iteration [  130/10000] | d_real_loss: 0.5468 | d_Y_loss: 0.6007 | d_X_loss:
0.7075 | d_fake_loss: 1.3082 | g_loss: 3.1894
Iteration [  140/10000] | d_real_loss: 0.5815 | d_Y_loss: 0.7086 | d_X_loss:
0.7033 | d_fake_loss: 1.4119 | g_loss: 2.9379
Iteration [  150/10000] | d_real_loss: 0.5518 | d_Y_loss: 0.6612 | d_X_loss:
0.7039 | d_fake_loss: 1.3652 | g_loss: 2.5536
Iteration [  160/10000] | d_real_loss: 0.5261 | d_Y_loss: 0.6374 | d_X_loss:
0.6487 | d_fake_loss: 1.2861 | g_loss: 3.1580
Iteration [  170/10000] | d_real_loss: 0.5381 | d_Y_loss: 0.6783 | d_X_loss:
0.6924 | d_fake_loss: 1.3707 | g_loss: 3.2834
Iteration [  180/10000] | d_real_loss: 0.5472 | d_Y_loss: 0.6303 | d_X_loss:
0.7089 | d_fake_loss: 1.3392 | g_loss: 3.3654
Iteration [  190/10000] | d_real_loss: 0.4772 | d_Y_loss: 0.6372 | d_X_loss:
0.6057 | d_fake_loss: 1.2429 | g_loss: 3.3676
Iteration [  200/10000] | d_real_loss: 0.4644 | d_Y_loss: 0.5980 | d_X_loss:
0.6074 | d_fake_loss: 1.2054 | g_loss: 3.0051
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000200-Y-X.png
Iteration [  210/10000] | d_real_loss: 0.4940 | d_Y_loss: 0.5868 | d_X_loss:
0.6446 | d_fake_loss: 1.2314 | g_loss: 3.0092
Iteration [  220/10000] | d_real_loss: 0.5281 | d_Y_loss: 0.5813 | d_X_loss:
0.7575 | d_fake_loss: 1.3388 | g_loss: 2.9880
Iteration [  230/10000] | d_real_loss: 0.4877 | d_Y_loss: 0.6018 | d_X_loss:
0.6545 | d_fake_loss: 1.2563 | g_loss: 3.4954
Iteration [  240/10000] | d_real_loss: 0.4997 | d_Y_loss: 0.5567 | d_X_loss:
0.6280 | d_fake_loss: 1.1847 | g_loss: 2.9353
Iteration [  250/10000] | d_real_loss: 0.5002 | d_Y_loss: 0.5674 | d_X_loss:
0.6630 | d_fake_loss: 1.2304 | g_loss: 3.1188
Iteration [  260/10000] | d_real_loss: 0.4961 | d_Y_loss: 0.5341 | d_X_loss:
0.5909 | d_fake_loss: 1.1249 | g_loss: 2.9086
Iteration [  270/10000] | d_real_loss: 0.4891 | d_Y_loss: 0.5586 | d_X_loss:
```

```
0.6398 | d_fake_loss: 1.1984 | g_loss: 3.1359
Iteration [  280/10000] | d_real_loss: 0.4651 | d_Y_loss: 0.5391 | d_X_loss:
0.6634 | d_fake_loss: 1.2025 | g_loss: 3.3282
Iteration [  290/10000] | d_real_loss: 0.4619 | d_Y_loss: 0.5087 | d_X_loss:
0.7511 | d_fake_loss: 1.2598 | g_loss: 2.8628
Iteration [  300/10000] | d_real_loss: 0.4442 | d_Y_loss: 0.5148 | d_X_loss:
0.5962 | d_fake_loss: 1.1109 | g_loss: 3.0176
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000300-Y-X.png
Iteration [  310/10000] | d_real_loss: 0.4405 | d_Y_loss: 0.5145 | d_X_loss:
0.6685 | d_fake_loss: 1.1830 | g_loss: 3.5022
Iteration [  320/10000] | d_real_loss: 0.4452 | d_Y_loss: 0.4708 | d_X_loss:
0.6465 | d_fake_loss: 1.1173 | g_loss: 3.3892
Iteration [  330/10000] | d_real_loss: 0.4345 | d_Y_loss: 0.5036 | d_X_loss:
0.6092 | d_fake_loss: 1.1127 | g_loss: 3.7443
Iteration [  340/10000] | d_real_loss: 0.4239 | d_Y_loss: 0.4637 | d_X_loss:
0.5962 | d_fake_loss: 1.0600 | g_loss: 3.3586
Iteration [  350/10000] | d_real_loss: 0.4980 | d_Y_loss: 0.4694 | d_X_loss:
0.5891 | d_fake_loss: 1.0585 | g_loss: 3.1673
Iteration [  360/10000] | d_real_loss: 0.4582 | d_Y_loss: 0.4448 | d_X_loss:
0.6625 | d_fake_loss: 1.1073 | g_loss: 3.4988
Iteration [  370/10000] | d_real_loss: 0.4201 | d_Y_loss: 0.4488 | d_X_loss:
0.4884 | d_fake_loss: 0.9373 | g_loss: 3.5134
Iteration [  380/10000] | d_real_loss: 0.4467 | d_Y_loss: 0.4780 | d_X_loss:
0.5989 | d_fake_loss: 1.0769 | g_loss: 3.4808
Iteration [  390/10000] | d_real_loss: 0.4017 | d_Y_loss: 0.4785 | d_X_loss:
0.5923 | d_fake_loss: 1.0708 | g_loss: 3.5494
Iteration [  400/10000] | d_real_loss: 0.4184 | d_Y_loss: 0.4331 | d_X_loss:
0.5297 | d_fake_loss: 0.9628 | g_loss: 3.3923
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000400-Y-X.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000400-Y-X.png
Iteration [  410/10000] | d_real_loss: 0.3978 | d_Y_loss: 0.4381 | d_X_loss:
0.6455 | d_fake_loss: 1.0836 | g_loss: 3.4743
Iteration [  420/10000] | d_real_loss: 0.3873 | d_Y_loss: 0.4079 | d_X_loss:
0.6151 | d_fake_loss: 1.0230 | g_loss: 3.4419
Iteration [  430/10000] | d_real_loss: 0.3592 | d_Y_loss: 0.3737 | d_X_loss:
0.5965 | d_fake_loss: 0.9702 | g_loss: 3.8074
Iteration [  440/10000] | d_real_loss: 0.3956 | d_Y_loss: 0.3776 | d_X_loss:
0.5151 | d_fake_loss: 0.8927 | g_loss: 3.7910
Iteration [  450/10000] | d_real_loss: 0.3882 | d_Y_loss: 0.4292 | d_X_loss:
```

```
0.6436 | d_fake_loss: 1.0728 | g_loss: 2.9742
Iteration [  460/10000] | d_real_loss: 0.3725 | d_Y_loss: 0.3565 | d_X_loss:
0.5859 | d_fake_loss: 0.9423 | g_loss: 3.2414
Iteration [  470/10000] | d_real_loss: 0.3742 | d_Y_loss: 0.3625 | d_X_loss:
0.5364 | d_fake_loss: 0.8989 | g_loss: 3.3331
Iteration [  480/10000] | d_real_loss: 0.3793 | d_Y_loss: 0.4033 | d_X_loss:
0.5819 | d_fake_loss: 0.9852 | g_loss: 3.1574
Iteration [  490/10000] | d_real_loss: 0.4352 | d_Y_loss: 0.3644 | d_X_loss:
0.4495 | d_fake_loss: 0.8139 | g_loss: 3.4012
Iteration [  500/10000] | d_real_loss: 0.3572 | d_Y_loss: 0.4188 | d_X_loss:
0.4413 | d_fake_loss: 0.8601 | g_loss: 3.5555
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000500-Y-X.png
Iteration [  510/10000] | d_real_loss: 0.3877 | d_Y_loss: 0.3716 | d_X_loss:
0.4698 | d_fake_loss: 0.8413 | g_loss: 3.8218
Iteration [  520/10000] | d_real_loss: 0.3847 | d_Y_loss: 0.3675 | d_X_loss:
0.4796 | d_fake_loss: 0.8471 | g_loss: 4.0977
Iteration [  530/10000] | d_real_loss: 0.3837 | d_Y_loss: 0.3817 | d_X_loss:
0.5894 | d_fake_loss: 0.9711 | g_loss: 3.7635
Iteration [  540/10000] | d_real_loss: 0.3983 | d_Y_loss: 0.3816 | d_X_loss:
0.4858 | d_fake_loss: 0.8674 | g_loss: 3.3761
Iteration [  550/10000] | d_real_loss: 0.3631 | d_Y_loss: 0.3841 | d_X_loss:
0.5241 | d_fake_loss: 0.9083 | g_loss: 3.5431
Iteration [  560/10000] | d_real_loss: 0.3681 | d_Y_loss: 0.3598 | d_X_loss:
0.4890 | d_fake_loss: 0.8488 | g_loss: 3.3560
Iteration [  570/10000] | d_real_loss: 0.3280 | d_Y_loss: 0.3186 | d_X_loss:
0.4876 | d_fake_loss: 0.8062 | g_loss: 3.7593
Iteration [  580/10000] | d_real_loss: 0.3458 | d_Y_loss: 0.3572 | d_X_loss:
0.5598 | d_fake_loss: 0.9170 | g_loss: 3.8133
Iteration [  590/10000] | d_real_loss: 0.3086 | d_Y_loss: 0.3300 | d_X_loss:
0.4189 | d_fake_loss: 0.7489 | g_loss: 3.4089
Iteration [  600/10000] | d_real_loss: 0.3043 | d_Y_loss: 0.2883 | d_X_loss:
0.5442 | d_fake_loss: 0.8326 | g_loss: 3.7084
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000600-Y-X.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000600-Y-X.png
Iteration [  610/10000] | d_real_loss: 0.3018 | d_Y_loss: 0.3335 | d_X_loss:
0.3958 | d_fake_loss: 0.7293 | g_loss: 3.8551
Iteration [  620/10000] | d_real_loss: 0.3221 | d_Y_loss: 0.3003 | d_X_loss:
0.5412 | d_fake_loss: 0.8416 | g_loss: 3.5964
Iteration [  630/10000] | d_real_loss: 0.3430 | d_Y_loss: 0.2908 | d_X_loss:
```

```
0.4282 | d_fake_loss: 0.7191 | g_loss: 3.8483
Iteration [  640/10000] | d_real_loss: 0.3275 | d_Y_loss: 0.3064 | d_X_loss:
0.4547 | d_fake_loss: 0.7611 | g_loss: 4.1487
Iteration [  650/10000] | d_real_loss: 0.3036 | d_Y_loss: 0.3252 | d_X_loss:
0.4788 | d_fake_loss: 0.8041 | g_loss: 3.6016
Iteration [  660/10000] | d_real_loss: 0.3086 | d_Y_loss: 0.2823 | d_X_loss:
0.3991 | d_fake_loss: 0.6814 | g_loss: 3.9229
Iteration [  670/10000] | d_real_loss: 0.3348 | d_Y_loss: 0.2523 | d_X_loss:
0.4217 | d_fake_loss: 0.6740 | g_loss: 4.0690
Iteration [  680/10000] | d_real_loss: 0.2985 | d_Y_loss: 0.3271 | d_X_loss:
0.4975 | d_fake_loss: 0.8246 | g_loss: 3.8214
Iteration [  690/10000] | d_real_loss: 0.3114 | d_Y_loss: 0.2567 | d_X_loss:
0.4465 | d_fake_loss: 0.7031 | g_loss: 3.6886
Iteration [  700/10000] | d_real_loss: 0.3637 | d_Y_loss: 0.2457 | d_X_loss:
0.3787 | d_fake_loss: 0.6244 | g_loss: 3.9466
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000700-Y-X.png
Iteration [  710/10000] | d_real_loss: 0.3484 | d_Y_loss: 0.2656 | d_X_loss:
0.3882 | d_fake_loss: 0.6538 | g_loss: 3.8494
Iteration [  720/10000] | d_real_loss: 0.2562 | d_Y_loss: 0.2629 | d_X_loss:
0.4226 | d_fake_loss: 0.6856 | g_loss: 3.6495
Iteration [  730/10000] | d_real_loss: 0.2921 | d_Y_loss: 0.2690 | d_X_loss:
0.4355 | d_fake_loss: 0.7045 | g_loss: 4.2612
Iteration [  740/10000] | d_real_loss: 0.2643 | d_Y_loss: 0.2693 | d_X_loss:
0.3453 | d_fake_loss: 0.6147 | g_loss: 3.7190
Iteration [  750/10000] | d_real_loss: 0.2698 | d_Y_loss: 0.2380 | d_X_loss:
0.4383 | d_fake_loss: 0.6763 | g_loss: 3.9847
Iteration [  760/10000] | d_real_loss: 0.2449 | d_Y_loss: 0.2370 | d_X_loss:
0.3624 | d_fake_loss: 0.5995 | g_loss: 3.7439
Iteration [  770/10000] | d_real_loss: 0.2647 | d_Y_loss: 0.2154 | d_X_loss:
0.4003 | d_fake_loss: 0.6158 | g_loss: 3.8970
Iteration [  780/10000] | d_real_loss: 0.2604 | d_Y_loss: 0.3032 | d_X_loss:
0.3025 | d_fake_loss: 0.6056 | g_loss: 3.8276
Iteration [  790/10000] | d_real_loss: 0.2979 | d_Y_loss: 0.2606 | d_X_loss:
0.3393 | d_fake_loss: 0.5999 | g_loss: 3.6135
Iteration [  800/10000] | d_real_loss: 0.2571 | d_Y_loss: 0.2149 | d_X_loss:
0.3329 | d_fake_loss: 0.5478 | g_loss: 3.7412
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000800-Y-X.png
Iteration [  810/10000] | d_real_loss: 0.2396 | d_Y_loss: 0.2061 | d_X_loss:
0.3036 | d_fake_loss: 0.5096 | g_loss: 3.8681
Iteration [  820/10000] | d_real_loss: 0.2415 | d_Y_loss: 0.2353 | d_X_loss:
0.3122 | d_fake_loss: 0.5475 | g_loss: 3.7305
Iteration [  830/10000] | d_real_loss: 0.2620 | d_Y_loss: 0.2165 | d_X_loss:
```

```
0.3279 | d_fake_loss: 0.5444 | g_loss: 3.7155
Iteration [  840/10000] | d_real_loss: 0.2097 | d_Y_loss: 0.2472 | d_X_loss:
0.3474 | d_fake_loss: 0.5946 | g_loss: 3.9841
Iteration [  850/10000] | d_real_loss: 0.2408 | d_Y_loss: 0.2187 | d_X_loss:
0.4735 | d_fake_loss: 0.6922 | g_loss: 3.9406
Iteration [  860/10000] | d_real_loss: 0.3054 | d_Y_loss: 0.2192 | d_X_loss:
0.3049 | d_fake_loss: 0.5241 | g_loss: 4.1947
Iteration [  870/10000] | d_real_loss: 0.2183 | d_Y_loss: 0.2287 | d_X_loss:
0.3422 | d_fake_loss: 0.5709 | g_loss: 4.4690
Iteration [  880/10000] | d_real_loss: 0.2081 | d_Y_loss: 0.2468 | d_X_loss:
0.3664 | d_fake_loss: 0.6132 | g_loss: 4.9543
Iteration [  890/10000] | d_real_loss: 0.2168 | d_Y_loss: 0.1921 | d_X_loss:
0.3085 | d_fake_loss: 0.5006 | g_loss: 4.9906
Iteration [  900/10000] | d_real_loss: 0.2297 | d_Y_loss: 0.2210 | d_X_loss:
0.2907 | d_fake_loss: 0.5117 | g_loss: 4.0168
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
000900-Y-X.png
Iteration [  910/10000] | d_real_loss: 0.3349 | d_Y_loss: 0.2494 | d_X_loss:
0.6598 | d_fake_loss: 0.9092 | g_loss: 4.1189
Iteration [  920/10000] | d_real_loss: 0.2344 | d_Y_loss: 0.2111 | d_X_loss:
0.2925 | d_fake_loss: 0.5037 | g_loss: 4.0640
Iteration [  930/10000] | d_real_loss: 0.2671 | d_Y_loss: 0.1975 | d_X_loss:
0.4296 | d_fake_loss: 0.6270 | g_loss: 3.9238
Iteration [  940/10000] | d_real_loss: 0.2385 | d_Y_loss: 0.1869 | d_X_loss:
0.3185 | d_fake_loss: 0.5054 | g_loss: 3.9729
Iteration [  950/10000] | d_real_loss: 0.1988 | d_Y_loss: 0.2068 | d_X_loss:
0.2682 | d_fake_loss: 0.4750 | g_loss: 3.6047
Iteration [  960/10000] | d_real_loss: 0.2457 | d_Y_loss: 0.1952 | d_X_loss:
0.2799 | d_fake_loss: 0.4751 | g_loss: 4.9838
Iteration [  970/10000] | d_real_loss: 0.2303 | d_Y_loss: 0.2737 | d_X_loss:
0.3291 | d_fake_loss: 0.6027 | g_loss: 4.1463
Iteration [  980/10000] | d_real_loss: 0.2124 | d_Y_loss: 0.1802 | d_X_loss:
0.2272 | d_fake_loss: 0.4074 | g_loss: 4.1086
Iteration [  990/10000] | d_real_loss: 0.2257 | d_Y_loss: 0.1672 | d_X_loss:
0.3239 | d_fake_loss: 0.4910 | g_loss: 4.2936
Iteration [ 1000/10000] | d_real_loss: 0.2802 | d_Y_loss: 0.2135 | d_X_loss:
0.5724 | d_fake_loss: 0.7860 | g_loss: 4.2785
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001000-Y-X.png
Iteration [ 1010/10000] | d_real_loss: 0.3556 | d_Y_loss: 0.5814 | d_X_loss:
0.5363 | d_fake_loss: 1.1178 | g_loss: 3.6654
Iteration [ 1020/10000] | d_real_loss: 0.2719 | d_Y_loss: 0.2951 | d_X_loss:
0.2720 | d_fake_loss: 0.5671 | g_loss: 4.1032
Iteration [ 1030/10000] | d_real_loss: 0.3165 | d_Y_loss: 0.2301 | d_X_loss:
```

```
0.2524 | d_fake_loss: 0.4825 | g_loss: 4.0629
Iteration [ 1040/10000] | d_real_loss: 0.2281 | d_Y_loss: 0.1972 | d_X_loss:
0.2722 | d_fake_loss: 0.4694 | g_loss: 4.0533
Iteration [ 1050/10000] | d_real_loss: 0.2459 | d_Y_loss: 0.1943 | d_X_loss:
0.2564 | d_fake_loss: 0.4506 | g_loss: 4.8541
Iteration [ 1060/10000] | d_real_loss: 0.2068 | d_Y_loss: 0.2196 | d_X_loss:
0.2570 | d_fake_loss: 0.4766 | g_loss: 4.7988
Iteration [ 1070/10000] | d_real_loss: 0.2032 | d_Y_loss: 0.1691 | d_X_loss:
0.2371 | d_fake_loss: 0.4062 | g_loss: 4.1321
Iteration [ 1080/10000] | d_real_loss: 0.2054 | d_Y_loss: 0.2894 | d_X_loss:
0.2399 | d_fake_loss: 0.5292 | g_loss: 4.4249
Iteration [ 1090/10000] | d_real_loss: 0.2651 | d_Y_loss: 0.2076 | d_X_loss:
1.3937 | d_fake_loss: 1.6013 | g_loss: 4.4007
Iteration [ 1100/10000] | d_real_loss: 0.2094 | d_Y_loss: 0.2415 | d_X_loss:
0.6622 | d_fake_loss: 0.9037 | g_loss: 4.0706
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001100-Y-X.png
Iteration [ 1110/10000] | d_real_loss: 0.3766 | d_Y_loss: 0.2641 | d_X_loss:
0.3671 | d_fake_loss: 0.6312 | g_loss: 4.1081
Iteration [ 1120/10000] | d_real_loss: 0.2759 | d_Y_loss: 0.2619 | d_X_loss:
0.3007 | d_fake_loss: 0.5627 | g_loss: 3.9105
Iteration [ 1130/10000] | d_real_loss: 0.2271 | d_Y_loss: 0.3515 | d_X_loss:
0.2735 | d_fake_loss: 0.6249 | g_loss: 4.0655
Iteration [ 1140/10000] | d_real_loss: 0.1696 | d_Y_loss: 0.3329 | d_X_loss:
0.2121 | d_fake_loss: 0.5450 | g_loss: 4.3146
Iteration [ 1150/10000] | d_real_loss: 0.1846 | d_Y_loss: 0.2177 | d_X_loss:
0.2106 | d_fake_loss: 0.4283 | g_loss: 4.1103
Iteration [ 1160/10000] | d_real_loss: 0.1725 | d_Y_loss: 0.3466 | d_X_loss:
0.2111 | d_fake_loss: 0.5577 | g_loss: 4.2574
Iteration [ 1170/10000] | d_real_loss: 0.3739 | d_Y_loss: 0.3764 | d_X_loss:
0.2657 | d_fake_loss: 0.6421 | g_loss: 3.8339
Iteration [ 1180/10000] | d_real_loss: 0.2186 | d_Y_loss: 0.3639 | d_X_loss:
0.2134 | d_fake_loss: 0.5773 | g_loss: 4.1516
Iteration [ 1190/10000] | d_real_loss: 0.2828 | d_Y_loss: 0.3778 | d_X_loss:
0.1949 | d_fake_loss: 0.5728 | g_loss: 3.8723
Iteration [ 1200/10000] | d_real_loss: 0.3168 | d_Y_loss: 0.6594 | d_X_loss:
0.8615 | d_fake_loss: 1.5209 | g_loss: 5.0229
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001200-Y-X.png
Iteration [ 1210/10000] | d_real_loss: 0.2417 | d_Y_loss: 0.2595 | d_X_loss:
0.2272 | d_fake_loss: 0.4867 | g_loss: 3.7438
Iteration [ 1220/10000] | d_real_loss: 0.2364 | d_Y_loss: 0.2391 | d_X_loss:
0.2779 | d_fake_loss: 0.5169 | g_loss: 4.3332
Iteration [ 1230/10000] | d_real_loss: 0.2709 | d_Y_loss: 0.2060 | d_X_loss:
```

```
0.3423 | d_fake_loss: 0.5483 | g_loss: 3.9691
Iteration [ 1240/10000] | d_real_loss: 0.2396 | d_Y_loss: 0.4457 | d_X_loss:
0.2149 | d_fake_loss: 0.6606 | g_loss: 3.9881
Iteration [ 1250/10000] | d_real_loss: 0.1930 | d_Y_loss: 0.2558 | d_X_loss:
0.2481 | d_fake_loss: 0.5038 | g_loss: 4.1239
Iteration [ 1260/10000] | d_real_loss: 0.2084 | d_Y_loss: 0.4143 | d_X_loss:
0.1944 | d_fake_loss: 0.6087 | g_loss: 3.8938
Iteration [ 1270/10000] | d_real_loss: 0.3403 | d_Y_loss: 0.4450 | d_X_loss:
0.2969 | d_fake_loss: 0.7419 | g_loss: 3.7571
Iteration [ 1280/10000] | d_real_loss: 0.4303 | d_Y_loss: 0.7546 | d_X_loss:
0.8251 | d_fake_loss: 1.5796 | g_loss: 3.7673
Iteration [ 1290/10000] | d_real_loss: 0.4770 | d_Y_loss: 0.8786 | d_X_loss:
0.4446 | d_fake_loss: 1.3233 | g_loss: 4.2254
Iteration [ 1300/10000] | d_real_loss: 0.2598 | d_Y_loss: 0.2831 | d_X_loss:
0.3117 | d_fake_loss: 0.5947 | g_loss: 4.0728
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001300-Y-X.png
Iteration [ 1310/10000] | d_real_loss: 0.3299 | d_Y_loss: 1.0943 | d_X_loss:
0.3037 | d_fake_loss: 1.3980 | g_loss: 4.4984
Iteration [ 1320/10000] | d_real_loss: 0.3289 | d_Y_loss: 0.5196 | d_X_loss:
0.6389 | d_fake_loss: 1.1585 | g_loss: 4.0758
Iteration [ 1330/10000] | d_real_loss: 0.2187 | d_Y_loss: 0.8605 | d_X_loss:
0.4383 | d_fake_loss: 1.2987 | g_loss: 3.9607
Iteration [ 1340/10000] | d_real_loss: 0.3576 | d_Y_loss: 0.5669 | d_X_loss:
0.2184 | d_fake_loss: 0.7853 | g_loss: 3.7982
Iteration [ 1350/10000] | d_real_loss: 0.3635 | d_Y_loss: 0.5887 | d_X_loss:
0.3127 | d_fake_loss: 0.9014 | g_loss: 4.0150
Iteration [ 1360/10000] | d_real_loss: 0.2842 | d_Y_loss: 0.4908 | d_X_loss:
0.2950 | d_fake_loss: 0.7859 | g_loss: 4.7968
Iteration [ 1370/10000] | d_real_loss: 0.3057 | d_Y_loss: 0.7234 | d_X_loss:
0.2017 | d_fake_loss: 0.9251 | g_loss: 3.6341
Iteration [ 1380/10000] | d_real_loss: 0.1983 | d_Y_loss: 0.9318 | d_X_loss:
0.2760 | d_fake_loss: 1.2078 | g_loss: 3.7159
Iteration [ 1390/10000] | d_real_loss: 0.3653 | d_Y_loss: 0.6138 | d_X_loss:
1.6148 | d_fake_loss: 2.2286 | g_loss: 3.5262
Iteration [ 1400/10000] | d_real_loss: 0.4268 | d_Y_loss: 0.7935 | d_X_loss:
0.3742 | d_fake_loss: 1.1677 | g_loss: 3.5755
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001400-Y-X.png
Iteration [ 1410/10000] | d_real_loss: 0.4775 | d_Y_loss: 0.6265 | d_X_loss:
0.6468 | d_fake_loss: 1.2733 | g_loss: 4.2635
Iteration [ 1420/10000] | d_real_loss: 0.3906 | d_Y_loss: 0.7013 | d_X_loss:
0.2014 | d_fake_loss: 0.9027 | g_loss: 4.1136
Iteration [ 1430/10000] | d_real_loss: 0.3031 | d_Y_loss: 0.4642 | d_X_loss:
```

