

Workshop

Setup Remote Jupyterhub Notebook

1. Signing into Jupyterhub

Login to JupyterHub with your cluster credentials [here](#)

Choose an XS or S slice

make sure to choose cuda 11.7 from the dropdown



A screenshot of a JupyterHub interface. At the top, there is a light gray header bar. Below it, the text 'Login to JupyterHub with your cluster credentials [here](#)' is displayed. Underneath, the instruction 'Choose an XS or S slice' is shown. A prominent orange banner contains the text 'make sure to choose cuda 11.7 from the dropdown'. Below the banner, a screenshot of the JupyterHub server selection interface is shown, featuring a 'Start My Server' button with a mouse cursor hovering over it.

2. Installing Stylegan3

```
conda init bash
```

```
source ~/.bashrc
```

```
git clone https://github.com/NVlabs/stylegan3.git
```

```
cd stylegan3
```

```
conda env create -f environment.yml
```

```
conda activate stylegan3
```

```
conda install cudatoolkit
```

```
pip install torch torchvision
```

```
conda install psutil
```

Training

For training your own datasets, you can follow these steps.

For your own dataset make sure that your training data has the correct resolution. You may use either 1024x1024, 512x512 or 256x256 resolution. The chosen resolution has to match with the pre-existing dataset that you want to train on.

You may start a dataset from scratch, just be aware that generally training your collected images on a pre-existing dataset will usually give better results (and faster ones too).

activating conda environment

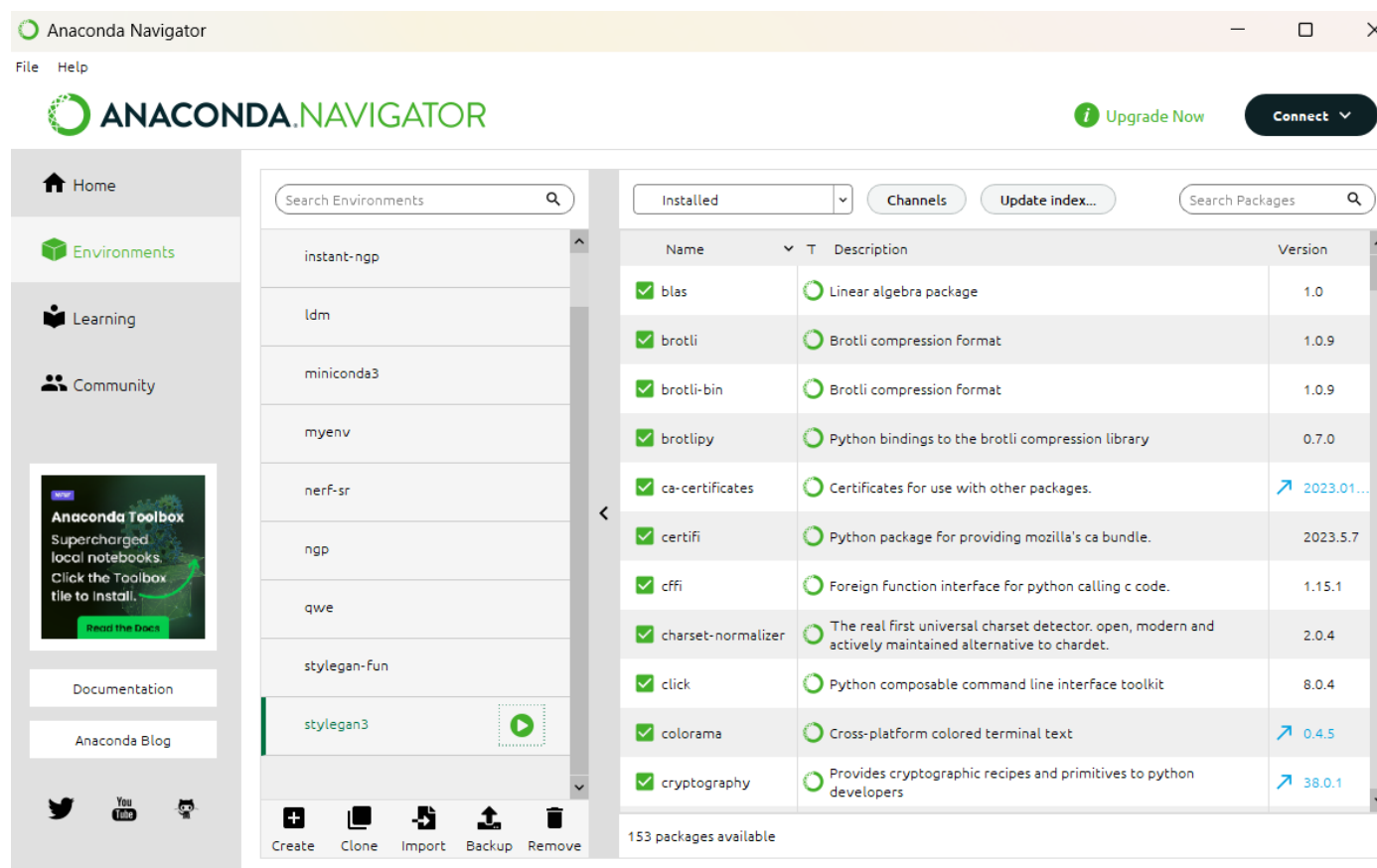
this needs to be done before every session if you want to use stylegan

```
conda init bash
```

```
source ~/.bashrc
```

```
conda activate stylegan3
```

