

In [27]:

```
!nvidia-smi  
#!/usr/bin/python3  
#use this code to install the driver
```

Fri May 31 10:46:25 2024

```
+  
| NVIDIA-SMI 535.104.05                 Driver Version: 535.104.05     CUDA Version: 12.2  
|  
+-----+-----+  
| GPU  Name                  Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC  
|  
| Fan  Temp     Perf      Pwr:Usage/Cap |          Memory-Usage | GPU-Util  Compute M.  
|  
|          |          |          |          |          |          |          MIG M.  
+-----+-----+  
| 0  Tesla T4                Off  | 00000000:00:04.0 Off  |  
| N/A   76C     P0      34W /  70W |  1781MiB / 15360MiB | 0%       Default  
|  
|          |          |          |          |          |          |          N/A  
+-----+-----+  
  
+-----+  
+  
| Processes:  
|  
| GPU  GI  CI      PID  Type  Process name          GPU Memory  
|  
|          ID  ID  
|  
+-----+-----+  
|  
|  
+-----+
```

Clip Architecture

In [28]:

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

```
Cloning into 'CLIP'...
remote: Enumerating objects: 251, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 251 (delta 3), reused 3 (delta 0), pack-reused 243
Receiving objects: 100% (251/251), 8.93 MiB | 23.56 MiB/s, done.
Resolving deltas: 100% (127/127), done.
```

TAMING-TRANSFORMER ARCHITECTURE

Tn [29]:

```
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  
Receiving objects: 100% (1342/1342), 409.77 MiB | 24.68 MiB/s, done.  
Resolving deltas: 100% (282/282), done.
```

```
In [30]: # We Need to install some more libraries as well
```

```
!pip install --no-deps ftfy regex tqdm  
!pip install omegaconf==2.0.0 pytorch-lightning==1.0.8  
!pip uninstall torchtext --yes  
!pip install einops
```

```
Requirement already satisfied: ftfy in /usr/local/lib/python3.10/dist-packages (6.2.0)  
Requirement already satisfied: regex in /usr/local/lib/python3.10/dist-packages (2023.1  
2.25)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.2)  
Requirement already satisfied: omegaconf==2.0.0 in /usr/local/lib/python3.10/dist-packages  
(2.0.0)  
Requirement already satisfied: pytorch-lightning==1.0.8 in /usr/local/lib/python3.10/dist-  
packages (1.0.8)  
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from o  
megaconf==2.0.0) (6.0.1)  
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-  
packages (from omegaconf==2.0.0) (4.11.0)  
Requirement already satisfied: numpy>=1.16.4 in /usr/local/lib/python3.10/dist-packages  
(from pytorch-lightning==1.0.8) (1.25.2)  
Requirement already satisfied: torch>=1.3 in /usr/local/lib/python3.10/dist-packages (fr  
om pytorch-lightning==1.0.8) (2.2.1+cu121)  
Requirement already satisfied: future>=0.17.1 in /usr/local/lib/python3.10/dist-packages  
(from pytorch-lightning==1.0.8) (0.18.3)  
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages  
(from pytorch-lightning==1.0.8) (4.66.2)  
Requirement already satisfied: fsspec>=0.8.0 in /usr/local/lib/python3.10/dist-packages  
(from pytorch-lightning==1.0.8) (2023.6.0)  
Requirement already satisfied: tensorboard>=2.2.0 in /usr/local/lib/python3.10/dist-  
packages (from pytorch-lightning==1.0.8) (2.15.2)  
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.10/dist-packages  
(from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (1.4.0)  
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.10/dist-packages  
(from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (1.63.0)  
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-p  
ackages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (2.27.0)  
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.1  
0/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (1.2.0)  
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-  
packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.6)  
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.10/di  
st-packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.20.3)  
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-p  
ackages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (2.31.0)  
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.10/dist-  
packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (67.7.2)  
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.10/dist-packages (from  
tensorboard>=2.2.0->pytorch-lightning==1.0.8) (1.16.0)  
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/p  
ython3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (0.7.2)  
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-  
packages (from tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.0.2)  
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from  
torch>=1.3->pytorch-lightning==1.0.8) (3.14.0)  
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from to  
rch>=1.3->pytorch-lightning==1.0.8) (1.12)  
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from  
torch>=1.3->pytorch-lightning==1.0.8) (3.3)  
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from t  
orch>=1.3->pytorch-lightning==1.0.8) (3.1.3)
```

```
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (8.9.2.26)
Requirement already satisfied: nvidia-cUBLAS-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.19.3 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (2.19.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (12.1.105)
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (2.2.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.3->pytorch-lightning==1.0.8) (12.5.40)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<2,>=0.5->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (2024.2.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.3->pytorch-lightning==1.0.8) (1.3.0)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5->tensorboard>=2.2.0->pytorch-lightning==1.0.8) (3.2.2)
WARNING: Skipping torchtext as it is not installed.
Requirement already satisfied: einops in /usr/local/lib/python3.10/dist-packages (0.8.0)
```

