StyleGAN2-ADA-PyTorch

Notes

- Training and Inference sections should be fairly stable. I'll slowly add new features but it should work for most mainstream use cases.
- Advanced Features are being documented toward the bottom of this notebook

---

If you find this notebook useful, consider signing up for my Patreon or YouTube channel. You can also send me a one-time payment on Venmo.

Setup

Let's start by checking to see what GPU we've been assigned. Ideally we get a V100, but a P100 is fine too. Other GPUs may lead to issues.

```
!nvidia-smi -L
```

```
GPU 0: Tesla T4 (UUID: GPU-c8a42fb2-11b9-7d51-c9dc-8149abfb0604)
```

Next let's connect our Google Drive account. This is optional but highly recommended.

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

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

Install repo

The next cell will install the StyleGAN repository in Google Drive. If you have already installed it it will just move into that folder. If you don't have Google Drive connected it will just install the necessary code in Colab.

```
import os
!pip install gdown --upgrade
```

reals.jp

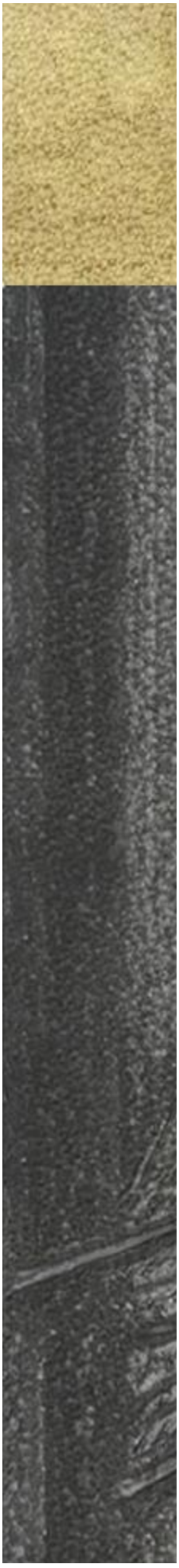
```
if os.path.isdir("/content/drive/MyDrive/colab-sg2-ada-pytorch"):
    %cd "/content/drive/MyDrive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch"
elif os.path.isdir("/content/drive/"):
    #install script
    %cd "/content/drive/MyDrive/"
    !mkdir colab-sg2-ada-pytorch
    %cd colab-sg2-ada-pytorch
    !git clone https://github.com/dvschultz/stylegan2-ada-pytorch
    %cd stylegan2-ada-pytorch
    !mkdir downloads
    !mkdir datasets
    !mkdir pretrained
    !gdown --id 1-5xZkD8ajXw1DdopTkH_rAoCsD72LhKU -O /content/drive/MyDrive/c
else:
    !git clone https://github.com/dvschultz/stylegan2-ada-pytorch
    %cd stylegan2-ada-pytorch
    !mkdir downloads
    !mkdir datasets
    !mkdir pretrained
    %cd pretrained
    !gdown --id 1-5xZkD8ajXw1DdopTkH_rAoCsD72LhKU
    %cd ../
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/c
Requirement already satisfied: gdown in /usr/local/lib/python3.7/dist-pa
Collecting gdown
  Downloading gdown-4.5.3.tar.gz (14 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: requests[socks] in /usr/local/lib/python3
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 i
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/pytho
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/pyth
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/
Building wheels for collected packages: gdown
  Building wheel for gdown (PEP 517) ... done
  Created wheel for gdown: filename=gdown-4.5.3-py3-none-any.whl size=14
  Stored in directory: /root/.cache/pip/wheels/94/8d/0b/bdcd83555c3555f9
Successfully built gdown
Installing collected packages: gdown
  Attempting uninstall: gdown
    Found existing installation: gdown 4.4.0
    Uninstalling gdown-4.4.0:
      Successfully uninstalled gdown-4.4.0
Successfully installed gdown-4.5.3
/content/drive/MyDrive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch
```

```
#Uninstall new JAX
!pip uninstall jax jaxlib -y
#GPU frontend
!pip install "jax[cuda11_cudnn805]==0.3.10" -f https://storage.googleapis.com
#CPU frontend
#!pip install jax[cpu]==0.3.10
#Downgrade Pytorch
!pip uninstall torch torchvision -y
!pip install torch==1.9.0+cu111 torchvision==0.10.0+cu111 -f https://download
!pip install timm==0.4.12 ftfy==6.1.1 ninja==1.10.2 opensimplex
```

```
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /usr/local/l
Building wheels for collected packages: jax
  Building wheel for jax (setup.py) ... done
  Created wheel for jax: filename=jax-0.3.10-py3-none-any.whl size=10
  Stored in directory: /root/.cache/pip/wheels/1c/05/23/36730377cd731
Successfully built jax
Installing collected packages: jaxlib, jax
Successfully installed jax-0.3.10 jaxlib-0.3.10+cuda11.cudnn805
Found existing installation: torch 1.12.1+cu113
Uninstalling torch-1.12.1+cu113:
  Successfully uninstalled torch-1.12.1+cu113
Found existing installation: torchvision 0.13.1+cu113
Uninstalling torchvision-0.13.1+cu113:
  Successfully uninstalled torchvision-0.13.1+cu113
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.de
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.9.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torch-1.9.0%2Bcu
     |████████████                    | 834.1 MB 1.3 MB/s eta 0:15:21
     |████████████                    | 1055.7 MB 1.3 MB/s eta 0:12:5
     |████████████                    | 1336.2 MB 1.2 MB/s eta 0:09:4
     |████████████                    | 1691.1 MB 1.2 MB/s eta 0:04:4
     |████████████                    | 2041.3 MB 1.2 MB/s eta 0:00:0
tcmalloc: large alloc 2551685120 bytes == 0x1e1486000 @  0x7f0bc243d6

     |████████████                    | 2041.3 MB 7.1 kB/s
Collecting torchvision==0.10.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torchvision-0.10
     |████████████                    | 23.2 MB 103.3 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/py
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist
Installing collected packages: torch, torchvision
ERROR: pip's dependency resolver does not currently take into account
torchtext 0.13.1 requires torch==1.12.1, but you have torch 1.9.0+cu1
torchaudio 0.12.1+cu113 requires torch==1.12.1, but you have torch 1.
Successfully installed torch-1.9.0+cu111 torchvision-0.10.0+cu111
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.de
Collecting timm==0.4.12
  Downloading timm-0.4.12-py3-none-any.whl (376 kB)
     |████████████                    | 376 kB 5.0 MB/s
Collecting ftfy==6.1.1
  Downloading ftfy-6.1.1-py3-none-any.whl (53 kB)
     |████████████                    | 53 kB 1.6 MB/s
```

```
Collecting ninja==1.10.2
  Downloading ninja-1.10.2-py2.py3-none-manylinux_2_5_x86_64.manylinu
     |████████████████████████████████| 108 kB 70.9 MB/s
Collecting opensimplex
  Downloading opensimplex-0.4.2-py3-none-any.whl (17 kB)
Requirement already satisfied: torch>=1.4 in /usr/local/lib/python3.7
Requirement already satisfied: torchvision in /usr/local/lib/python3.
Requirement already satisfied: wcwidth>=0.2.5 in /usr/local/lib/pytho
Requirement already satisfied: typing-extensions in /usr/local/lib/py
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.
Requirement already satisfied: pillow>=5.3.0 in /usr/local/lib/python
Installing collected packages: timm, opensimplex, ninja, ftfy
Successfully installed ftfy-6.1.1 ninja-1.10.2 opensimplex-0.4.2 timm
```