```
0.4221 | d_fake_loss: 0.8863 | g_loss: 3.8221
Iteration [ 1440/10000] | d_real_loss: 0.3288 | d_Y_loss: 0.5528 | d_X_loss:
0.2665 | d_fake_loss: 0.8193 | g_loss: 4.4168
Iteration [ 1450/10000] | d_real_loss: 0.5989 | d_Y_loss: 0.7662 | d_X_loss:
0.3899 | d_fake_loss: 1.1561 | g_loss: 4.0855
Iteration [ 1460/10000] | d_real_loss: 0.2447 | d_Y_loss: 0.6246 | d_X_loss:
0.1784 | d_fake_loss: 0.8030 | g_loss: 4.0996
Iteration [ 1470/10000] | d_real_loss: 0.3037 | d_Y_loss: 0.7333 | d_X_loss:
0.2158 | d_fake_loss: 0.9491 | g_loss: 4.6934
Iteration [ 1480/10000] | d_real_loss: 0.2384 | d_Y_loss: 0.4596 | d_X_loss:
0.1936 | d_fake_loss: 0.6531 | g_loss: 4.0651
Iteration [ 1490/10000] | d_real_loss: 0.3190 | d_Y_loss: 0.6310 | d_X_loss:
0.2109 | d_fake_loss: 0.8419 | g_loss: 3.3470
Iteration [ 1500/10000] | d_real_loss: 0.2482 | d_Y_loss: 0.6667 | d_X_loss:
0.5490 | d_fake_loss: 1.2157 | g_loss: 4.1965
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001500-Y-X.png
Iteration [ 1510/10000] | d_real_loss: 0.4158 | d_Y_loss: 0.5711 | d_X_loss:
0.1625 | d_fake_loss: 0.7336 | g_loss: 4.1799
Iteration [ 1520/10000] | d_real_loss: 0.3701 | d_Y_loss: 0.3341 | d_X_loss:
0.2513 | d_fake_loss: 0.5854 | g_loss: 3.7504
Iteration [ 1530/10000] | d_real_loss: 0.3334 | d_Y_loss: 0.5432 | d_X_loss:
0.1991 | d_fake_loss: 0.7423 | g_loss: 3.9913
Iteration [ 1540/10000] | d_real_loss: 0.2390 | d_Y_loss: 0.7366 | d_X_loss:
0.6708 | d_fake_loss: 1.4073 | g_loss: 3.7760
Iteration [ 1550/10000] | d_real_loss: 0.4940 | d_Y_loss: 0.7660 | d_X_loss:
0.8348 | d_fake_loss: 1.6008 | g_loss: 3.7300
Iteration [ 1560/10000] | d_real_loss: 0.4916 | d_Y_loss: 0.6588 | d_X_loss:
0.4977 | d_fake_loss: 1.1565 | g_loss: 4.1903
Iteration [ 1570/10000] | d_real_loss: 0.4003 | d_Y_loss: 0.5751 | d_X_loss:
0.4174 | d_fake_loss: 0.9925 | g_loss: 4.1389
Iteration [ 1580/10000] | d_real_loss: 0.3706 | d_Y_loss: 0.5834 | d_X_loss:
0.4465 | d_fake_loss: 1.0299 | g_loss: 4.1161
Iteration [ 1590/10000] | d_real_loss: 0.4376 | d_Y_loss: 0.4753 | d_X_loss:
0.9133 | d_fake_loss: 1.3886 | g_loss: 3.8178
Iteration [ 1600/10000] | d_real_loss: 0.4513 | d_Y_loss: 0.6988 | d_X_loss:
0.4504 | d_fake_loss: 1.1492 | g_loss: 4.2155
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001600-Y-X.png
Iteration [ 1610/10000] | d_real_loss: 0.3378 | d_Y_loss: 0.6441 | d_X_loss:
0.2217 | d_fake_loss: 0.8658 | g_loss: 4.1566
Iteration [ 1620/10000] | d_real_loss: 0.4319 | d_Y_loss: 0.6586 | d_X_loss:
0.3785 | d_fake_loss: 1.0372 | g_loss: 4.1650
Iteration [ 1630/10000] | d_real_loss: 0.2864 | d_Y_loss: 0.6719 | d_X_loss:
```

```
0.2813 | d_fake_loss: 0.9532 | g_loss: 4.0403
Iteration [ 1640/10000] | d_real_loss: 0.2669 | d_Y_loss: 0.4919 | d_X_loss:
0.1899 | d_fake_loss: 0.6818 | g_loss: 4.0812
Iteration [ 1650/10000] | d_real_loss: 0.4318 | d_Y_loss: 0.4055 | d_X_loss:
0.3576 | d_fake_loss: 0.7631 | g_loss: 3.6673
Iteration [ 1660/10000] | d_real_loss: 0.3561 | d_Y_loss: 0.6205 | d_X_loss:
0.2260 | d_fake_loss: 0.8465 | g_loss: 3.8420
Iteration [ 1670/10000] | d_real_loss: 0.4131 | d_Y_loss: 0.4166 | d_X_loss:
0.2202 | d_fake_loss: 0.6367 | g_loss: 3.7477
Iteration [ 1680/10000] | d_real_loss: 0.3268 | d_Y_loss: 0.6802 | d_X_loss:
0.4113 | d_fake_loss: 1.0915 | g_loss: 4.0938
Iteration [ 1690/10000] | d_real_loss: 0.2828 | d_Y_loss: 0.5615 | d_X_loss:
0.1454 | d_fake_loss: 0.7069 | g_loss: 4.3339
Iteration [ 1700/10000] | d_real_loss: 0.3583 | d_Y_loss: 0.4490 | d_X_loss:
0.1676 | d_fake_loss: 0.6166 | g_loss: 4.0235
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001700-Y-X.png
Iteration [ 1710/10000] | d_real_loss: 0.2719 | d_Y_loss: 0.5048 | d_X_loss:
0.1853 | d_fake_loss: 0.6900 | g_loss: 4.4773
Iteration [ 1720/10000] | d_real_loss: 0.2805 | d_Y_loss: 0.6420 | d_X_loss:
0.2302 | d_fake_loss: 0.8722 | g_loss: 4.2073
Iteration [ 1730/10000] | d_real_loss: 0.3361 | d_Y_loss: 0.6399 | d_X_loss:
0.1697 | d_fake_loss: 0.8096 | g_loss: 4.3729
Iteration [ 1740/10000] | d_real_loss: 0.4642 | d_Y_loss: 0.5247 | d_X_loss:
0.6990 | d_fake_loss: 1.2237 | g_loss: 3.8543
Iteration [ 1750/10000] | d_real_loss: 0.4017 | d_Y_loss: 0.5440 | d_X_loss:
0.1902 | d_fake_loss: 0.7342 | g_loss: 4.2522
Iteration [ 1760/10000] | d_real_loss: 0.2742 | d_Y_loss: 0.7137 | d_X_loss:
0.2352 | d_fake_loss: 0.9489 | g_loss: 3.5288
Iteration [ 1770/10000] | d_real_loss: 0.3311 | d_Y_loss: 0.6015 | d_X_loss:
0.3446 | d_fake_loss: 0.9461 | g_loss: 3.6851
Iteration [ 1780/10000] | d_real_loss: 0.2864 | d_Y_loss: 0.5976 | d_X_loss:
0.1548 | d_fake_loss: 0.7525 | g_loss: 4.3145
Iteration [ 1790/10000] | d_real_loss: 0.4425 | d_Y_loss: 0.5324 | d_X_loss:
0.1480 | d_fake_loss: 0.6804 | g_loss: 3.7917
Iteration [ 1800/10000] | d_real_loss: 0.3479 | d_Y_loss: 0.5715 | d_X_loss:
0.1496 | d_fake_loss: 0.7212 | g_loss: 4.2688
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001800-Y-X.png
Iteration [ 1810/10000] | d_real_loss: 0.8460 | d_Y_loss: 0.5358 | d_X_loss:
0.9038 | d_fake_loss: 1.4396 | g_loss: 3.9816
Iteration [ 1820/10000] | d_real_loss: 0.4909 | d_Y_loss: 0.4782 | d_X_loss:
0.9777 | d_fake_loss: 1.4559 | g_loss: 3.8204
Iteration [ 1830/10000] | d_real_loss: 0.5495 | d_Y_loss: 0.4819 | d_X_loss:
```

```
0.7008 | d_fake_loss: 1.1827 | g_loss: 3.4801
Iteration [ 1840/10000] | d_real_loss: 0.5827 | d_Y_loss: 0.6231 | d_X_loss:
0.5973 | d_fake_loss: 1.2204 | g_loss: 3.9247
Iteration [ 1850/10000] | d_real_loss: 0.3219 | d_Y_loss: 0.7946 | d_X_loss:
0.4220 | d_fake_loss: 1.2166 | g_loss: 3.7946
Iteration [ 1860/10000] | d_real_loss: 0.4132 | d_Y_loss: 0.5115 | d_X_loss:
0.1971 | d_fake_loss: 0.7086 | g_loss: 3.7431
Iteration [ 1870/10000] | d_real_loss: 0.2933 | d_Y_loss: 0.5412 | d_X_loss:
0.1884 | d_fake_loss: 0.7296 | g_loss: 3.8904
Iteration [ 1880/10000] | d_real_loss: 0.3553 | d_Y_loss: 0.9188 | d_X_loss:
0.4198 | d_fake_loss: 1.3386 | g_loss: 4.4747
Iteration [ 1890/10000] | d_real_loss: 0.3103 | d_Y_loss: 0.5917 | d_X_loss:
0.1551 | d_fake_loss: 0.7469 | g_loss: 3.8729
Iteration [ 1900/10000] | d_real_loss: 0.4340 | d_Y_loss: 0.5270 | d_X_loss:
0.1798 | d_fake_loss: 0.7067 | g_loss: 4.0358
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
001900-Y-X.png
Iteration [ 1910/10000] | d_real_loss: 0.2993 | d_Y_loss: 0.4230 | d_X_loss:
0.2956 | d_fake_loss: 0.7187 | g_loss: 3.6890
Iteration [ 1920/10000] | d_real_loss: 0.7141 | d_Y_loss: 0.5074 | d_X_loss:
0.1967 | d_fake_loss: 0.7041 | g_loss: 3.6816
Iteration [ 1930/10000] | d_real_loss: 0.3047 | d_Y_loss: 0.4320 | d_X_loss:
0.3413 | d_fake_loss: 0.7733 | g_loss: 3.9125
Iteration [ 1940/10000] | d_real_loss: 0.3937 | d_Y_loss: 0.6183 | d_X_loss:
0.2709 | d_fake_loss: 0.8892 | g_loss: 4.1327
Iteration [ 1950/10000] | d_real_loss: 0.3503 | d_Y_loss: 0.4304 | d_X_loss:
0.1889 | d_fake_loss: 0.6194 | g_loss: 4.1240
Iteration [ 1960/10000] | d_real_loss: 0.3029 | d_Y_loss: 0.6294 | d_X_loss:
0.1376 | d_fake_loss: 0.7670 | g_loss: 3.9501
Iteration [ 1970/10000] | d_real_loss: 0.2127 | d_Y_loss: 0.5755 | d_X_loss:
0.1360 | d_fake_loss: 0.7115 | g_loss: 4.0580
Iteration [ 1980/10000] | d_real_loss: 0.2748 | d_Y_loss: 0.5795 | d_X_loss:
0.1552 | d_fake_loss: 0.7347 | g_loss: 3.8926
Iteration [ 1990/10000] | d_real_loss: 0.2558 | d_Y_loss: 0.7648 | d_X_loss:
0.1498 | d_fake_loss: 0.9146 | g_loss: 3.5465
Iteration [ 2000/10000] | d_real_loss: 0.3463 | d_Y_loss: 0.4224 | d_X_loss:
0.1954 | d_fake_loss: 0.6178 | g_loss: 3.6778
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002000-Y-X.png
Iteration [ 2010/10000] | d_real_loss: 0.2362 | d_Y_loss: 0.4068 | d_X_loss:
0.1901 | d_fake_loss: 0.5969 | g_loss: 3.8091
Iteration [ 2020/10000] | d_real_loss: 0.4469 | d_Y_loss: 0.2650 | d_X_loss:
0.1930 | d_fake_loss: 0.4580 | g_loss: 4.0017
Iteration [ 2030/10000] | d_real_loss: 0.1844 | d_Y_loss: 0.6949 | d_X_loss:
```

```
0.1656 | d_fake_loss: 0.8605 | g_loss: 3.9803
Iteration [ 2040/10000] | d_real_loss: 0.2125 | d_Y_loss: 0.7624 | d_X_loss:
0.1633 | d_fake_loss: 0.9257 | g_loss: 4.3875
Iteration [ 2050/10000] | d_real_loss: 0.5014 | d_Y_loss: 0.7612 | d_X_loss:
0.1529 | d_fake_loss: 0.9141 | g_loss: 3.9552
Iteration [ 2060/10000] | d_real_loss: 0.2979 | d_Y_loss: 0.7026 | d_X_loss:
0.1017 | d_fake_loss: 0.8043 | g_loss: 3.8291
Iteration [ 2070/10000] | d_real_loss: 0.2946 | d_Y_loss: 0.6445 | d_X_loss:
0.1212 | d_fake_loss: 0.7657 | g_loss: 3.6344
Iteration [ 2080/10000] | d_real_loss: 0.3247 | d_Y_loss: 0.7174 | d_X_loss:
0.2092 | d_fake_loss: 0.9266 | g_loss: 3.6660
Iteration [ 2090/10000] | d_real_loss: 0.3047 | d_Y_loss: 0.8638 | d_X_loss:
0.9253 | d_fake_loss: 1.7891 | g_loss: 4.5961
Iteration [ 2100/10000] | d_real_loss: 0.5559 | d_Y_loss: 0.5426 | d_X_loss:
0.2575 | d_fake_loss: 0.8001 | g_loss: 3.5848
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002100-Y-X.png
Iteration [ 2110/10000] | d_real_loss: 0.3108 | d_Y_loss: 0.6856 | d_X_loss:
0.8067 | d_fake_loss: 1.4923 | g_loss: 3.4351
Iteration [ 2120/10000] | d_real_loss: 0.3977 | d_Y_loss: 0.5528 | d_X_loss:
0.2745 | d_fake_loss: 0.8273 | g_loss: 3.8303
Iteration [ 2130/10000] | d_real_loss: 0.4624 | d_Y_loss: 0.5984 | d_X_loss:
0.8174 | d_fake_loss: 1.4158 | g_loss: 3.8546
Iteration [ 2140/10000] | d_real_loss: 0.4276 | d_Y_loss: 0.6278 | d_X_loss:
0.3692 | d_fake_loss: 0.9970 | g_loss: 4.2624
Iteration [ 2150/10000] | d_real_loss: 0.3213 | d_Y_loss: 0.6199 | d_X_loss:
0.1356 | d_fake_loss: 0.7555 | g_loss: 4.3132
Iteration [ 2160/10000] | d_real_loss: 0.3825 | d_Y_loss: 0.7184 | d_X_loss:
0.4721 | d_fake_loss: 1.1905 | g_loss: 4.0250
Iteration [ 2170/10000] | d_real_loss: 0.2796 | d_Y_loss: 0.5346 | d_X_loss:
0.2856 | d_fake_loss: 0.8202 | g_loss: 3.9492
Iteration [ 2180/10000] | d_real_loss: 0.2957 | d_Y_loss: 0.4046 | d_X_loss:
0.6043 | d_fake_loss: 1.0089 | g_loss: 4.3520
Iteration [ 2190/10000] | d_real_loss: 0.3610 | d_Y_loss: 0.4944 | d_X_loss:
0.9950 | d_fake_loss: 1.4894 | g_loss: 3.9239
Iteration [ 2200/10000] | d_real_loss: 0.3468 | d_Y_loss: 0.6773 | d_X_loss:
0.3714 | d_fake_loss: 1.0487 | g_loss: 3.7155
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002200-Y-X.png
Iteration [ 2210/10000] | d_real_loss: 0.2864 | d_Y_loss: 0.4638 | d_X_loss:
0.2129 | d_fake_loss: 0.6767 | g_loss: 3.6199
Iteration [ 2220/10000] | d_real_loss: 0.2553 | d_Y_loss: 0.7224 | d_X_loss:
0.1404 | d_fake_loss: 0.8628 | g_loss: 3.9153
Iteration [ 2230/10000] | d_real_loss: 0.3191 | d_Y_loss: 0.3975 | d_X_loss:
```

```
0.6181 | d_fake_loss: 1.0156 | g_loss: 4.2288
Iteration [ 2240/10000] | d_real_loss: 0.3863 | d_Y_loss: 0.3396 | d_X_loss:
0.1508 | d_fake_loss: 0.4904 | g_loss: 3.9750
Iteration [ 2250/10000] | d_real_loss: 0.2069 | d_Y_loss: 0.4944 | d_X_loss:
0.2677 | d_fake_loss: 0.7622 | g_loss: 4.0097
Iteration [ 2260/10000] | d_real_loss: 0.3107 | d_Y_loss: 0.4779 | d_X_loss:
0.2970 | d_fake_loss: 0.7749 | g_loss: 3.6935
Iteration [ 2270/10000] | d_real_loss: 0.4897 | d_Y_loss: 0.6795 | d_X_loss:
0.1356 | d_fake_loss: 0.8151 | g_loss: 3.8738
Iteration [ 2280/10000] | d_real_loss: 0.3783 | d_Y_loss: 0.4186 | d_X_loss:
0.4630 | d_fake_loss: 0.8815 | g_loss: 3.8233
Iteration [ 2290/10000] | d_real_loss: 0.7360 | d_Y_loss: 0.4520 | d_X_loss:
0.1952 | d_fake_loss: 0.6472 | g_loss: 3.9586
Iteration [ 2300/10000] | d_real_loss: 0.2918 | d_Y_loss: 0.5826 | d_X_loss:
0.4924 | d_fake_loss: 1.0750 | g_loss: 3.6867
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002300-Y-X.png
Iteration [ 2310/10000] | d_real_loss: 0.4103 | d_Y_loss: 0.4532 | d_X_loss:
0.2338 | d_fake_loss: 0.6870 | g_loss: 4.1502
Iteration [ 2320/10000] | d_real_loss: 0.3162 | d_Y_loss: 0.4403 | d_X_loss:
0.2210 | d_fake_loss: 0.6612 | g_loss: 4.0945
Iteration [ 2330/10000] | d_real_loss: 0.3336 | d_Y_loss: 0.4958 | d_X_loss:
0.1743 | d_fake_loss: 0.6701 | g_loss: 4.0893
Iteration [ 2340/10000] | d_real_loss: 0.4469 | d_Y_loss: 0.6722 | d_X_loss:
0.3760 | d_fake_loss: 1.0481 | g_loss: 3.8535
Iteration [ 2350/10000] | d_real_loss: 0.4270 | d_Y_loss: 0.6304 | d_X_loss:
0.4395 | d_fake_loss: 1.0699 | g_loss: 4.0015
Iteration [ 2360/10000] | d_real_loss: 0.3403 | d_Y_loss: 0.7249 | d_X_loss:
0.2132 | d_fake_loss: 0.9381 | g_loss: 3.7782
Iteration [ 2370/10000] | d_real_loss: 0.3796 | d_Y_loss: 0.5448 | d_X_loss:
0.1842 | d_fake_loss: 0.7290 | g_loss: 4.0756
Iteration [ 2380/10000] | d_real_loss: 0.4120 | d_Y_loss: 0.4816 | d_X_loss:
0.4456 | d_fake_loss: 0.9272 | g_loss: 3.9125
Iteration [ 2390/10000] | d_real_loss: 0.2549 | d_Y_loss: 0.5295 | d_X_loss:
0.2205 | d_fake_loss: 0.7501 | g_loss: 3.8778
Iteration [ 2400/10000] | d_real_loss: 0.3734 | d_Y_loss: 0.8379 | d_X_loss:
0.3728 | d_fake_loss: 1.2107 | g_loss: 3.7469
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002400-Y-X.png
Iteration [ 2410/10000] | d_real_loss: 0.2925 | d_Y_loss: 0.5228 | d_X_loss:
0.1577 | d_fake_loss: 0.6805 | g_loss: 3.7328
Iteration [ 2420/10000] | d_real_loss: 0.4866 | d_Y_loss: 0.7941 | d_X_loss:
0.4162 | d_fake_loss: 1.2103 | g_loss: 3.7687
Iteration [ 2430/10000] | d_real_loss: 0.3042 | d_Y_loss: 0.7050 | d_X_loss:
```

```
0.8038 | d_fake_loss: 1.5088 | g_loss: 4.1336
Iteration [ 2440/10000] | d_real_loss: 0.3348 | d_Y_loss: 0.4631 | d_X_loss:
0.6964 | d_fake_loss: 1.1595 | g_loss: 3.8084
Iteration [ 2450/10000] | d_real_loss: 0.3638 | d_Y_loss: 0.3479 | d_X_loss:
0.5039 | d_fake_loss: 0.8519 | g_loss: 3.9855
Iteration [ 2460/10000] | d_real_loss: 0.4045 | d_Y_loss: 0.8386 | d_X_loss:
0.5856 | d_fake_loss: 1.4242 | g_loss: 3.7462
Iteration [ 2470/10000] | d_real_loss: 0.3916 | d_Y_loss: 0.7251 | d_X_loss:
0.3810 | d_fake_loss: 1.1061 | g_loss: 3.5996
Iteration [ 2480/10000] | d_real_loss: 0.4734 | d_Y_loss: 0.5784 | d_X_loss:
0.9542 | d_fake_loss: 1.5326 | g_loss: 3.9450
Iteration [ 2490/10000] | d_real_loss: 0.4012 | d_Y_loss: 0.5949 | d_X_loss:
0.6288 | d_fake_loss: 1.2237 | g_loss: 4.0429
Iteration [ 2500/10000] | d_real_loss: 0.5699 | d_Y_loss: 0.4788 | d_X_loss:
0.2590 | d_fake_loss: 0.7378 | g_loss: 3.7814
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002500-Y-X.png
Iteration [ 2510/10000] | d_real_loss: 0.3988 | d_Y_loss: 0.9626 | d_X_loss:
0.2361 | d_fake_loss: 1.1987 | g_loss: 3.7734
Iteration [ 2520/10000] | d_real_loss: 0.4207 | d_Y_loss: 0.3289 | d_X_loss:
0.2830 | d_fake_loss: 0.6119 | g_loss: 4.4864
Iteration [ 2530/10000] | d_real_loss: 0.4532 | d_Y_loss: 0.4228 | d_X_loss:
0.1827 | d_fake_loss: 0.6055 | g_loss: 3.7084
Iteration [ 2540/10000] | d_real_loss: 0.3956 | d_Y_loss: 0.4197 | d_X_loss:
0.2291 | d_fake_loss: 0.6488 | g_loss: 3.7092
Iteration [ 2550/10000] | d_real_loss: 0.3388 | d_Y_loss: 0.5079 | d_X_loss:
0.4232 | d_fake_loss: 0.9311 | g_loss: 3.7785
Iteration [ 2560/10000] | d_real_loss: 0.3667 | d_Y_loss: 0.5629 | d_X_loss:
0.7135 | d_fake_loss: 1.2764 | g_loss: 3.7721
Iteration [ 2570/10000] | d_real_loss: 0.3310 | d_Y_loss: 0.3567 | d_X_loss:
0.3296 | d_fake_loss: 0.6863 | g_loss: 3.4631
Iteration [ 2580/10000] | d_real_loss: 0.3497 | d_Y_loss: 0.6789 | d_X_loss:
0.6197 | d_fake_loss: 1.2987 | g_loss: 3.5436
Iteration [ 2590/10000] | d_real_loss: 0.4068 | d_Y_loss: 0.4664 | d_X_loss:
0.5040 | d_fake_loss: 0.9704 | g_loss: 3.7762
Iteration [ 2600/10000] | d_real_loss: 0.3058 | d_Y_loss: 0.3996 | d_X_loss:
0.6853 | d_fake_loss: 1.0850 | g_loss: 3.7694
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002600-Y-X.png
Iteration [ 2610/10000] | d_real_loss: 0.3762 | d_Y_loss: 0.5917 | d_X_loss:
0.4106 | d_fake_loss: 1.0024 | g_loss: 3.7765
Iteration [ 2620/10000] | d_real_loss: 0.4105 | d_Y_loss: 0.6143 | d_X_loss:
0.3695 | d_fake_loss: 0.9838 | g_loss: 3.7079
Iteration [ 2630/10000] | d_real_loss: 0.4730 | d_Y_loss: 0.3866 | d_X_loss:
```

