```
# mount drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# clip architecture
!git clone https://github.com/openai/CLIP
```

```
Cloning into 'CLIP'...
remote: Enumerating objects: 256, done.
remote: Counting objects: 100% (154/154), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 256 (delta 126), reused 110 (delta 110), pack-reused 102 (from 1)
Receiving objects: 100% (256/256), 8.86 MiB | 31.96 MiB/s, done.
Resolving deltas: 100% (140/140), done.
```

```
# taming transformer
!git clone https://github.com/CompVis/taming-transformers
```

```
Cloning into 'taming-transformers'...
remote: Enumerating objects: 1342, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 1342 (delta 0), reused 1 (delta 0), pack-reused 1340 (from 1)
Receiving objects: 100% (1342/1342), 409.77 MiB | 37.00 MiB/s, done.
Resolving deltas: 100% (282/282), done.
```

```
# install all the library!
!pip install --no-deps ftfy regex tqdm
!pip install omegaconf==2.0.0 pytorch-lightning==1.0.8
!pip uninstall torchtext --yes

!pip install torch-fidelity einops
```

```
Downloading ftfy-6.3.1-py3-none-any.whl (44 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.8/44.8 kB 2.3 MB/s eta 0:00:00
Installing collected packages: ftfy
Successfully installed ftfy-6.3.1
Collecting omegaconf==2.0.0
  Downloading omegaconf-2.0.0-py3-none-any.whl.metadata (3.5 kB)
Collecting pytorch-lightning==1.0.8
  Downloading pytorch_lightning-1.0.8-py3-none-any.whl.metadata (26 kB)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from omegaconf==2.0.0) (6.0.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from omegaconf==2.0.0) (4.12.2)
Requirement already satisfied: numpy>=1.16.4 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (1.26.4)
Requirement already satisfied: torch>=1.3 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2.5.1+cu12
Requirement already satisfied: future>=0.17.1 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (1.0.0)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (4.66.6)
Requirement already satisfied: fsspec>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2024.10
Requirement already satisfied: tensorboard>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2.
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightni
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-light
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-ligh
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning=
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->py
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-l
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboar
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-ligh
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (3.
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch>=1.3->py
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard>=
Downloading omegaconf-2.0.0-py3-none-any.whl (33 kB)
Downloading pytorch_lightning-1.0.8-py3-none-any.whl (561 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 561.4/561.4 kB 16.5 MB/s eta 0:00:00
Installing collected packages: omegaconf, pytorch-lightning
Successfully installed omegaconf-2.0.0 pytorch-lightning-1.0.8
WARNING: Skipping torchtext as it is not installed.
Collecting torch-fidelity
  Downloading torch_fidelity-0.3.0-py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: einops in /usr/local/lib/python3.10/dist-packages (0.8.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (1.26.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (11.0.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (1.13.1)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (2.5.1+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (0.20.1+cu121)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from torch-fidelity) (4.66.6)
```

```
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch->torch-fidelity) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch->torch-f
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch->torch-fidelity) (
Downloading torch_fidelity-0.3.0-py3-none-any.whl (37 kB)
Installing collected packages: torch-fidelity
Successfully installed torch-fidelity-0.3.0
```

```
import PIL
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
import torch, os, imageio,pdb,math
import torchvision
import torchvision.transforms as T
import torchvision.transforms.functional as TF
```

```
import yaml
from omegaconf import OmegaConf

from CLIP import clip
import warnings
warnings.filterwarnings("ignore")
```

