HOMEWORK 2

16824 VISUAL LEARNING AND RECOGNITION (FALL 2025)
https://piazza.com/cmu/fall2025/16824a/

RELEASED: Friday, 3rd Oct. 2025
DUE: 11:59 PM ET, Wednesday, 22nd Oct. 2025
Instructor: Jun-Yan Zhu
TAs: Ananya Bal, Eungyeup Kim, Jay Karhade, Zhixuan Liu

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 5** late days across all homework submissions. Submissions that use additional late days will incur a 10% penalty per late day.
- **Submitting your work:**
  - We will be using Gradescope (`https://gradescope.com/`) to submit the Problem Sets. Please use the provided template only. You do **not** need any additional packages and using them is **strongly discouraged**. 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 accuracy as long as you include partial results in this pdf.

1 Generative Adversarial Networks (50 points)

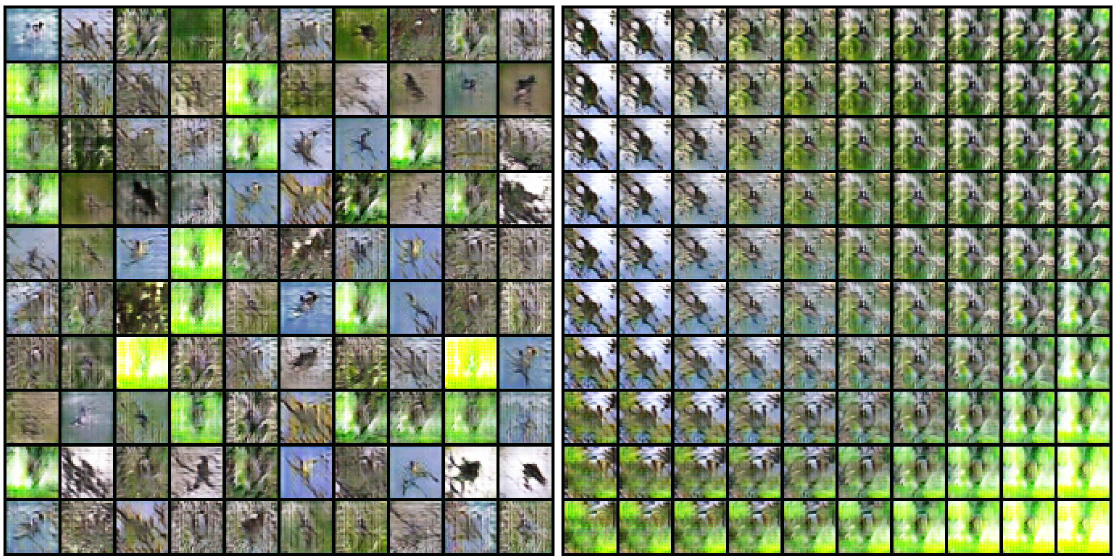
We will be training Generative Adversarial Networks (GAN) on the CUB 2011 Dataset.

- **Setup:** Run the following command to set up everything you need for the assignment: `./setup.sh /path/to/python_env/lib/python3.8/site-packages`. Please use the pytorch installation from the previous homework.
- **Question:** Follow the instructions in the `README.md` file in the `gan/` folder to complete the implementation of GANs.
- **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.
- **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 printout for "Final FID (Full 50K)".

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

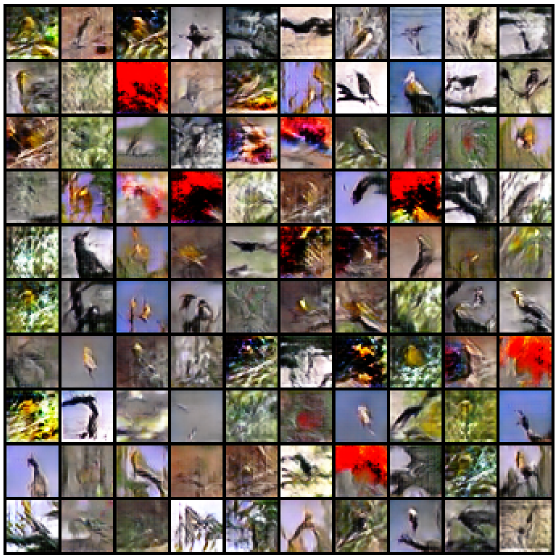
Solution:
Vanilla GAN FID: 142.18 (Reached down to about 63 but then went back up, and I could not keep it from collapsing back up)