```
0.4464 | d_fake_loss: 0.8330 | g_loss: 3.9470
Iteration [ 2640/10000] | d_real_loss: 0.2721 | d_Y_loss: 0.4496 | d_X_loss:
0.1438 | d_fake_loss: 0.5934 | g_loss: 3.9337
Iteration [ 2650/10000] | d_real_loss: 0.2805 | d_Y_loss: 0.6841 | d_X_loss:
0.1450 | d_fake_loss: 0.8291 | g_loss: 3.7228
Iteration [ 2660/10000] | d_real_loss: 0.2843 | d_Y_loss: 0.4871 | d_X_loss:
0.2747 | d_fake_loss: 0.7618 | g_loss: 3.7271
Iteration [ 2670/10000] | d_real_loss: 0.4365 | d_Y_loss: 0.3995 | d_X_loss:
0.4386 | d_fake_loss: 0.8381 | g_loss: 3.8497
Iteration [ 2680/10000] | d_real_loss: 0.2804 | d_Y_loss: 0.4264 | d_X_loss:
0.1925 | d_fake_loss: 0.6190 | g_loss: 4.1821
Iteration [ 2690/10000] | d_real_loss: 0.2539 | d_Y_loss: 0.4630 | d_X_loss:
0.1217 | d_fake_loss: 0.5846 | g_loss: 4.1497
Iteration [ 2700/10000] | d_real_loss: 0.2832 | d_Y_loss: 0.5727 | d_X_loss:
1.0125 | d_fake_loss: 1.5852 | g_loss: 3.6828
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002700-Y-X.png
Iteration [ 2710/10000] | d_real_loss: 0.4260 | d_Y_loss: 1.2082 | d_X_loss:
0.1721 | d_fake_loss: 1.3803 | g_loss: 3.6891
Iteration [ 2720/10000] | d_real_loss: 0.3222 | d_Y_loss: 0.4897 | d_X_loss:
0.3515 | d_fake_loss: 0.8412 | g_loss: 3.7875
Iteration [ 2730/10000] | d_real_loss: 0.3986 | d_Y_loss: 0.3176 | d_X_loss:
0.6288 | d_fake_loss: 0.9464 | g_loss: 4.2837
Iteration [ 2740/10000] | d_real_loss: 0.3816 | d_Y_loss: 0.5166 | d_X_loss:
0.4198 | d_fake_loss: 0.9364 | g_loss: 3.8688
Iteration [ 2750/10000] | d_real_loss: 0.4279 | d_Y_loss: 0.3785 | d_X_loss:
0.2594 | d_fake_loss: 0.6379 | g_loss: 3.7346
Iteration [ 2760/10000] | d_real_loss: 0.3605 | d_Y_loss: 0.6474 | d_X_loss:
0.1928 | d_fake_loss: 0.8401 | g_loss: 4.0709
Iteration [ 2770/10000] | d_real_loss: 0.4505 | d_Y_loss: 0.5676 | d_X_loss:
0.4468 | d_fake_loss: 1.0145 | g_loss: 3.5035
Iteration [ 2780/10000] | d_real_loss: 0.4359 | d_Y_loss: 0.6713 | d_X_loss:
1.0222 | d_fake_loss: 1.6936 | g_loss: 4.0249
Iteration [ 2790/10000] | d_real_loss: 0.4775 | d_Y_loss: 0.6035 | d_X_loss:
0.4845 | d_fake_loss: 1.0880 | g_loss: 3.8094
Iteration [ 2800/10000] | d_real_loss: 0.3672 | d_Y_loss: 0.7073 | d_X_loss:
0.4892 | d_fake_loss: 1.1965 | g_loss: 4.0137
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002800-Y-X.png
Iteration [ 2810/10000] | d_real_loss: 0.3821 | d_Y_loss: 0.8503 | d_X_loss:
0.4815 | d_fake_loss: 1.3318 | g_loss: 3.5797
Iteration [ 2820/10000] | d_real_loss: 0.5653 | d_Y_loss: 0.7278 | d_X_loss:
0.6327 | d_fake_loss: 1.3605 | g_loss: 3.6196
Iteration [ 2830/10000] | d_real_loss: 0.5530 | d_Y_loss: 0.5101 | d_X_loss:
```

```
0.3837 | d_fake_loss: 0.8938 | g_loss: 3.6719
Iteration [ 2840/10000] | d_real_loss: 0.2549 | d_Y_loss: 0.6492 | d_X_loss:
0.2570 | d_fake_loss: 0.9062 | g_loss: 3.6858
Iteration [ 2850/10000] | d_real_loss: 0.3669 | d_Y_loss: 0.4861 | d_X_loss:
0.2377 | d_fake_loss: 0.7238 | g_loss: 4.2029
Iteration [ 2860/10000] | d_real_loss: 0.3629 | d_Y_loss: 0.8045 | d_X_loss:
0.7648 | d_fake_loss: 1.5693 | g_loss: 3.5585
Iteration [ 2870/10000] | d_real_loss: 0.2773 | d_Y_loss: 0.5241 | d_X_loss:
0.5914 | d_fake_loss: 1.1155 | g_loss: 3.8787
Iteration [ 2880/10000] | d_real_loss: 0.2928 | d_Y_loss: 0.6899 | d_X_loss:
0.6807 | d_fake_loss: 1.3706 | g_loss: 3.9398
Iteration [ 2890/10000] | d_real_loss: 0.3373 | d_Y_loss: 0.7808 | d_X_loss:
0.7534 | d_fake_loss: 1.5342 | g_loss: 3.4014
Iteration [ 2900/10000] | d_real_loss: 0.5526 | d_Y_loss: 0.7853 | d_X_loss:
0.2849 | d_fake_loss: 1.0702 | g_loss: 3.3442
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
002900-Y-X.png
Iteration [ 2910/10000] | d_real_loss: 0.5454 | d_Y_loss: 0.7039 | d_X_loss:
0.5159 | d_fake_loss: 1.2199 | g_loss: 4.0865
Iteration [ 2920/10000] | d_real_loss: 0.4497 | d_Y_loss: 0.5378 | d_X_loss:
0.6765 | d_fake_loss: 1.2143 | g_loss: 4.6774
Iteration [ 2930/10000] | d_real_loss: 0.2691 | d_Y_loss: 0.3247 | d_X_loss:
0.4516 | d_fake_loss: 0.7763 | g_loss: 4.1109
Iteration [ 2940/10000] | d_real_loss: 0.5256 | d_Y_loss: 0.7492 | d_X_loss:
0.8322 | d_fake_loss: 1.5815 | g_loss: 3.6094
Iteration [ 2950/10000] | d_real_loss: 0.4526 | d_Y_loss: 0.5469 | d_X_loss:
0.8070 | d_fake_loss: 1.3539 | g_loss: 3.6652
Iteration [ 2960/10000] | d_real_loss: 0.4542 | d_Y_loss: 0.5222 | d_X_loss:
0.4966 | d_fake_loss: 1.0187 | g_loss: 3.8414
Iteration [ 2970/10000] | d_real_loss: 0.4346 | d_Y_loss: 0.5538 | d_X_loss:
0.3433 | d_fake_loss: 0.8971 | g_loss: 4.2082
Iteration [ 2980/10000] | d_real_loss: 0.3655 | d_Y_loss: 0.6042 | d_X_loss:
0.8343 | d_fake_loss: 1.4385 | g_loss: 3.6096
Iteration [ 2990/10000] | d_real_loss: 0.5142 | d_Y_loss: 0.5718 | d_X_loss:
0.3374 | d_fake_loss: 0.9092 | g_loss: 3.9478
Iteration [ 3000/10000] | d_real_loss: 0.4631 | d_Y_loss: 0.7527 | d_X_loss:
0.6354 | d_fake_loss: 1.3880 | g_loss: 3.3211
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003000-Y-X.png
Iteration [ 3010/10000] | d_real_loss: 0.3027 | d_Y_loss: 0.4782 | d_X_loss:
0.1994 | d_fake_loss: 0.6776 | g_loss: 4.1392
Iteration [ 3020/10000] | d_real_loss: 0.3773 | d_Y_loss: 0.6188 | d_X_loss:
0.3030 | d_fake_loss: 0.9217 | g_loss: 3.9892
Iteration [ 3030/10000] | d_real_loss: 0.5625 | d_Y_loss: 0.6071 | d_X_loss:
```

```
0.6186 | d_fake_loss: 1.2257 | g_loss: 3.6578
Iteration [ 3040/10000] | d_real_loss: 0.3836 | d_Y_loss: 0.5775 | d_X_loss:
0.6115 | d_fake_loss: 1.1890 | g_loss: 3.7339
Iteration [ 3050/10000] | d_real_loss: 0.3889 | d_Y_loss: 0.4790 | d_X_loss:
1.0191 | d_fake_loss: 1.4981 | g_loss: 4.0695
Iteration [ 3060/10000] | d_real_loss: 0.3664 | d_Y_loss: 0.4955 | d_X_loss:
0.8759 | d_fake_loss: 1.3714 | g_loss: 3.5914
Iteration [ 3070/10000] | d_real_loss: 0.4339 | d_Y_loss: 0.5708 | d_X_loss:
0.8644 | d_fake_loss: 1.4352 | g_loss: 4.4304
Iteration [ 3080/10000] | d_real_loss: 0.4455 | d_Y_loss: 0.4748 | d_X_loss:
0.4892 | d_fake_loss: 0.9640 | g_loss: 4.0684
Iteration [ 3090/10000] | d_real_loss: 0.4819 | d_Y_loss: 0.4675 | d_X_loss:
0.4314 | d_fake_loss: 0.8989 | g_loss: 3.8516
Iteration [ 3100/10000] | d_real_loss: 0.4169 | d_Y_loss: 0.5062 | d_X_loss:
0.3347 | d_fake_loss: 0.8408 | g_loss: 3.9183
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003100-Y-X.png
Iteration [ 3110/10000] | d_real_loss: 0.4029 | d_Y_loss: 0.5815 | d_X_loss:
1.0198 | d_fake_loss: 1.6013 | g_loss: 4.0233
Iteration [ 3120/10000] | d_real_loss: 0.5416 | d_Y_loss: 0.5762 | d_X_loss:
0.3070 | d_fake_loss: 0.8832 | g_loss: 3.8321
Iteration [ 3130/10000] | d_real_loss: 0.5702 | d_Y_loss: 0.5725 | d_X_loss:
0.3090 | d_fake_loss: 0.8815 | g_loss: 4.0031
Iteration [ 3140/10000] | d_real_loss: 0.3218 | d_Y_loss: 0.6097 | d_X_loss:
0.3043 | d_fake_loss: 0.9140 | g_loss: 3.7555
Iteration [ 3150/10000] | d_real_loss: 0.3211 | d_Y_loss: 0.5677 | d_X_loss:
0.3491 | d_fake_loss: 0.9168 | g_loss: 3.6605
Iteration [ 3160/10000] | d_real_loss: 0.2472 | d_Y_loss: 0.8539 | d_X_loss:
0.6214 | d_fake_loss: 1.4753 | g_loss: 3.6576
Iteration [ 3170/10000] | d_real_loss: 0.3498 | d_Y_loss: 0.4797 | d_X_loss:
0.3788 | d_fake_loss: 0.8585 | g_loss: 3.6839
Iteration [ 3180/10000] | d_real_loss: 0.6818 | d_Y_loss: 0.6532 | d_X_loss:
0.4729 | d_fake_loss: 1.1261 | g_loss: 4.1122
Iteration [ 3190/10000] | d_real_loss: 0.3776 | d_Y_loss: 0.4785 | d_X_loss:
0.8186 | d_fake_loss: 1.2971 | g_loss: 3.6908
Iteration [ 3200/10000] | d_real_loss: 0.4022 | d_Y_loss: 0.5056 | d_X_loss:
0.5270 | d_fake_loss: 1.0326 | g_loss: 3.9032
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003200-Y-X.png
Iteration [ 3210/10000] | d_real_loss: 0.5658 | d_Y_loss: 0.4828 | d_X_loss:
0.1904 | d_fake_loss: 0.6732 | g_loss: 4.0203
Iteration [ 3220/10000] | d_real_loss: 0.4301 | d_Y_loss: 0.6764 | d_X_loss:
0.2757 | d_fake_loss: 0.9521 | g_loss: 3.6463
Iteration [ 3230/10000] | d_real_loss: 0.6072 | d_Y_loss: 0.5538 | d_X_loss:
```

```
1.1040 | d_fake_loss: 1.6578 | g_loss: 3.7266
Iteration [ 3240/10000] | d_real_loss: 0.3894 | d_Y_loss: 0.6116 | d_X_loss:
1.0182 | d_fake_loss: 1.6299 | g_loss: 3.8075
Iteration [ 3250/10000] | d_real_loss: 0.3239 | d_Y_loss: 0.7420 | d_X_loss:
0.4535 | d_fake_loss: 1.1955 | g_loss: 3.6871
Iteration [ 3260/10000] | d_real_loss: 0.3798 | d_Y_loss: 0.7722 | d_X_loss:
0.3947 | d_fake_loss: 1.1669 | g_loss: 3.7045
Iteration [ 3270/10000] | d_real_loss: 0.4126 | d_Y_loss: 0.9269 | d_X_loss:
0.3816 | d_fake_loss: 1.3085 | g_loss: 3.3648
Iteration [ 3280/10000] | d_real_loss: 0.5206 | d_Y_loss: 0.4835 | d_X_loss:
0.4442 | d_fake_loss: 0.9277 | g_loss: 3.7362
Iteration [ 3290/10000] | d_real_loss: 0.2050 | d_Y_loss: 0.6996 | d_X_loss:
0.1138 | d_fake_loss: 0.8133 | g_loss: 4.1093
Iteration [ 3300/10000] | d_real_loss: 0.4151 | d_Y_loss: 0.5577 | d_X_loss:
0.1003 | d_fake_loss: 0.6580 | g_loss: 3.7208
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003300-Y-X.png
Iteration [ 3310/10000] | d_real_loss: 0.2250 | d_Y_loss: 0.6465 | d_X_loss:
0.0961 | d_fake_loss: 0.7426 | g_loss: 4.0263
Iteration [ 3320/10000] | d_real_loss: 0.4035 | d_Y_loss: 0.5793 | d_X_loss:
0.1019 | d_fake_loss: 0.6812 | g_loss: 3.9378
Iteration [ 3330/10000] | d_real_loss: 0.2447 | d_Y_loss: 0.6820 | d_X_loss:
0.1266 | d_fake_loss: 0.8086 | g_loss: 4.1616
Iteration [ 3340/10000] | d_real_loss: 0.1709 | d_Y_loss: 0.5828 | d_X_loss:
0.1462 | d_fake_loss: 0.7290 | g_loss: 3.9073
Iteration [ 3350/10000] | d_real_loss: 0.2078 | d_Y_loss: 0.7269 | d_X_loss:
0.1789 | d_fake_loss: 0.9058 | g_loss: 3.6527
Iteration [ 3360/10000] | d_real_loss: 0.2698 | d_Y_loss: 0.6229 | d_X_loss:
0.1220 | d_fake_loss: 0.7450 | g_loss: 3.7325
Iteration [ 3370/10000] | d_real_loss: 0.2257 | d_Y_loss: 0.4834 | d_X_loss:
0.1401 | d_fake_loss: 0.6235 | g_loss: 3.7815
Iteration [ 3380/10000] | d_real_loss: 0.2086 | d_Y_loss: 0.6998 | d_X_loss:
0.1386 | d_fake_loss: 0.8385 | g_loss: 4.2391
Iteration [ 3390/10000] | d_real_loss: 0.2992 | d_Y_loss: 0.4849 | d_X_loss:
0.1244 | d_fake_loss: 0.6093 | g_loss: 3.8976
Iteration [ 3400/10000] | d_real_loss: 0.2665 | d_Y_loss: 0.4630 | d_X_loss:
0.1522 | d_fake_loss: 0.6152 | g_loss: 3.9933
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003400-Y-X.png
Iteration [ 3410/10000] | d_real_loss: 0.3296 | d_Y_loss: 0.2339 | d_X_loss:
0.1164 | d_fake_loss: 0.3503 | g_loss: 4.4631
Iteration [ 3420/10000] | d_real_loss: 0.1966 | d_Y_loss: 0.4590 | d_X_loss:
0.5778 | d_fake_loss: 1.0368 | g_loss: 3.8707
Iteration [ 3430/10000] | d_real_loss: 0.2593 | d_Y_loss: 0.8155 | d_X_loss:
```

```
1.2814 | d_fake_loss: 2.0969 | g_loss: 4.3169
Iteration [ 3440/10000] | d_real_loss: 0.3929 | d_Y_loss: 0.5379 | d_X_loss:
0.2290 | d_fake_loss: 0.7669 | g_loss: 3.7392
Iteration [ 3450/10000] | d_real_loss: 0.3157 | d_Y_loss: 0.6770 | d_X_loss:
0.2530 | d_fake_loss: 0.9300 | g_loss: 3.8905
Iteration [ 3460/10000] | d_real_loss: 0.4439 | d_Y_loss: 0.5540 | d_X_loss:
0.4015 | d_fake_loss: 0.9555 | g_loss: 3.9564
Iteration [ 3470/10000] | d_real_loss: 0.3274 | d_Y_loss: 0.5343 | d_X_loss:
0.1379 | d_fake_loss: 0.6722 | g_loss: 4.0541
Iteration [ 3480/10000] | d_real_loss: 0.3183 | d_Y_loss: 0.5695 | d_X_loss:
0.7268 | d_fake_loss: 1.2962 | g_loss: 4.0590
Iteration [ 3490/10000] | d_real_loss: 0.3682 | d_Y_loss: 0.5374 | d_X_loss:
0.8991 | d_fake_loss: 1.4364 | g_loss: 3.7796
Iteration [ 3500/10000] | d_real_loss: 0.2728 | d_Y_loss: 0.7035 | d_X_loss:
0.2985 | d_fake_loss: 1.0020 | g_loss: 4.0268
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003500-Y-X.png
Iteration [ 3510/10000] | d_real_loss: 0.4687 | d_Y_loss: 0.7748 | d_X_loss:
0.6593 | d_fake_loss: 1.4341 | g_loss: 3.4343
Iteration [ 3520/10000] | d_real_loss: 0.4708 | d_Y_loss: 0.7874 | d_X_loss:
0.2622 | d_fake_loss: 1.0496 | g_loss: 3.7299
Iteration [ 3530/10000] | d_real_loss: 0.4957 | d_Y_loss: 0.4561 | d_X_loss:
0.4030 | d_fake_loss: 0.8591 | g_loss: 3.9368
Iteration [ 3540/10000] | d_real_loss: 0.4181 | d_Y_loss: 0.7265 | d_X_loss:
0.4871 | d_fake_loss: 1.2136 | g_loss: 4.2842
Iteration [ 3550/10000] | d_real_loss: 0.3508 | d_Y_loss: 0.8164 | d_X_loss:
1.0899 | d_fake_loss: 1.9063 | g_loss: 4.0955
Iteration [ 3560/10000] | d_real_loss: 0.6593 | d_Y_loss: 0.4597 | d_X_loss:
0.6598 | d_fake_loss: 1.1195 | g_loss: 3.9094
Iteration [ 3570/10000] | d_real_loss: 0.4740 | d_Y_loss: 0.6895 | d_X_loss:
0.4678 | d_fake_loss: 1.1573 | g_loss: 3.8664
Iteration [ 3580/10000] | d_real_loss: 0.5856 | d_Y_loss: 0.9050 | d_X_loss:
0.2517 | d_fake_loss: 1.1567 | g_loss: 4.1138
Iteration [ 3590/10000] | d_real_loss: 0.3171 | d_Y_loss: 0.6613 | d_X_loss:
0.3389 | d_fake_loss: 1.0001 | g_loss: 3.8184
Iteration [ 3600/10000] | d_real_loss: 0.5098 | d_Y_loss: 0.4658 | d_X_loss:
0.2291 | d_fake_loss: 0.6949 | g_loss: 3.9267
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003600-Y-X.png
Iteration [ 3610/10000] | d_real_loss: 0.3629 | d_Y_loss: 0.8029 | d_X_loss:
0.7397 | d_fake_loss: 1.5425 | g_loss: 4.0148
Iteration [ 3620/10000] | d_real_loss: 0.5613 | d_Y_loss: 0.5462 | d_X_loss:
0.7266 | d_fake_loss: 1.2728 | g_loss: 3.5998
Iteration [ 3630/10000] | d_real_loss: 0.6451 | d_Y_loss: 0.6014 | d_X_loss:
```

```
0.3976 | d_fake_loss: 0.9990 | g_loss: 3.7139
Iteration [ 3640/10000] | d_real_loss: 0.3290 | d_Y_loss: 0.7029 | d_X_loss:
0.5876 | d_fake_loss: 1.2904 | g_loss: 3.6524
Iteration [ 3650/10000] | d_real_loss: 0.4072 | d_Y_loss: 0.8613 | d_X_loss:
0.2959 | d_fake_loss: 1.1573 | g_loss: 3.8570
Iteration [ 3660/10000] | d_real_loss: 0.4415 | d_Y_loss: 0.4403 | d_X_loss:
0.4132 | d_fake_loss: 0.8535 | g_loss: 3.5991
Iteration [ 3670/10000] | d_real_loss: 0.3972 | d_Y_loss: 0.7511 | d_X_loss:
0.3749 | d_fake_loss: 1.1260 | g_loss: 3.3261
Iteration [ 3680/10000] | d_real_loss: 0.4443 | d_Y_loss: 0.7596 | d_X_loss:
0.3340 | d_fake_loss: 1.0937 | g_loss: 3.6470
Iteration [ 3690/10000] | d_real_loss: 0.4534 | d_Y_loss: 0.4816 | d_X_loss:
0.6622 | d_fake_loss: 1.1437 | g_loss: 4.1082
Iteration [ 3700/10000] | d_real_loss: 0.4473 | d_Y_loss: 0.7163 | d_X_loss:
0.8587 | d_fake_loss: 1.5750 | g_loss: 3.6362
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003700-Y-X.png
Iteration [ 3710/10000] | d_real_loss: 0.5452 | d_Y_loss: 0.6435 | d_X_loss:
0.6936 | d_fake_loss: 1.3371 | g_loss: 3.6961
Iteration [ 3720/10000] | d_real_loss: 0.5942 | d_Y_loss: 0.5205 | d_X_loss:
0.3141 | d_fake_loss: 0.8346 | g_loss: 3.8948
Iteration [ 3730/10000] | d_real_loss: 0.4235 | d_Y_loss: 0.4862 | d_X_loss:
0.6615 | d_fake_loss: 1.1477 | g_loss: 3.5160
Iteration [ 3740/10000] | d_real_loss: 0.5595 | d_Y_loss: 0.6575 | d_X_loss:
0.3841 | d_fake_loss: 1.0416 | g_loss: 3.7886
Iteration [ 3750/10000] | d_real_loss: 0.2799 | d_Y_loss: 0.8779 | d_X_loss:
0.3080 | d_fake_loss: 1.1859 | g_loss: 3.8347
Iteration [ 3760/10000] | d_real_loss: 0.4313 | d_Y_loss: 0.5152 | d_X_loss:
0.2383 | d_fake_loss: 0.7536 | g_loss: 3.8059
Iteration [ 3770/10000] | d_real_loss: 0.5559 | d_Y_loss: 0.5725 | d_X_loss:
0.3797 | d_fake_loss: 0.9522 | g_loss: 3.9983
Iteration [ 3780/10000] | d_real_loss: 0.3652 | d_Y_loss: 0.6993 | d_X_loss:
0.6306 | d_fake_loss: 1.3299 | g_loss: 3.7019
Iteration [ 3790/10000] | d_real_loss: 0.3557 | d_Y_loss: 0.5503 | d_X_loss:
0.9072 | d_fake_loss: 1.4575 | g_loss: 3.6075
Iteration [ 3800/10000] | d_real_loss: 0.7594 | d_Y_loss: 0.4229 | d_X_loss:
1.2077 | d_fake_loss: 1.6306 | g_loss: 3.7708
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003800-Y-X.png
Iteration [ 3810/10000] | d_real_loss: 0.4012 | d_Y_loss: 0.5167 | d_X_loss:
0.4980 | d_fake_loss: 1.0146 | g_loss: 3.8933
Iteration [ 3820/10000] | d_real_loss: 0.4284 | d_Y_loss: 0.4762 | d_X_loss:
0.4731 | d_fake_loss: 0.9493 | g_loss: 3.5221
Iteration [ 3830/10000] | d_real_loss: 0.4138 | d_Y_loss: 0.8240 | d_X_loss:
```

```
0.5124 | d_fake_loss: 1.3364 | g_loss: 3.6272
Iteration [ 3840/10000] | d_real_loss: 0.5051 | d_Y_loss: 0.5383 | d_X_loss:
0.9630 | d_fake_loss: 1.5013 | g_loss: 3.6166
Iteration [ 3850/10000] | d_real_loss: 0.4907 | d_Y_loss: 0.5707 | d_X_loss:
0.4796 | d_fake_loss: 1.0503 | g_loss: 3.7975
Iteration [ 3860/10000] | d_real_loss: 0.4394 | d_Y_loss: 0.8924 | d_X_loss:
0.7434 | d_fake_loss: 1.6358 | g_loss: 3.5291
Iteration [ 3870/10000] | d_real_loss: 0.4679 | d_Y_loss: 0.5281 | d_X_loss:
0.3423 | d_fake_loss: 0.8705 | g_loss: 3.4172
Iteration [ 3880/10000] | d_real_loss: 0.5154 | d_Y_loss: 0.8784 | d_X_loss:
0.3745 | d_fake_loss: 1.2529 | g_loss: 3.4619
Iteration [ 3890/10000] | d_real_loss: 0.4549 | d_Y_loss: 0.5948 | d_X_loss:
0.5673 | d_fake_loss: 1.1621 | g_loss: 3.7404
Iteration [ 3900/10000] | d_real_loss: 0.5486 | d_Y_loss: 0.5628 | d_X_loss:
0.4089 | d_fake_loss: 0.9716 | g_loss: 3.7692
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
003900-Y-X.png
Iteration [ 3910/10000] | d_real_loss: 0.5539 | d_Y_loss: 0.4498 | d_X_loss:
0.5963 | d_fake_loss: 1.0460 | g_loss: 3.6410
Iteration [ 3920/10000] | d_real_loss: 0.4882 | d_Y_loss: 0.6541 | d_X_loss:
0.6346 | d_fake_loss: 1.2888 | g_loss: 4.1097
Iteration [ 3930/10000] | d_real_loss: 0.3121 | d_Y_loss: 0.8875 | d_X_loss:
0.5316 | d_fake_loss: 1.4191 | g_loss: 3.4933
Iteration [ 3940/10000] | d_real_loss: 0.4536 | d_Y_loss: 0.8717 | d_X_loss:
0.3146 | d_fake_loss: 1.1863 | g_loss: 3.5476
Iteration [ 3950/10000] | d_real_loss: 0.5831 | d_Y_loss: 0.6075 | d_X_loss:
0.2655 | d_fake_loss: 0.8731 | g_loss: 3.9954
Iteration [ 3960/10000] | d_real_loss: 0.3863 | d_Y_loss: 0.8070 | d_X_loss:
1.1341 | d_fake_loss: 1.9411 | g_loss: 3.6372
Iteration [ 3970/10000] | d_real_loss: 0.4770 | d_Y_loss: 0.6953 | d_X_loss:
0.3149 | d_fake_loss: 1.0102 | g_loss: 3.5389
Iteration [ 3980/10000] | d_real_loss: 0.3824 | d_Y_loss: 0.6623 | d_X_loss:
0.5256 | d_fake_loss: 1.1880 | g_loss: 4.2761
Iteration [ 3990/10000] | d_real_loss: 0.5361 | d_Y_loss: 0.6932 | d_X_loss:
0.4419 | d_fake_loss: 1.1351 | g_loss: 3.8923
Iteration [ 4000/10000] | d_real_loss: 0.4329 | d_Y_loss: 0.8217 | d_X_loss:
0.4495 | d_fake_loss: 1.2712 | g_loss: 3.5962
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004000-Y-X.png
Iteration [ 4010/10000] | d_real_loss: 0.4475 | d_Y_loss: 0.5216 | d_X_loss:
0.7067 | d_fake_loss: 1.2283 | g_loss: 3.6088
Iteration [ 4020/10000] | d_real_loss: 0.5380 | d_Y_loss: 0.8048 | d_X_loss:
0.9060 | d_fake_loss: 1.7108 | g_loss: 4.0421
Iteration [ 4030/10000] | d_real_loss: 0.3834 | d_Y_loss: 1.2478 | d_X_loss:
```