Double-click (or enter) to edit

```
## helper functions

def show_from_tensor(tensor):
  img = tensor.clone()
  img = img.mul(255).byte()
  img = img.cpu().numpy().transpose((1,2,0))

  plt.figure(figsize=(10,7))
  plt.axis('off')
  plt.imshow(img)
  plt.show()

def norm_data(data):
  return (data.clip(-1,1)+1)/2 ### range between 0 and 1 in the result

### Parameters
learning_rate = .5
batch_size = 1
wd = .1
noise_factor = .22

total_iter=400
im_shape = [450, 450, 3] # height, width, channel
size1, size2, channels = im_shape


clipmodel,_=clip.load("ViT-B/32", jit=False)
clipmodel.eval()
print(clip.available_models())

print("clip model visual input resolution: ", clipmodel.visual.input_resolution)

device=torch.device("cuda:0")
torch.cuda.empty_cache()
```

```
100%|████████████████████████████████| 338M/338M [00:05<00:00, 67.5MiB/s]
['RN50', 'RN101', 'RN50x4', 'RN50x16', 'RN50x64', 'ViT-B/32', 'ViT-B/16', 'ViT-L/14', 'ViT-L/14@336px']
clip model visual input resolution:  224
```

```
## Taming transformer instantiation

%cd taming-transformers/

!mkdir -p models/vqgan_imagenet_f16_16384/checkpoints
!mkdir -p models/vqgan_imagenet_f16_16384/configs

if len(os.listdir('models/vqgan_imagenet_f16_16384/checkpoints/')) == 0:
   !wget 'https://heibox.uni-heidelberg.de/f/867b05fc8c4841768640/?dl=1' -O 'models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt'
   !wget 'https://heibox.uni-heidelberg.de/f/274fb24ed38341bfa753/?dl=1' -O 'models/vqgan_imagenet_f16_16384/configs/model.yaml'
```

```
/content/taming-transformers
--2024-12-16 14:28:30--  https://heibox.uni-heidelberg.de/f/867b05fc8c4841768640/?dl=1
```

```
Resolving heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)... 129.206.7.113
Connecting to heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)|129.206.7.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://heibox.uni-heidelberg.de/seafhttp/files/1cc44988-a856-4939-b408-1c91717b5a6a/last.ckpt [following]
--2024-12-16 14:28:31--  https://heibox.uni-heidelberg.de/seafhttp/files/1cc44988-a856-4939-b408-1c91717b5a6a/last.ckpt
Reusing existing connection to heibox.uni-heidelberg.de:443.
HTTP request sent, awaiting response... 200 OK
Length: 980092370 (935M) [application/octet-stream]
Saving to: ‘models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt’

models/vqgan_imagen 100%[==================>] 934.69M  9.53MB/s    in 1m 40s

2024-12-16 14:30:11 (9.39 MB/s) - ‘models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt’ saved [980092370/980092370]

--2024-12-16 14:30:11--  https://heibox.uni-heidelberg.de/f/274fb24ed38341bfa753/?dl=1
Resolving heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)... 129.206.7.113
Connecting to heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)|129.206.7.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://heibox.uni-heidelberg.de/seafhttp/files/009cd4d0-f547-4d30-aa83-cbed62217c4e/model.yaml [following]
--2024-12-16 14:30:12--  https://heibox.uni-heidelberg.de/seafhttp/files/009cd4d0-f547-4d30-aa83-cbed62217c4e/model.yaml
Reusing existing connection to heibox.uni-heidelberg.de:443.
HTTP request sent, awaiting response... 200 OK
Length: 692 [application/octet-stream]
Saving to: ‘models/vqgan_imagenet_f16_16384/configs/model.yaml’

models/vqgan_imagen 100%[==================>]     692  --.-KB/s    in 0s

2024-12-16 14:30:12 (482 MB/s) - ‘models/vqgan_imagenet_f16_16384/configs/model.yaml’ saved [692/692]
```

```
from taming.models.vqgan import VQModel

def load_config(config_path, display=False):