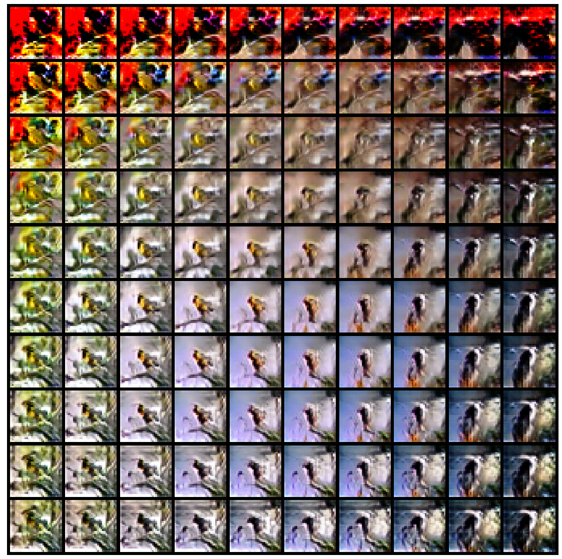
(a) Vanilla GAN Samples (b) Vanilla GAN Latent Space Interpolations

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

Solution:
LS-GAN FID: 72



(a) LS-GAN Samples

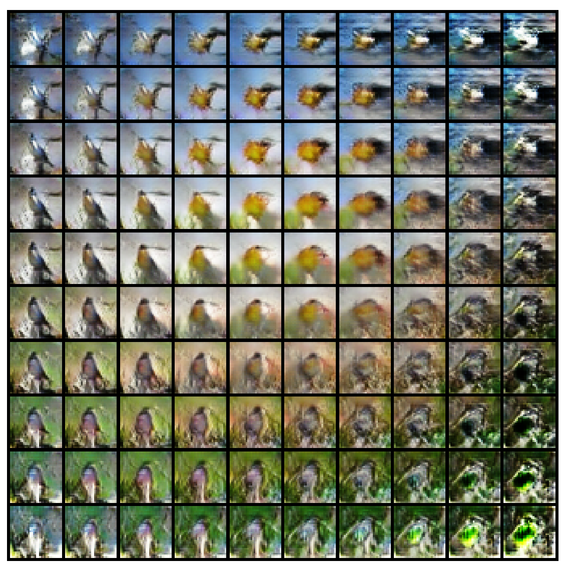


(b) LS-GAN Latent Space Interpolations

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

Solution:
WGAN-GP FID: 45

(a) WGAN-GP Samples



(b) WGAN-GP Latent Space Interpolations

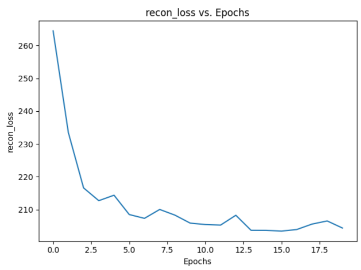
2 Variational Autoencoders (30 pts)

We will be training Autoencoders and Variational Auto-Encoders (VAE) on the CIFAR-10 dataset.

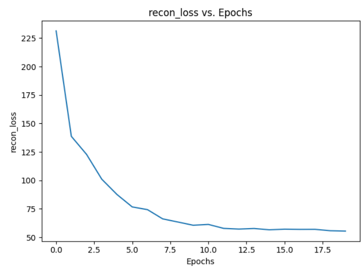
- **Question:** Follow the instructions in the `README.md` file in the `vae/` folder to complete the implementation of VAEs.
- **Debugging Tips:**
  - Make sure the auto-encoder can produce good-quality reconstructions before moving on to the VAE. While the VAE reconstructions might not be clear and 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.
- **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).