You probably don't need to run this, but this will update your repo to the latest and greatest.

```
%cd "/content/drive/My Drive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch"
!git config --global user.name "test"
!git config --global user.email "test@test.com"
!git fetch origin
!git pull
!git stash
!git checkout origin/main -- train.py generate.py legacy.py closed_form_facto
```

```
/content/drive/My Drive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch
Already up to date.
Saved working directory and index state WIP on main: 59e05bb added opens
```

Dataset Preparation

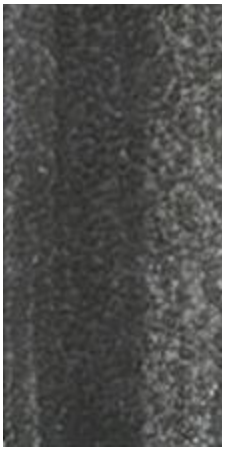
Upload a .zip of square images to the `datasets` folder. Previously you had to convert your model to .tfrecords. That's no longer needed :)

Train model

Below are a series of variables you need to set to run the training. You probably won't need to touch most of them.

- `dataset_path` : this is the path to your .zip file
- `resume_from` : if you're starting a new dataset I recommend `'ffhq1024'` or `'./pretrained/wikiart.pkl'`

- `mirror_x` and `mirror_y`: Allow the dataset to use horizontal or vertical mirroring.



```
#required: definitely edit these!
dataset_path = '/content/drive/MyDrive/stylegan_x1/datasets/landscape.zip'
resume_from = './pretrained/wikiart.pkl'
aug_strength = 0.0
train_count = 0
mirror_x = True
mirror_y = False

#optional: you might not need to edit these
gamma_value = 50.0
augs = 'bg'
config = '11gb-gpu'
snapshot_count = 4


!python train.py --gpus=1 --cfg=$config --metrics=None --outdir=./results --d
```

```
Training options:
{
  "num_gpus": 1,
  "image_snapshot_ticks": 4,
  "network_snapshot_ticks": 4,
  "metrics": [],
  "random_seed": 0,
  "training_set_kwargs": {
    "class_name": "training.dataset.ImageFolderDataset",
    "path": "/content/drive/MyDrive/stylegan_x1/datasets/landscape.zi
    "use_labels": false,
    "max_size": 10,
    "xflip": true,
    "resolution": 1024
  },
  "data_loader_kwargs": {
    "pin_memory": true,
    "num_workers": 3,
    "prefetch_factor": 2
  },
  "G_kwargs": {
    "class_name": "training.networks.Generator",
    "z_dim": 512,
    "w_dim": 512,
    "mapping_kwargs": {
      "num_layers": 8
    },
    "synthesis_kwargs": {
      "channel_base": 32768,
      "channel_max": 512,
      "num_fp16_res": 4,
```

```
    "conv_clamp": 256
  }
},
"D_kwargs": {
  "class_name": "training.networks.Discriminator",
  "block_kwargs": {},
  "mapping_kwargs": {},
  "epilogue_kwargs": {
    "mbstd_group_size": 4
  },
  "channel_base": 32768,
  "channel_max": 512,
  "num_fp16_res": 4,
  "conv_clamp": 256
},
"G_opt_kwargs": {
  "class_name": "torch.optim.Adam",
  "lr": 0.002,
  "betas": [
    0,
    0.99
  ],
  "eps": 1e-08
},
```

Resume Training

Once Colab has shutdown, you'll need to resume your training. Reset the variables above, particularly the `resume_from` and `aug_strength` settings.

1. Point `resume_from` to the last .pkl you trained (you'll find these in the `results` folder)
2. Update `aug_strength` to match the augment value of the last pkl file. Often you'll see this in the console, but you may need to look at the `log.txt`. Updating this makes sure training stays as stable as possible.
3. You may want to update `train_count` to keep track of your training progress.

Once all of this has been reset, run that variable cell and the training command cell after it.