   config_data = OmegaConf.load(config_path)
   if display:
     print(yaml.dump(OmegaConf.to_container(config_data)))
   return config_data

def load_vqgan(config, chk_path=None):
  model = VQModel(**config.model.params)
  if chk_path is not None:
    state_dict = torch.load(chk_path, map_location="cpu")["state_dict"]
    missing, unexpected = model.load_state_dict(state_dict, strict=False)
  return model.eval()

def generator(x):
  x = taming_model.post_quant_conv(x)
  x = taming_model.decoder(x)
  return x

taming_config = load_config("./models/vqgan_imagenet_f16_16384/configs/model.yaml", display=True)
taming_model = load_vqgan(taming_config, chk_path="./models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt").to(device)
```

```
model:
  base_learning_rate: 4.5e-06
  params:
    ddconfig:
      attn_resolutions:
      - 16
      ch: 128
      ch_mult:
      - 1
      - 1
      - 2
      - 2
      - 4
      double_z: false
      dropout: 0.0
      in_channels: 3
      num_res_blocks: 2
      out_ch: 3
      resolution: 256
      z_channels: 256
    embed_dim: 256
    lossconfig:
      params:
        codebook_weight: 1.0
        disc_conditional: false
        disc_in_channels: 3
        disc_num_layers: 2
        disc_start: 0
        disc_weight: 0.75
      target: taming.modules.losses.vqperceptual.VQLPIPSWithDiscriminator
    monitor: val/rec_loss
    n_embed: 16384
  target: taming.models.vqgan.VQModel
```

```
Working with z of shape (1, 256, 16, 16) = 65536 dimensions.
Downloading: "https://download.pytorch.org/models/vgg16-397923af.pth" to /root/.cache/torch/hub/checkpoints/vgg16-397923af.pth
100%|██████████| 528M/528M [00:03<00:00, 170MB/s]
Downloading vgg_lpips model from https://heibox.uni-heidelberg.de/f/607503859c864bc1b30b/?dl=1 to taming/modules/autoencoder/lpips/v
8.19kB [00:00, 620kB/s]
loaded pretrained LPIPS loss from taming/modules/autoencoder/lpips/vgg.pth
VQLPIPSWithDiscriminator running with hinge loss.
```

```
### Declare the values that we are going to optimize

class Parameters(torch.nn.Module):
  def __init__(self):
    super(Parameters, self).__init__()
    self.data = .5*torch.randn(batch_size, 256, size1//16, size2//16).cuda() # 1x256x14x15 (225/16, 400/16)
    self.data = torch.nn.Parameter(torch.sin(self.data))

  def forward(self):
    return self.data

def init_params():
  params=Parameters().cuda()
  optimizer = torch.optim.AdamW([{'params':[params.data], 'lr': learning_rate}], weight_decay=wd)
  return params, optimizer


### Encoding prompts and a few more things
normalize = torchvision.transforms.Normalize((0.48145466, 0.4578275, 0.40821073), (0.26862954, 0.26130258, 0.27577711))

def encodeText(text):
  t=clip.tokenize(text).cuda()
  t=clipmodel.encode_text(t).detach().clone()
  return t

def createEncodings(include, exclude, extras):
  include_enc=[]
  for text in include:
    include_enc.append(encodeText(text))
  exclude_enc=encodeText(exclude) if exclude != '' else 0
  extras_enc=encodeText(extras) if extras !='' else 0

  return include_enc, exclude_enc, extras_enc

augTransform = torch.nn.Sequential(
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomAffine(30, (.2, .2), fill=0)
).cuda()

Params, optimizer = init_params()

with torch.no_grad():
  print(Params().shape)
  img= norm_data(generator(Params()).cpu()) # 1 x 3 x 224 x 400 [225 x 400]
  print("img dimensions: ",img.shape)
  show_from_tensor(img[0])
```

```
torch.Size([1, 256, 28, 28])
img dimensions:  torch.Size([1, 3, 448, 448])
```

```
### create crops

def create_crops(img, num_crops=32):
