HOMEWORK 2

16824 VISUAL LEARNING AND RECOGNITION (SPRING 2023)
https://piazza.com/class/lcy4ow5l5xp2fl

RELEASED: Thurs, 23rd Feb 2023
DUE: Wed, 15th March 2023
Instructor: Deepak Pathak
TAs: Ananye Agarwal, Rohan Choudhury, Murtaza Dalal, Russell Mendonca

START HERE: Instructions

- **Collaboration policy:** All are encouraged to work together BUT you must do your own work (code and write up). If you work with someone, please include their name in your write-up and cite any code that has been discussed. If we find highly identical write-ups or code or lack of proper accreditation of collaborators, we will take action according to strict university policies. See the Academic Integrity Section detailed in the initial lecture for more information.
- **Late Submission Policy:** There are a **total of 7** late days across all homework submissions. Submissions more than 7 days after the deadline will receive a 0.
- **Submitting your work:**
  - We will be using Gradescope (`https://gradescope.com/`) to submit the Problem Sets. Please use the provided template only. Submissions must be written in LaTeX. All submissions not adhering to the template will not be graded and receive a zero.
  - **Deliverables:** Please submit all the `.py` files. Add all relevant plots and text answers in the boxes provided in this file. TO include plots you can simply modify the already provided latex code. Submit the compiled `.pdf` report as well.

NOTE: Partial points will be given for implementing parts of the homework even if you don't get the mentioned numbers as long as you include partial results in this pdf.

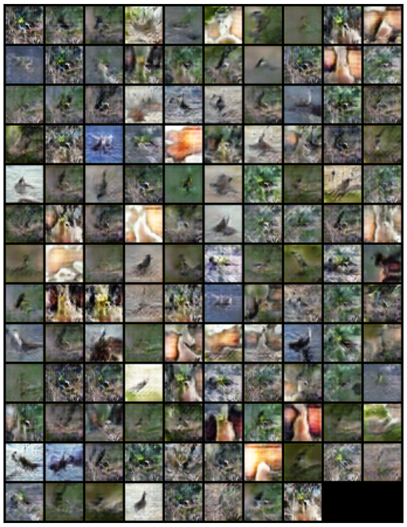
1 Generative Adversarial Networks (50 points)

We will be training Generative Adversarial Networks (GAN) on the CUB 2011 Dataset.

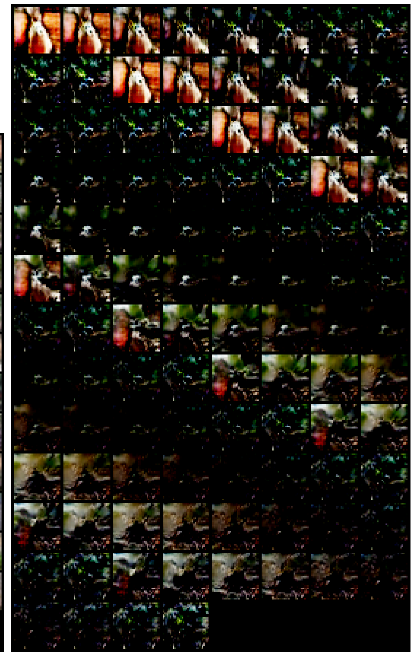
- **Setup:** Run the following command to setup everything you need for the assignment:
  `./setup.sh /path/to/python_env/lib/python3.8/site-packages`
- **Question:** Follow the instructions in the `README.md` file in the `gan/` folder to complete the implementation of GANs.
- **Deliverables:** The code will log plots to `gan/data_gan`, `gan/data_ls_gan`, and `gan/data_wgan_gp`. Extract plots and paste them into the appropriate section below. Note for all questions, we ask for final FID. Final FID is computed using 50K samples, at the very end of training. See the final print out for "Final FID (Full 50K): ".
- **Debugging Tips:**
  - GAN losses are pretty much meaningless! If you want to understand if your network is learning, visualize the samples. The FID score should generally be going down as well.
  - Do NOT change the hyper-parameters at all, they have been carefully tuned to ensure the networks will train stably. If things aren't working its a bug in your code.
  - For debugging, disable JIT using `export PYTORCH_JIT=0 python ...` and disable AMP by using the flag `--disable_amp`. However, do note that disabling JIT will cause the FID calculation to fail. So only disable JIT to make sure that your network code runs correctly, then re-enable when training. If you observe any errors involving type mismatches and tensors that have half types, it is due to AMP, you may need to explicitly cast the tensor using `.half()`.
  - Here is a sample image from WGAN-GP at the end of training. The other networks may have variations but should look similar:

- **Expected results:**
  - Vanilla GAN: Final FID should be less than 110.
  - LS-GAN: Final FID should be less than 90.
  - WGAN-GP: Final FID should be less than 70.