Testing/Inference

Also known as "Inference", "Evaluation" or "Testing" the model. This is the process of usinng your trained model to generate new material, usually images or videos.

Generate Single Images

`--network` : Make sure the `--network` argument points to your .pkl file. (My preferred method is to right click on the file in the Files pane to your left and choose `Copy Path`, then paste that into the argument after the `=` sign).

`--seeds` : This allows you to choose random seeds from the model. Remember that our input to StyleGAN is a 512-dimensional array. These seeds will generate those 512 values. Each seed will generate a different, random array. The same seed value will also always generate the same random array, so we can later use it for other purposes like interpolation.

`--truncation` : Truncation, well, truncates the latent space. This can have a subtle or dramatic affect on your images depending on the value you use. The smaller the number the more realistic your images should appear, but this will also affect diversity. Most people choose between 0.5 and 1.0, but technically it's infinite.

```
!python generate.py --outdir=/content/out/images/ --trunc=0.8 --seeds=0 --net
```

```
Loading networks from "/content/drive/MyDrive/colab-sg2-ada-pytorch/styl
Generating image for seed 0 (0/1) ...
Setting up PyTorch plugin "bias_act_plugin"... Done.
Setting up PyTorch plugin "upfirdn2d_plugin"... Done.
```

Non-Square outputs

We can modify the model to output images that are not square. This isn't as good as training a rectangular model, but with the right model it can still look nice.

- `--size` size takes in a value of `xdim-ydim`. For example, to generate a 1920x1080 image use `1920-1080`
- `--scale-type` This determines the padding style to apply in the additional space. There are four options: `pad`, `padside`, `symm`, and `symmside`. I recommend trying each one to see what works best with your images.

```
!python generate.py --outdir=/content/out/images/ --trunc=0.7 --size=1820-102
```

```
Generating image for seed 211 (211/500) ...
Generating image for seed 212 (212/500) ...
Generating image for seed 213 (213/500) ...
Generating image for seed 214 (214/500) ...
Generating image for seed 215 (215/500) ...
Generating image for seed 216 (216/500) ...
Generating image for seed 217 (217/500) ...
Generating image for seed 218 (218/500) ...
```

```
Generating image for seed 219 (219/500) ...
Generating image for seed 220 (220/500) ...
Generating image for seed 221 (221/500) ...
Generating image for seed 222 (222/500) ...
Generating image for seed 223 (223/500) ...
Generating image for seed 224 (224/500) ...
Generating image for seed 225 (225/500) ...
Generating image for seed 226 (226/500) ...
Generating image for seed 227 (227/500) ...
Generating image for seed 228 (228/500) ...
Generating image for seed 229 (229/500) ...
Generating image for seed 230 (230/500) ...
Generating image for seed 231 (231/500) ...
Generating image for seed 232 (232/500) ...
Generating image for seed 233 (233/500) ...
Generating image for seed 234 (234/500) ...
Generating image for seed 235 (235/500) ...
Generating image for seed 236 (236/500) ...
Generating image for seed 237 (237/500) ...
Generating image for seed 238 (238/500) ...
Generating image for seed 239 (239/500) ...
Generating image for seed 240 (240/500) ...
Generating image for seed 241 (241/500) ...
Generating image for seed 242 (242/500) ...
Generating image for seed 243 (243/500) ...
Generating image for seed 244 (244/500) ...
Generating image for seed 245 (245/500) ...
Generating image for seed 246 (246/500) ...
Generating image for seed 247 (247/500) ...
Generating image for seed 248 (248/500) ...
Generating image for seed 249 (249/500) ...
Generating image for seed 250 (250/500) ...
Generating image for seed 251 (251/500) ...
Generating image for seed 252 (252/500) ...
Generating image for seed 253 (253/500) ...
Generating image for seed 254 (254/500) ...
Generating image for seed 255 (255/500) ...
Generating image for seed 256 (256/500) ...
Generating image for seed 257 (257/500) ...
Generating image for seed 258 (258/500) ...
Generating image for seed 259 (259/500) ...
Generating image for seed 260 (260/500) ...
Generating image for seed 261 (261/500) ...
Generating image for seed 262 (262/500) ...
Generating image for seed 263 (263/500) ...
Generating image for seed 264 (264/500) ...
Generating image for seed 265 (265/500) ...
Generating image for seed 266 (266/500) ...
Generating image for seed 267 (267/500) ...
```

We can use these options for any image or video generation commands (excluding projection).

Truncation Traversal

Below you can take one seed and look at the changes to it across any truncation amount. -1 to 1 will be pretty realistic images, but the further out you get the weirder it gets.

Options

`--network` : Again, this should be the path to your .pkl file.

`--seeds` : Pass this only one seed. Pick a favorite from your generated images.

`--start` : Starting truncation value.

`--stop` : Stopping truncation value. This should be larger than the start value. (Will probably break if its not).

`--increment` : How much each frame should increment the truncation value. Make this really small if you want a long, slow interpolation. (stop-start/increment=total frames)

```
!python generate.py --process="truncation" --outdir=/content/out/trunc-trav-3
```

```
Generating truncation 0.76
Generating truncation 0.78
Generating truncation 0.80
Generating truncation 0.82
Generating truncation 0.84
Generating truncation 0.86
Generating truncation 0.88
Generating truncation 0.90
Generating truncation 0.92
Generating truncation 0.94
Generating truncation 0.96
Generating truncation 0.98
Generating truncation 1.00
Generating truncation 1.02
Generating truncation 1.04
Generating truncation 1.06
Generating truncation 1.08
Generating truncation 1.10
Generating truncation 1.12
Generating truncation 1.14
Generating truncation 1.16
Generating truncation 1.18
Generating truncation 1.20
Generating truncation 1.22
Generating truncation 1.24
Generating truncation 1.26
Generating truncation 1.28
Generating truncation 1.30
Generating truncation 1.32
```

```
Generating truncation 1.34
Generating truncation 1.36
Generating truncation 1.38
Generating truncation 1.40
Generating truncation 1.42
Generating truncation 1.44
Generating truncation 1.46
Generating truncation 1.48
Generating truncation 1.50
Generating truncation 1.52
Generating truncation 1.54
Generating truncation 1.56
Generating truncation 1.58
Generating truncation 1.60
Generating truncation 1.62
Generating truncation 1.64
Generating truncation 1.66
Generating truncation 1.68
Generating truncation 1.70
Generating truncation 1.72
Generating truncation 1.74
Generating truncation 1.76
Generating truncation 1.78
Generating truncation 1.80
Generating truncation 1.82
Generating truncation 1.84
Generating truncation 1.86
Generating truncation 1.88
```