  p=size1//2
  img = torch.nn.functional.pad(img, (p,p,p,p), mode='constant', value=0) # 1 x 3 x 448 x 624 (adding 112*2 on all sides to 224x400)

  img = augTransform(img) #RandomHorizontalFlip and RandomAffine

  crop_set = []
  for ch in range(num_crops):
    gap1= int(torch.normal(1.2, .3, ()).clip(.43, 1.9) * size1)
    offsetx = torch.randint(0, int(size1*2-gap1),())
    offsety = torch.randint(0, int(size1*2-gap1),())

    crop=img[:,:,offsetx:offsetx+gap1, offsety:offsety+gap1]

    crop = torch.nn.functional.interpolate(crop,(224,224), mode='bilinear', align_corners=True)
    crop_set.append(crop)

  img_crops=torch.cat(crop_set,0) ## 30 x 3 x 224 x 224

  randnormal = torch.randn_like(img_crops, requires_grad=False)
  num_rands=0
  randstotal=torch.rand((img_crops.shape[0],1,1,1)).cuda() #32

  for ns in range(num_rands):
    randstotal*=torch.rand((img_crops.shape[0],1,1,1)).cuda()

  img_crops = img_crops + noise_factor*randstotal*randnormal

  return img_crops


### Show current state of generation

def showme(Params, show_crop):
  with torch.no_grad():
    generated = generator(Params())

    if (show_crop):
      print("Augmented cropped example")
      aug_gen = generated.float() # 1 x 3 x 224 x 400
      aug_gen = create_crops(aug_gen, num_crops=1)
      aug_gen_norm = norm_data(aug_gen[0])
      show_from_tensor(aug_gen_norm)

    print("Generation")
    latest_gen=norm_data(generated.cpu()) # 1 x 3 x 224 x 400
    show_from_tensor(latest_gen[0])
```

```
  return (latest_gen[0])


# Optimization process

def optimize_result(Params, prompt):
  alpha=1 ## the importance of the include encodings
  beta=.5 ## the importance of the exclude encodings

  ## image encoding
  out = generator(Params())
  out = norm_data(out)
  out = create_crops(out)
  out = normalize(out) # 30 x 3 x 224 x 224
  image_enc=clipmodel.encode_image(out) ## 30 x 512

  ## text encoding  w1 and w2
  final_enc = w1*prompt + w1*extras_enc # prompt and extras_enc : 1 x 512
  final_text_include_enc = final_enc / final_enc.norm(dim=-1, keepdim=True) # 1 x 512
  final_text_exclude_enc = exclude_enc

  ## calculate the loss
  main_loss = torch.cosine_similarity(final_text_include_enc, image_enc, -1) # 30
  penalize_loss = torch.cosine_similarity(final_text_exclude_enc, image_enc, -1) # 30

  final_loss = -alpha*main_loss + beta*penalize_loss

  return final_loss

def optimize(Params, optimizer, prompt):
  loss = optimize_result(Params, prompt).mean()
  optimizer.zero_grad()
  loss.backward()
  optimizer.step()
  return loss


### training loop

def training_loop(Params, optimizer, show_crop=False):
  res_img=[]
  res_z=[]

  for prompt in include_enc:
    iteration=0
    Params, optimizer = init_params() # 1 x 256 x 14 x 25 (225/16, 400/16)

    for it in range(total_iter):
      loss = optimize(Params, optimizer, prompt)

      if iteration>=80 and iteration%show_step == 0:
        new_img = showme(Params, show_crop)
        res_img.append(new_img)
        res_z.append(Params()) # 1 x 256 x 14 x 25
        print("loss:", loss.item(), "\niteration:",iteration)

      iteration+=1
    torch.cuda.empty_cache()
  return res_img, res_z


torch.cuda.empty_cache()
include=['beautifulll Bird in Sky']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor= .22
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example

Generation

loss: -0.2080078125
iteration: 80
Augmented cropped example

Generation

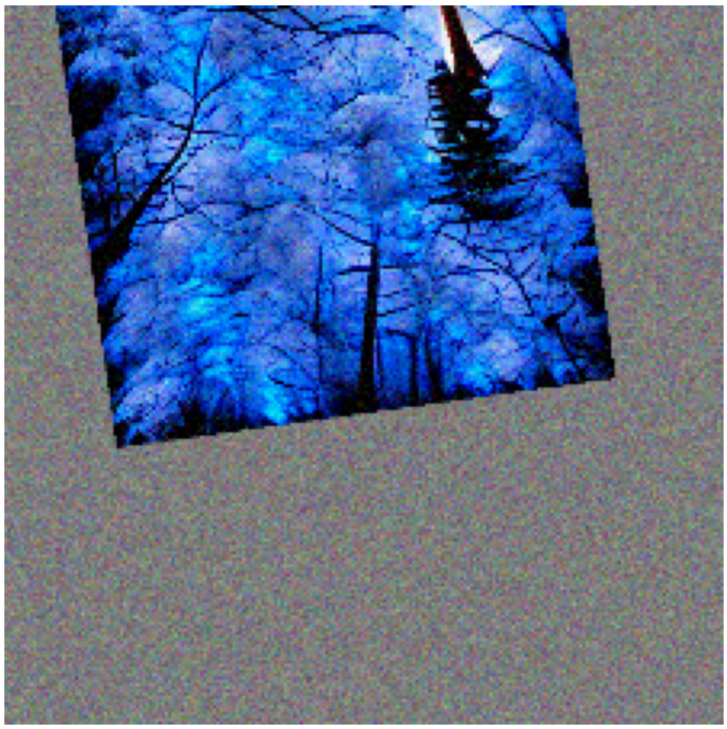
loss: -0.2022705078125
iteration: 90
Augmented cropped example

Generation