```
0.4243 | d_fake_loss: 1.6721 | g_loss: 3.9764
Iteration [ 4040/10000] | d_real_loss: 0.4336 | d_Y_loss: 0.7378 | d_X_loss:
0.3806 | d_fake_loss: 1.1184 | g_loss: 3.8748
Iteration [ 4050/10000] | d_real_loss: 0.3155 | d_Y_loss: 0.5464 | d_X_loss:
0.4583 | d_fake_loss: 1.0047 | g_loss: 3.9161
Iteration [ 4060/10000] | d_real_loss: 0.4250 | d_Y_loss: 0.7472 | d_X_loss:
0.8637 | d_fake_loss: 1.6109 | g_loss: 3.9142
Iteration [ 4070/10000] | d_real_loss: 0.4776 | d_Y_loss: 0.6312 | d_X_loss:
0.9082 | d_fake_loss: 1.5395 | g_loss: 3.7458
Iteration [ 4080/10000] | d_real_loss: 0.5022 | d_Y_loss: 0.5204 | d_X_loss:
0.2704 | d_fake_loss: 0.7908 | g_loss: 3.8775
Iteration [ 4090/10000] | d_real_loss: 0.5570 | d_Y_loss: 0.8005 | d_X_loss:
0.3993 | d_fake_loss: 1.1999 | g_loss: 3.6664
Iteration [ 4100/10000] | d_real_loss: 0.6008 | d_Y_loss: 0.4602 | d_X_loss:
0.7408 | d_fake_loss: 1.2009 | g_loss: 4.0867
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004100-Y-X.png
Iteration [ 4110/10000] | d_real_loss: 0.5500 | d_Y_loss: 0.6999 | d_X_loss:
0.7755 | d_fake_loss: 1.4754 | g_loss: 3.4953
Iteration [ 4120/10000] | d_real_loss: 0.3823 | d_Y_loss: 0.8308 | d_X_loss:
0.2637 | d_fake_loss: 1.0945 | g_loss: 3.8578
Iteration [ 4130/10000] | d_real_loss: 0.4332 | d_Y_loss: 0.6421 | d_X_loss:
0.6007 | d_fake_loss: 1.2428 | g_loss: 3.9567
Iteration [ 4140/10000] | d_real_loss: 0.4916 | d_Y_loss: 1.0308 | d_X_loss:
0.5096 | d_fake_loss: 1.5404 | g_loss: 3.4944
Iteration [ 4150/10000] | d_real_loss: 0.5747 | d_Y_loss: 0.6311 | d_X_loss:
0.7309 | d_fake_loss: 1.3620 | g_loss: 3.7270
Iteration [ 4160/10000] | d_real_loss: 0.4890 | d_Y_loss: 0.5976 | d_X_loss:
0.7605 | d_fake_loss: 1.3581 | g_loss: 3.7571
Iteration [ 4170/10000] | d_real_loss: 0.2926 | d_Y_loss: 0.4887 | d_X_loss:
0.6211 | d_fake_loss: 1.1098 | g_loss: 4.0900
Iteration [ 4180/10000] | d_real_loss: 0.3447 | d_Y_loss: 0.7584 | d_X_loss:
0.6878 | d_fake_loss: 1.4462 | g_loss: 3.6959
Iteration [ 4190/10000] | d_real_loss: 0.4702 | d_Y_loss: 0.5473 | d_X_loss:
0.2920 | d_fake_loss: 0.8394 | g_loss: 4.0021
Iteration [ 4200/10000] | d_real_loss: 0.3356 | d_Y_loss: 0.6259 | d_X_loss:
0.6506 | d_fake_loss: 1.2765 | g_loss: 3.5218
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004200-Y-X.png
Iteration [ 4210/10000] | d_real_loss: 0.5339 | d_Y_loss: 0.4641 | d_X_loss:
0.6883 | d_fake_loss: 1.1524 | g_loss: 3.7034
Iteration [ 4220/10000] | d_real_loss: 0.5803 | d_Y_loss: 0.9452 | d_X_loss:
0.3855 | d_fake_loss: 1.3308 | g_loss: 3.7039
Iteration [ 4230/10000] | d_real_loss: 0.3848 | d_Y_loss: 0.7055 | d_X_loss:
```

```
0.7301 | d_fake_loss: 1.4356 | g_loss: 3.7138
Iteration [ 4240/10000] | d_real_loss: 0.5200 | d_Y_loss: 0.3406 | d_X_loss:
0.2904 | d_fake_loss: 0.6310 | g_loss: 4.0003
Iteration [ 4250/10000] | d_real_loss: 0.4307 | d_Y_loss: 0.8175 | d_X_loss:
0.3471 | d_fake_loss: 1.1646 | g_loss: 3.7816
Iteration [ 4260/10000] | d_real_loss: 0.5549 | d_Y_loss: 0.6784 | d_X_loss:
0.4925 | d_fake_loss: 1.1708 | g_loss: 3.6945
Iteration [ 4270/10000] | d_real_loss: 0.5997 | d_Y_loss: 0.4338 | d_X_loss:
0.6162 | d_fake_loss: 1.0500 | g_loss: 3.6194
Iteration [ 4280/10000] | d_real_loss: 0.4419 | d_Y_loss: 0.6877 | d_X_loss:
0.5189 | d_fake_loss: 1.2066 | g_loss: 3.6524
Iteration [ 4290/10000] | d_real_loss: 0.4559 | d_Y_loss: 0.6365 | d_X_loss:
0.3943 | d_fake_loss: 1.0308 | g_loss: 4.3619
Iteration [ 4300/10000] | d_real_loss: 0.4245 | d_Y_loss: 0.5507 | d_X_loss:
0.5921 | d_fake_loss: 1.1428 | g_loss: 3.9107
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004300-Y-X.png
Iteration [ 4310/10000] | d_real_loss: 0.5454 | d_Y_loss: 0.5963 | d_X_loss:
0.5868 | d_fake_loss: 1.1831 | g_loss: 4.2184
Iteration [ 4320/10000] | d_real_loss: 0.4026 | d_Y_loss: 0.7449 | d_X_loss:
0.5353 | d_fake_loss: 1.2802 | g_loss: 3.7656
Iteration [ 4330/10000] | d_real_loss: 0.5824 | d_Y_loss: 0.6340 | d_X_loss:
0.7782 | d_fake_loss: 1.4122 | g_loss: 3.8323
Iteration [ 4340/10000] | d_real_loss: 0.4428 | d_Y_loss: 0.6653 | d_X_loss:
0.5564 | d_fake_loss: 1.2217 | g_loss: 3.5284
Iteration [ 4350/10000] | d_real_loss: 0.5489 | d_Y_loss: 0.5201 | d_X_loss:
0.3670 | d_fake_loss: 0.8871 | g_loss: 3.9199
Iteration [ 4360/10000] | d_real_loss: 0.3564 | d_Y_loss: 0.7362 | d_X_loss:
0.4425 | d_fake_loss: 1.1788 | g_loss: 4.0064
Iteration [ 4370/10000] | d_real_loss: 0.4848 | d_Y_loss: 0.8160 | d_X_loss:
0.3121 | d_fake_loss: 1.1281 | g_loss: 3.5670
Iteration [ 4380/10000] | d_real_loss: 0.3508 | d_Y_loss: 0.6491 | d_X_loss:
0.4545 | d_fake_loss: 1.1036 | g_loss: 3.8549
Iteration [ 4390/10000] | d_real_loss: 0.4667 | d_Y_loss: 0.3270 | d_X_loss:
0.6321 | d_fake_loss: 0.9591 | g_loss: 3.5012
Iteration [ 4400/10000] | d_real_loss: 0.4055 | d_Y_loss: 0.6543 | d_X_loss:
0.5423 | d_fake_loss: 1.1967 | g_loss: 3.6093
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004400-Y-X.png
Iteration [ 4410/10000] | d_real_loss: 0.4549 | d_Y_loss: 0.5643 | d_X_loss:
0.7340 | d_fake_loss: 1.2983 | g_loss: 4.3050
Iteration [ 4420/10000] | d_real_loss: 0.5717 | d_Y_loss: 0.7977 | d_X_loss:
0.3786 | d_fake_loss: 1.1763 | g_loss: 3.6952
Iteration [ 4430/10000] | d_real_loss: 0.4499 | d_Y_loss: 0.5180 | d_X_loss:
```

```
0.4289 | d_fake_loss: 0.9469 | g_loss: 4.0556
Iteration [ 4440/10000] | d_real_loss: 0.5385 | d_Y_loss: 0.5478 | d_X_loss:
0.7615 | d_fake_loss: 1.3093 | g_loss: 3.8810
Iteration [ 4450/10000] | d_real_loss: 0.3053 | d_Y_loss: 0.5077 | d_X_loss:
0.3802 | d_fake_loss: 0.8879 | g_loss: 3.8369
Iteration [ 4460/10000] | d_real_loss: 0.3590 | d_Y_loss: 0.7077 | d_X_loss:
0.8519 | d_fake_loss: 1.5596 | g_loss: 3.5895
Iteration [ 4470/10000] | d_real_loss: 0.5516 | d_Y_loss: 0.4524 | d_X_loss:
0.4944 | d_fake_loss: 0.9467 | g_loss: 3.8021
Iteration [ 4480/10000] | d_real_loss: 0.4368 | d_Y_loss: 0.7185 | d_X_loss:
0.8433 | d_fake_loss: 1.5618 | g_loss: 3.3715
Iteration [ 4490/10000] | d_real_loss: 0.4030 | d_Y_loss: 0.5989 | d_X_loss:
1.0789 | d_fake_loss: 1.6779 | g_loss: 3.9083
Iteration [ 4500/10000] | d_real_loss: 0.3512 | d_Y_loss: 0.6347 | d_X_loss:
0.5474 | d_fake_loss: 1.1821 | g_loss: 3.8555
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004500-Y-X.png
Iteration [ 4510/10000] | d_real_loss: 0.4036 | d_Y_loss: 0.4419 | d_X_loss:
0.1977 | d_fake_loss: 0.6396 | g_loss: 4.0028
Iteration [ 4520/10000] | d_real_loss: 0.7087 | d_Y_loss: 0.4975 | d_X_loss:
0.7708 | d_fake_loss: 1.2683 | g_loss: 3.8042
Iteration [ 4530/10000] | d_real_loss: 0.5857 | d_Y_loss: 0.4735 | d_X_loss:
0.2857 | d_fake_loss: 0.7593 | g_loss: 3.7495
Iteration [ 4540/10000] | d_real_loss: 0.5119 | d_Y_loss: 0.5051 | d_X_loss:
1.0222 | d_fake_loss: 1.5274 | g_loss: 3.6920
Iteration [ 4550/10000] | d_real_loss: 0.5630 | d_Y_loss: 0.7753 | d_X_loss:
0.2302 | d_fake_loss: 1.0054 | g_loss: 4.1984
Iteration [ 4560/10000] | d_real_loss: 0.3541 | d_Y_loss: 0.4872 | d_X_loss:
0.6568 | d_fake_loss: 1.1440 | g_loss: 4.0041
Iteration [ 4570/10000] | d_real_loss: 0.3902 | d_Y_loss: 0.8802 | d_X_loss:
0.2637 | d_fake_loss: 1.1439 | g_loss: 3.8020
Iteration [ 4580/10000] | d_real_loss: 0.5388 | d_Y_loss: 0.4939 | d_X_loss:
0.2569 | d_fake_loss: 0.7508 | g_loss: 3.7138
Iteration [ 4590/10000] | d_real_loss: 0.4105 | d_Y_loss: 0.6369 | d_X_loss:
1.2815 | d_fake_loss: 1.9184 | g_loss: 3.6591
Iteration [ 4600/10000] | d_real_loss: 0.4297 | d_Y_loss: 0.5527 | d_X_loss:
0.3724 | d_fake_loss: 0.9251 | g_loss: 3.7054
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004600-Y-X.png
Iteration [ 4610/10000] | d_real_loss: 0.4779 | d_Y_loss: 0.5134 | d_X_loss:
0.7102 | d_fake_loss: 1.2236 | g_loss: 3.6059
Iteration [ 4620/10000] | d_real_loss: 0.4295 | d_Y_loss: 0.3777 | d_X_loss:
0.5168 | d_fake_loss: 0.8945 | g_loss: 3.8186
Iteration [ 4630/10000] | d_real_loss: 0.3854 | d_Y_loss: 0.6153 | d_X_loss:
```

```
0.4463 | d_fake_loss: 1.0616 | g_loss: 4.1730
Iteration [ 4640/10000] | d_real_loss: 0.3648 | d_Y_loss: 0.3448 | d_X_loss:
0.8741 | d_fake_loss: 1.2190 | g_loss: 4.2454
Iteration [ 4650/10000] | d_real_loss: 0.3222 | d_Y_loss: 0.5997 | d_X_loss:
0.7784 | d_fake_loss: 1.3781 | g_loss: 4.0371
Iteration [ 4660/10000] | d_real_loss: 0.7386 | d_Y_loss: 0.4699 | d_X_loss:
0.1682 | d_fake_loss: 0.6381 | g_loss: 3.7260
Iteration [ 4670/10000] | d_real_loss: 0.4269 | d_Y_loss: 0.5877 | d_X_loss:
0.9102 | d_fake_loss: 1.4979 | g_loss: 3.5214
Iteration [ 4680/10000] | d_real_loss: 0.3916 | d_Y_loss: 0.5373 | d_X_loss:
0.6144 | d_fake_loss: 1.1516 | g_loss: 3.5036
Iteration [ 4690/10000] | d_real_loss: 0.5019 | d_Y_loss: 0.7599 | d_X_loss:
0.4420 | d_fake_loss: 1.2019 | g_loss: 4.0270
Iteration [ 4700/10000] | d_real_loss: 0.7316 | d_Y_loss: 0.5444 | d_X_loss:
0.7240 | d_fake_loss: 1.2684 | g_loss: 3.6554
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004700-Y-X.png
Iteration [ 4710/10000] | d_real_loss: 0.4127 | d_Y_loss: 0.5203 | d_X_loss:
0.6156 | d_fake_loss: 1.1359 | g_loss: 3.5882
Iteration [ 4720/10000] | d_real_loss: 0.7173 | d_Y_loss: 0.5235 | d_X_loss:
0.4273 | d_fake_loss: 0.9508 | g_loss: 4.0475
Iteration [ 4730/10000] | d_real_loss: 0.4154 | d_Y_loss: 0.6542 | d_X_loss:
0.7190 | d_fake_loss: 1.3732 | g_loss: 3.8441
Iteration [ 4740/10000] | d_real_loss: 0.3775 | d_Y_loss: 0.4932 | d_X_loss:
0.8198 | d_fake_loss: 1.3130 | g_loss: 3.8347
Iteration [ 4750/10000] | d_real_loss: 0.4551 | d_Y_loss: 0.5907 | d_X_loss:
0.7546 | d_fake_loss: 1.3452 | g_loss: 3.6743
Iteration [ 4760/10000] | d_real_loss: 0.4480 | d_Y_loss: 0.6646 | d_X_loss:
0.6129 | d_fake_loss: 1.2775 | g_loss: 4.1354
Iteration [ 4770/10000] | d_real_loss: 0.5605 | d_Y_loss: 0.6828 | d_X_loss:
0.2098 | d_fake_loss: 0.8925 | g_loss: 4.0665
Iteration [ 4780/10000] | d_real_loss: 0.6391 | d_Y_loss: 0.7855 | d_X_loss:
0.4944 | d_fake_loss: 1.2798 | g_loss: 3.2892
Iteration [ 4790/10000] | d_real_loss: 0.3824 | d_Y_loss: 0.6430 | d_X_loss:
0.5778 | d_fake_loss: 1.2207 | g_loss: 4.7047
Iteration [ 4800/10000] | d_real_loss: 0.3896 | d_Y_loss: 0.7764 | d_X_loss:
0.4524 | d_fake_loss: 1.2288 | g_loss: 3.8400
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004800-Y-X.png
Iteration [ 4810/10000] | d_real_loss: 0.3382 | d_Y_loss: 0.2387 | d_X_loss:
1.5448 | d_fake_loss: 1.7835 | g_loss: 4.1031
Iteration [ 4820/10000] | d_real_loss: 0.4042 | d_Y_loss: 0.5705 | d_X_loss:
0.7868 | d_fake_loss: 1.3573 | g_loss: 3.8407
Iteration [ 4830/10000] | d_real_loss: 0.3531 | d_Y_loss: 0.3987 | d_X_loss:
```

```
0.7221 | d_fake_loss: 1.1208 | g_loss: 4.0200
Iteration [ 4840/10000] | d_real_loss: 0.4641 | d_Y_loss: 0.6863 | d_X_loss:
0.7337 | d_fake_loss: 1.4200 | g_loss: 3.5537
Iteration [ 4850/10000] | d_real_loss: 0.2989 | d_Y_loss: 0.6155 | d_X_loss:
0.4624 | d_fake_loss: 1.0779 | g_loss: 4.0914
Iteration [ 4860/10000] | d_real_loss: 0.4470 | d_Y_loss: 0.3818 | d_X_loss:
1.2262 | d_fake_loss: 1.6079 | g_loss: 3.8981
Iteration [ 4870/10000] | d_real_loss: 0.3499 | d_Y_loss: 0.5925 | d_X_loss:
0.8253 | d_fake_loss: 1.4178 | g_loss: 3.7889
Iteration [ 4880/10000] | d_real_loss: 0.4092 | d_Y_loss: 0.7292 | d_X_loss:
0.3470 | d_fake_loss: 1.0763 | g_loss: 3.5937
Iteration [ 4890/10000] | d_real_loss: 0.3201 | d_Y_loss: 0.5158 | d_X_loss:
0.8446 | d_fake_loss: 1.3604 | g_loss: 4.2856
Iteration [ 4900/10000] | d_real_loss: 0.4152 | d_Y_loss: 0.7422 | d_X_loss:
0.5913 | d_fake_loss: 1.3335 | g_loss: 3.7326
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
004900-Y-X.png
Iteration [ 4910/10000] | d_real_loss: 0.6261 | d_Y_loss: 0.5685 | d_X_loss:
0.3394 | d_fake_loss: 0.9079 | g_loss: 4.1137
Iteration [ 4920/10000] | d_real_loss: 0.4194 | d_Y_loss: 0.9792 | d_X_loss:
1.1039 | d_fake_loss: 2.0831 | g_loss: 3.4260
Iteration [ 4930/10000] | d_real_loss: 0.5150 | d_Y_loss: 0.6828 | d_X_loss:
0.8118 | d_fake_loss: 1.4946 | g_loss: 3.5899
Iteration [ 4940/10000] | d_real_loss: 0.3304 | d_Y_loss: 0.9929 | d_X_loss:
0.3965 | d_fake_loss: 1.3894 | g_loss: 4.0667
Iteration [ 4950/10000] | d_real_loss: 0.5158 | d_Y_loss: 0.9054 | d_X_loss:
0.8924 | d_fake_loss: 1.7979 | g_loss: 3.1689
Iteration [ 4960/10000] | d_real_loss: 0.4517 | d_Y_loss: 0.5233 | d_X_loss:
1.0193 | d_fake_loss: 1.5426 | g_loss: 3.7490
Iteration [ 4970/10000] | d_real_loss: 0.3577 | d_Y_loss: 0.6339 | d_X_loss:
0.7582 | d_fake_loss: 1.3920 | g_loss: 3.5312
Iteration [ 4980/10000] | d_real_loss: 0.4595 | d_Y_loss: 0.5595 | d_X_loss:
0.3595 | d_fake_loss: 0.9191 | g_loss: 3.8339
Iteration [ 4990/10000] | d_real_loss: 0.3651 | d_Y_loss: 0.4810 | d_X_loss:
0.6882 | d_fake_loss: 1.1692 | g_loss: 3.9988
Iteration [ 5000/10000] | d_real_loss: 0.3646 | d_Y_loss: 0.4400 | d_X_loss:
0.3540 | d_fake_loss: 0.7939 | g_loss: 3.7097
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005000-Y-X.png
Iteration [ 5010/10000] | d_real_loss: 0.3801 | d_Y_loss: 0.8852 | d_X_loss:
0.5846 | d_fake_loss: 1.4698 | g_loss: 3.9849
Iteration [ 5020/10000] | d_real_loss: 0.4945 | d_Y_loss: 0.7604 | d_X_loss:
0.6224 | d_fake_loss: 1.3828 | g_loss: 3.5870
Iteration [ 5030/10000] | d_real_loss: 0.4518 | d_Y_loss: 0.9073 | d_X_loss:
```

```
0.7607 | d_fake_loss: 1.6680 | g_loss: 3.5716
Iteration [ 5040/10000] | d_real_loss: 0.6396 | d_Y_loss: 0.6659 | d_X_loss:
0.4487 | d_fake_loss: 1.1146 | g_loss: 3.8153
Iteration [ 5050/10000] | d_real_loss: 0.4556 | d_Y_loss: 0.4064 | d_X_loss:
0.5024 | d_fake_loss: 0.9088 | g_loss: 3.7908
Iteration [ 5060/10000] | d_real_loss: 0.5489 | d_Y_loss: 0.5597 | d_X_loss:
0.4805 | d_fake_loss: 1.0401 | g_loss: 3.7609
Iteration [ 5070/10000] | d_real_loss: 0.6079 | d_Y_loss: 0.5770 | d_X_loss:
0.7642 | d_fake_loss: 1.3412 | g_loss: 3.5516
Iteration [ 5080/10000] | d_real_loss: 0.4622 | d_Y_loss: 0.5537 | d_X_loss:
0.5648 | d_fake_loss: 1.1186 | g_loss: 3.9023
Iteration [ 5090/10000] | d_real_loss: 0.3511 | d_Y_loss: 0.7820 | d_X_loss:
0.5746 | d_fake_loss: 1.3566 | g_loss: 4.1045
Iteration [ 5100/10000] | d_real_loss: 0.3842 | d_Y_loss: 0.4129 | d_X_loss:
0.8811 | d_fake_loss: 1.2939 | g_loss: 3.5251
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005100-Y-X.png
Iteration [ 5110/10000] | d_real_loss: 0.6507 | d_Y_loss: 0.9307 | d_X_loss:
0.1699 | d_fake_loss: 1.1007 | g_loss: 4.1054
Iteration [ 5120/10000] | d_real_loss: 0.5499 | d_Y_loss: 0.6907 | d_X_loss:
0.9852 | d_fake_loss: 1.6759 | g_loss: 3.8717
Iteration [ 5130/10000] | d_real_loss: 0.4983 | d_Y_loss: 0.4245 | d_X_loss:
0.3967 | d_fake_loss: 0.8212 | g_loss: 4.7296
Iteration [ 5140/10000] | d_real_loss: 0.7264 | d_Y_loss: 0.5096 | d_X_loss:
0.2922 | d_fake_loss: 0.8018 | g_loss: 3.9462
Iteration [ 5150/10000] | d_real_loss: 0.4130 | d_Y_loss: 0.6399 | d_X_loss:
1.1268 | d_fake_loss: 1.7667 | g_loss: 4.0385
Iteration [ 5160/10000] | d_real_loss: 0.4932 | d_Y_loss: 0.8816 | d_X_loss:
0.2770 | d_fake_loss: 1.1586 | g_loss: 4.1531
Iteration [ 5170/10000] | d_real_loss: 0.3082 | d_Y_loss: 0.5109 | d_X_loss:
0.4896 | d_fake_loss: 1.0005 | g_loss: 3.4832
Iteration [ 5180/10000] | d_real_loss: 0.3825 | d_Y_loss: 0.4904 | d_X_loss:
0.2412 | d_fake_loss: 0.7316 | g_loss: 4.1602
Iteration [ 5190/10000] | d_real_loss: 0.4990 | d_Y_loss: 0.5485 | d_X_loss:
0.7827 | d_fake_loss: 1.3311 | g_loss: 3.5091
Iteration [ 5200/10000] | d_real_loss: 0.5737 | d_Y_loss: 0.4120 | d_X_loss:
0.8837 | d_fake_loss: 1.2957 | g_loss: 3.7644
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005200-Y-X.png
Iteration [ 5210/10000] | d_real_loss: 0.4339 | d_Y_loss: 0.7753 | d_X_loss:
0.5553 | d_fake_loss: 1.3306 | g_loss: 3.8136
Iteration [ 5220/10000] | d_real_loss: 0.4337 | d_Y_loss: 0.7510 | d_X_loss:
0.5885 | d_fake_loss: 1.3395 | g_loss: 3.7544
Iteration [ 5230/10000] | d_real_loss: 0.8415 | d_Y_loss: 0.6628 | d_X_loss:
```