Interpolations

Interpolation is the process of generating very small changes to a vector in order to make it appear animated from frame to frame.

We'll look at different examples of interpolation below.

Options

`--network` : path to your .pkl file

`--interpolation` : Walk type defines the type of interpolation you want. In some cases it can also specify whether you want the z space or the w space.

`--frames` : How many frames you want to produce. Use this to manage the length of your video.

`--trunc` : truncation value

Linear Interpolation

```
!python generate.py --outdir=/content/out/video1-w-0.5/ --space="z" --trunc=0
```

```
Generating image for frame 228/240 ...
Generating image for frame 229/240 ...
Generating image for frame 230/240 ...
Generating image for frame 231/240 ...
Generating image for frame 232/240 ...
Generating image for frame 233/240 ...
Generating image for frame 234/240 ...
Generating image for frame 235/240 ...
Generating image for frame 236/240 ...
Generating image for frame 237/240 ...
Generating image for frame 238/240 ...
Generating image for frame 239/240 ...
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg d
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, image2, from '/content/out/video1-w-0.5/frames/frame%04d.pn
  Duration: 00:00:09.60, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: png, rgb24(pc), 1024x1024, 25 fps, 25 tbr, 25
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x55abeff63e00] using cpu capabilities: MMX2 SSE2Fast SSSE
[libx264 @ 0x55abeff63e00] profile High, level 3.2
[libx264 @ 0x55abeff63e00] 264 - core 152 r2854 e9a5903 - H.264/MPEG-
Output #0, mp4, to '/content/out/video1-w-0.5//interpolation-linear-s
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661), yuv420p,
    Metadata:
      encoder         : Lavc57.107.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=  240 fps= 13 q=-1.0 Lsize=    7946kB time=00:00:09.87 bitrate=
video:7944kB audio:0kB subtitle:0kB other streams:0kB global headers:
[libx264 @ 0x55abeff63e00] frame I:1     Avg QP:22.52  size:287043
[libx264 @ 0x55abeff63e00] frame P:239   Avg QP:23.29  size: 32833
[libx264 @ 0x55abeff63e00] mb I  I16..4:  3.0% 60.1% 36.9%
[libx264 @ 0x55abeff63e00] mb P  I16..4:  0.0%  0.0%  0.0%  P16..4: 5

[libx264 @ 0x55abeff63e00] 8x8 transform intra:60.8% inter:50.3%
[libx264 @ 0x55abeff63e00] coded y,uvDC,uvAC intra: 100.0% 80.6% 37.4
[libx264 @ 0x55abeff63e00] i16 v,h,dc,p:  0%  2% 74% 24%
[libx264 @ 0x55abeff63e00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 10% 12% 27%
[libx264 @ 0x55abeff63e00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu:  8% 13% 16%
[libx264 @ 0x55abeff63e00] i8c dc,h,v,p: 52% 20% 19%  8%
[libx264 @ 0x55abeff63e00] Weighted P-Frames: Y:0.0% UV:0.0%
```

```
[libx264 @ 0x55abeff63e00] ref P L0: 76.1% 23.3%  0.5%  0.0%
[libx264 @ 0x55abeff63e00] kb/s:6507.36
```

```
!python generate.py --outdir=out/video1-w/ --space="w" --trunc=1 --process="i
```

```
Generating image for frame 712/720 ...
Generating image for frame 713/720 ...
Generating image for frame 714/720 ...
Generating image for frame 715/720 ...
Generating image for frame 716/720 ...
Generating image for frame 717/720 ...
Generating image for frame 718/720 ...
Generating image for frame 719/720 ...
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg d
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, image2, from 'out/video1-w/frames/frame%04d.png':
  Duration: 00:00:28.80, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: png, rgb24(pc), 1024x1024, 25 fps, 25 tbr, 25
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x5641c0d9fe00] using cpu capabilities: MMX2 SSE2Fast SSSE
[libx264 @ 0x5641c0d9fe00] profile High, level 3.2
[libx264 @ 0x5641c0d9fe00] 264 - core 152 r2854 e9a5903 - H.264/MPEG-
Output #0, mp4, to 'out/video1-w//interpolation-linear-seeds_85_265_2
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661), yuv420p,
    Metadata:
      encoder         : Lavc57.107.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=  720 fps= 12 q=-1.0 Lsize=   31115kB time=00:00:29.87 bitrate=
video:31111kB audio:0kB subtitle:0kB other streams:0kB global headers
[libx264 @ 0x5641c0d9fe00] frame I:3     Avg QP:21.01  size:292381
[libx264 @ 0x5641c0d9fe00] frame P:708   Avg QP:23.35  size: 43706
[libx264 @ 0x5641c0d9fe00] frame B:9     Avg QP:28.21  size:  3959
[libx264 @ 0x5641c0d9fe00] consecutive B-frames: 97.9%  0.8%  1.2%  0
[libx264 @ 0x5641c0d9fe00] mb I  I16..4:  1.1% 65.4% 33.5%
[libx264 @ 0x5641c0d9fe00] mb P  I16..4:  0.0%  0.1%  0.0%  P16..4: 5
[libx264 @ 0x5641c0d9fe00] mb B  I16..4:  0.0%  0.0%  0.0%  B16..8: 2
[libx264 @ 0x5641c0d9fe00] 8x8 transform intra:69.6% inter:51.4%
[libx264 @ 0x5641c0d9fe00] coded y,uvDC,uvAC intra: 99.6% 89.5% 53.3%
[libx264 @ 0x5641c0d9fe00] i16 v,h,dc,p:  2% 13% 46% 38%
```

```
[libx264 @ 0x5641c0d9fe00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu:  8% 17% 24%
[libx264 @ 0x5641c0d9fe00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu:  8% 17% 15%
[libx264 @ 0x5641c0d9fe00] i8c dc,h,v,p: 42% 26% 16% 16%
[libx264 @ 0x5641c0d9fe00] Weighted P-Frames: Y:8.5% UV:0.4%
[libx264 @ 0x5641c0d9fe00] ref P L0: 73.5% 23.2%  3.2%  0.1%  0.0%
[libx264 @ 0x5641c0d9fe00] ref B L0: 98.3%  1.3%  0.4%
[libx264 @ 0x5641c0d9fe00] kb/s:8495.08
```