alternatively if you are not able to activate stylegan3 through the terminal on our workstations, you can use the anaconda Navigator and start the terminal with the environment activated:



The screenshot shows the Anaconda Navigator application window. The left sidebar contains navigation options: Home, Environments, Learning, and Community. The 'Environments' section is active, displaying a list of environments: instant-ngp, ldm, miniconda3, myenv, nerf-sr, ngp, qwe, stylegan-fun, and stylegan3. The 'stylegan3' environment is selected and highlighted with a green play button icon. Below the environment list are buttons for 'Create', 'Clone', 'Import', 'Backup', and 'Remove'. The main panel shows the 'Installed' tab with a table of installed packages. The table has columns for Name, Description, and Version. The packages listed are: blas, brotli, brotli-bin, brotli-py, ca-certificates, certifi, cffi, charset-normalizer, click, colorama, and cryptography. At the bottom of the package list, it says '153 packages available'.

Name	Description	Version
blas	Linear algebra package	1.0
brotli	Brotli compression format	1.0.9
brotli-bin	Brotli compression format	1.0.9
brotli-py	Python bindings to the brotli compression library	0.7.0
ca-certificates	Certificates for use with other packages.	2023.01...
certifi	Python package for providing mozilla's ca bundle.	2023.5.7
cffi	Foreign Function interface for python calling c code.	1.15.1
charset-normalizer	The real first universal charset detector. open, modern and actively maintained alternative to chardet.	2.0.4
click	Python composable command line interface toolkit	8.0.4
colorama	Cross-platform colored terminal text	0.4.5
cryptography	Provides cryptographic recipes and primitives to python developers	38.0.1

Download Training data

If not done before, set up a new directory for the training data

```
mkdir trainingdata
```

in the trainingdata folder: set up a new folder (for example called imgs) and upload your images

If you want to download your trainingdata from a sciebo folder, you may use this code. Instead of the given link you may use your own one.

```
wget --no-check-certificate --content-disposition https://th-koeln.sciebo.de/s/7SzJ55ZroKPf5zY/download -O trainingdata/group01.zip
```

```
wget --no-check-certificate --content-disposition https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/stylegan3-t-afhqv2-512x512.pkl -O datasets/stylegan3-t-afhqv2-512x512.pkl
```

<https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/stylegan3-t-afhqv2-512x512.pkl>

Prepare training data

Before training, it is highly recommended to check your dataset through the given stylegan3 check-up. It can resize your images too, although it is usually better to do it before on your own. (Adobe Bridge is a great tool for batch processing.)

```
python dataset_tool.py --source=trainingdata/imgs/ --dest=trainingdata/group01.zip --resolution=512x512
```

--source= your directory with the given files

--destination= the output directory, dont forget to add .zip at the end

--resolution= resolution you want your images to be saved in. (1024x1024, 512x512 or 256x256 resolution)

start training

```
python train.py --outdir=/home/jovyan/stylegan3/training-runs --cfg=stylegan3-t --data=/home/jovyan/stylegan3/trainingdata/group01.zip --gpus=1 --batch=8 --gamma=2 --snap=20 --resume="https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/stylegan3-t-afhqv2-512x512.pkl"
```