```
0.8085 | d_fake_loss: 1.4713 | g_loss: 3.8492
Iteration [ 5240/10000] | d_real_loss: 0.4153 | d_Y_loss: 0.8276 | d_X_loss:
0.4887 | d_fake_loss: 1.3164 | g_loss: 3.4001
Iteration [ 5250/10000] | d_real_loss: 0.3973 | d_Y_loss: 0.5408 | d_X_loss:
0.4185 | d_fake_loss: 0.9594 | g_loss: 3.9638
Iteration [ 5260/10000] | d_real_loss: 0.5391 | d_Y_loss: 0.6608 | d_X_loss:
0.8579 | d_fake_loss: 1.5187 | g_loss: 4.2152
Iteration [ 5270/10000] | d_real_loss: 0.5885 | d_Y_loss: 0.5487 | d_X_loss:
0.3072 | d_fake_loss: 0.8559 | g_loss: 3.9597
Iteration [ 5280/10000] | d_real_loss: 0.4703 | d_Y_loss: 0.4157 | d_X_loss:
0.4240 | d_fake_loss: 0.8396 | g_loss: 4.0469
Iteration [ 5290/10000] | d_real_loss: 0.4603 | d_Y_loss: 0.3349 | d_X_loss:
0.7194 | d_fake_loss: 1.0543 | g_loss: 4.0717
Iteration [ 5300/10000] | d_real_loss: 0.5297 | d_Y_loss: 0.4657 | d_X_loss:
0.4258 | d_fake_loss: 0.8916 | g_loss: 3.9902
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005300-Y-X.png
Iteration [ 5310/10000] | d_real_loss: 0.3978 | d_Y_loss: 0.4913 | d_X_loss:
0.5586 | d_fake_loss: 1.0499 | g_loss: 4.4813
Iteration [ 5320/10000] | d_real_loss: 0.4134 | d_Y_loss: 0.7863 | d_X_loss:
0.4981 | d_fake_loss: 1.2844 | g_loss: 4.0560
Iteration [ 5330/10000] | d_real_loss: 0.4573 | d_Y_loss: 0.7417 | d_X_loss:
0.6497 | d_fake_loss: 1.3913 | g_loss: 3.7981
Iteration [ 5340/10000] | d_real_loss: 0.4647 | d_Y_loss: 0.6051 | d_X_loss:
0.3941 | d_fake_loss: 0.9992 | g_loss: 3.7396
Iteration [ 5350/10000] | d_real_loss: 0.5501 | d_Y_loss: 0.7028 | d_X_loss:
0.4006 | d_fake_loss: 1.1034 | g_loss: 3.8474
Iteration [ 5360/10000] | d_real_loss: 0.3693 | d_Y_loss: 0.6078 | d_X_loss:
0.4509 | d_fake_loss: 1.0586 | g_loss: 3.9707
Iteration [ 5370/10000] | d_real_loss: 0.5111 | d_Y_loss: 0.5214 | d_X_loss:
0.7892 | d_fake_loss: 1.3106 | g_loss: 3.7101
Iteration [ 5380/10000] | d_real_loss: 0.6703 | d_Y_loss: 0.8041 | d_X_loss:
0.3351 | d_fake_loss: 1.1391 | g_loss: 3.9003
Iteration [ 5390/10000] | d_real_loss: 0.3447 | d_Y_loss: 0.6552 | d_X_loss:
0.5477 | d_fake_loss: 1.2029 | g_loss: 3.5062
Iteration [ 5400/10000] | d_real_loss: 0.4714 | d_Y_loss: 0.6521 | d_X_loss:
0.8555 | d_fake_loss: 1.5076 | g_loss: 3.8587
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005400-Y-X.png
Iteration [ 5410/10000] | d_real_loss: 0.4563 | d_Y_loss: 0.3402 | d_X_loss:
0.4950 | d_fake_loss: 0.8352 | g_loss: 3.9910
Iteration [ 5420/10000] | d_real_loss: 0.3071 | d_Y_loss: 0.6009 | d_X_loss:
0.6317 | d_fake_loss: 1.2326 | g_loss: 3.8954
Iteration [ 5430/10000] | d_real_loss: 0.4382 | d_Y_loss: 0.4926 | d_X_loss:
```

```
0.4786 | d_fake_loss: 0.9712 | g_loss: 3.9800
Iteration [ 5440/10000] | d_real_loss: 0.5579 | d_Y_loss: 0.5194 | d_X_loss:
0.7804 | d_fake_loss: 1.2999 | g_loss: 4.0319
Iteration [ 5450/10000] | d_real_loss: 0.5299 | d_Y_loss: 0.5488 | d_X_loss:
0.5234 | d_fake_loss: 1.0722 | g_loss: 3.6288
Iteration [ 5460/10000] | d_real_loss: 0.4933 | d_Y_loss: 0.6690 | d_X_loss:
0.3225 | d_fake_loss: 0.9915 | g_loss: 3.7581
Iteration [ 5470/10000] | d_real_loss: 0.4401 | d_Y_loss: 0.7733 | d_X_loss:
0.6115 | d_fake_loss: 1.3849 | g_loss: 3.8006
Iteration [ 5480/10000] | d_real_loss: 0.3315 | d_Y_loss: 0.7976 | d_X_loss:
0.4660 | d_fake_loss: 1.2635 | g_loss: 4.3766
Iteration [ 5490/10000] | d_real_loss: 0.6095 | d_Y_loss: 0.2975 | d_X_loss:
0.5350 | d_fake_loss: 0.8325 | g_loss: 3.6222
Iteration [ 5500/10000] | d_real_loss: 0.6106 | d_Y_loss: 0.5501 | d_X_loss:
0.4234 | d_fake_loss: 0.9735 | g_loss: 4.0308
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005500-Y-X.png
Iteration [ 5510/10000] | d_real_loss: 0.4097 | d_Y_loss: 0.5159 | d_X_loss:
0.5792 | d_fake_loss: 1.0951 | g_loss: 3.9745
Iteration [ 5520/10000] | d_real_loss: 0.3708 | d_Y_loss: 0.6101 | d_X_loss:
0.6872 | d_fake_loss: 1.2973 | g_loss: 3.9289
Iteration [ 5530/10000] | d_real_loss: 0.5887 | d_Y_loss: 0.5555 | d_X_loss:
0.4852 | d_fake_loss: 1.0407 | g_loss: 3.5661
Iteration [ 5540/10000] | d_real_loss: 0.4124 | d_Y_loss: 0.7090 | d_X_loss:
0.4706 | d_fake_loss: 1.1796 | g_loss: 3.6926
Iteration [ 5550/10000] | d_real_loss: 0.3604 | d_Y_loss: 0.5832 | d_X_loss:
0.6711 | d_fake_loss: 1.2543 | g_loss: 3.9526
Iteration [ 5560/10000] | d_real_loss: 0.4412 | d_Y_loss: 0.5407 | d_X_loss:
0.5685 | d_fake_loss: 1.1092 | g_loss: 4.0486
Iteration [ 5570/10000] | d_real_loss: 0.5031 | d_Y_loss: 0.6044 | d_X_loss:
0.5799 | d_fake_loss: 1.1842 | g_loss: 4.1129
Iteration [ 5580/10000] | d_real_loss: 0.4083 | d_Y_loss: 0.5801 | d_X_loss:
0.5284 | d_fake_loss: 1.1085 | g_loss: 3.7713
Iteration [ 5590/10000] | d_real_loss: 0.5457 | d_Y_loss: 0.4777 | d_X_loss:
0.8576 | d_fake_loss: 1.3353 | g_loss: 3.9965
Iteration [ 5600/10000] | d_real_loss: 0.3672 | d_Y_loss: 0.6914 | d_X_loss:
0.2280 | d_fake_loss: 0.9194 | g_loss: 3.8372
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005600-Y-X.png
Iteration [ 5610/10000] | d_real_loss: 0.2831 | d_Y_loss: 0.7979 | d_X_loss:
0.2991 | d_fake_loss: 1.0969 | g_loss: 4.3059
Iteration [ 5620/10000] | d_real_loss: 0.6092 | d_Y_loss: 0.7638 | d_X_loss:
1.5655 | d_fake_loss: 2.3294 | g_loss: 3.7721
Iteration [ 5630/10000] | d_real_loss: 0.5676 | d_Y_loss: 0.9811 | d_X_loss:
```

```
0.2936 | d_fake_loss: 1.2747 | g_loss: 3.5181
Iteration [ 5640/10000] | d_real_loss: 0.6805 | d_Y_loss: 0.5095 | d_X_loss:
0.5648 | d_fake_loss: 1.0743 | g_loss: 3.4910
Iteration [ 5650/10000] | d_real_loss: 0.4449 | d_Y_loss: 0.4497 | d_X_loss:
1.1593 | d_fake_loss: 1.6090 | g_loss: 3.4883
Iteration [ 5660/10000] | d_real_loss: 0.4998 | d_Y_loss: 0.7521 | d_X_loss:
0.4321 | d_fake_loss: 1.1842 | g_loss: 4.0290
Iteration [ 5670/10000] | d_real_loss: 0.4111 | d_Y_loss: 0.4829 | d_X_loss:
0.6494 | d_fake_loss: 1.1322 | g_loss: 3.9536
Iteration [ 5680/10000] | d_real_loss: 0.4371 | d_Y_loss: 0.5586 | d_X_loss:
0.8065 | d_fake_loss: 1.3651 | g_loss: 3.8607
Iteration [ 5690/10000] | d_real_loss: 0.5018 | d_Y_loss: 0.6293 | d_X_loss:
0.4036 | d_fake_loss: 1.0329 | g_loss: 4.0012
Iteration [ 5700/10000] | d_real_loss: 0.3255 | d_Y_loss: 0.8747 | d_X_loss:
0.9320 | d_fake_loss: 1.8067 | g_loss: 3.8937
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005700-Y-X.png
Iteration [ 5710/10000] | d_real_loss: 0.4516 | d_Y_loss: 0.9482 | d_X_loss:
0.8202 | d_fake_loss: 1.7684 | g_loss: 3.6719
Iteration [ 5720/10000] | d_real_loss: 0.4307 | d_Y_loss: 0.6310 | d_X_loss:
0.6766 | d_fake_loss: 1.3076 | g_loss: 3.7492
Iteration [ 5730/10000] | d_real_loss: 0.3355 | d_Y_loss: 0.5121 | d_X_loss:
0.4326 | d_fake_loss: 0.9447 | g_loss: 3.7095
Iteration [ 5740/10000] | d_real_loss: 0.4736 | d_Y_loss: 0.7790 | d_X_loss:
0.4325 | d_fake_loss: 1.2114 | g_loss: 3.6314
Iteration [ 5750/10000] | d_real_loss: 0.3153 | d_Y_loss: 0.7039 | d_X_loss:
0.3261 | d_fake_loss: 1.0300 | g_loss: 3.6090
Iteration [ 5760/10000] | d_real_loss: 0.5103 | d_Y_loss: 0.6368 | d_X_loss:
0.3972 | d_fake_loss: 1.0341 | g_loss: 4.0075
Iteration [ 5770/10000] | d_real_loss: 0.4809 | d_Y_loss: 0.5173 | d_X_loss:
0.6564 | d_fake_loss: 1.1737 | g_loss: 3.6782
Iteration [ 5780/10000] | d_real_loss: 0.6372 | d_Y_loss: 0.6431 | d_X_loss:
0.3466 | d_fake_loss: 0.9897 | g_loss: 3.9192
Iteration [ 5790/10000] | d_real_loss: 0.4975 | d_Y_loss: 0.4759 | d_X_loss:
0.5928 | d_fake_loss: 1.0686 | g_loss: 3.9957
Iteration [ 5800/10000] | d_real_loss: 0.4234 | d_Y_loss: 0.4291 | d_X_loss:
0.7299 | d_fake_loss: 1.1590 | g_loss: 3.8824
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005800-Y-X.png
Iteration [ 5810/10000] | d_real_loss: 0.3508 | d_Y_loss: 0.6458 | d_X_loss:
0.5924 | d_fake_loss: 1.2383 | g_loss: 3.7244
Iteration [ 5820/10000] | d_real_loss: 0.4008 | d_Y_loss: 0.6240 | d_X_loss:
0.4069 | d_fake_loss: 1.0309 | g_loss: 3.9900
Iteration [ 5830/10000] | d_real_loss: 0.3887 | d_Y_loss: 0.4202 | d_X_loss:
```

```
0.6036 | d_fake_loss: 1.0238 | g_loss: 4.1854
Iteration [ 5840/10000] | d_real_loss: 0.4544 | d_Y_loss: 0.6005 | d_X_loss:
0.3948 | d_fake_loss: 0.9953 | g_loss: 3.9285
Iteration [ 5850/10000] | d_real_loss: 0.3447 | d_Y_loss: 0.4751 | d_X_loss:
0.3917 | d_fake_loss: 0.8668 | g_loss: 3.8262
Iteration [ 5860/10000] | d_real_loss: 0.4634 | d_Y_loss: 0.5938 | d_X_loss:
0.3001 | d_fake_loss: 0.8939 | g_loss: 3.8376
Iteration [ 5870/10000] | d_real_loss: 0.5694 | d_Y_loss: 0.3505 | d_X_loss:
0.4400 | d_fake_loss: 0.7904 | g_loss: 4.0197
Iteration [ 5880/10000] | d_real_loss: 0.3931 | d_Y_loss: 0.5666 | d_X_loss:
0.4244 | d_fake_loss: 0.9910 | g_loss: 3.7350
Iteration [ 5890/10000] | d_real_loss: 0.3796 | d_Y_loss: 0.7197 | d_X_loss:
0.4683 | d_fake_loss: 1.1881 | g_loss: 3.6081
Iteration [ 5900/10000] | d_real_loss: 0.5377 | d_Y_loss: 0.8817 | d_X_loss:
0.3445 | d_fake_loss: 1.2262 | g_loss: 3.7587
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
005900-Y-X.png
Iteration [ 5910/10000] | d_real_loss: 0.4623 | d_Y_loss: 0.5752 | d_X_loss:
0.6088 | d_fake_loss: 1.1841 | g_loss: 3.8406
Iteration [ 5920/10000] | d_real_loss: 0.8412 | d_Y_loss: 0.5955 | d_X_loss:
0.2259 | d_fake_loss: 0.8214 | g_loss: 4.2044
Iteration [ 5930/10000] | d_real_loss: 0.5196 | d_Y_loss: 0.6803 | d_X_loss:
0.5810 | d_fake_loss: 1.2614 | g_loss: 4.1105
Iteration [ 5940/10000] | d_real_loss: 0.4610 | d_Y_loss: 0.8241 | d_X_loss:
0.6077 | d_fake_loss: 1.4318 | g_loss: 3.6987
Iteration [ 5950/10000] | d_real_loss: 0.5096 | d_Y_loss: 0.4590 | d_X_loss:
0.4144 | d_fake_loss: 0.8734 | g_loss: 3.8657
Iteration [ 5960/10000] | d_real_loss: 0.4991 | d_Y_loss: 0.5134 | d_X_loss:
0.5492 | d_fake_loss: 1.0626 | g_loss: 3.7949
Iteration [ 5970/10000] | d_real_loss: 0.5267 | d_Y_loss: 0.7051 | d_X_loss:
0.6327 | d_fake_loss: 1.3378 | g_loss: 3.2905
Iteration [ 5980/10000] | d_real_loss: 0.7266 | d_Y_loss: 0.5307 | d_X_loss:
0.3888 | d_fake_loss: 0.9196 | g_loss: 3.7371
Iteration [ 5990/10000] | d_real_loss: 0.6250 | d_Y_loss: 0.7992 | d_X_loss:
0.4366 | d_fake_loss: 1.2358 | g_loss: 4.2283
Iteration [ 6000/10000] | d_real_loss: 0.4538 | d_Y_loss: 0.6433 | d_X_loss:
0.3972 | d_fake_loss: 1.0404 | g_loss: 3.8844
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006000-Y-X.png
Iteration [ 6010/10000] | d_real_loss: 0.4153 | d_Y_loss: 0.6171 | d_X_loss:
0.6397 | d_fake_loss: 1.2568 | g_loss: 3.5186
Iteration [ 6020/10000] | d_real_loss: 0.6396 | d_Y_loss: 0.5410 | d_X_loss:
0.5126 | d_fake_loss: 1.0537 | g_loss: 3.2666
Iteration [ 6030/10000] | d_real_loss: 0.5599 | d_Y_loss: 0.7641 | d_X_loss:
```

```
0.6432 | d_fake_loss: 1.4073 | g_loss: 3.5505
Iteration [ 6040/10000] | d_real_loss: 0.5298 | d_Y_loss: 0.7764 | d_X_loss:
0.4401 | d_fake_loss: 1.2165 | g_loss: 4.0248
Iteration [ 6050/10000] | d_real_loss: 0.5795 | d_Y_loss: 0.5001 | d_X_loss:
0.8263 | d_fake_loss: 1.3264 | g_loss: 4.2501
Iteration [ 6060/10000] | d_real_loss: 0.2975 | d_Y_loss: 0.7012 | d_X_loss:
0.8902 | d_fake_loss: 1.5914 | g_loss: 3.9915
Iteration [ 6070/10000] | d_real_loss: 0.4525 | d_Y_loss: 0.7734 | d_X_loss:
0.6568 | d_fake_loss: 1.4302 | g_loss: 3.7663
Iteration [ 6080/10000] | d_real_loss: 0.4291 | d_Y_loss: 0.4842 | d_X_loss:
0.5333 | d_fake_loss: 1.0176 | g_loss: 3.7758
Iteration [ 6090/10000] | d_real_loss: 0.4320 | d_Y_loss: 0.7258 | d_X_loss:
0.4569 | d_fake_loss: 1.1827 | g_loss: 3.6482
Iteration [ 6100/10000] | d_real_loss: 0.6055 | d_Y_loss: 0.5505 | d_X_loss:
0.6204 | d_fake_loss: 1.1709 | g_loss: 3.8079
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006100-Y-X.png
Iteration [ 6110/10000] | d_real_loss: 0.5655 | d_Y_loss: 0.6799 | d_X_loss:
0.3202 | d_fake_loss: 1.0001 | g_loss: 3.6640
Iteration [ 6120/10000] | d_real_loss: 0.4621 | d_Y_loss: 0.5799 | d_X_loss:
0.5448 | d_fake_loss: 1.1247 | g_loss: 3.7922
Iteration [ 6130/10000] | d_real_loss: 0.5599 | d_Y_loss: 0.7630 | d_X_loss:
0.4522 | d_fake_loss: 1.2152 | g_loss: 3.5429
Iteration [ 6140/10000] | d_real_loss: 0.5521 | d_Y_loss: 0.5611 | d_X_loss:
0.4098 | d_fake_loss: 0.9709 | g_loss: 3.2381
Iteration [ 6150/10000] | d_real_loss: 0.6194 | d_Y_loss: 0.8256 | d_X_loss:
0.5170 | d_fake_loss: 1.3426 | g_loss: 3.5450
Iteration [ 6160/10000] | d_real_loss: 0.5744 | d_Y_loss: 0.6231 | d_X_loss:
0.4530 | d_fake_loss: 1.0761 | g_loss: 3.7920
Iteration [ 6170/10000] | d_real_loss: 0.6800 | d_Y_loss: 0.8792 | d_X_loss:
0.5385 | d_fake_loss: 1.4177 | g_loss: 3.8300
Iteration [ 6180/10000] | d_real_loss: 0.5462 | d_Y_loss: 0.6773 | d_X_loss:
0.6354 | d_fake_loss: 1.3126 | g_loss: 3.5311
Iteration [ 6190/10000] | d_real_loss: 0.3489 | d_Y_loss: 0.7396 | d_X_loss:
0.6261 | d_fake_loss: 1.3657 | g_loss: 3.3979
Iteration [ 6200/10000] | d_real_loss: 0.5414 | d_Y_loss: 0.6855 | d_X_loss:
0.9233 | d_fake_loss: 1.6089 | g_loss: 3.7713
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006200-Y-X.png
Iteration [ 6210/10000] | d_real_loss: 0.6681 | d_Y_loss: 0.8566 | d_X_loss:
0.4283 | d_fake_loss: 1.2849 | g_loss: 4.3082
Iteration [ 6220/10000] | d_real_loss: 0.3236 | d_Y_loss: 0.6121 | d_X_loss:
0.7818 | d_fake_loss: 1.3939 | g_loss: 3.4659
Iteration [ 6230/10000] | d_real_loss: 0.4693 | d_Y_loss: 0.6945 | d_X_loss:
```

```
0.4858 | d_fake_loss: 1.1803 | g_loss: 3.5355
Iteration [ 6240/10000] | d_real_loss: 0.4289 | d_Y_loss: 0.5501 | d_X_loss:
0.5639 | d_fake_loss: 1.1140 | g_loss: 4.2130
Iteration [ 6250/10000] | d_real_loss: 0.6143 | d_Y_loss: 0.7973 | d_X_loss:
0.2217 | d_fake_loss: 1.0191 | g_loss: 3.3929
Iteration [ 6260/10000] | d_real_loss: 0.6029 | d_Y_loss: 0.5778 | d_X_loss:
0.2425 | d_fake_loss: 0.8203 | g_loss: 3.9686
Iteration [ 6270/10000] | d_real_loss: 0.6251 | d_Y_loss: 0.4710 | d_X_loss:
0.7022 | d_fake_loss: 1.1732 | g_loss: 3.9749
Iteration [ 6280/10000] | d_real_loss: 0.3796 | d_Y_loss: 0.6206 | d_X_loss:
0.4139 | d_fake_loss: 1.0345 | g_loss: 3.6771
Iteration [ 6290/10000] | d_real_loss: 0.6604 | d_Y_loss: 0.8098 | d_X_loss:
0.4163 | d_fake_loss: 1.2261 | g_loss: 3.5482
Iteration [ 6300/10000] | d_real_loss: 0.3758 | d_Y_loss: 0.6621 | d_X_loss:
0.9797 | d_fake_loss: 1.6418 | g_loss: 4.0304
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006300-Y-X.png
Iteration [ 6310/10000] | d_real_loss: 0.4025 | d_Y_loss: 0.3539 | d_X_loss:
0.7839 | d_fake_loss: 1.1378 | g_loss: 3.6766
Iteration [ 6320/10000] | d_real_loss: 0.4402 | d_Y_loss: 0.4644 | d_X_loss:
0.4244 | d_fake_loss: 0.8888 | g_loss: 3.7282
Iteration [ 6330/10000] | d_real_loss: 0.4593 | d_Y_loss: 0.7730 | d_X_loss:
0.8004 | d_fake_loss: 1.5733 | g_loss: 3.7673
Iteration [ 6340/10000] | d_real_loss: 0.6638 | d_Y_loss: 0.6231 | d_X_loss:
0.8290 | d_fake_loss: 1.4520 | g_loss: 3.9842
Iteration [ 6350/10000] | d_real_loss: 0.6152 | d_Y_loss: 0.4091 | d_X_loss:
0.3862 | d_fake_loss: 0.7953 | g_loss: 3.5310
Iteration [ 6360/10000] | d_real_loss: 0.5200 | d_Y_loss: 0.6446 | d_X_loss:
0.5388 | d_fake_loss: 1.1835 | g_loss: 3.9675
Iteration [ 6370/10000] | d_real_loss: 0.4061 | d_Y_loss: 0.7119 | d_X_loss:
0.5645 | d_fake_loss: 1.2764 | g_loss: 3.8340
Iteration [ 6380/10000] | d_real_loss: 0.5957 | d_Y_loss: 0.5659 | d_X_loss:
0.4823 | d_fake_loss: 1.0482 | g_loss: 3.6269
Iteration [ 6390/10000] | d_real_loss: 0.4392 | d_Y_loss: 0.6743 | d_X_loss:
0.7316 | d_fake_loss: 1.4059 | g_loss: 4.0450
Iteration [ 6400/10000] | d_real_loss: 0.7631 | d_Y_loss: 0.4162 | d_X_loss:
0.4003 | d_fake_loss: 0.8164 | g_loss: 3.8653
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006400-Y-X.png
Iteration [ 6410/10000] | d_real_loss: 0.4847 | d_Y_loss: 0.6397 | d_X_loss:
0.7304 | d_fake_loss: 1.3701 | g_loss: 3.4728
Iteration [ 6420/10000] | d_real_loss: 0.5784 | d_Y_loss: 0.6278 | d_X_loss:
0.4250 | d_fake_loss: 1.0528 | g_loss: 3.5300
Iteration [ 6430/10000] | d_real_loss: 0.3623 | d_Y_loss: 0.5814 | d_X_loss:
```