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

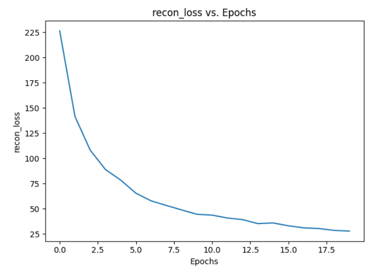
Solution:



(a) Loss (size 16)



(b) Loss: (size 128)



(c) Loss: (size 1024)

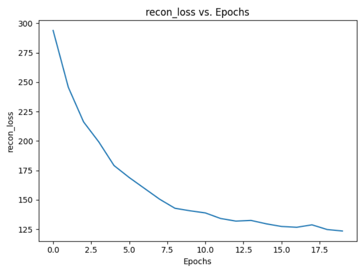
(d) Reconstructions: (size 16)

(e) Reconstructions: (size 128)

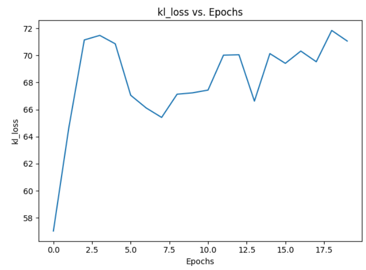
(f) Reconstructions: (size 1024)

2. VAE: Choose the β that results in the best sample quality, β^*. Report the β^* you used in your experiments. In addition, paste the reconstruction, KL loss curve plots, and the sample images corresponding to the VAE trained using constant β^* and the VAE trained using β annealing scheme with β^*.

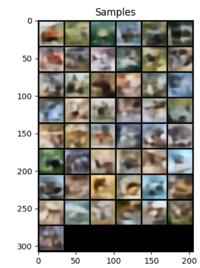
Solution:
β^* value: 0.8



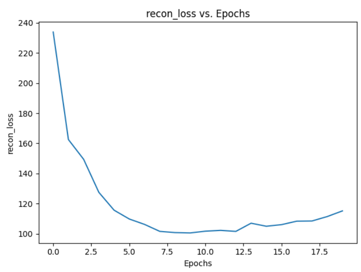
(a) Recon. Loss: constant β^*



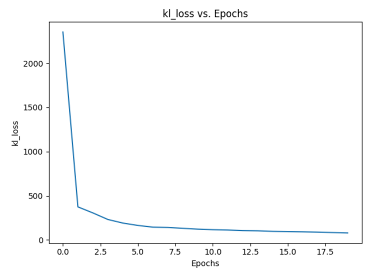
(b) KL Loss: constant β^*



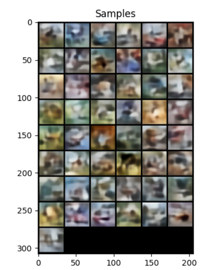
(c) Samples: constant β^*



(d) Recon. Loss: β annealed



(e) KL Loss: β annealed



(f) Samples: β annealed

3 Diffusion Models (20 points)

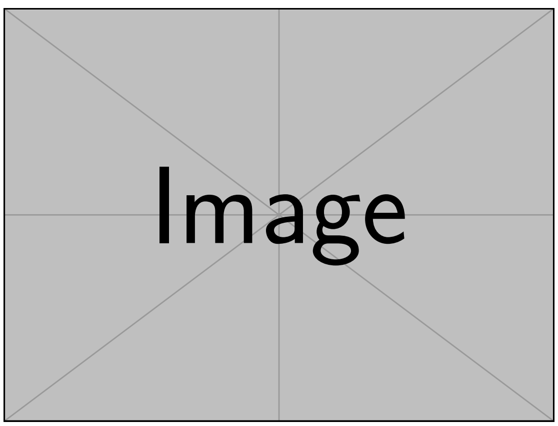
NOTE: You only need to complete **ONE** of the following two sections: either Section 3 (Diffusion Models) OR Section 4 (Flow Matching Models). Both sections are worth 20 points.