1. Paste your plot of the samples and latent space interpolations from Vanilla GAN as well as the *final* FID score you obtained.
   FID: **98.58848398694346**



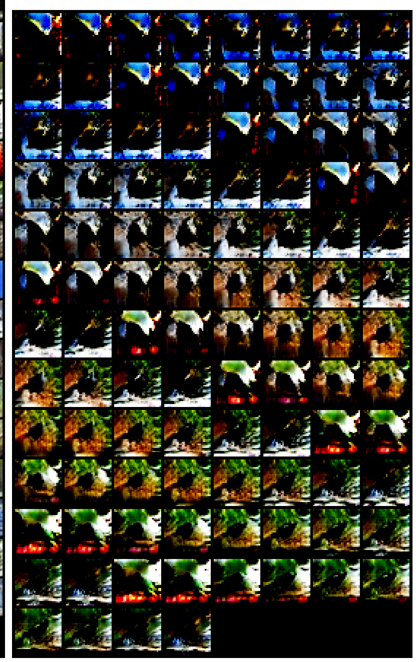
(a) Samples



(b) Latent Space Interpolations

2. Paste your plot of the samples and latent space interpolations from LS-GAN as well as the *final* FID score you obtained.
   FID: **81.7076792077882**

(a) Samples



(b) Latent Space Interpolations

3. Paste your plot of the samples and latent space interpolations from WGAN-GP as well as the *final* FID score you obtained.
   FID: **49.37464583949486**

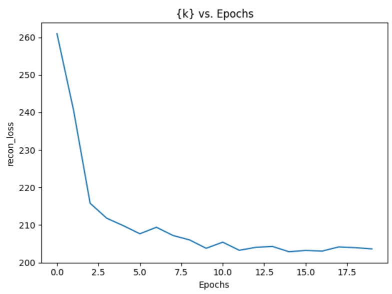
(a) Samples (b) Latent Space Interpolations

2 Variational Autoencoders (30 pts)

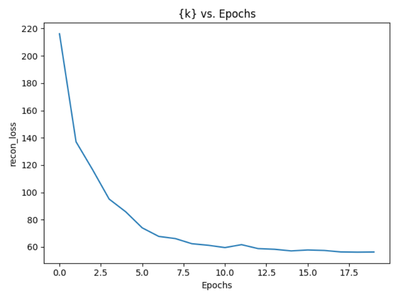
We will be training AutoEncoders and Variational Auto-Encoders (VAE) on the CIFAR10 dataset.

- **Question:** Follow the instructions in the `README.md` file in the `vae/` folder to complete the implementation of VAEs.
- **Deliverables:** The code will log plots to different folders in `vae`. Please paste the plots into the appropriate place for the questions below. Note for ALL questions, use the reconstructions and samples from the final epoch (epoch 19).
- **Debugging Tips:**
  - Make sure the auto-encoder can produce good quality reconstructions before moving on to the VAE. While the VAE reconstructions might not be clear and the VAE samples even less so, the auto-encoder reconstructions should be very clear.
  - If you are struggling to get the VAE portion working: debug the KL loss independently of the reconstruction loss to ensure the learned distribution matches standard normal.
- **Expected results:**
  - AE: reconstruction loss should be < 40, reconstructions should look similar to original image.
  - VAE: reconstruction loss should be < 145 ($\beta = 1$ case).
  - VAE: reconstruction loss should be < 125 when annealing β.

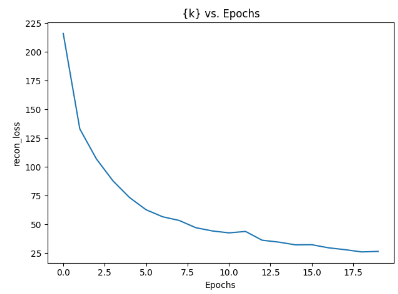
1. Autoencoder: For each latent size, paste your plot of the reconstruction loss curve and reconstructions.



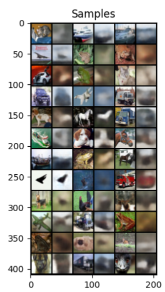
(a) Loss: latent size 16



(b) Loss: latent size 128



(c) Loss: latent size 1024

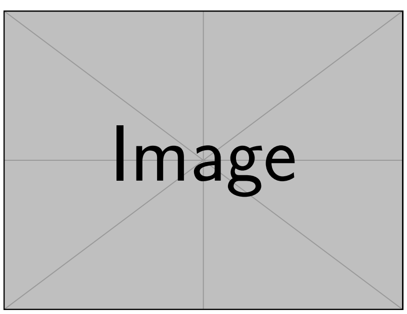


(d) Reconstructions: latent size 16

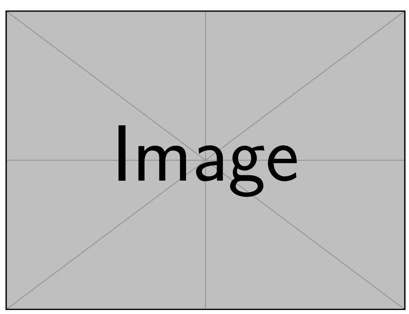
(e) Reconstructions: latent size 128

(f) Reconstructions: latent size 1024

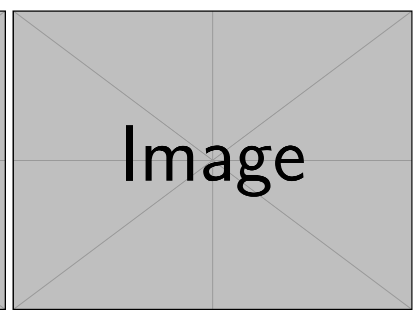
2. VAE: Choose the β that results in the best sample quality, β^*. Paste the reconstruction and kl loss curve plots as well as the sample images corresponding to the VAE trained using constant β^* and the VAE trained using β annealing scheme with β^*.