```
0.4703 | d_fake_loss: 1.0517 | g_loss: 3.5057
Iteration [ 6440/10000] | d_real_loss: 0.6068 | d_Y_loss: 0.7423 | d_X_loss:
0.9681 | d_fake_loss: 1.7103 | g_loss: 3.5512
Iteration [ 6450/10000] | d_real_loss: 0.3830 | d_Y_loss: 0.4097 | d_X_loss:
0.4798 | d_fake_loss: 0.8895 | g_loss: 3.7909
Iteration [ 6460/10000] | d_real_loss: 0.6276 | d_Y_loss: 0.7342 | d_X_loss:
0.5467 | d_fake_loss: 1.2808 | g_loss: 3.7366
Iteration [ 6470/10000] | d_real_loss: 0.7771 | d_Y_loss: 0.6288 | d_X_loss:
0.4475 | d_fake_loss: 1.0762 | g_loss: 3.6434
Iteration [ 6480/10000] | d_real_loss: 0.4533 | d_Y_loss: 0.4841 | d_X_loss:
0.7109 | d_fake_loss: 1.1950 | g_loss: 3.9615
Iteration [ 6490/10000] | d_real_loss: 0.5853 | d_Y_loss: 0.6949 | d_X_loss:
0.9872 | d_fake_loss: 1.6821 | g_loss: 4.1025
Iteration [ 6500/10000] | d_real_loss: 0.4091 | d_Y_loss: 0.3773 | d_X_loss:
0.9300 | d_fake_loss: 1.3073 | g_loss: 3.9092
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006500-Y-X.png
Iteration [ 6510/10000] | d_real_loss: 0.4588 | d_Y_loss: 0.7126 | d_X_loss:
0.5692 | d_fake_loss: 1.2818 | g_loss: 3.5999
Iteration [ 6520/10000] | d_real_loss: 0.6539 | d_Y_loss: 0.7029 | d_X_loss:
0.3759 | d_fake_loss: 1.0789 | g_loss: 3.7880
Iteration [ 6530/10000] | d_real_loss: 0.3367 | d_Y_loss: 0.4339 | d_X_loss:
0.4914 | d_fake_loss: 0.9253 | g_loss: 3.8886
Iteration [ 6540/10000] | d_real_loss: 0.4562 | d_Y_loss: 0.7026 | d_X_loss:
0.8482 | d_fake_loss: 1.5508 | g_loss: 3.5112
Iteration [ 6550/10000] | d_real_loss: 0.3120 | d_Y_loss: 0.8483 | d_X_loss:
0.7486 | d_fake_loss: 1.5968 | g_loss: 3.5701
Iteration [ 6560/10000] | d_real_loss: 0.6220 | d_Y_loss: 0.5286 | d_X_loss:
0.5188 | d_fake_loss: 1.0474 | g_loss: 3.8937
Iteration [ 6570/10000] | d_real_loss: 0.2626 | d_Y_loss: 0.5144 | d_X_loss:
0.2766 | d_fake_loss: 0.7910 | g_loss: 3.9196
Iteration [ 6580/10000] | d_real_loss: 0.5241 | d_Y_loss: 0.6953 | d_X_loss:
0.8422 | d_fake_loss: 1.5375 | g_loss: 3.8221
Iteration [ 6590/10000] | d_real_loss: 0.4071 | d_Y_loss: 0.4153 | d_X_loss:
1.1673 | d_fake_loss: 1.5826 | g_loss: 3.8368
Iteration [ 6600/10000] | d_real_loss: 0.3814 | d_Y_loss: 0.7889 | d_X_loss:
0.6984 | d_fake_loss: 1.4874 | g_loss: 4.2717
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006600-Y-X.png
Iteration [ 6610/10000] | d_real_loss: 0.7226 | d_Y_loss: 0.4369 | d_X_loss:
0.6036 | d_fake_loss: 1.0405 | g_loss: 4.0366
Iteration [ 6620/10000] | d_real_loss: 0.3597 | d_Y_loss: 0.8663 | d_X_loss:
0.4642 | d_fake_loss: 1.3305 | g_loss: 3.9056
Iteration [ 6630/10000] | d_real_loss: 0.4665 | d_Y_loss: 0.6201 | d_X_loss:
```

```
0.8072 | d_fake_loss: 1.4273 | g_loss: 3.8933
Iteration [ 6640/10000] | d_real_loss: 0.5370 | d_Y_loss: 0.8174 | d_X_loss:
0.7953 | d_fake_loss: 1.6128 | g_loss: 3.5084
Iteration [ 6650/10000] | d_real_loss: 0.5108 | d_Y_loss: 0.5929 | d_X_loss:
1.0151 | d_fake_loss: 1.6080 | g_loss: 3.5959
Iteration [ 6660/10000] | d_real_loss: 0.3655 | d_Y_loss: 0.5360 | d_X_loss:
0.2168 | d_fake_loss: 0.7528 | g_loss: 3.9440
Iteration [ 6670/10000] | d_real_loss: 0.5045 | d_Y_loss: 1.1286 | d_X_loss:
0.5247 | d_fake_loss: 1.6533 | g_loss: 3.6593
Iteration [ 6680/10000] | d_real_loss: 0.5300 | d_Y_loss: 0.5605 | d_X_loss:
0.4456 | d_fake_loss: 1.0061 | g_loss: 4.0265
Iteration [ 6690/10000] | d_real_loss: 0.4410 | d_Y_loss: 0.4394 | d_X_loss:
0.3762 | d_fake_loss: 0.8156 | g_loss: 3.8178
Iteration [ 6700/10000] | d_real_loss: 0.6191 | d_Y_loss: 0.8715 | d_X_loss:
0.5582 | d_fake_loss: 1.4298 | g_loss: 3.6551
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006700-Y-X.png
Iteration [ 6710/10000] | d_real_loss: 0.4244 | d_Y_loss: 1.1025 | d_X_loss:
0.2675 | d_fake_loss: 1.3700 | g_loss: 3.4791
Iteration [ 6720/10000] | d_real_loss: 0.5392 | d_Y_loss: 0.5951 | d_X_loss:
0.7199 | d_fake_loss: 1.3150 | g_loss: 3.6449
Iteration [ 6730/10000] | d_real_loss: 0.4846 | d_Y_loss: 1.0378 | d_X_loss:
0.1937 | d_fake_loss: 1.2315 | g_loss: 3.6769
Iteration [ 6740/10000] | d_real_loss: 0.5259 | d_Y_loss: 0.6178 | d_X_loss:
0.3817 | d_fake_loss: 0.9995 | g_loss: 3.8933
Iteration [ 6750/10000] | d_real_loss: 0.4656 | d_Y_loss: 0.8761 | d_X_loss:
0.7627 | d_fake_loss: 1.6389 | g_loss: 3.8410
Iteration [ 6760/10000] | d_real_loss: 0.6946 | d_Y_loss: 0.7685 | d_X_loss:
0.7741 | d_fake_loss: 1.5426 | g_loss: 3.6320
Iteration [ 6770/10000] | d_real_loss: 0.4060 | d_Y_loss: 0.6161 | d_X_loss:
0.9119 | d_fake_loss: 1.5279 | g_loss: 3.5598
Iteration [ 6780/10000] | d_real_loss: 0.5328 | d_Y_loss: 0.8064 | d_X_loss:
0.3908 | d_fake_loss: 1.1972 | g_loss: 3.7768
Iteration [ 6790/10000] | d_real_loss: 0.4616 | d_Y_loss: 0.6442 | d_X_loss:
0.6508 | d_fake_loss: 1.2951 | g_loss: 4.1022
Iteration [ 6800/10000] | d_real_loss: 0.3676 | d_Y_loss: 0.4803 | d_X_loss:
0.2839 | d_fake_loss: 0.7642 | g_loss: 3.4716
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006800-Y-X.png
Iteration [ 6810/10000] | d_real_loss: 0.4250 | d_Y_loss: 0.5054 | d_X_loss:
0.8539 | d_fake_loss: 1.3593 | g_loss: 3.9081
Iteration [ 6820/10000] | d_real_loss: 0.5947 | d_Y_loss: 0.8002 | d_X_loss:
0.8153 | d_fake_loss: 1.6154 | g_loss: 3.5672
Iteration [ 6830/10000] | d_real_loss: 0.5792 | d_Y_loss: 0.6989 | d_X_loss:
```

```
0.6783 | d_fake_loss: 1.3772 | g_loss: 4.2330
Iteration [ 6840/10000] | d_real_loss: 0.5265 | d_Y_loss: 0.6216 | d_X_loss:
0.5215 | d_fake_loss: 1.1431 | g_loss: 3.6899
Iteration [ 6850/10000] | d_real_loss: 0.4997 | d_Y_loss: 0.6827 | d_X_loss:
0.8744 | d_fake_loss: 1.5571 | g_loss: 4.3693
Iteration [ 6860/10000] | d_real_loss: 0.6445 | d_Y_loss: 0.8832 | d_X_loss:
0.6060 | d_fake_loss: 1.4892 | g_loss: 3.9443
Iteration [ 6870/10000] | d_real_loss: 0.3768 | d_Y_loss: 0.4813 | d_X_loss:
0.5764 | d_fake_loss: 1.0577 | g_loss: 3.7428
Iteration [ 6880/10000] | d_real_loss: 0.5363 | d_Y_loss: 0.7980 | d_X_loss:
0.3588 | d_fake_loss: 1.1568 | g_loss: 3.5307
Iteration [ 6890/10000] | d_real_loss: 0.4879 | d_Y_loss: 0.7301 | d_X_loss:
0.4122 | d_fake_loss: 1.1423 | g_loss: 3.6250
Iteration [ 6900/10000] | d_real_loss: 0.4114 | d_Y_loss: 0.9467 | d_X_loss:
0.3677 | d_fake_loss: 1.3144 | g_loss: 3.7254
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
006900-Y-X.png
Iteration [ 6910/10000] | d_real_loss: 0.4162 | d_Y_loss: 0.5051 | d_X_loss:
0.8664 | d_fake_loss: 1.3716 | g_loss: 3.3948
Iteration [ 6920/10000] | d_real_loss: 0.4792 | d_Y_loss: 0.4988 | d_X_loss:
0.4572 | d_fake_loss: 0.9560 | g_loss: 4.0347
Iteration [ 6930/10000] | d_real_loss: 0.4303 | d_Y_loss: 0.7767 | d_X_loss:
0.4490 | d_fake_loss: 1.2257 | g_loss: 4.0624
Iteration [ 6940/10000] | d_real_loss: 0.2607 | d_Y_loss: 0.8318 | d_X_loss:
0.8520 | d_fake_loss: 1.6837 | g_loss: 3.8528
Iteration [ 6950/10000] | d_real_loss: 0.5263 | d_Y_loss: 1.4714 | d_X_loss:
0.6388 | d_fake_loss: 2.1101 | g_loss: 3.7449
Iteration [ 6960/10000] | d_real_loss: 0.6379 | d_Y_loss: 0.7154 | d_X_loss:
0.4346 | d_fake_loss: 1.1500 | g_loss: 3.4824
Iteration [ 6970/10000] | d_real_loss: 0.4895 | d_Y_loss: 0.9382 | d_X_loss:
0.6572 | d_fake_loss: 1.5954 | g_loss: 3.5065
Iteration [ 6980/10000] | d_real_loss: 0.5382 | d_Y_loss: 0.5513 | d_X_loss:
0.6739 | d_fake_loss: 1.2252 | g_loss: 3.6419
Iteration [ 6990/10000] | d_real_loss: 0.4921 | d_Y_loss: 0.4826 | d_X_loss:
0.6471 | d_fake_loss: 1.1297 | g_loss: 4.0506
Iteration [ 7000/10000] | d_real_loss: 0.4450 | d_Y_loss: 0.5976 | d_X_loss:
0.4979 | d_fake_loss: 1.0954 | g_loss: 3.5635
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007000-Y-X.png
Iteration [ 7010/10000] | d_real_loss: 0.5277 | d_Y_loss: 0.5951 | d_X_loss:
0.7977 | d_fake_loss: 1.3927 | g_loss: 3.7158
Iteration [ 7020/10000] | d_real_loss: 0.6430 | d_Y_loss: 0.5918 | d_X_loss:
0.5094 | d_fake_loss: 1.1012 | g_loss: 3.6158
Iteration [ 7030/10000] | d_real_loss: 0.4417 | d_Y_loss: 0.4960 | d_X_loss:
```

```
0.9913 | d_fake_loss: 1.4873 | g_loss: 3.8458
Iteration [ 7040/10000] | d_real_loss: 0.4375 | d_Y_loss: 0.7001 | d_X_loss:
0.6134 | d_fake_loss: 1.3135 | g_loss: 3.6073
Iteration [ 7050/10000] | d_real_loss: 0.5334 | d_Y_loss: 0.5852 | d_X_loss:
0.4440 | d_fake_loss: 1.0292 | g_loss: 3.6169
Iteration [ 7060/10000] | d_real_loss: 0.4427 | d_Y_loss: 0.5147 | d_X_loss:
0.5326 | d_fake_loss: 1.0473 | g_loss: 3.5684
Iteration [ 7070/10000] | d_real_loss: 0.6552 | d_Y_loss: 0.9220 | d_X_loss:
0.4084 | d_fake_loss: 1.3305 | g_loss: 3.8424
Iteration [ 7080/10000] | d_real_loss: 0.3935 | d_Y_loss: 0.8383 | d_X_loss:
0.4466 | d_fake_loss: 1.2848 | g_loss: 3.4387
Iteration [ 7090/10000] | d_real_loss: 0.5383 | d_Y_loss: 0.4511 | d_X_loss:
0.7063 | d_fake_loss: 1.1574 | g_loss: 3.7242
Iteration [ 7100/10000] | d_real_loss: 0.5195 | d_Y_loss: 0.5236 | d_X_loss:
0.5037 | d_fake_loss: 1.0273 | g_loss: 3.4264
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007100-Y-X.png
Iteration [ 7110/10000] | d_real_loss: 0.4384 | d_Y_loss: 0.5258 | d_X_loss:
0.7720 | d_fake_loss: 1.2978 | g_loss: 3.5264
Iteration [ 7120/10000] | d_real_loss: 0.4409 | d_Y_loss: 0.8759 | d_X_loss:
0.3997 | d_fake_loss: 1.2756 | g_loss: 3.3830
Iteration [ 7130/10000] | d_real_loss: 0.5159 | d_Y_loss: 0.7705 | d_X_loss:
0.5636 | d_fake_loss: 1.3341 | g_loss: 3.4747
Iteration [ 7140/10000] | d_real_loss: 0.4569 | d_Y_loss: 0.7382 | d_X_loss:
0.6497 | d_fake_loss: 1.3879 | g_loss: 4.1322
Iteration [ 7150/10000] | d_real_loss: 0.5635 | d_Y_loss: 0.4514 | d_X_loss:
0.2402 | d_fake_loss: 0.6917 | g_loss: 3.6285
Iteration [ 7160/10000] | d_real_loss: 0.5706 | d_Y_loss: 0.6335 | d_X_loss:
0.4499 | d_fake_loss: 1.0833 | g_loss: 3.4006
Iteration [ 7170/10000] | d_real_loss: 0.4068 | d_Y_loss: 0.9503 | d_X_loss:
0.6088 | d_fake_loss: 1.5591 | g_loss: 3.3702
Iteration [ 7180/10000] | d_real_loss: 0.6045 | d_Y_loss: 0.7277 | d_X_loss:
1.1492 | d_fake_loss: 1.8769 | g_loss: 3.3588
Iteration [ 7190/10000] | d_real_loss: 0.6786 | d_Y_loss: 0.6791 | d_X_loss:
0.4118 | d_fake_loss: 1.0908 | g_loss: 3.5973
Iteration [ 7200/10000] | d_real_loss: 0.5503 | d_Y_loss: 0.7035 | d_X_loss:
0.7593 | d_fake_loss: 1.4627 | g_loss: 3.9028
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007200-Y-X.png
Iteration [ 7210/10000] | d_real_loss: 0.5300 | d_Y_loss: 0.6190 | d_X_loss:
0.5220 | d_fake_loss: 1.1410 | g_loss: 3.6053
Iteration [ 7220/10000] | d_real_loss: 0.6414 | d_Y_loss: 0.8073 | d_X_loss:
0.6491 | d_fake_loss: 1.4564 | g_loss: 3.6916
Iteration [ 7230/10000] | d_real_loss: 0.5022 | d_Y_loss: 0.7401 | d_X_loss:
```

```
0.5820 | d_fake_loss: 1.3221 | g_loss: 3.8860
Iteration [ 7240/10000] | d_real_loss: 0.3919 | d_Y_loss: 0.5190 | d_X_loss:
0.9193 | d_fake_loss: 1.4383 | g_loss: 3.8039
Iteration [ 7250/10000] | d_real_loss: 0.5024 | d_Y_loss: 0.7322 | d_X_loss:
0.5032 | d_fake_loss: 1.2354 | g_loss: 3.3500
Iteration [ 7260/10000] | d_real_loss: 0.3153 | d_Y_loss: 0.5456 | d_X_loss:
0.7921 | d_fake_loss: 1.3376 | g_loss: 3.5454
Iteration [ 7270/10000] | d_real_loss: 0.4387 | d_Y_loss: 0.6415 | d_X_loss:
0.6731 | d_fake_loss: 1.3146 | g_loss: 3.6791
Iteration [ 7280/10000] | d_real_loss: 0.5751 | d_Y_loss: 0.5787 | d_X_loss:
0.4963 | d_fake_loss: 1.0751 | g_loss: 3.7080
Iteration [ 7290/10000] | d_real_loss: 0.4227 | d_Y_loss: 0.5646 | d_X_loss:
0.7452 | d_fake_loss: 1.3098 | g_loss: 4.0020
Iteration [ 7300/10000] | d_real_loss: 0.3869 | d_Y_loss: 0.6015 | d_X_loss:
0.5318 | d_fake_loss: 1.1333 | g_loss: 4.2946
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007300-Y-X.png
Iteration [ 7310/10000] | d_real_loss: 0.6615 | d_Y_loss: 0.5636 | d_X_loss:
0.4416 | d_fake_loss: 1.0052 | g_loss: 3.8728
Iteration [ 7320/10000] | d_real_loss: 0.4286 | d_Y_loss: 0.6858 | d_X_loss:
0.5165 | d_fake_loss: 1.2023 | g_loss: 3.9766
Iteration [ 7330/10000] | d_real_loss: 0.6237 | d_Y_loss: 0.5923 | d_X_loss:
0.6259 | d_fake_loss: 1.2182 | g_loss: 3.5185
Iteration [ 7340/10000] | d_real_loss: 0.3432 | d_Y_loss: 0.4759 | d_X_loss:
0.4741 | d_fake_loss: 0.9500 | g_loss: 3.6221
Iteration [ 7350/10000] | d_real_loss: 0.4350 | d_Y_loss: 0.8236 | d_X_loss:
0.5841 | d_fake_loss: 1.4077 | g_loss: 3.8140
Iteration [ 7360/10000] | d_real_loss: 0.4229 | d_Y_loss: 0.7169 | d_X_loss:
0.5907 | d_fake_loss: 1.3076 | g_loss: 3.6697
Iteration [ 7370/10000] | d_real_loss: 0.3884 | d_Y_loss: 0.5327 | d_X_loss:
0.5286 | d_fake_loss: 1.0614 | g_loss: 3.6864
Iteration [ 7380/10000] | d_real_loss: 0.5063 | d_Y_loss: 0.4219 | d_X_loss:
0.6088 | d_fake_loss: 1.0307 | g_loss: 3.8629
Iteration [ 7390/10000] | d_real_loss: 0.3519 | d_Y_loss: 0.7135 | d_X_loss:
0.5015 | d_fake_loss: 1.2149 | g_loss: 3.7616
Iteration [ 7400/10000] | d_real_loss: 0.7277 | d_Y_loss: 0.8104 | d_X_loss:
0.3892 | d_fake_loss: 1.1996 | g_loss: 4.2136
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007400-Y-X.png
Iteration [ 7410/10000] | d_real_loss: 0.5210 | d_Y_loss: 0.5917 | d_X_loss:
0.5360 | d_fake_loss: 1.1277 | g_loss: 3.6194
Iteration [ 7420/10000] | d_real_loss: 0.5634 | d_Y_loss: 0.5690 | d_X_loss:
0.5636 | d_fake_loss: 1.1326 | g_loss: 3.7306
Iteration [ 7430/10000] | d_real_loss: 0.4511 | d_Y_loss: 0.5796 | d_X_loss:
```

```
1.0456 | d_fake_loss: 1.6252 | g_loss: 3.2674
Iteration [ 7440/10000] | d_real_loss: 0.6700 | d_Y_loss: 0.6667 | d_X_loss:
0.5588 | d_fake_loss: 1.2254 | g_loss: 3.8029
Iteration [ 7450/10000] | d_real_loss: 0.5262 | d_Y_loss: 0.6973 | d_X_loss:
0.5643 | d_fake_loss: 1.2617 | g_loss: 3.5300
Iteration [ 7460/10000] | d_real_loss: 0.4292 | d_Y_loss: 0.5035 | d_X_loss:
0.9157 | d_fake_loss: 1.4192 | g_loss: 4.0116
Iteration [ 7470/10000] | d_real_loss: 0.5990 | d_Y_loss: 0.7496 | d_X_loss:
0.2976 | d_fake_loss: 1.0473 | g_loss: 3.7000
Iteration [ 7480/10000] | d_real_loss: 0.4616 | d_Y_loss: 0.7201 | d_X_loss:
0.7520 | d_fake_loss: 1.4721 | g_loss: 3.7738
Iteration [ 7490/10000] | d_real_loss: 0.4097 | d_Y_loss: 0.4950 | d_X_loss:
0.5874 | d_fake_loss: 1.0824 | g_loss: 4.2860
Iteration [ 7500/10000] | d_real_loss: 0.4228 | d_Y_loss: 0.6841 | d_X_loss:
0.5067 | d_fake_loss: 1.1908 | g_loss: 3.7479
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007500-Y-X.png
Iteration [ 7510/10000] | d_real_loss: 0.6299 | d_Y_loss: 0.7637 | d_X_loss:
0.4034 | d_fake_loss: 1.1671 | g_loss: 3.7911
Iteration [ 7520/10000] | d_real_loss: 0.4422 | d_Y_loss: 0.5518 | d_X_loss:
0.4758 | d_fake_loss: 1.0275 | g_loss: 3.7234
Iteration [ 7530/10000] | d_real_loss: 0.5753 | d_Y_loss: 0.4795 | d_X_loss:
0.6867 | d_fake_loss: 1.1662 | g_loss: 3.8471
Iteration [ 7540/10000] | d_real_loss: 0.4290 | d_Y_loss: 0.7074 | d_X_loss:
0.2510 | d_fake_loss: 0.9584 | g_loss: 3.9675
Iteration [ 7550/10000] | d_real_loss: 0.4769 | d_Y_loss: 0.6392 | d_X_loss:
0.7072 | d_fake_loss: 1.3464 | g_loss: 3.5301
Iteration [ 7560/10000] | d_real_loss: 0.5156 | d_Y_loss: 0.8617 | d_X_loss:
0.6536 | d_fake_loss: 1.5153 | g_loss: 3.3861
Iteration [ 7570/10000] | d_real_loss: 0.4350 | d_Y_loss: 0.3814 | d_X_loss:
0.5982 | d_fake_loss: 0.9796 | g_loss: 4.1135
Iteration [ 7580/10000] | d_real_loss: 0.4766 | d_Y_loss: 0.7083 | d_X_loss:
0.1962 | d_fake_loss: 0.9045 | g_loss: 3.7850
Iteration [ 7590/10000] | d_real_loss: 0.5256 | d_Y_loss: 0.5993 | d_X_loss:
0.6583 | d_fake_loss: 1.2576 | g_loss: 3.8036
Iteration [ 7600/10000] | d_real_loss: 0.5792 | d_Y_loss: 0.7040 | d_X_loss:
0.6278 | d_fake_loss: 1.3318 | g_loss: 3.7697
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007600-Y-X.png
Iteration [ 7610/10000] | d_real_loss: 0.6204 | d_Y_loss: 0.8675 | d_X_loss:
0.6430 | d_fake_loss: 1.5105 | g_loss: 3.7960
Iteration [ 7620/10000] | d_real_loss: 0.4291 | d_Y_loss: 0.7192 | d_X_loss:
0.7366 | d_fake_loss: 1.4558 | g_loss: 3.7889
Iteration [ 7630/10000] | d_real_loss: 0.5678 | d_Y_loss: 0.5595 | d_X_loss:
```

```
0.5617 | d_fake_loss: 1.1211 | g_loss: 3.7663
Iteration [ 7640/10000] | d_real_loss: 0.6459 | d_Y_loss: 0.9155 | d_X_loss:
0.6987 | d_fake_loss: 1.6142 | g_loss: 4.0081
Iteration [ 7650/10000] | d_real_loss: 0.5649 | d_Y_loss: 0.8148 | d_X_loss:
0.6085 | d_fake_loss: 1.4233 | g_loss: 3.2721
Iteration [ 7660/10000] | d_real_loss: 0.4750 | d_Y_loss: 0.5736 | d_X_loss:
1.1867 | d_fake_loss: 1.7603 | g_loss: 3.6773
Iteration [ 7670/10000] | d_real_loss: 0.3921 | d_Y_loss: 0.6228 | d_X_loss:
0.4535 | d_fake_loss: 1.0763 | g_loss: 3.7782
Iteration [ 7680/10000] | d_real_loss: 0.5021 | d_Y_loss: 0.6930 | d_X_loss:
0.5840 | d_fake_loss: 1.2770 | g_loss: 3.7164
Iteration [ 7690/10000] | d_real_loss: 0.4930 | d_Y_loss: 0.5116 | d_X_loss:
0.8078 | d_fake_loss: 1.3193 | g_loss: 3.6301
Iteration [ 7700/10000] | d_real_loss: 0.5162 | d_Y_loss: 0.6358 | d_X_loss:
0.7280 | d_fake_loss: 1.3639 | g_loss: 3.9861
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007700-Y-X.png
Iteration [ 7710/10000] | d_real_loss: 0.3135 | d_Y_loss: 0.6083 | d_X_loss:
0.6982 | d_fake_loss: 1.3065 | g_loss: 3.5410
Iteration [ 7720/10000] | d_real_loss: 0.4562 | d_Y_loss: 0.7163 | d_X_loss:
0.4289 | d_fake_loss: 1.1452 | g_loss: 3.9180
Iteration [ 7730/10000] | d_real_loss: 0.5015 | d_Y_loss: 0.7273 | d_X_loss:
0.9587 | d_fake_loss: 1.6860 | g_loss: 4.4259
Iteration [ 7740/10000] | d_real_loss: 0.6522 | d_Y_loss: 0.3972 | d_X_loss:
0.4230 | d_fake_loss: 0.8201 | g_loss: 4.1476
Iteration [ 7750/10000] | d_real_loss: 0.4834 | d_Y_loss: 0.9746 | d_X_loss:
0.4998 | d_fake_loss: 1.4744 | g_loss: 4.0455
Iteration [ 7760/10000] | d_real_loss: 0.3273 | d_Y_loss: 0.7587 | d_X_loss:
0.5832 | d_fake_loss: 1.3419 | g_loss: 3.8446
Iteration [ 7770/10000] | d_real_loss: 0.5621 | d_Y_loss: 0.6111 | d_X_loss:
0.4890 | d_fake_loss: 1.1002 | g_loss: 3.5149
Iteration [ 7780/10000] | d_real_loss: 0.3079 | d_Y_loss: 0.7723 | d_X_loss:
0.3854 | d_fake_loss: 1.1578 | g_loss: 3.8533
Iteration [ 7790/10000] | d_real_loss: 0.3664 | d_Y_loss: 0.6677 | d_X_loss:
0.6201 | d_fake_loss: 1.2878 | g_loss: 3.4007
Iteration [ 7800/10000] | d_real_loss: 0.2773 | d_Y_loss: 0.6624 | d_X_loss:
0.7403 | d_fake_loss: 1.4027 | g_loss: 3.6563
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007800-Y-X.png
Iteration [ 7810/10000] | d_real_loss: 0.5190 | d_Y_loss: 0.6106 | d_X_loss:
0.7044 | d_fake_loss: 1.3150 | g_loss: 3.5477
Iteration [ 7820/10000] | d_real_loss: 0.5516 | d_Y_loss: 0.7432 | d_X_loss:
0.3623 | d_fake_loss: 1.1055 | g_loss: 3.7931
Iteration [ 7830/10000] | d_real_loss: 0.2539 | d_Y_loss: 0.4057 | d_X_loss:
```