Slerp Interpolation

This gets a little heady, but technically linear interpolations are not the best in high-dimensional GANs. [This github link](#) is one of the more popular explanations ad discussions.

In reality I do not find a huge difference between linear and spherical interpolations (the difference in z- and w-space is enough in many cases), but I've implemented slerp here for anyone interested.

```
!python generate.py --outdir=out/slerp-w/ --space="w" --trunc=1 --process="in
```

```
Generating image for frame 29/36 ...
Generating image for frame 30/36 ...
Generating image for frame 31/36 ...
Generating image for frame 32/36 ...
Generating image for frame 33/36 ...
Generating image for frame 34/36 ...
Generating image for frame 35/36 ...
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg d
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, image2, from 'out/slerp-w/frames/frame%04d.png':
  Duration: 00:00:01.44, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: png, rgb24(pc), 1024x1024, 25 fps, 25 tbr, 25
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x560dd8083e00] using cpu capabilities: MMX2 SSE2Fast SSSE
[libx264 @ 0x560dd8083e00] profile High, level 3.2
[libx264 @ 0x560dd8083e00] 264 - core 152 r2854 e9a5903 - H.264/MPEG-
Output #0, mp4, to 'out/slerp-w//interpolation-slerp-seeds_85_265_297
  Metadata:
    encoder         : Lavf57.83.100
```

```
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661), yuv420p,
    Metadata:
      encoder         : Lavc57.107.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=   36 fps=6.2 q=-1.0 Lsize=    4769kB time=00:00:01.37 bitrate=
video:4768kB audio:0kB subtitle:0kB other streams:0kB global headers:
[libx264 @ 0x560dd8083e00] frame I:1     Avg QP:25.94  size: 81748
[libx264 @ 0x560dd8083e00] frame P:9     Avg QP:29.26  size:188583
[libx264 @ 0x560dd8083e00] frame B:26    Avg QP:29.68  size:119332
[libx264 @ 0x560dd8083e00] consecutive B-frames:  2.8%  0.0%  8.3% 88
[libx264 @ 0x560dd8083e00] mb I  I16..4:  0.7% 89.0% 10.3%
[libx264 @ 0x560dd8083e00] mb P  I16..4:  0.9% 66.6% 23.1%  P16..4:
[libx264 @ 0x560dd8083e00] mb B  I16..4:  0.2% 13.2%  9.2%  B16..8: 1
[libx264 @ 0x560dd8083e00] 8x8 transform intra:68.6% inter:57.8%
[libx264 @ 0x560dd8083e00] coded y,uvDC,uvAC intra: 99.2% 76.6% 40.1%
[libx264 @ 0x560dd8083e00] i16 v,h,dc,p:  5% 15% 43% 37%
[libx264 @ 0x560dd8083e00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu:  8% 17% 29%
[libx264 @ 0x560dd8083e00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 10% 15% 17%
[libx264 @ 0x560dd8083e00] i8c dc,h,v,p: 48% 24% 17% 11%
[libx264 @ 0x560dd8083e00] Weighted P-Frames: Y:44.4% UV:44.4%
[libx264 @ 0x560dd8083e00] ref P L0: 40.9% 34.2% 13.6%  7.7%  3.6%
[libx264 @ 0x560dd8083e00] ref B L0: 89.2%  8.4%  2.4%
[libx264 @ 0x560dd8083e00] ref B L1: 97.8%  2.2%
[libx264 @ 0x560dd8083e00] kb/s:26035.33
```

Double-click (or enter) to edit

Noise Loop

If you want to just make a random but fun interpolation of your model the noise loop is the way to go. It creates a random path thru the z space to show you a diverse set of images.

`--interpolation="noiseloop"` : set this to use the noise loop funtion

`--diameter` : This controls how "wide" the loop is. Make it smaller to show a less diverse range of samples. Make it larger to cover a lot of samples. This plus `--frames` can help determine how fast the video feels.

`--random_seed` : this allows you to change your starting place in the z space. Note: this value has nothing to do with the seeds you use to generate images. It just allows you to randomize your start point (and if you want to return to it you can use the same seed multiple times).