```
loss: -0.228759765625
```

```
torch.cuda.empty_cache()
include=['A BLUE TREE IN THE FOREST','KIDS PLAYING IN MOON','FLOWERS DANCING']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor= .22
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example



Generation

loss: -0.2169189453125
iteration: 80
Augmented cropped example

```
Generation
```

```
loss: -0.2320556640625
iteration: 90
Augmented cropped example
```

```
Generation
```

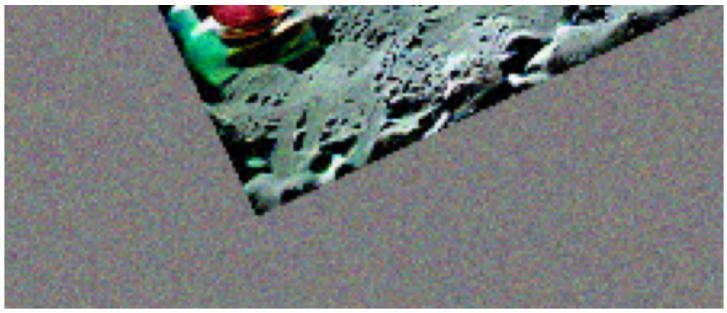
```
loss: -0.2237548828125
iteration: 100
Augmented cropped example
```

```
Generation
```

```
loss: -0.32861328125
iteration: 80
Augmented cropped example
```

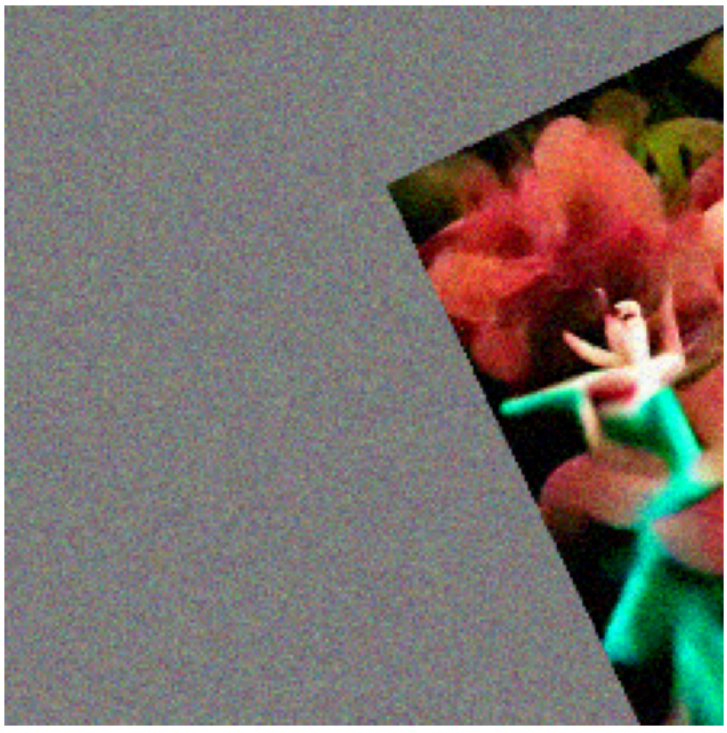
```
Generation
```

```
loss: -0.31298828125
iteration: 90
Augmented cropped example
```

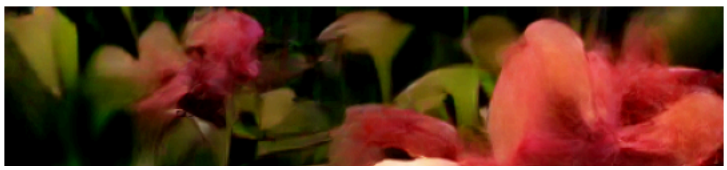


Generation

loss: -0.303466796875
iteration: 100
Augmented cropped example



Generation



```
loss: -0.262939453125
iteration: 80
Augmented cropped example
```