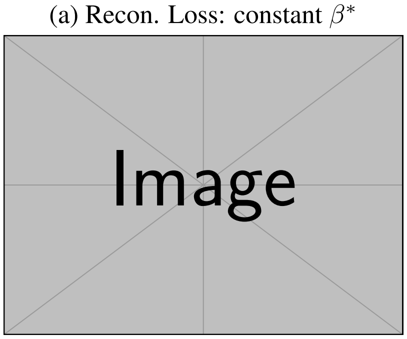
(a) Recon. Loss: constant β^*



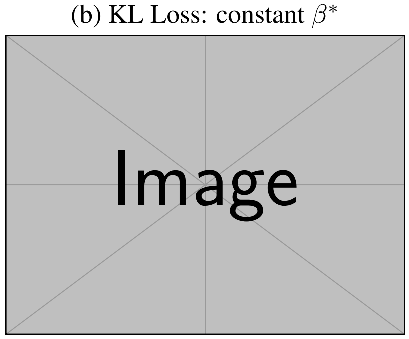
(b) KL Loss: constant β^*



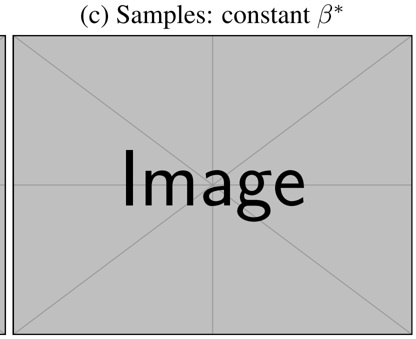
(c) Samples: constant β^*



(d) Recon. Loss: β annealed



(e) KL Loss: β annealed



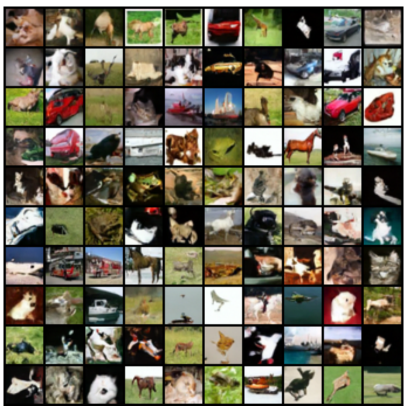
(f) Samples: β annealed

3 Diffusion Models (20 points)

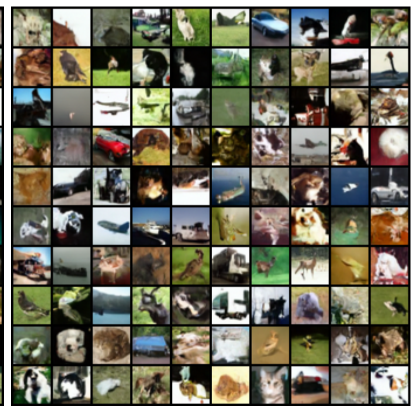
We will be running inference using a pre-trained diffusion model (DDPM) on CIFAR-10.

- **Setup:** Download our pre-trained checkpoint for DDPM from https://drive.google.com/file/d/1gtn9Jv9jBUol7iJw-94hw4j6KfpG3SZE/view?usp=sharing.
- **Question:** Follow the instructions in the `README.md` file in the `diffusion/` folder to complete the implementation of the sampling procedures for Diffusion Models.
- **Deliverables:** The code will log plots to `diffusion/data_ddpm` and `diffusion/data_ddim`. Extract plots and paste them into the appropriate section below.
- **Expected results:**
  - FID of less than 60 for DDPM and DDIM

1. Paste your plots of the DDPM and DDIM samples.



(a) DDPM Samples



(b) DDIM Samples

2. Paste in the FID score you obtained from running inference using DDPM and DDIM.
   DDPM FID: **31.243336221617994**
   DDIM FID: **35.01378734353398**

Collaboration Survey Please answer the following:

1. Did you receive any help whatsoever from anyone in solving this assignment?

   ◯ Yes

   ◯ No

   - If you answered 'Yes', give full details:
   - (e.g. "Jane Doe explained to me what is asked in Question 3.4")

2. Did you give any help whatsoever to anyone in solving this assignment?

   ◯ Yes

   ◯ No

   - If you answered 'Yes', give full details:
   - (e.g. "I pointed Joe Smith to section 2.3 since he didn't know how to proceed with Question 2")

3. Note that copying code or writeup even from a collaborator or anywhere on the internet violates the Academic Integrity Code of Conduct.