Noise loops currently only work in z space.

```
!python generate.py --outdir=out/video-noiseloop-0.9d/ --trunc=0.8 --process=
Generating image for frame 233/240 ...
Generating image for frame 234/240 ...
Generating image for frame 235/240 ...
Generating image for frame 236/240 ...
Generating image for frame 237/240 ...
Generating image for frame 238/240 ...
Generating image for frame 239/240 ...
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg d
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, image2, from 'out/video-noiseloop-0.9d/frames/frame%04d.png
  Duration: 00:00:09.60, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: png, rgb24(pc), 1024x1024, 25 fps, 25 tbr, 25
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x55e8ff231e00] using cpu capabilities: MMX2 SSE2Fast SSSE
[libx264 @ 0x55e8ff231e00] profile High, level 3.2
[libx264 @ 0x55e8ff231e00] 264 - core 152 r2854 e9a5903 - H.264/MPEG-
Output #0, mp4, to 'out/video-noiseloop-0.9d//interpolation-noiseloop
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661), yuv420p,
    Metadata:
      encoder         : Lavc57.107.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=  240 fps=9.3 q=-1.0 Lsize=   10110kB time=00:00:09.87 bitrate=
video:10106kB audio:0kB subtitle:0kB other streams:0kB global headers
[libx264 @ 0x55e8ff231e00] frame I:2     Avg QP:24.02  size:251932
[libx264 @ 0x55e8ff231e00] frame P:116   Avg QP:25.25  size: 60820
[libx264 @ 0x55e8ff231e00] frame B:122   Avg QP:28.86  size: 22864
[libx264 @ 0x55e8ff231e00] consecutive B-frames: 31.7%  0.8%  2.5% 65

[libx264 @ 0x55e8ff231e00] mb I  I16..4:  4.7% 66.7% 28.6%
[libx264 @ 0x55e8ff231e00] mb P  I16..4:  0.1% 10.8%  3.1%  P16..4: 4
[libx264 @ 0x55e8ff231e00] mb B  I16..4:  0.0%  1.1%  0.7%  B16..8: 1
[libx264 @ 0x55e8ff231e00] 8x8 transform intra:74.6% inter:54.8%
[libx264 @ 0x55e8ff231e00] coded y,uvDC,uvAC intra: 98.6% 79.9% 36.0%
[libx264 @ 0x55e8ff231e00] i16 v,h,dc,p:  5%  6% 62% 27%
[libx264 @ 0x55e8ff231e00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 12% 12% 19%
[libx264 @ 0x55e8ff231e00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 11% 11% 16%
[libx264 @ 0x55e8ff231e00] i8c dc,h,v,p: 50% 20% 18% 12%
[libx264 @ 0x55e8ff231e00] Weighted P-Frames: Y:65.5% UV:51.7%
[libx264 @ 0x55e8ff231e00] ref P L0: 51.9% 31.2% 15.5%  1.0%  0.4%
[libx264 @ 0x55e8ff231e00] ref B L0: 95.5%  3.2%  1.3%
```

```
[libx264 @ 0x55e8ff231e00] ref B L1: 98.6%  1.4%
[libx264 @ 0x55e8ff231e00] kb/s:8278.63
```

Circular Loop

The noise loop is, well, noisy. This circular loop will feel much more even, while still providing a random loop.

I recommend using a higher `--diameter` value than you do with noise loops. Something between `50.0` and `500.0` alongside `--frames` can help control speed and diversity.

```
!python generate.py --outdir=out/video-circularloop/ --trunc=1 --process="int
```

```
Generating image for frame 714/721 ...
Generating image for frame 715/721 ...
Generating image for frame 716/721 ...
Generating image for frame 717/721 ...
Generating image for frame 718/721 ...
Generating image for frame 719/721 ...
Generating image for frame 720/721 ...
ffmpeg version 3.4.11-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg d
  built with gcc 7 (Ubuntu 7.5.0-3ubuntu1~18.04)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain
  libavutil      55. 78.100 / 55. 78.100
  libavcodec     57.107.100 / 57.107.100
  libavformat    57. 83.100 / 57. 83.100
  libavdevice    57. 10.100 / 57. 10.100
  libavfilter     6.107.100 /  6.107.100
  libavresample   3.  7.  0 /  3.  7.  0
  libswscale      4.  8.100 /  4.  8.100
  libswresample   2.  9.100 /  2.  9.100
  libpostproc    54.  7.100 / 54.  7.100
Input #0, image2, from 'out/video-circularloop/frames/frame%04d.png':
  Duration: 00:00:28.84, start: 0.000000, bitrate: N/A
    Stream #0:0: Video: png, rgb24(pc), 1024x1024, 25 fps, 25 tbr, 25
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x55619946be00] using cpu capabilities: MMX2 SSE2Fast SSSE
[libx264 @ 0x55619946be00] profile High, level 3.2
[libx264 @ 0x55619946be00] 264 - core 152 r2854 e9a5903 - H.264/MPEG-
Output #0, mp4, to 'out/video-circularloop//interpolation-circularloo
  Metadata:
    encoder         : Lavf57.83.100
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661), yuv420p,
    Metadata:
      encoder         : Lavc57.107.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=  721 fps= 10 q=-1.0 Lsize=   27803kB time=00:00:29.91 bitrate=
```

```
video:2779kB audio:0kB subtitle:0kB other streams:0kB global headers
[libx264 @ 0x55619946be00] frame I:3     Avg QP:21.56  size:233778
[libx264 @ 0x55619946be00] frame P:676   Avg QP:23.89  size: 40483
[libx264 @ 0x55619946be00] frame B:42    Avg QP:28.87  size:  9445
[libx264 @ 0x55619946be00] consecutive B-frames: 91.3%  2.5%  1.2%  5
[libx264 @ 0x55619946be00] mb I  I16..4:  1.5% 71.6% 26.9%
[libx264 @ 0x55619946be00] mb P  I16..4:  0.0%  0.7%  0.2%  P16..4: 5
[libx264 @ 0x55619946be00] mb B  I16..4:  0.0%  0.2%  0.2%  B16..8: 1
[libx264 @ 0x55619946be00] 8x8 transform intra:74.8% inter:54.9%
[libx264 @ 0x55619946be00] coded y,uvDC,uvAC intra: 97.8% 83.6% 47.7%
[libx264 @ 0x55619946be00] i16 v,h,dc,p: 28% 22% 22% 28%
[libx264 @ 0x55619946be00] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 13% 15% 16%
[libx264 @ 0x55619946be00] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 13% 14% 14%
[libx264 @ 0x55619946be00] i8c dc,h,v,p: 46% 22% 20% 12%
[libx264 @ 0x55619946be00] Weighted P-Frames: Y:20.6% UV:10.4%
[libx264 @ 0x55619946be00] ref P L0: 66.3% 29.0%  4.6%  0.1%  0.0%
[libx264 @ 0x55619946be00] ref B L0: 98.6%  1.0%  0.4%
[libx264 @ 0x55619946be00] ref B L1: 99.9%  0.1%
[libx264 @ 0x55619946be00] kb/s:7580.03
```