```
0.9606 | d_fake_loss: 1.3663 | g_loss: 4.3058
Iteration [ 7840/10000] | d_real_loss: 0.5142 | d_Y_loss: 0.6327 | d_X_loss:
0.6974 | d_fake_loss: 1.3301 | g_loss: 3.9998
Iteration [ 7850/10000] | d_real_loss: 0.4784 | d_Y_loss: 0.6510 | d_X_loss:
0.4318 | d_fake_loss: 1.0828 | g_loss: 3.7317
Iteration [ 7860/10000] | d_real_loss: 0.4433 | d_Y_loss: 0.7230 | d_X_loss:
0.8866 | d_fake_loss: 1.6095 | g_loss: 3.8579
Iteration [ 7870/10000] | d_real_loss: 0.5468 | d_Y_loss: 0.6940 | d_X_loss:
0.5159 | d_fake_loss: 1.2099 | g_loss: 3.8571
Iteration [ 7880/10000] | d_real_loss: 0.7715 | d_Y_loss: 0.4870 | d_X_loss:
0.3885 | d_fake_loss: 0.8756 | g_loss: 3.7728
Iteration [ 7890/10000] | d_real_loss: 0.4984 | d_Y_loss: 0.8817 | d_X_loss:
0.6052 | d_fake_loss: 1.4869 | g_loss: 3.5586
Iteration [ 7900/10000] | d_real_loss: 0.4291 | d_Y_loss: 0.4820 | d_X_loss:
0.3952 | d_fake_loss: 0.8772 | g_loss: 3.5046
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
007900-Y-X.png
Iteration [ 7910/10000] | d_real_loss: 0.4579 | d_Y_loss: 0.7054 | d_X_loss:
0.4620 | d_fake_loss: 1.1674 | g_loss: 3.9624
Iteration [ 7920/10000] | d_real_loss: 0.6405 | d_Y_loss: 0.5112 | d_X_loss:
0.8761 | d_fake_loss: 1.3873 | g_loss: 3.5223
Iteration [ 7930/10000] | d_real_loss: 0.4192 | d_Y_loss: 0.6248 | d_X_loss:
0.5714 | d_fake_loss: 1.1962 | g_loss: 4.1798
Iteration [ 7940/10000] | d_real_loss: 0.5000 | d_Y_loss: 0.7882 | d_X_loss:
0.7735 | d_fake_loss: 1.5617 | g_loss: 3.8755
Iteration [ 7950/10000] | d_real_loss: 0.5006 | d_Y_loss: 0.9597 | d_X_loss:
0.5980 | d_fake_loss: 1.5577 | g_loss: 3.6523
Iteration [ 7960/10000] | d_real_loss: 0.4740 | d_Y_loss: 0.7138 | d_X_loss:
0.5959 | d_fake_loss: 1.3097 | g_loss: 4.0213
Iteration [ 7970/10000] | d_real_loss: 0.6014 | d_Y_loss: 0.5068 | d_X_loss:
0.3821 | d_fake_loss: 0.8889 | g_loss: 3.7518
Iteration [ 7980/10000] | d_real_loss: 0.6218 | d_Y_loss: 0.6030 | d_X_loss:
0.2702 | d_fake_loss: 0.8733 | g_loss: 3.7444
Iteration [ 7990/10000] | d_real_loss: 0.6062 | d_Y_loss: 0.5713 | d_X_loss:
0.4958 | d_fake_loss: 1.0671 | g_loss: 3.8101
Iteration [ 8000/10000] | d_real_loss: 0.5426 | d_Y_loss: 0.5530 | d_X_loss:
0.7691 | d_fake_loss: 1.3221 | g_loss: 3.6616
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008000-Y-X.png
Iteration [ 8010/10000] | d_real_loss: 0.4143 | d_Y_loss: 0.5066 | d_X_loss:
0.8307 | d_fake_loss: 1.3373 | g_loss: 3.8019
Iteration [ 8020/10000] | d_real_loss: 0.5580 | d_Y_loss: 0.5608 | d_X_loss:
0.6933 | d_fake_loss: 1.2541 | g_loss: 3.9431
Iteration [ 8030/10000] | d_real_loss: 0.6527 | d_Y_loss: 0.4594 | d_X_loss:
```

```
0.4015 | d_fake_loss: 0.8609 | g_loss: 4.0694
Iteration [ 8040/10000] | d_real_loss: 0.4279 | d_Y_loss: 0.7271 | d_X_loss:
1.0822 | d_fake_loss: 1.8093 | g_loss: 3.3656
Iteration [ 8050/10000] | d_real_loss: 0.4324 | d_Y_loss: 0.6682 | d_X_loss:
0.4746 | d_fake_loss: 1.1428 | g_loss: 4.0591
Iteration [ 8060/10000] | d_real_loss: 0.4547 | d_Y_loss: 0.5565 | d_X_loss:
0.5040 | d_fake_loss: 1.0605 | g_loss: 3.8083
Iteration [ 8070/10000] | d_real_loss: 0.4399 | d_Y_loss: 0.7162 | d_X_loss:
0.4348 | d_fake_loss: 1.1510 | g_loss: 4.2146
Iteration [ 8080/10000] | d_real_loss: 0.4719 | d_Y_loss: 0.5430 | d_X_loss:
0.5749 | d_fake_loss: 1.1179 | g_loss: 3.5556
Iteration [ 8090/10000] | d_real_loss: 0.5917 | d_Y_loss: 0.4700 | d_X_loss:
0.5435 | d_fake_loss: 1.0135 | g_loss: 4.1386
Iteration [ 8100/10000] | d_real_loss: 0.4618 | d_Y_loss: 0.6799 | d_X_loss:
0.7046 | d_fake_loss: 1.3845 | g_loss: 3.8679
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008100-Y-X.png
Iteration [ 8110/10000] | d_real_loss: 0.5273 | d_Y_loss: 0.7483 | d_X_loss:
0.3693 | d_fake_loss: 1.1177 | g_loss: 4.0235
Iteration [ 8120/10000] | d_real_loss: 0.5701 | d_Y_loss: 0.3924 | d_X_loss:
1.0472 | d_fake_loss: 1.4396 | g_loss: 3.6625
Iteration [ 8130/10000] | d_real_loss: 0.4724 | d_Y_loss: 0.7946 | d_X_loss:
0.5320 | d_fake_loss: 1.3266 | g_loss: 3.9194
Iteration [ 8140/10000] | d_real_loss: 0.6017 | d_Y_loss: 0.9705 | d_X_loss:
0.3726 | d_fake_loss: 1.3431 | g_loss: 3.6823
Iteration [ 8150/10000] | d_real_loss: 0.5503 | d_Y_loss: 0.6704 | d_X_loss:
0.8224 | d_fake_loss: 1.4927 | g_loss: 3.5687
Iteration [ 8160/10000] | d_real_loss: 0.5620 | d_Y_loss: 0.7439 | d_X_loss:
0.5030 | d_fake_loss: 1.2469 | g_loss: 3.5537
Iteration [ 8170/10000] | d_real_loss: 0.5767 | d_Y_loss: 0.5493 | d_X_loss:
0.7350 | d_fake_loss: 1.2843 | g_loss: 3.6283
Iteration [ 8180/10000] | d_real_loss: 0.3902 | d_Y_loss: 0.4705 | d_X_loss:
0.5479 | d_fake_loss: 1.0184 | g_loss: 3.9806
Iteration [ 8190/10000] | d_real_loss: 0.5030 | d_Y_loss: 0.8729 | d_X_loss:
0.6664 | d_fake_loss: 1.5393 | g_loss: 3.6721
Iteration [ 8200/10000] | d_real_loss: 0.4972 | d_Y_loss: 0.9513 | d_X_loss:
0.5123 | d_fake_loss: 1.4636 | g_loss: 3.6450
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008200-Y-X.png
Iteration [ 8210/10000] | d_real_loss: 0.3138 | d_Y_loss: 0.7137 | d_X_loss:
0.3011 | d_fake_loss: 1.0147 | g_loss: 3.5173
Iteration [ 8220/10000] | d_real_loss: 0.4519 | d_Y_loss: 0.5417 | d_X_loss:
0.5898 | d_fake_loss: 1.1315 | g_loss: 3.6031
Iteration [ 8230/10000] | d_real_loss: 0.6352 | d_Y_loss: 0.4466 | d_X_loss:
```

```
0.5036 | d_fake_loss: 0.9502 | g_loss: 3.5977
Iteration [ 8240/10000] | d_real_loss: 0.5843 | d_Y_loss: 0.6817 | d_X_loss:
0.5415 | d_fake_loss: 1.2232 | g_loss: 3.3768
Iteration [ 8250/10000] | d_real_loss: 0.5545 | d_Y_loss: 0.5225 | d_X_loss:
0.3873 | d_fake_loss: 0.9098 | g_loss: 3.1640
Iteration [ 8260/10000] | d_real_loss: 0.3746 | d_Y_loss: 0.9507 | d_X_loss:
1.4135 | d_fake_loss: 2.3642 | g_loss: 3.8299
Iteration [ 8270/10000] | d_real_loss: 0.4536 | d_Y_loss: 0.5736 | d_X_loss:
0.7476 | d_fake_loss: 1.3212 | g_loss: 3.6916
Iteration [ 8280/10000] | d_real_loss: 0.4709 | d_Y_loss: 0.5985 | d_X_loss:
0.5314 | d_fake_loss: 1.1299 | g_loss: 3.9012
Iteration [ 8290/10000] | d_real_loss: 0.4619 | d_Y_loss: 0.6069 | d_X_loss:
0.8200 | d_fake_loss: 1.4269 | g_loss: 3.7293
Iteration [ 8300/10000] | d_real_loss: 0.6622 | d_Y_loss: 1.0158 | d_X_loss:
0.5513 | d_fake_loss: 1.5672 | g_loss: 3.9851
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008300-Y-X.png
Iteration [ 8310/10000] | d_real_loss: 0.5389 | d_Y_loss: 0.9247 | d_X_loss:
0.3906 | d_fake_loss: 1.3153 | g_loss: 3.9863
Iteration [ 8320/10000] | d_real_loss: 0.4327 | d_Y_loss: 0.8973 | d_X_loss:
0.8512 | d_fake_loss: 1.7485 | g_loss: 3.6797
Iteration [ 8330/10000] | d_real_loss: 0.5189 | d_Y_loss: 0.5300 | d_X_loss:
0.2724 | d_fake_loss: 0.8024 | g_loss: 4.0814
Iteration [ 8340/10000] | d_real_loss: 0.4562 | d_Y_loss: 0.6346 | d_X_loss:
0.4892 | d_fake_loss: 1.1239 | g_loss: 3.8133
Iteration [ 8350/10000] | d_real_loss: 0.7692 | d_Y_loss: 0.6463 | d_X_loss:
0.5504 | d_fake_loss: 1.1966 | g_loss: 3.6586
Iteration [ 8360/10000] | d_real_loss: 0.5133 | d_Y_loss: 0.4996 | d_X_loss:
0.5763 | d_fake_loss: 1.0759 | g_loss: 4.0763
Iteration [ 8370/10000] | d_real_loss: 0.4416 | d_Y_loss: 0.9575 | d_X_loss:
0.4341 | d_fake_loss: 1.3916 | g_loss: 3.9740
Iteration [ 8380/10000] | d_real_loss: 0.6309 | d_Y_loss: 0.5679 | d_X_loss:
0.7059 | d_fake_loss: 1.2738 | g_loss: 3.3891
Iteration [ 8390/10000] | d_real_loss: 0.5863 | d_Y_loss: 1.0079 | d_X_loss:
0.5976 | d_fake_loss: 1.6055 | g_loss: 3.4931
Iteration [ 8400/10000] | d_real_loss: 0.5788 | d_Y_loss: 0.6644 | d_X_loss:
0.7103 | d_fake_loss: 1.3746 | g_loss: 3.8020
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008400-Y-X.png
Iteration [ 8410/10000] | d_real_loss: 0.4010 | d_Y_loss: 0.4744 | d_X_loss:
0.9610 | d_fake_loss: 1.4354 | g_loss: 3.7833
Iteration [ 8420/10000] | d_real_loss: 0.4711 | d_Y_loss: 0.9631 | d_X_loss:
0.4078 | d_fake_loss: 1.3709 | g_loss: 3.7405
Iteration [ 8430/10000] | d_real_loss: 0.5792 | d_Y_loss: 0.3856 | d_X_loss:
```

```
0.5407 | d_fake_loss: 0.9264 | g_loss: 3.8598
Iteration [ 8440/10000] | d_real_loss: 0.4705 | d_Y_loss: 0.6855 | d_X_loss:
0.4481 | d_fake_loss: 1.1336 | g_loss: 3.6086
Iteration [ 8450/10000] | d_real_loss: 0.4807 | d_Y_loss: 0.7347 | d_X_loss:
0.6084 | d_fake_loss: 1.3432 | g_loss: 3.7885
Iteration [ 8460/10000] | d_real_loss: 0.4271 | d_Y_loss: 0.9334 | d_X_loss:
0.6187 | d_fake_loss: 1.5521 | g_loss: 3.5366
Iteration [ 8470/10000] | d_real_loss: 0.7485 | d_Y_loss: 0.7172 | d_X_loss:
0.5057 | d_fake_loss: 1.2229 | g_loss: 3.5257
Iteration [ 8480/10000] | d_real_loss: 0.4101 | d_Y_loss: 0.6569 | d_X_loss:
0.6932 | d_fake_loss: 1.3501 | g_loss: 3.4804
Iteration [ 8490/10000] | d_real_loss: 0.4642 | d_Y_loss: 0.5463 | d_X_loss:
0.4210 | d_fake_loss: 0.9673 | g_loss: 3.6809
Iteration [ 8500/10000] | d_real_loss: 0.6314 | d_Y_loss: 0.7670 | d_X_loss:
0.3270 | d_fake_loss: 1.0940 | g_loss: 3.7341
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008500-Y-X.png
Iteration [ 8510/10000] | d_real_loss: 0.4798 | d_Y_loss: 0.6555 | d_X_loss:
0.6970 | d_fake_loss: 1.3525 | g_loss: 3.6476
Iteration [ 8520/10000] | d_real_loss: 0.5974 | d_Y_loss: 0.7485 | d_X_loss:
0.4133 | d_fake_loss: 1.1617 | g_loss: 3.5161
Iteration [ 8530/10000] | d_real_loss: 0.5427 | d_Y_loss: 0.6204 | d_X_loss:
0.2241 | d_fake_loss: 0.8445 | g_loss: 3.7581
Iteration [ 8540/10000] | d_real_loss: 0.3855 | d_Y_loss: 0.6365 | d_X_loss:
0.8296 | d_fake_loss: 1.4661 | g_loss: 3.6442
Iteration [ 8550/10000] | d_real_loss: 0.3622 | d_Y_loss: 0.6730 | d_X_loss:
0.6685 | d_fake_loss: 1.3415 | g_loss: 3.7520
Iteration [ 8560/10000] | d_real_loss: 0.6025 | d_Y_loss: 0.3529 | d_X_loss:
0.4680 | d_fake_loss: 0.8209 | g_loss: 3.7208
Iteration [ 8570/10000] | d_real_loss: 0.4788 | d_Y_loss: 0.4433 | d_X_loss:
0.3425 | d_fake_loss: 0.7858 | g_loss: 3.7053
Iteration [ 8580/10000] | d_real_loss: 0.3856 | d_Y_loss: 0.4469 | d_X_loss:
0.5661 | d_fake_loss: 1.0131 | g_loss: 4.1189
Iteration [ 8590/10000] | d_real_loss: 0.5762 | d_Y_loss: 1.0478 | d_X_loss:
0.4431 | d_fake_loss: 1.4909 | g_loss: 3.8598
Iteration [ 8600/10000] | d_real_loss: 0.4850 | d_Y_loss: 0.6809 | d_X_loss:
0.4098 | d_fake_loss: 1.0907 | g_loss: 4.0766
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008600-Y-X.png
Iteration [ 8610/10000] | d_real_loss: 0.5690 | d_Y_loss: 0.6428 | d_X_loss:
0.6134 | d_fake_loss: 1.2562 | g_loss: 3.7471
Iteration [ 8620/10000] | d_real_loss: 0.2868 | d_Y_loss: 0.5306 | d_X_loss:
0.7121 | d_fake_loss: 1.2427 | g_loss: 3.9914
Iteration [ 8630/10000] | d_real_loss: 0.4055 | d_Y_loss: 0.7352 | d_X_loss:
```

```
0.7083 | d_fake_loss: 1.4435 | g_loss: 3.7785
Iteration [ 8640/10000] | d_real_loss: 0.3770 | d_Y_loss: 0.6341 | d_X_loss:
0.5735 | d_fake_loss: 1.2076 | g_loss: 3.9168
Iteration [ 8650/10000] | d_real_loss: 0.4740 | d_Y_loss: 0.6853 | d_X_loss:
0.3624 | d_fake_loss: 1.0477 | g_loss: 3.9260
Iteration [ 8660/10000] | d_real_loss: 0.4631 | d_Y_loss: 0.8178 | d_X_loss:
0.5773 | d_fake_loss: 1.3951 | g_loss: 3.6493
Iteration [ 8670/10000] | d_real_loss: 0.6039 | d_Y_loss: 0.7057 | d_X_loss:
0.8204 | d_fake_loss: 1.5261 | g_loss: 4.0630
Iteration [ 8680/10000] | d_real_loss: 0.4877 | d_Y_loss: 0.5307 | d_X_loss:
0.3884 | d_fake_loss: 0.9191 | g_loss: 4.0972
Iteration [ 8690/10000] | d_real_loss: 0.5419 | d_Y_loss: 0.8373 | d_X_loss:
0.6091 | d_fake_loss: 1.4464 | g_loss: 3.3089
Iteration [ 8700/10000] | d_real_loss: 0.4305 | d_Y_loss: 0.4915 | d_X_loss:
0.6467 | d_fake_loss: 1.1382 | g_loss: 3.9381
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008700-Y-X.png
Iteration [ 8710/10000] | d_real_loss: 0.6209 | d_Y_loss: 0.8934 | d_X_loss:
0.8736 | d_fake_loss: 1.7670 | g_loss: 3.2253
Iteration [ 8720/10000] | d_real_loss: 0.4611 | d_Y_loss: 0.8036 | d_X_loss:
0.4559 | d_fake_loss: 1.2595 | g_loss: 3.4253
Iteration [ 8730/10000] | d_real_loss: 0.4341 | d_Y_loss: 0.5508 | d_X_loss:
0.9087 | d_fake_loss: 1.4595 | g_loss: 3.8762
Iteration [ 8740/10000] | d_real_loss: 0.3848 | d_Y_loss: 0.5534 | d_X_loss:
0.6252 | d_fake_loss: 1.1786 | g_loss: 3.7563
Iteration [ 8750/10000] | d_real_loss: 0.4620 | d_Y_loss: 0.6290 | d_X_loss:
0.9683 | d_fake_loss: 1.5973 | g_loss: 3.8114
Iteration [ 8760/10000] | d_real_loss: 0.4391 | d_Y_loss: 0.9847 | d_X_loss:
0.7449 | d_fake_loss: 1.7295 | g_loss: 3.4126
Iteration [ 8770/10000] | d_real_loss: 0.6183 | d_Y_loss: 0.5665 | d_X_loss:
0.8695 | d_fake_loss: 1.4360 | g_loss: 3.7320
Iteration [ 8780/10000] | d_real_loss: 0.4689 | d_Y_loss: 0.6739 | d_X_loss:
0.5387 | d_fake_loss: 1.2126 | g_loss: 3.5275
Iteration [ 8790/10000] | d_real_loss: 0.4043 | d_Y_loss: 0.5371 | d_X_loss:
0.8096 | d_fake_loss: 1.3467 | g_loss: 3.9826
Iteration [ 8800/10000] | d_real_loss: 0.8613 | d_Y_loss: 0.5774 | d_X_loss:
0.4752 | d_fake_loss: 1.0526 | g_loss: 3.7771
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008800-Y-X.png
Iteration [ 8810/10000] | d_real_loss: 0.3295 | d_Y_loss: 0.6956 | d_X_loss:
0.8021 | d_fake_loss: 1.4977 | g_loss: 3.8508
Iteration [ 8820/10000] | d_real_loss: 0.7101 | d_Y_loss: 0.6474 | d_X_loss:
0.5979 | d_fake_loss: 1.2453 | g_loss: 3.5357
Iteration [ 8830/10000] | d_real_loss: 0.6341 | d_Y_loss: 0.4693 | d_X_loss:
```