--outdir= direction where you want to save the trained data

--data= your dataset you want to use

--gpus= depends on the gpus you have. usually 1 will be

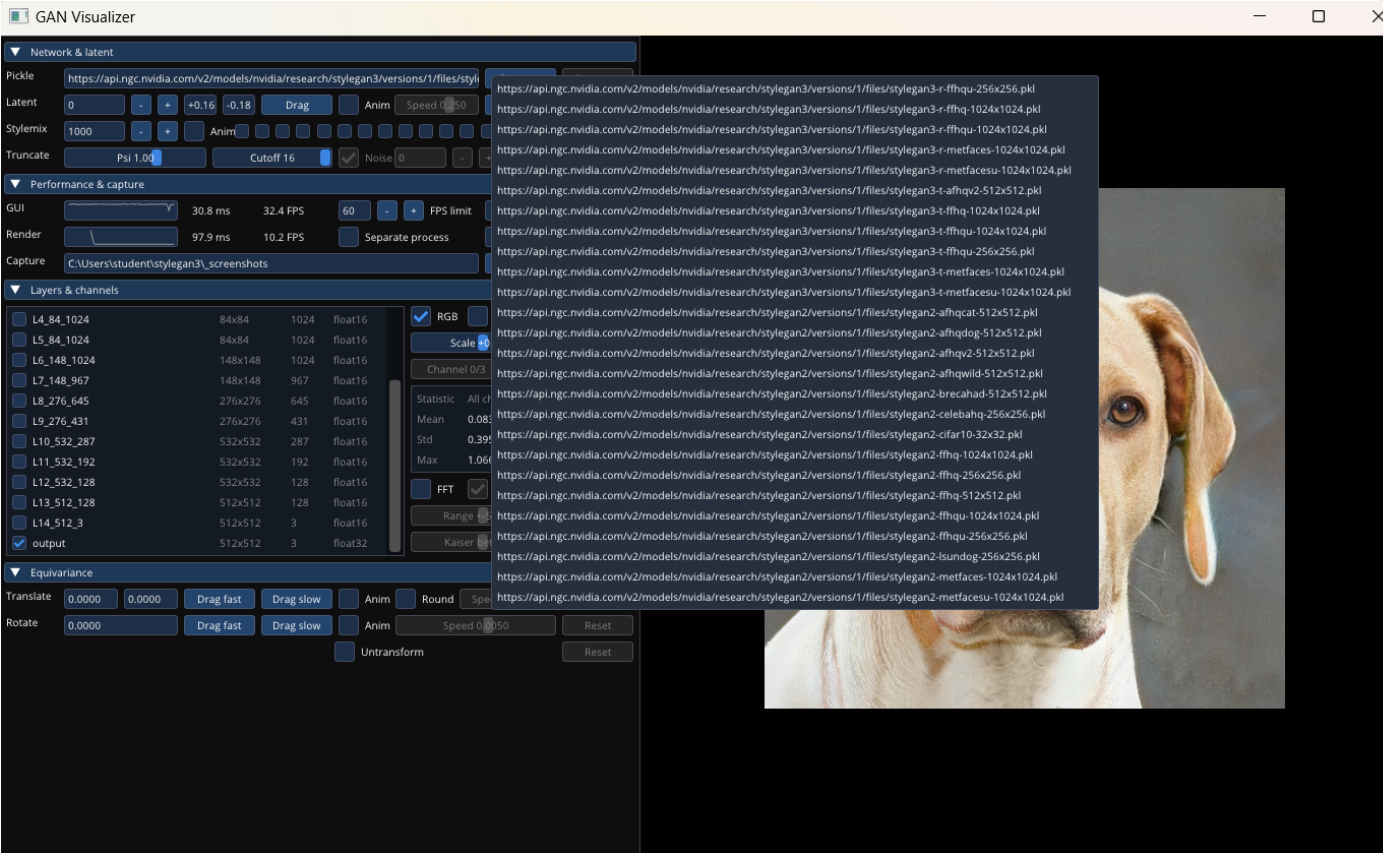
--batch=

Opening the Python Visualizer

Python Visualizer

The following script will open the python Visualizer

```
python visualizer.py
```



You can open one of the .pkl files that NVIDIA provided or paste a local URL to your own trainingdata to visualize and play around.