Projection

Basic Projector

- `--target` : this is a path to the image file that you want to "find" in your model. This image must be the exact same size as your model.
- `--num-steps` : how many iterations the projctor should run for. Lower will mean less steps and less likelihood of a good projection. Higher will take longer but will likely produce better images.

```
!python projector.py --help
```

```
Usage: projector.py [OPTIONS]

  Project given image to the latent space of
  pretrained network pickle.

  Examples:

  python projector.py --outdir=out --target=~/mytargetimg.png \
      --network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/p

Options:
  --network TEXT        Network pickle filename
                        [required]

  --target FILE         Target image file to
                        project to  [required]
```

```
  --num-steps INTEGER   Number of optimization
                        steps  [default: 1000]

  --seed INTEGER        Random seed  [default:
                        303]

  --save-video BOOLEAN  Save an mp4 video of
                        optimization progress
                        [default: True]

  --outdir DIR          Where to save the output
                        images  [required]

  --help                Show this message and
                        exit.
```

```
!python projector.py --network=/content/drive/MyDrive/colab-sg2-ada-pytorch/s
```

```
Loading networks from "/content/drive/MyDrive/colab-sg2-ada-pytorch/s
Computing W midpoint and stddev using 10000 samples...
Setting up PyTorch plugin "bias_act_plugin"... Done.
Downloading https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorch/pr
/usr/local/lib/python3.7/dist-packages/torch/nn/modules/module.py:105
  return forward_call(*input, **kwargs)
Setting up PyTorch plugin "upfirdn2d_plugin"... Done.
step    1/200: dist 0.82 loss 9558.62
step    2/200: dist 0.73 loss 13250.25
step    3/200: dist 0.77 loss 11540.60
step    4/200: dist 0.73 loss 8739.42
step    5/200: dist 0.74 loss 5764.26
step    6/200: dist 0.71 loss 3211.07
step    7/200: dist 0.77 loss 2035.84
step    8/200: dist 0.72 loss 2417.17
step    9/200: dist 0.73 loss 4066.37
step   10/200: dist 0.72 loss 5851.79
step   11/200: dist 0.72 loss 6889.45
step   12/200: dist 0.71 loss 6590.75
step   13/200: dist 0.71 loss 5817.12
step   14/200: dist 0.73 loss 4632.40
step   15/200: dist 0.71 loss 3482.05
step   16/200: dist 0.71 loss 2572.83
step   17/200: dist 0.71 loss 2101.17
step   18/200: dist 0.72 loss 1800.98
step   19/200: dist 0.72 loss 1558.86
step   20/200: dist 0.71 loss 1510.41
step   21/200: dist 0.70 loss 1586.13
step   22/200: dist 0.71 loss 1611.23
step   23/200: dist 0.71 loss 1512.92
step   24/200: dist 0.69 loss 1406.51
step   25/200: dist 0.70 loss 1375.06
step   26/200: dist 0.70 loss 1328.12
step   27/200: dist 0.69 loss 1202.93
step   28/200: dist 0.69 loss 998.09
```

```
step   29/200: dist 0.69 loss 784.73
step   30/200: dist 0.70 loss 589.29
step   31/200: dist 0.70 loss 447.32
step   32/200: dist 0.70 loss 354.61
step   33/200: dist 0.69 loss 333.43
step   34/200: dist 0.69 loss 388.53
step   35/200: dist 0.69 loss 449.49
step   36/200: dist 0.70 loss 491.45
step   37/200: dist 0.69 loss 497.62
step   38/200: dist 0.69 loss 438.31
step   39/200: dist 0.69 loss 344.12
step   40/200: dist 0.70 loss 257.43
step   41/200: dist 0.69 loss 179.54
step   42/200: dist 0.69 loss 127.03
step   43/200: dist 0.69 loss 111.30
step   44/200: dist 0.69 loss 140.25
step   45/200: dist 0.69 loss 173.13
step   46/200: dist 0.69 loss 184.86
step   47/200: dist 0.68 loss 167.57
step   48/200: dist 0.68 loss 141.47
step   49/200: dist 0.68 loss 109.74
step   50/200: dist 0.68 loss 84.22
```

Peter Baylies' Projector

```
!python /content/drive/MyDrive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch/pb
```