In [31]:

```
#import IMAGE, NUMPY, PANDAS, MATPLOTLIB Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import PIL
```

In [32]:

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

In [33]:

```
import yaml
from omegaconf import OmegaConf

from CLIP import clip

import warnings
warnings.filterwarnings('ignore')
```

In [34]:

```
#create helper function for CLIP

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
```

In [35]:

```
##CLIP MODEL
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()

['RN50', 'RN101', 'RN50x4', 'RN50x16', 'RN50x64', 'ViT-B/32', 'ViT-B/16', 'ViT-L/14', 'ViT-L/14@336px']
Clip model visual input resolution: 224
```

In [36]:

```
## 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/vqgan_imagenet_f16_16384.pt'
    !wget 'https://heibox.uni-heidelberg.de/f/274fb24ed38341bfa753/?dl=1' -O 'models/vqgan_imagenet_f16_16384/configs/vqgan_imagenet_f16_16384.yaml'
```

```

/content/taming-transformers/taming-transformers
--2024-05-31 10:47:05-- 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/03201b7b-948f-4884-bf3c-07a100
b2c4bf/last.ckpt [following]
--2024-05-31 10:47:05-- https://heibox.uni-heidelberg.de/seafhttp/files/03201b7b-948f-4
884-bf3c-07a100b2c4bf/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 9.76MB/s in 1m 44s

2024-05-31 10:48:49 (9.02 MB/s) - 'models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt' saved [980092370/980092370]

--2024-05-31 10:48:49-- 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/313d3304-3350-4956-a66b-4d2efb
1d240a/model.yaml [following]
--2024-05-31 10:48:49-- https://heibox.uni-heidelberg.de/seafhttp/files/313d3304-3350-4
956-a66b-4d2efb1d240a/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

2024-05-31 10:48:49 (825 MB/s) - 'models/vqgan_imagenet_f16_16384/configs/model.yaml' sa
ved [692/692]

```

In [37]:

```

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", disp
taming_model = load_vqgan(taming_config, chk_path="./models/vqgan_imagenet_f16_16384/che

#ERROR -> Module_not found (torch-six) || FIX -> click the link above above error messag
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 vgg_lpips model from <https://heibox.uni-heidelberg.de/f/607503859c864bc1b30b/?dl=1> to taming/modules/autoencoder/lpips/vgg.pth
 8.19kB [00:00, 38.8kB/s]
 loaded pretrained LPIPS loss from taming/modules/autoencoder/lpips/vgg.pth
 VQLPIPSWithDiscriminator running with hinge loss.

In [38]: *### 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()
    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_d
  return params, optimizer

```

In [39]: *### Encoding prompts and a few more things*

```

normalize = torchvision.transforms.Normalize((0.48145466, 0.4578275, 0.40821073), (0.268
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])



In [40]: *### 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
    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_corner)
    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

```

In [41]: *### 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])

```

In [42]: *# 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

```

In [43]: *### 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

```

In [44]:

```

torch.cuda.empty_cache()
include=['A LAMP IN THE ROOM']
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: -0.23876953125

iteration: 80

Augmented cropped example



Generation



loss: -0.2364501953125

iteration: 90

Augmented cropped example



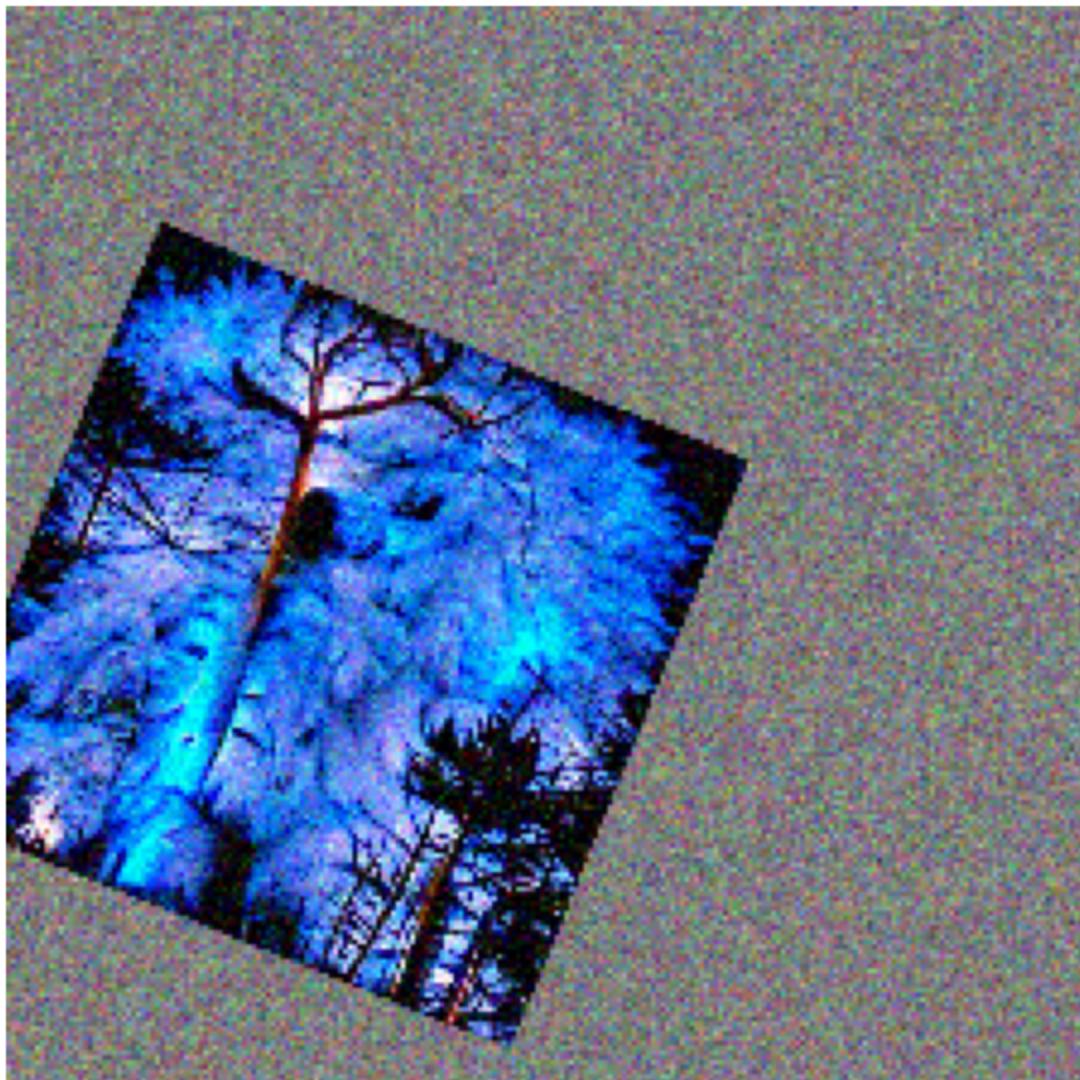
Generation



```
loss: -0.238525390625
iteration: 100
```

```
In [45]: torch.cuda.empty_cache()
include=["A Blue tree in the forest"]
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



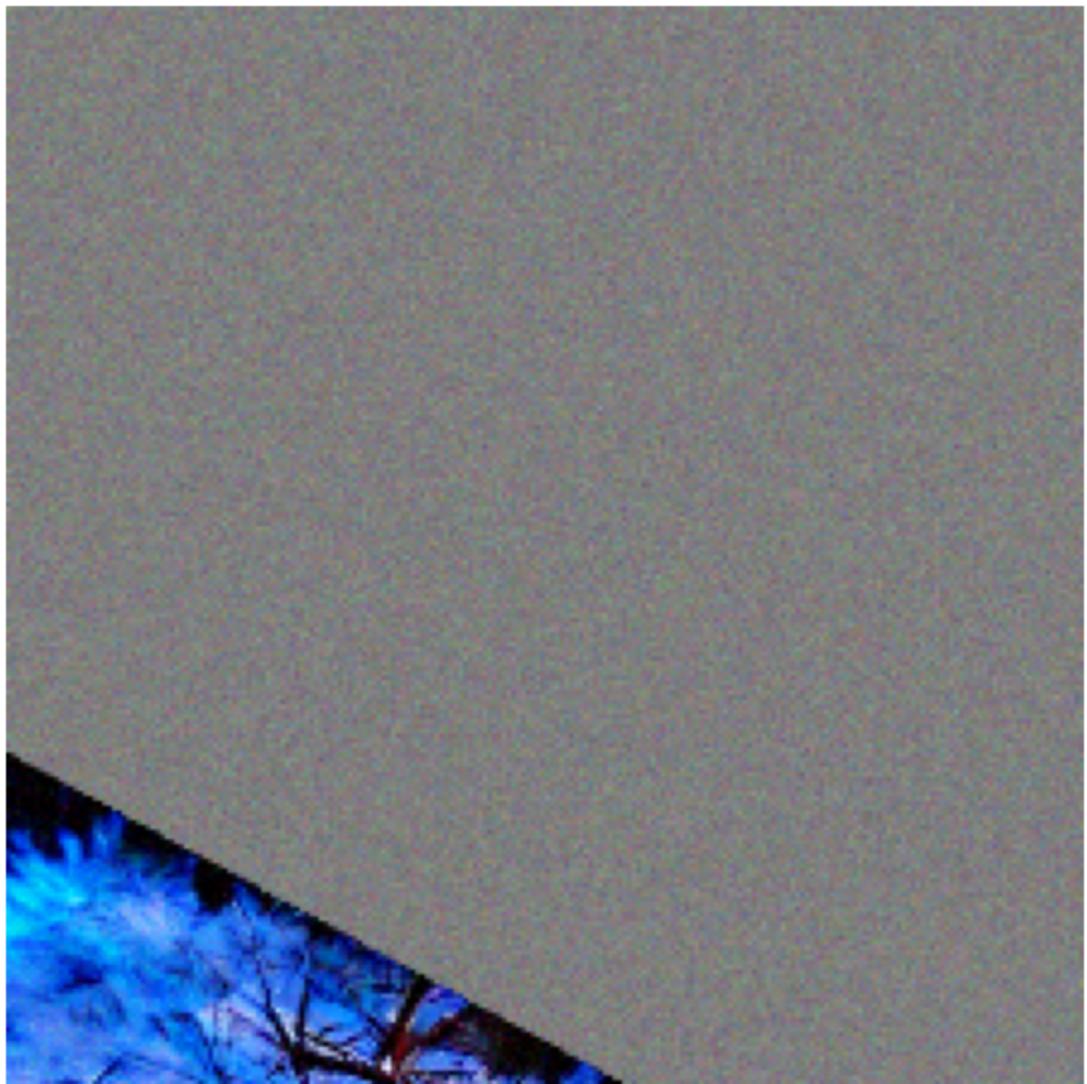
