WORK LOG

1. Create 2 pickle files, 1 for train one for test as in the code specifies,
2. SSH into instruction GPU 2

y "ssh [your cs account name]@[instgpu-01.cs.wisc.edu](http://instgpu-01.cs.wisc.edu/)", and give it your CS account password.

[instgpu-01.cs.wisc.edu](http://instgpu-01.cs.wisc.edu/) to [instgpu-04.cs.wisc.edu](http://instgpu-04.cs.wisc.edu/).

1. Nvidia-smi shows you current available GPU
2. Download anaconda Linux version using wget (web address)
3. df -h shows usage
4. Bash anaconda … to install anaconda using bash?
5. Store in no backup on GPU -> can only ssh into that account. Store in home->can access everywhere
6. Copy: scp (file name in the directory you’re in)
7. Opt.niter is the parameter for number of epoch

To do: 1. download all data sets and classify them into folders

3. Try to run the experiment using return feature = True (using the hint Jifan uses)

4. change the data loader code to use the setting of the code base OOD … -> don’t need separate pickle files [scone/CIFAR/train.py at main · deeplearning-wisc/scone](https://github.com/deeplearning-wisc/scone/blob/main/CIFAR/train.py)

5. run on GPU

6. first experiment run only one other generator

7. Todo: figure out the val.pickle file thing (right now it’s the same as train.pickle)

opt.real\_list\_path = "/path/to/real\_picked\_files"

opt.fake\_list\_path = "/path/to/fake\_picked\_files"

opt.data\_mode = "ours" # since you are using your own dataset

opt.batch\_size = 32 # adjust based on your GPU memory

opt.num\_threads = 4 # adjust based on your machine

**Optimizing for Large Datasets:**

* **Increase num\_threads**: The DataLoader can use multiple worker threads to load images from disk in parallel (num\_workers=int(opt.num\_threads)).
* **Adjust batch\_size**: Depending on your GPU memory, you might need to experiment with the batch size for optimal performance.

python train.py --data\_mode ours --real\_list\_path /path/to/real\_picked\_files --fake\_list\_path /path/to/fake\_picked\_files --batch\_size 32 --num\_threads 4

* current command:

~~python train.py --name=clip\_vitl14 --arch=CLIP:ViT-L/14 --fix\_backbone --data\_mode ours --real\_list\_path .\datasets\_real\_and\_fake\real --fake\_list\_path .\datasets\_real\_and\_fake\fake --batch\_size 32 --num\_threads 4~~

python train.py --name=clip\_vitl14 --arch=CLIP:ViT-L/14 --fix\_backbone --data\_mode ours --real\_path .\datasets\_real\_and\_fake\real --fake\_path .\datasets\_real\_and\_fake\fake\firefly\_synthbuster\_1k .\datasets\_real\_and\_fake\fake\glide\_synthbuster\_1k --real\_sample\_size 1000 --fake\_sample\_sizes 1000 1000 --experiment\_name Oct\_29\_first --batch\_size 32 --num\_threads 4 --seed 42

print out: CUDA is not available or no GPU IDs specified. Running on CPU.

CUDA is not available or no GPU IDs specified. Running on CPU.

Created pickle file for real images: train\_real\_Oct\_29\_first.pickle

Created pickle file for fake images: train\_fake\_Oct\_29\_first.pickle

Model moved to CPU

**Todo**: figure out the val.pickle file thing (right now it’s the same as train.pickle in train.py after main)

**Note**: the current val.pickle file in both the real folder and the fake folder are just copies of the train.pickle folder. I’ll create an actual val.pickle file later. Essentially, in both the real and fake path, there need to be both train.pickle and val.pickle files