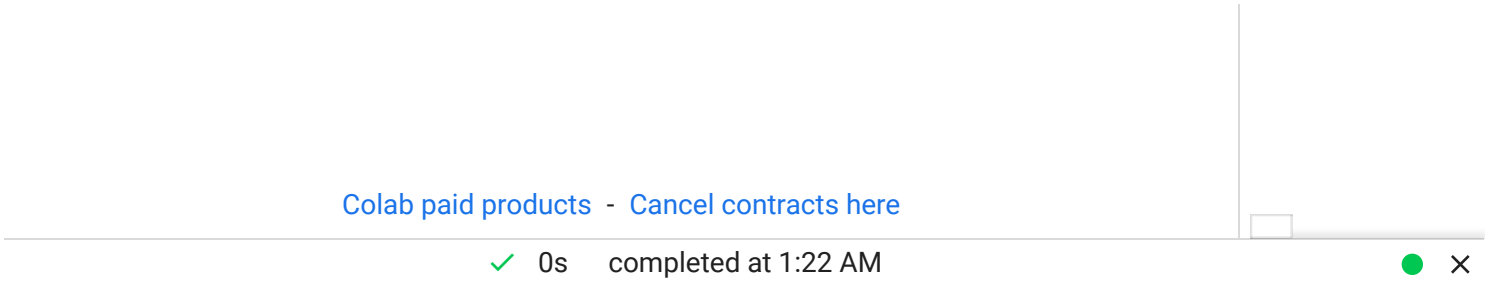
```
    Project given image to the latent space of
    pretrained network pickle.

    Examples:

    python projector.py --outdir=out --target=~/mytargetimg.png \
        --network=https://nvlabs-fi-cdn.nvidia.com/stylegan2-ada-pytorc

  Options:
    --network TEXT          Network pickle filename
                            [required]

    --target-image FILE     Target image file to
                            project to

    --target-text TEXT      Target text to project to
    --initial-latent TEXT   Initial latent
    --lr FLOAT              Learning rate  [default:
                            0.1]

    --num-steps INTEGER     Number of optimization
                            steps  [default: 1000]

    --seed INTEGER          Random seed  [default:
                            303]

    --save-video BOOLEAN    Save an mp4 video of
```

```
                            optimization progress
                            [default: True]

  --outdir DIR              Where to save the output
                            images  [required]

  --use-vgg BOOLEAN         Use VGG16 in the loss
                            [default: True]

  --use-clip BOOLEAN        Use CLIP in the loss
                            [default: True]

  --use-pixel BOOLEAN       Use L1/L2 distance on
                            pixels in the loss
                            [default: True]

  --use-penalty BOOLEAN     Use a penalty on latent
                            values distance from the
                            mean in the loss
                            [default: True]

  --use-center BOOLEAN      Optimize against an
                            additional center image
                            crop  [default: True]

  --use-kmeans BOOLEAN      Perform kmeans clustering
                            for selecting initial
                            latents  [default: True]

  --help                    Show this message and
                            exit.
```

```
!python /content/drive/MyDrive/colab-sg2-ada-pytorch/stylegan2-ada-pytorch/pb
```

```
step  288/3000: dist 13.41 loss 14.73
step  289/3000: dist 13.21 loss 13.46
step  290/3000: dist 12.88 loss 13.58
step  291/3000: dist 12.85 loss 13.96
step  292/3000: dist 13.13 loss 13.55
step  293/3000: dist 12.73 loss 13.14
step  294/3000: dist 12.78 loss 13.62
step  295/3000: dist 13.23 loss 13.63
step  296/3000: dist 12.79 loss 13.08
step  297/3000: dist 12.89 loss 13.59
step  298/3000: dist 12.77 loss 13.29
step  299/3000: dist 12.89 loss 13.60
step  300/3000: dist 12.53 loss 14.02
step  301/3000: dist 12.89 loss 14.64
step  302/3000: dist 12.63 loss 14.10
step  303/3000: dist 13.06 loss 13.87
step  304/3000: dist 12.76 loss 13.24
step  305/3000: dist 12.57 loss 13.66
step  306/3000: dist 12.88 loss 14.05
step  307/3000: dist 12.49 loss 12.79
step  308/3000: dist 13.10 loss 13.54
```

```
step  309/3000: dist 13.16 loss 14.07
step  310/3000: dist 12.38 loss 12.71
step  311/3000: dist 12.22 loss 12.46
step  312/3000: dist 12.62 loss 13.48
step  313/3000: dist 12.71 loss 13.45
step  314/3000: dist 12.53 loss 13.26
step  315/3000: dist 12.16 loss 13.59
step  316/3000: dist 12.35 loss 13.76
step  317/3000: dist 12.94 loss 13.59
step  318/3000: dist 12.73 loss 13.08
step  319/3000: dist 12.11 loss 12.65
step  320/3000: dist 12.14 loss 12.95
step  321/3000: dist 12.36 loss 13.39
step  322/3000: dist 12.22 loss 13.15
step  323/3000: dist 12.51 loss 13.49
step  324/3000: dist 12.38 loss 13.69
step  325/3000: dist 12.24 loss 13.24
step  326/3000: dist 14.26 loss 14.44
step  327/3000: dist 12.50 loss 12.67
step  328/3000: dist 12.45 loss 13.25
step  329/3000: dist 12.24 loss 13.10
step  330/3000: dist 12.04 loss 12.42
step  331/3000: dist 12.60 loss 12.98
step  332/3000: dist 11.93 loss 12.87
step  333/3000: dist 11.93 loss 13.30
step  334/3000: dist 12.79 loss 14.10
step  335/3000: dist 12.43 loss 13.25
step  336/3000: dist 12.22 loss 12.59
step  337/3000: dist 12.12 loss 12.56
step  338/3000: dist 12.09 loss 12.89
step  339/3000: dist 12.60 loss 13.47
step  340/3000: dist 11.60 loss 12.32
step  341/3000: dist 12.13 loss 13.00
step  342/3000: dist 12.20 loss 13.57
step  343/3000: dist 12.15 loss 13.66
step  344/3000: dist 11.83 loss 12.95
```


Colab paid products - Cancel contracts here
0s completed at 1:22 AM