Generation



loss: -0.223388671875

iteration: 80

Augmented cropped example



Generation



loss: -0.2176513671875

iteration: 90

Augmented cropped example



Generation



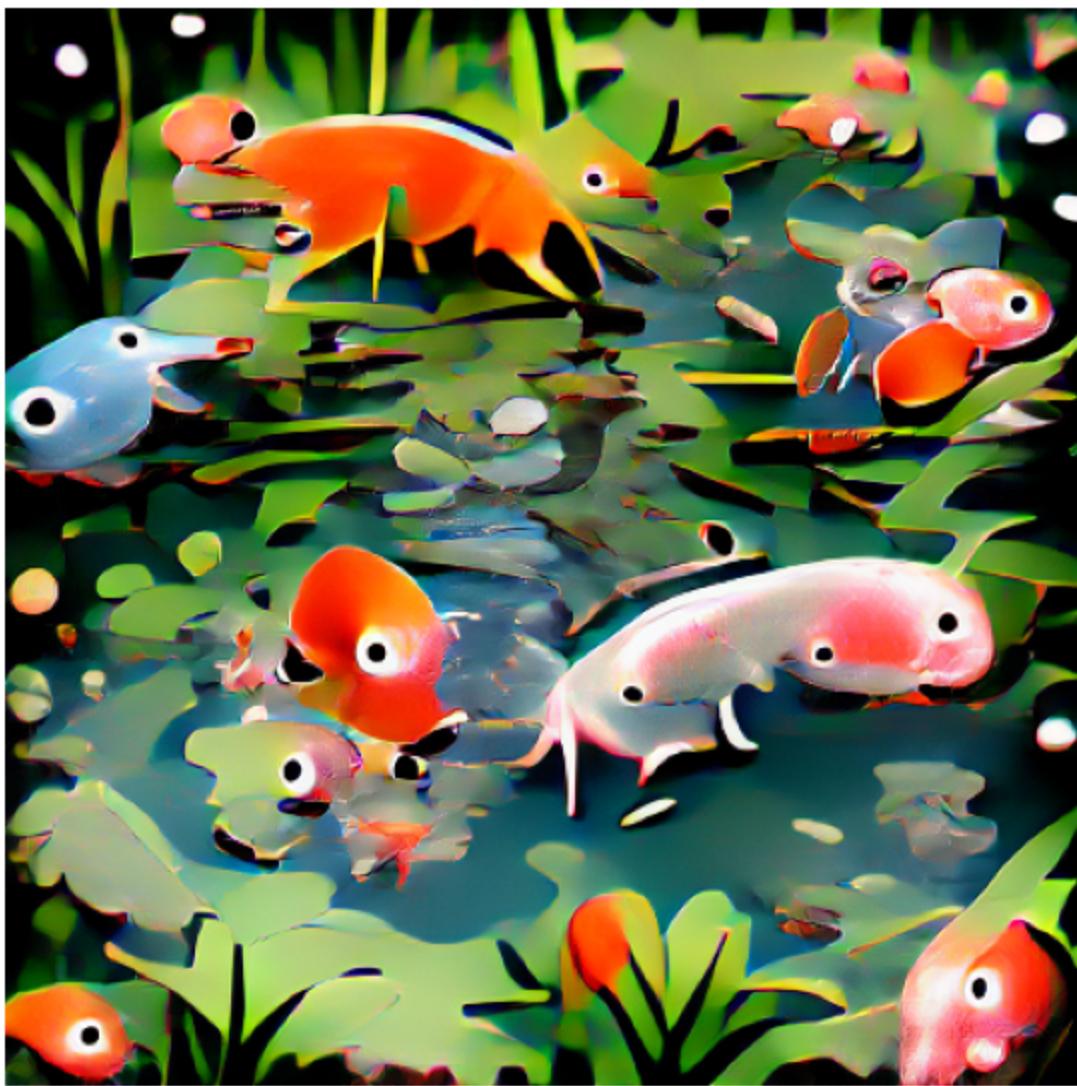
```
loss: -0.236083984375
iteration: 100
```