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

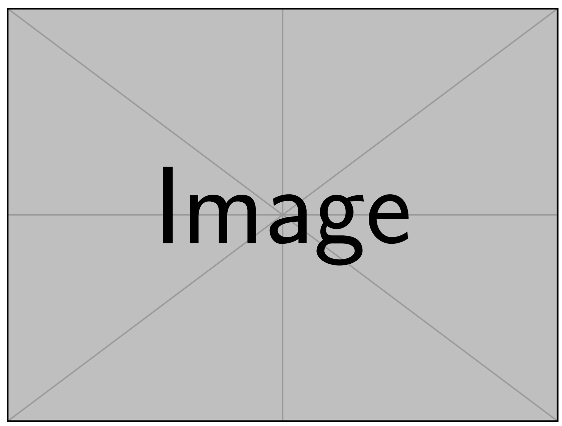
- **Setup:** Download our pre-trained checkpoint for DDPM from here.
- **Question:** Follow the instructions in the `README.md` file in the `diffusion/` folder to complete the implementation of the sampling procedures for Diffusion Models.
- **Expected results:**
  - FID of less than 60 for DDPM and DDIM
- **Deliverables:** The code will log plots to `diffusion/data_ddpm` and `diffusion/data_ddim`. Extract plots and paste them into the appropriate section below.

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

Solution:



(a) DDPM Samples



(b) DDIM Samples

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

Solution:

- DDPM FID:
- DDIM FID:

3. Compare the generated samples from your VAE and diffusion models. What do you observe? Does one generate better pictures than the other? Does the quality of the generated images match with the FID differences?

Solution:

4 Flow Matching Models (20 points)

NOTE: You only need to complete **ONE** of the following two sections: either Section 3 (Diffusion Models) OR Section 4 (Flow Matching Models). Both sections are worth 20 points.

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

- **Setup:** Download our pre-trained checkpoint for flow matching model from here.
- **Question:** Follow the instructions in the `README.md` file in the `flow_matching/` folder to complete the implementation of the sampling procedures for Flow Matching Models.
- **Expected results:**
  - FID of less than 60 for Euler ODE sampling
- **Deliverables:** The code will log plots to `flow_matching/flow_euler`. Extract the plot and paste it into the section below.

1. Paste your plots of the Euler ODE samples.

Solution:

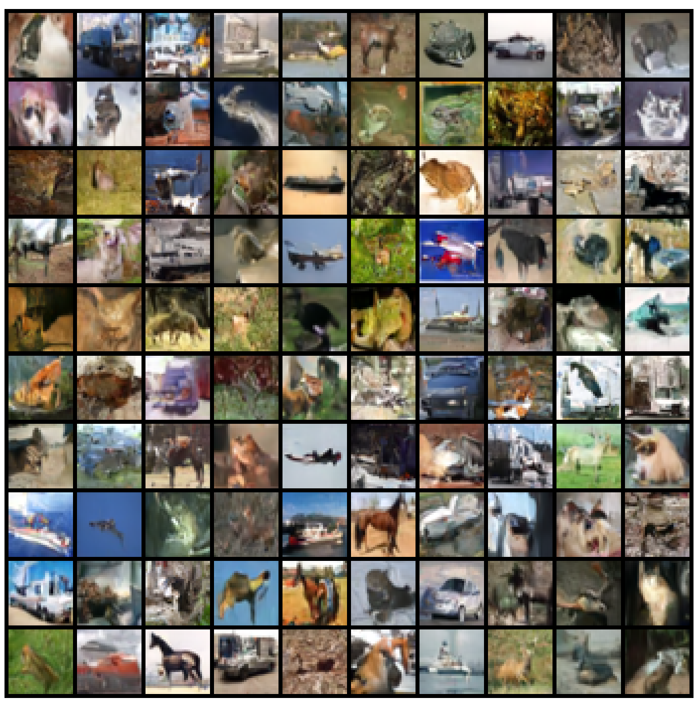


Figure 4.1: Flow Matching Euler ODE Samples

2. Paste in the FID score you obtained from running inference using Euler ODE sampling.

Solution:

- Flow Matching FID: 30.598

3. Compare the generated samples from your VAE and flow matching models. What do you observe? Does one generate better pictures than the other?

Solution:
The flow matching model seems to generate far sharper pictures than the VAE. However, given the resolution it is hard to tell the difference between the regular AE and the flow matching model though the flow matching model does seem a bit sharper though they both look similar. Interestingly, both are far better than the VAE with KL-Divergence and the Beta term, though it would be interesting to try to limit Beta toward the end following a warmup or attempt a cyclical annealing method.

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.