```
0.6253 | d_fake_loss: 1.0947 | g_loss: 3.7439
Iteration [ 8840/10000] | d_real_loss: 0.6656 | d_Y_loss: 0.4886 | d_X_loss:
0.4604 | d_fake_loss: 0.9490 | g_loss: 3.4733
Iteration [ 8850/10000] | d_real_loss: 0.4886 | d_Y_loss: 0.8350 | d_X_loss:
0.5656 | d_fake_loss: 1.4006 | g_loss: 3.5832
Iteration [ 8860/10000] | d_real_loss: 0.4618 | d_Y_loss: 0.3198 | d_X_loss:
0.9090 | d_fake_loss: 1.2288 | g_loss: 4.0786
Iteration [ 8870/10000] | d_real_loss: 0.6827 | d_Y_loss: 0.7096 | d_X_loss:
0.9026 | d_fake_loss: 1.6122 | g_loss: 3.9252
Iteration [ 8880/10000] | d_real_loss: 0.5438 | d_Y_loss: 0.7733 | d_X_loss:
0.5254 | d_fake_loss: 1.2987 | g_loss: 4.1578
Iteration [ 8890/10000] | d_real_loss: 0.3743 | d_Y_loss: 0.6479 | d_X_loss:
0.8160 | d_fake_loss: 1.4639 | g_loss: 3.9972
Iteration [ 8900/10000] | d_real_loss: 0.4872 | d_Y_loss: 0.5362 | d_X_loss:
0.5623 | d_fake_loss: 1.0986 | g_loss: 3.8745
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
008900-Y-X.png
Iteration [ 8910/10000] | d_real_loss: 0.4792 | d_Y_loss: 1.0332 | d_X_loss:
0.5469 | d_fake_loss: 1.5801 | g_loss: 3.7076
Iteration [ 8920/10000] | d_real_loss: 0.4861 | d_Y_loss: 0.8261 | d_X_loss:
0.4451 | d_fake_loss: 1.2711 | g_loss: 3.7439
Iteration [ 8930/10000] | d_real_loss: 0.7601 | d_Y_loss: 0.5064 | d_X_loss:
0.4916 | d_fake_loss: 0.9979 | g_loss: 3.6265
Iteration [ 8940/10000] | d_real_loss: 0.6574 | d_Y_loss: 0.9285 | d_X_loss:
0.6575 | d_fake_loss: 1.5860 | g_loss: 3.7205
Iteration [ 8950/10000] | d_real_loss: 0.5880 | d_Y_loss: 0.7413 | d_X_loss:
0.5914 | d_fake_loss: 1.3326 | g_loss: 3.5236
Iteration [ 8960/10000] | d_real_loss: 0.5529 | d_Y_loss: 0.7694 | d_X_loss:
0.4301 | d_fake_loss: 1.1995 | g_loss: 3.5152
Iteration [ 8970/10000] | d_real_loss: 0.5422 | d_Y_loss: 0.3218 | d_X_loss:
0.4908 | d_fake_loss: 0.8127 | g_loss: 3.5179
Iteration [ 8980/10000] | d_real_loss: 0.4652 | d_Y_loss: 0.5449 | d_X_loss:
0.5873 | d_fake_loss: 1.1322 | g_loss: 3.7440
Iteration [ 8990/10000] | d_real_loss: 0.5489 | d_Y_loss: 0.6875 | d_X_loss:
0.5742 | d_fake_loss: 1.2617 | g_loss: 3.6272
Iteration [ 9000/10000] | d_real_loss: 0.5157 | d_Y_loss: 0.5847 | d_X_loss:
0.6614 | d_fake_loss: 1.2461 | g_loss: 3.7308
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009000-Y-X.png
Iteration [ 9010/10000] | d_real_loss: 0.5384 | d_Y_loss: 0.3879 | d_X_loss:
0.6147 | d_fake_loss: 1.0027 | g_loss: 4.2250
Iteration [ 9020/10000] | d_real_loss: 0.3965 | d_Y_loss: 0.5967 | d_X_loss:
0.7627 | d_fake_loss: 1.3594 | g_loss: 3.7857
Iteration [ 9030/10000] | d_real_loss: 0.6176 | d_Y_loss: 0.7720 | d_X_loss:
```

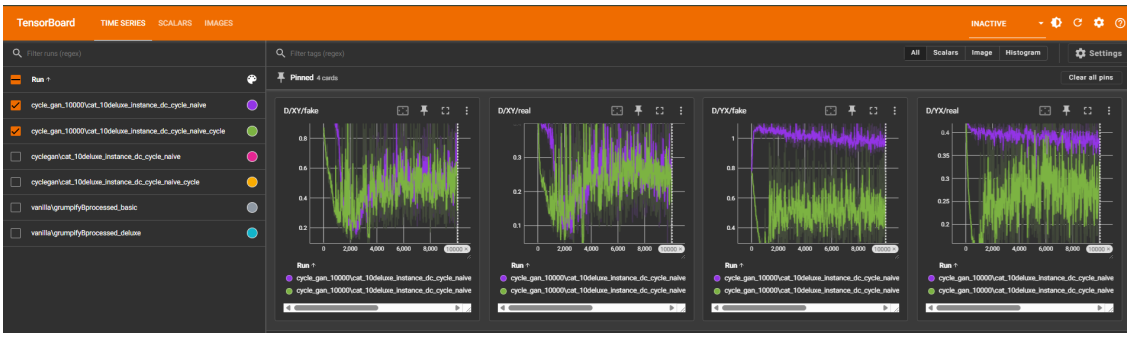
```
0.3142 | d_fake_loss: 1.0862 | g_loss: 3.8470
Iteration [ 9040/10000] | d_real_loss: 0.5869 | d_Y_loss: 0.5004 | d_X_loss:
0.4059 | d_fake_loss: 0.9063 | g_loss: 4.0458
Iteration [ 9050/10000] | d_real_loss: 0.6601 | d_Y_loss: 0.4513 | d_X_loss:
0.4394 | d_fake_loss: 0.8907 | g_loss: 3.8830
Iteration [ 9060/10000] | d_real_loss: 0.5513 | d_Y_loss: 0.6108 | d_X_loss:
0.5201 | d_fake_loss: 1.1309 | g_loss: 3.6988
Iteration [ 9070/10000] | d_real_loss: 0.4005 | d_Y_loss: 0.8148 | d_X_loss:
0.5735 | d_fake_loss: 1.3883 | g_loss: 3.6952
Iteration [ 9080/10000] | d_real_loss: 0.3414 | d_Y_loss: 0.5811 | d_X_loss:
0.4792 | d_fake_loss: 1.0603 | g_loss: 4.0442
Iteration [ 9090/10000] | d_real_loss: 0.4834 | d_Y_loss: 0.5908 | d_X_loss:
0.7577 | d_fake_loss: 1.3485 | g_loss: 4.0534
Iteration [ 9100/10000] | d_real_loss: 0.3440 | d_Y_loss: 0.5646 | d_X_loss:
0.6425 | d_fake_loss: 1.2071 | g_loss: 3.8692
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009100-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009100-Y-X.png
Iteration [ 9110/10000] | d_real_loss: 0.3267 | d_Y_loss: 0.9843 | d_X_loss:
0.5852 | d_fake_loss: 1.5695 | g_loss: 3.9291
Iteration [ 9120/10000] | d_real_loss: 0.3034 | d_Y_loss: 0.3364 | d_X_loss:
0.3409 | d_fake_loss: 0.6772 | g_loss: 4.0520
Iteration [ 9130/10000] | d_real_loss: 0.3969 | d_Y_loss: 0.5459 | d_X_loss:
0.8139 | d_fake_loss: 1.3598 | g_loss: 3.9620
Iteration [ 9140/10000] | d_real_loss: 0.4373 | d_Y_loss: 0.7043 | d_X_loss:
0.3765 | d_fake_loss: 1.0808 | g_loss: 3.9551
Iteration [ 9150/10000] | d_real_loss: 0.4953 | d_Y_loss: 0.5861 | d_X_loss:
0.5758 | d_fake_loss: 1.1619 | g_loss: 3.6564
Iteration [ 9160/10000] | d_real_loss: 0.4454 | d_Y_loss: 0.7644 | d_X_loss:
0.3588 | d_fake_loss: 1.1232 | g_loss: 3.9083
Iteration [ 9170/10000] | d_real_loss: 0.3517 | d_Y_loss: 0.6177 | d_X_loss:
0.8526 | d_fake_loss: 1.4704 | g_loss: 4.3405
Iteration [ 9180/10000] | d_real_loss: 0.4353 | d_Y_loss: 0.6842 | d_X_loss:
1.0738 | d_fake_loss: 1.7580 | g_loss: 3.9306
Iteration [ 9190/10000] | d_real_loss: 0.6589 | d_Y_loss: 0.5936 | d_X_loss:
0.5227 | d_fake_loss: 1.1163 | g_loss: 3.9436
Iteration [ 9200/10000] | d_real_loss: 0.6796 | d_Y_loss: 0.4221 | d_X_loss:
0.3879 | d_fake_loss: 0.8100 | g_loss: 3.5526
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009200-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009200-Y-X.png
Iteration [ 9210/10000] | d_real_loss: 0.3046 | d_Y_loss: 0.6080 | d_X_loss:
0.5584 | d_fake_loss: 1.1665 | g_loss: 4.0554
Iteration [ 9220/10000] | d_real_loss: 0.5103 | d_Y_loss: 0.8032 | d_X_loss:
0.9379 | d_fake_loss: 1.7411 | g_loss: 3.6432
Iteration [ 9230/10000] | d_real_loss: 0.5165 | d_Y_loss: 0.4815 | d_X_loss:
```

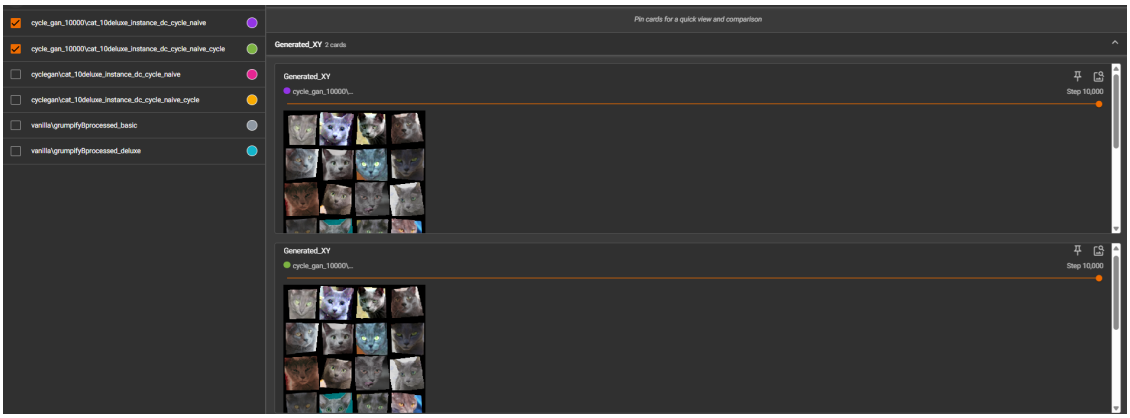
```
0.6587 | d_fake_loss: 1.1401 | g_loss: 3.9218
Iteration [ 9240/10000] | d_real_loss: 0.4760 | d_Y_loss: 0.7387 | d_X_loss:
0.5489 | d_fake_loss: 1.2876 | g_loss: 3.9194
Iteration [ 9250/10000] | d_real_loss: 0.4207 | d_Y_loss: 0.4925 | d_X_loss:
0.4470 | d_fake_loss: 0.9395 | g_loss: 3.7113
Iteration [ 9260/10000] | d_real_loss: 0.3678 | d_Y_loss: 0.7529 | d_X_loss:
0.5276 | d_fake_loss: 1.2805 | g_loss: 3.5294
Iteration [ 9270/10000] | d_real_loss: 0.5170 | d_Y_loss: 0.5749 | d_X_loss:
0.8378 | d_fake_loss: 1.4126 | g_loss: 3.8923
Iteration [ 9280/10000] | d_real_loss: 0.4027 | d_Y_loss: 0.4754 | d_X_loss:
0.4726 | d_fake_loss: 0.9480 | g_loss: 3.9200
Iteration [ 9290/10000] | d_real_loss: 0.4816 | d_Y_loss: 0.7343 | d_X_loss:
0.5526 | d_fake_loss: 1.2869 | g_loss: 3.6140
Iteration [ 9300/10000] | d_real_loss: 0.5010 | d_Y_loss: 0.5198 | d_X_loss:
0.3087 | d_fake_loss: 0.8285 | g_loss: 3.4384
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009300-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009300-Y-X.png
Iteration [ 9310/10000] | d_real_loss: 0.5022 | d_Y_loss: 0.5799 | d_X_loss:
0.7359 | d_fake_loss: 1.3158 | g_loss: 3.7368
Iteration [ 9320/10000] | d_real_loss: 0.4810 | d_Y_loss: 0.4606 | d_X_loss:
0.5332 | d_fake_loss: 0.9937 | g_loss: 4.5512
Iteration [ 9330/10000] | d_real_loss: 0.3490 | d_Y_loss: 0.6148 | d_X_loss:
0.5448 | d_fake_loss: 1.1596 | g_loss: 3.8210
Iteration [ 9340/10000] | d_real_loss: 0.4279 | d_Y_loss: 0.4234 | d_X_loss:
0.4701 | d_fake_loss: 0.8935 | g_loss: 3.5920
Iteration [ 9350/10000] | d_real_loss: 0.3333 | d_Y_loss: 0.6377 | d_X_loss:
0.5651 | d_fake_loss: 1.2028 | g_loss: 4.1865
Iteration [ 9360/10000] | d_real_loss: 0.2958 | d_Y_loss: 0.6312 | d_X_loss:
0.9187 | d_fake_loss: 1.5500 | g_loss: 4.2262
Iteration [ 9370/10000] | d_real_loss: 0.5345 | d_Y_loss: 0.4817 | d_X_loss:
0.8151 | d_fake_loss: 1.2968 | g_loss: 4.5094
Iteration [ 9380/10000] | d_real_loss: 0.7312 | d_Y_loss: 0.8880 | d_X_loss:
0.3869 | d_fake_loss: 1.2748 | g_loss: 3.7911
Iteration [ 9390/10000] | d_real_loss: 0.4351 | d_Y_loss: 0.6482 | d_X_loss:
0.8291 | d_fake_loss: 1.4773 | g_loss: 3.8284
Iteration [ 9400/10000] | d_real_loss: 0.4887 | d_Y_loss: 0.7220 | d_X_loss:
0.4546 | d_fake_loss: 1.1767 | g_loss: 3.9468
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009400-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009400-Y-X.png
Iteration [ 9410/10000] | d_real_loss: 0.3279 | d_Y_loss: 1.2191 | d_X_loss:
1.1357 | d_fake_loss: 2.3548 | g_loss: 3.7194
Iteration [ 9420/10000] | d_real_loss: 0.5278 | d_Y_loss: 0.9724 | d_X_loss:
1.0270 | d_fake_loss: 1.9994 | g_loss: 3.5610
Iteration [ 9430/10000] | d_real_loss: 0.5741 | d_Y_loss: 0.6570 | d_X_loss:
```

```
0.7870 | d_fake_loss: 1.4440 | g_loss: 3.8409
Iteration [ 9440/10000] | d_real_loss: 0.4919 | d_Y_loss: 0.5838 | d_X_loss:
0.6166 | d_fake_loss: 1.2004 | g_loss: 3.7491
Iteration [ 9450/10000] | d_real_loss: 0.6688 | d_Y_loss: 0.4414 | d_X_loss:
0.5836 | d_fake_loss: 1.0250 | g_loss: 3.5766
Iteration [ 9460/10000] | d_real_loss: 0.6551 | d_Y_loss: 0.6198 | d_X_loss:
0.5084 | d_fake_loss: 1.1283 | g_loss: 3.9185
Iteration [ 9470/10000] | d_real_loss: 0.3330 | d_Y_loss: 0.6549 | d_X_loss:
0.4143 | d_fake_loss: 1.0692 | g_loss: 3.7213
Iteration [ 9480/10000] | d_real_loss: 0.4707 | d_Y_loss: 0.8030 | d_X_loss:
0.5552 | d_fake_loss: 1.3582 | g_loss: 3.9891
Iteration [ 9490/10000] | d_real_loss: 0.4782 | d_Y_loss: 0.4492 | d_X_loss:
0.7439 | d_fake_loss: 1.1930 | g_loss: 4.2791
Iteration [ 9500/10000] | d_real_loss: 0.4835 | d_Y_loss: 0.5694 | d_X_loss:
0.4096 | d_fake_loss: 0.9790 | g_loss: 3.3408
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009500-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009500-Y-X.png
Iteration [ 9510/10000] | d_real_loss: 0.4163 | d_Y_loss: 0.7912 | d_X_loss:
0.5296 | d_fake_loss: 1.3208 | g_loss: 3.7801
Iteration [ 9520/10000] | d_real_loss: 0.4363 | d_Y_loss: 0.5446 | d_X_loss:
0.4636 | d_fake_loss: 1.0081 | g_loss: 3.7540
Iteration [ 9530/10000] | d_real_loss: 0.5223 | d_Y_loss: 0.7650 | d_X_loss:
0.4550 | d_fake_loss: 1.2200 | g_loss: 4.2398
Iteration [ 9540/10000] | d_real_loss: 0.4023 | d_Y_loss: 0.7253 | d_X_loss:
0.8472 | d_fake_loss: 1.5725 | g_loss: 3.5697
Iteration [ 9550/10000] | d_real_loss: 0.5647 | d_Y_loss: 0.4915 | d_X_loss:
0.3799 | d_fake_loss: 0.8714 | g_loss: 3.7300
Iteration [ 9560/10000] | d_real_loss: 0.6213 | d_Y_loss: 0.4531 | d_X_loss:
0.2749 | d_fake_loss: 0.7281 | g_loss: 4.2465
Iteration [ 9570/10000] | d_real_loss: 0.4303 | d_Y_loss: 0.6652 | d_X_loss:
0.4266 | d_fake_loss: 1.0919 | g_loss: 3.6521
Iteration [ 9580/10000] | d_real_loss: 0.7801 | d_Y_loss: 0.6341 | d_X_loss:
0.7442 | d_fake_loss: 1.3782 | g_loss: 3.6100
Iteration [ 9590/10000] | d_real_loss: 0.4171 | d_Y_loss: 0.4929 | d_X_loss:
0.3197 | d_fake_loss: 0.8127 | g_loss: 3.9022
Iteration [ 9600/10000] | d_real_loss: 0.4637 | d_Y_loss: 0.5411 | d_X_loss:
0.3816 | d_fake_loss: 0.9227 | g_loss: 3.9262
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009600-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009600-Y-X.png
Iteration [ 9610/10000] | d_real_loss: 0.3826 | d_Y_loss: 0.5069 | d_X_loss:
1.0793 | d_fake_loss: 1.5862 | g_loss: 3.5820
Iteration [ 9620/10000] | d_real_loss: 0.4713 | d_Y_loss: 0.6057 | d_X_loss:
0.5424 | d_fake_loss: 1.1481 | g_loss: 3.8203
Iteration [ 9630/10000] | d_real_loss: 0.5512 | d_Y_loss: 0.4640 | d_X_loss:
```

```
0.5651 | d_fake_loss: 1.0291 | g_loss: 4.2185
Iteration [ 9640/10000] | d_real_loss: 0.5093 | d_Y_loss: 0.6803 | d_X_loss:
0.6450 | d_fake_loss: 1.3253 | g_loss: 4.0814
Iteration [ 9650/10000] | d_real_loss: 0.4399 | d_Y_loss: 0.7053 | d_X_loss:
0.6111 | d_fake_loss: 1.3164 | g_loss: 3.8201
Iteration [ 9660/10000] | d_real_loss: 0.4986 | d_Y_loss: 0.4694 | d_X_loss:
0.5581 | d_fake_loss: 1.0275 | g_loss: 3.8351
Iteration [ 9670/10000] | d_real_loss: 0.3775 | d_Y_loss: 0.6236 | d_X_loss:
0.6823 | d_fake_loss: 1.3059 | g_loss: 4.2733
Iteration [ 9680/10000] | d_real_loss: 0.4701 | d_Y_loss: 0.4231 | d_X_loss:
0.8876 | d_fake_loss: 1.3107 | g_loss: 3.8131
Iteration [ 9690/10000] | d_real_loss: 0.3847 | d_Y_loss: 0.8229 | d_X_loss:
0.4530 | d_fake_loss: 1.2760 | g_loss: 3.4821
Iteration [ 9700/10000] | d_real_loss: 0.3090 | d_Y_loss: 0.6188 | d_X_loss:
0.8634 | d_fake_loss: 1.4822 | g_loss: 4.1277
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009700-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009700-Y-X.png
Iteration [ 9710/10000] | d_real_loss: 0.4274 | d_Y_loss: 0.9995 | d_X_loss:
0.3524 | d_fake_loss: 1.3520 | g_loss: 4.0930
Iteration [ 9720/10000] | d_real_loss: 0.3773 | d_Y_loss: 0.5714 | d_X_loss:
0.2781 | d_fake_loss: 0.8495 | g_loss: 3.9705
Iteration [ 9730/10000] | d_real_loss: 0.4083 | d_Y_loss: 0.5538 | d_X_loss:
0.4194 | d_fake_loss: 0.9731 | g_loss: 3.6141
Iteration [ 9740/10000] | d_real_loss: 0.4926 | d_Y_loss: 0.4783 | d_X_loss:
0.5080 | d_fake_loss: 0.9863 | g_loss: 3.8728
Iteration [ 9750/10000] | d_real_loss: 0.3568 | d_Y_loss: 0.4381 | d_X_loss:
0.5349 | d_fake_loss: 0.9730 | g_loss: 4.2097
Iteration [ 9760/10000] | d_real_loss: 0.3994 | d_Y_loss: 0.5455 | d_X_loss:
0.7462 | d_fake_loss: 1.2917 | g_loss: 3.7201
Iteration [ 9770/10000] | d_real_loss: 0.5669 | d_Y_loss: 0.4767 | d_X_loss:
0.4901 | d_fake_loss: 0.9668 | g_loss: 3.6113
Iteration [ 9780/10000] | d_real_loss: 0.5059 | d_Y_loss: 0.3521 | d_X_loss:
0.3663 | d_fake_loss: 0.7184 | g_loss: 3.7359
Iteration [ 9790/10000] | d_real_loss: 0.5473 | d_Y_loss: 0.6832 | d_X_loss:
0.6358 | d_fake_loss: 1.3191 | g_loss: 3.7878
Iteration [ 9800/10000] | d_real_loss: 0.2765 | d_Y_loss: 0.5744 | d_X_loss:
0.4333 | d_fake_loss: 1.0076 | g_loss: 3.6368
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009800-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009800-Y-X.png
Iteration [ 9810/10000] | d_real_loss: 0.7489 | d_Y_loss: 0.9323 | d_X_loss:
0.3145 | d_fake_loss: 1.2468 | g_loss: 3.6234
Iteration [ 9820/10000] | d_real_loss: 0.7595 | d_Y_loss: 0.5466 | d_X_loss:
0.6585 | d_fake_loss: 1.2051 | g_loss: 3.8104
Iteration [ 9830/10000] | d_real_loss: 0.5016 | d_Y_loss: 0.6701 | d_X_loss:
```

```
0.4169 | d_fake_loss: 1.0870 | g_loss: 3.6302
Iteration [ 9840/10000] | d_real_loss: 0.5098 | d_Y_loss: 0.5405 | d_X_loss:
0.4650 | d_fake_loss: 1.0055 | g_loss: 3.7358
Iteration [ 9850/10000] | d_real_loss: 0.5959 | d_Y_loss: 0.5886 | d_X_loss:
0.3116 | d_fake_loss: 0.9002 | g_loss: 3.5640
Iteration [ 9860/10000] | d_real_loss: 0.4003 | d_Y_loss: 0.5960 | d_X_loss:
0.4246 | d_fake_loss: 1.0206 | g_loss: 3.6858
Iteration [ 9870/10000] | d_real_loss: 0.5502 | d_Y_loss: 0.6977 | d_X_loss:
0.7211 | d_fake_loss: 1.4188 | g_loss: 4.2639
Iteration [ 9880/10000] | d_real_loss: 0.5739 | d_Y_loss: 0.7770 | d_X_loss:
0.4492 | d_fake_loss: 1.2261 | g_loss: 3.6837
Iteration [ 9890/10000] | d_real_loss: 0.6483 | d_Y_loss: 0.7414 | d_X_loss:
0.6284 | d_fake_loss: 1.3698 | g_loss: 3.9599
Iteration [ 9900/10000] | d_real_loss: 0.4320 | d_Y_loss: 0.7268 | d_X_loss:
0.5329 | d_fake_loss: 1.2597 | g_loss: 3.5986
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009900-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
009900-Y-X.png
Iteration [ 9910/10000] | d_real_loss: 0.4128 | d_Y_loss: 0.8064 | d_X_loss:
0.5517 | d_fake_loss: 1.3582 | g_loss: 3.5222
Iteration [ 9920/10000] | d_real_loss: 0.5408 | d_Y_loss: 0.5320 | d_X_loss:
0.4429 | d_fake_loss: 0.9749 | g_loss: 3.7478
Iteration [ 9930/10000] | d_real_loss: 0.5529 | d_Y_loss: 0.8200 | d_X_loss:
0.3423 | d_fake_loss: 1.1622 | g_loss: 3.3273
Iteration [ 9940/10000] | d_real_loss: 0.4093 | d_Y_loss: 0.7032 | d_X_loss:
0.9037 | d_fake_loss: 1.6070 | g_loss: 3.7446
Iteration [ 9950/10000] | d_real_loss: 0.3754 | d_Y_loss: 0.8682 | d_X_loss:
0.4176 | d_fake_loss: 1.2858 | g_loss: 3.3597
Iteration [ 9960/10000] | d_real_loss: 0.4472 | d_Y_loss: 0.4199 | d_X_loss:
0.5675 | d_fake_loss: 0.9874 | g_loss: 3.7111
Iteration [ 9970/10000] | d_real_loss: 0.4387 | d_Y_loss: 0.5292 | d_X_loss:
0.4599 | d_fake_loss: 0.9891 | g_loss: 3.3953
Iteration [ 9980/10000] | d_real_loss: 0.4532 | d_Y_loss: 0.7249 | d_X_loss:
0.4057 | d_fake_loss: 1.1307 | g_loss: 4.3037
Iteration [ 9990/10000] | d_real_loss: 0.2387 | d_Y_loss: 0.5045 | d_X_loss:
0.5601 | d_fake_loss: 1.0646 | g_loss: 3.5318
Iteration [10000/10000] | d_real_loss: 0.7040 | d_Y_loss: 0.7817 | d_X_loss:
0.2932 | d_fake_loss: 1.0750 | g_loss: 3.7031
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
010000-X-Y.png
Saved output/cycle_gan_10000\cat_10deluxe_instance_dc_cycle_naive_cycle\sample-
010000-Y-X.png
```





Results TensorBoard

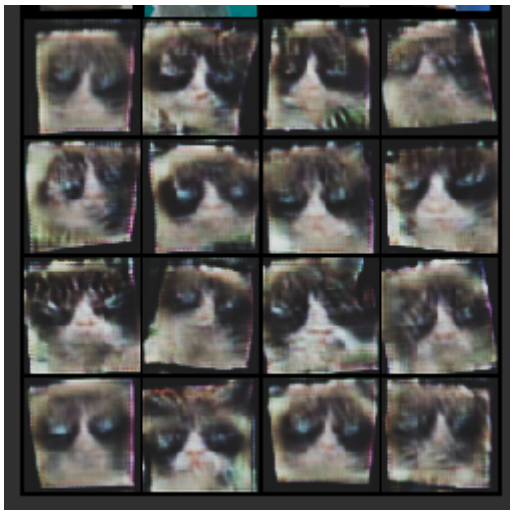
Can you account for these differences? **Answer:** We can see that when we use cycle consistency, the loss curves tend to be more stable, with gradual slopes and fewer extreme flactuations. This probably happens because the consistency loss acts as a guiding principle that ensures that the networks not only create fake images to "fool" the other network, but also thet they maintain coherence when an image is modified and then reverted to its original state. This guiding principle helps the training to be more stable and helps the networks to learn in a more organized way, instead of simply trying to "fool" each other all the time. The generator seems to learn better when following this principle.

Provide explanations as to why there might or might not be a noticeable difference between the two sets of results. **Answer:** Analyzing the images, we do not see a big difference

in the final images. This could be because the two types of "grumpy cats" we are using are not different in style. If the transformation we want to make is not too significant, the network can probably still learn to perform it correctly, even without the cycle coherence rule. Also, the rule depends on its weight parameter (lambda), and if this parameter needs to be tuned.

In this case, we used L1 loss, as it was recommended in the original paper. Perhaps, if we trained the model longer or used larger networks, or if the difference between cat styles was more noticeable, we would see a significant improvement in the images produced when we use the cycle coherence rule.

Any differences appear to be difficult to detect and would probably require closer examination; at a glance the images generated are the same. This visual similarity supports the explanation given above about the possible reasons for a possible large difference in the final result for this particular data set and training configuration.



To observe a noticeable difference, we can compare the images generated at iteration 1900 with the one from the final iteration. We notice that the images generated in this iteration, show a significant improvenment in quality, which leads us to conclude that the Generator has learned to

create better images. Finally, this comparison allows us to conclude that CycleGan performs much better that VanillaGan at generating high-quality images.