```
In [46]: torch.cuda.empty_cache()
include=["Small fishes in the pond"]
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



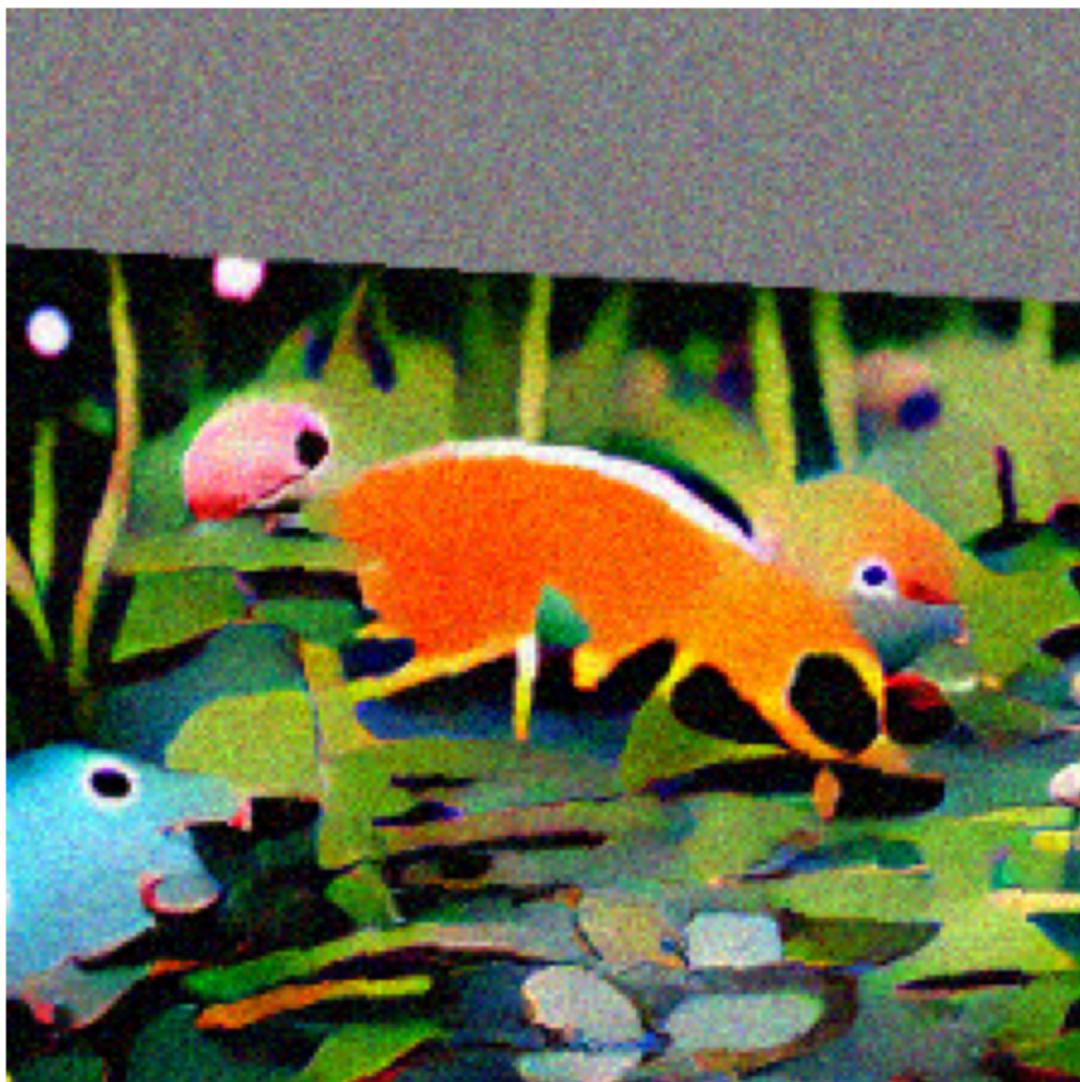
Generation



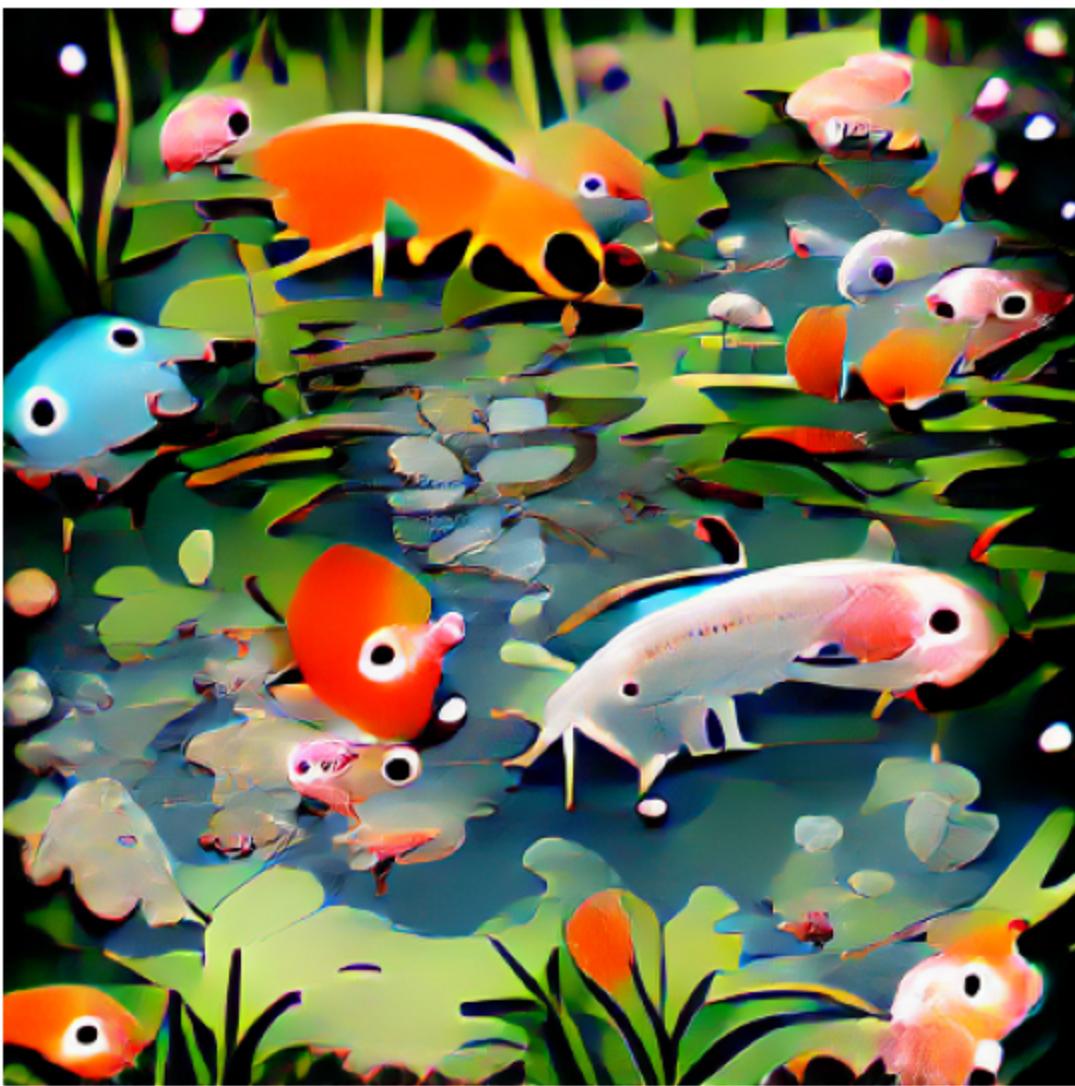
loss: -0.256103515625

iteration: 80

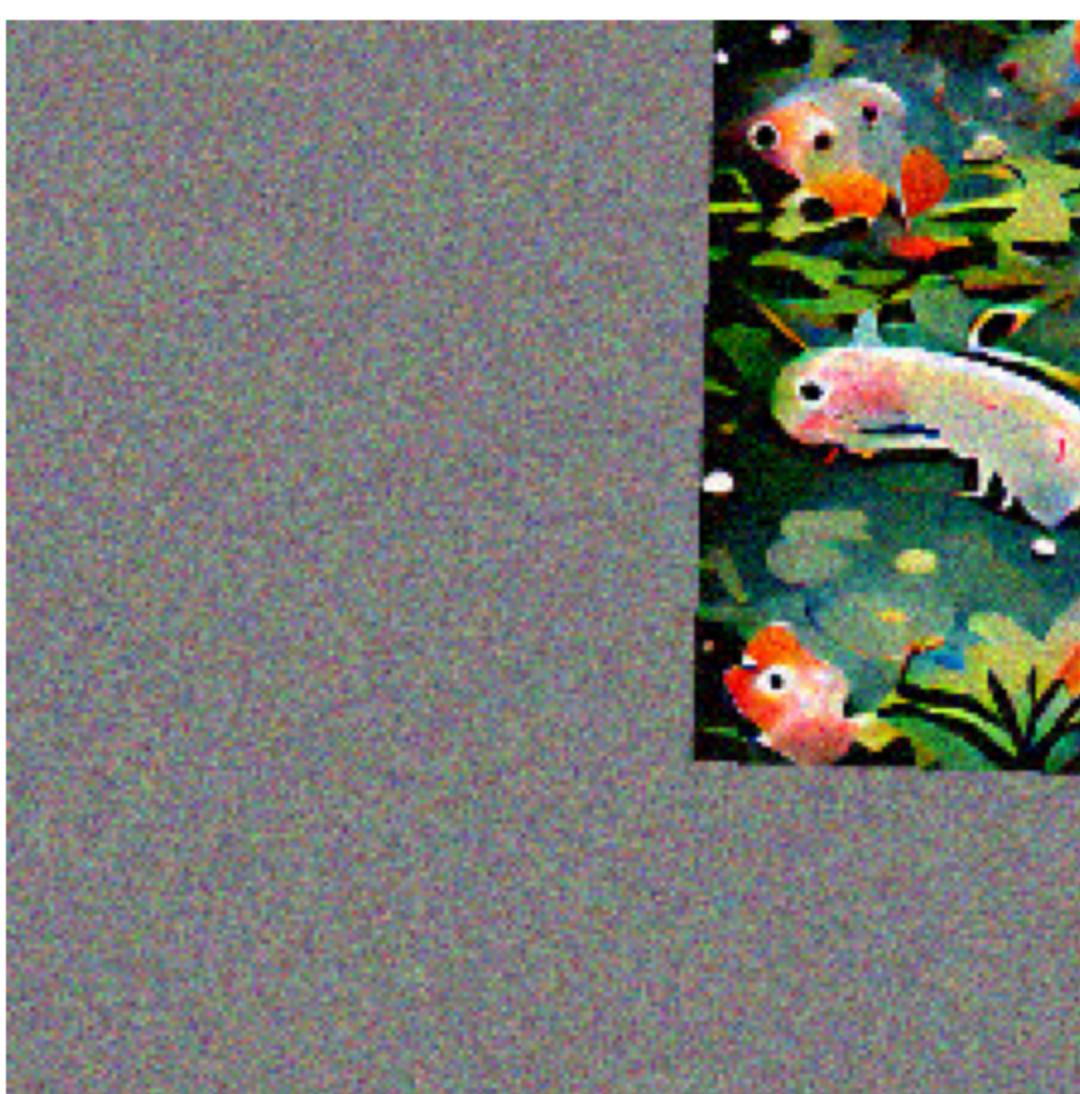
Augmented cropped example



Generation



loss: -0.26611328125
iteration: 90
Augmented cropped example



Generation



loss: -0.2431640625
iteration: 100

In [46]: