Spring 2025
Introduction to Artificial Intelligence

Homework 5 : Generative AI

Due Date: TBD

Introduction

Generative AI (Gen AI) has become a trending topic worldwide, especially after the release of ChatGPT. Gen AI can generate various types of content, including text, images, audio, and video. Recently, in the field of image generation, a novel technique called the Diffusion Model has gained significant attention, which this assignment focuses on. In this task, you will implement the diffusion process and train your own diffusion model from scratch. Additionally, you will generate a special-effect image using a pretrained text-to-image diffusion model. This special effect requires an additional denoising operation without requiring further training or fine-tuning.

The goal of this programming assignment is to:

1) Understand the architecture and design of U-Net.
2) Implement a diffusion model and train it from scratch.
3) Learn how to evaluate generative models (e.g., FID score).
4) Utilize existing (diffusion) models available on Hugging Face.
5) Explore a training-free method for diffusion.

Requirements

1. Please modify the code in source code between # Begin your code and # End your code. If you modify the other code (e.g. Add HorizontalFlip() for data augmentation), please specify in the corresponding section (e.g. Part1-4) or appendix.

2. This assignment requires high-level hardware resource (GPU), we suggest to run your code on Colab or Kaggle. In our perspective, it's better to use Kaggle since it allows a free account to access GPU T4 up to 30 hours per week.

   Codes on Kaggle Notebook: Part1 | Part2 | FID&AFD Evaluation

3. It's forbidden to use additional data, pretrained weights, other packages in your implementation. Besides, do not submit the dataset as the generated output, otherwise you might receive no score in that part.

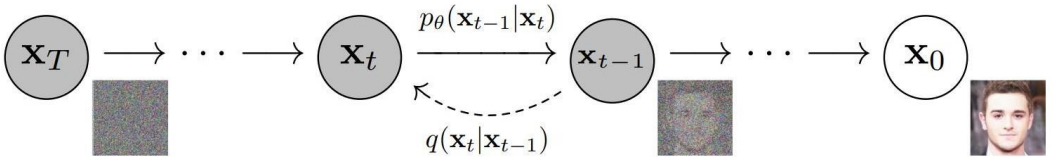
4. Please make sure your code can be reproduced. If our reproduced results are much different from yours (on evaluation), the score will be reduced.

Implementation (70%)

Part 1: Anime Face Generation (45%)

Part 1-1: Denoising Process (15%)

- In a diffusion model, there are two distinct processes. The first is the forward process, which adds noise to an image, with the noise sampled from a Gaussian distribution. The second is the backward process, where the model predicts the added noise from the image at time t.



(The image depicts the DDPM process, but we use the DDIM formula instead.)

Initialization

$$Given\ \beta_t;\ \alpha_t = 1 - \beta_t,\ \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

Forward

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

Backward

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_t(x_t), \sigma_t^2 I)$$

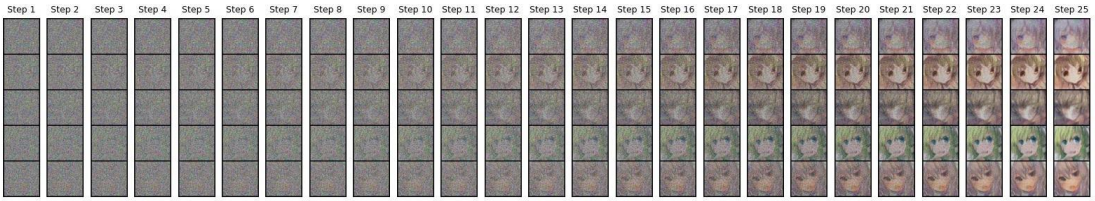
$$\mu_t(x_t) = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t, t)$$

$$\sigma_t = \eta \cdot \sqrt{\frac{(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}} \cdot \sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$$

In this section, you are required to complete the denoising process according to the above formula. The formula includes initialization, forward, and backward, which corresponding to init(), add_noise_forward(), and denoise_backward() functions in a class named **GaussianDiffusion**.

Part 1-2: Result Visualization (10%)

- To keep track of the capabilities of the current diffusion model, it's helpful to visualize some results to analyze. First, you need to display the denoising progress for five different images. By plotting intermediate steps, we can analyze efficiency and stability and identify potential issues like over-smoothing or artifacts. Second, you need to plot the loss curve for each training loop. The loss curve can help us to identify whether the model converged.



Part 1-3: Evaluation Baseline (20%)

- In the image generation task, FID scores are commonly used to evaluate the quality of generated images by measuring their similarity to real images. A lower FID score indicates better quality. In our task, we also apply Anime Face Detection (AFD) to help determine whether an image contains a recognizable anime-style face. Using a pre-trained model, AFD ensures that generated faces are valid and coherent. The grading criteria are as follows:

Standard	FID Score	AFD Rate	Grade
Default	≤ 160.0	≥ 0.60	0%
Simple	≤ 120.0	≥ 0.70	4%
Normal	≤ 100.0	≥ 0.80	4%
Medium	≤ 90.0	≥ 0.85	4%
Hard	≤ 80.0	≥ 0.90	4%
Boss	≤ 70.0	≥ 0.95	4%

(You must satisfy both the FID score and AFD rate to receive the full score.)
(e.g. FID 93.4, AFD 0.88 ➤ Simple✓ Normal✓ Medium✗ Hard✗ Boss✗)

Since original code doesn't have a great performance, you need to do some adjustments to improve. Here are some strategies you might want to attempt:

1. **Hyper-Parameter**
2. **Data augmentation**
3. **Deeper model architecture**
4. **Change the beta schedule**
5. **Others**

Please specify what strategies you've used, and your final evaluation results (FID score and AFD rate) in the report.

Part 2: Optical Illusion Generation (25%)

Part 2-1: Prompt Design (5%)

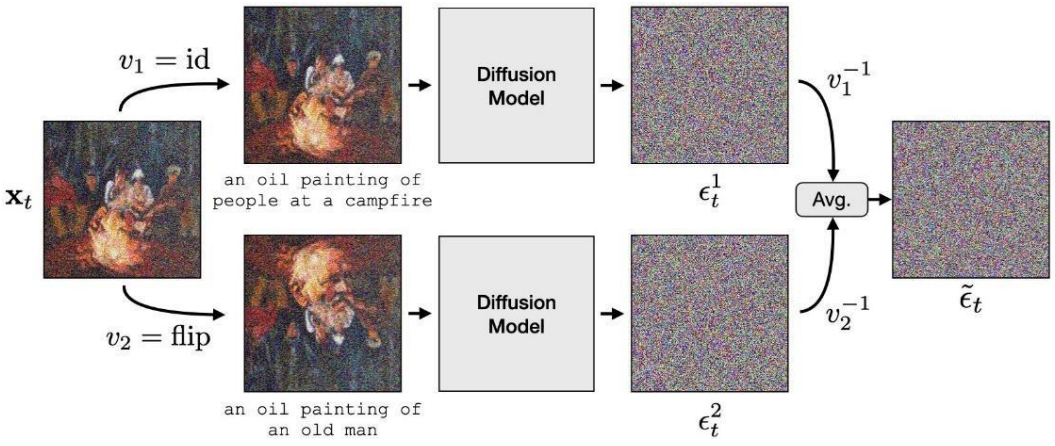
- Before generating an image, you must first determine its content using a text prompt. You need to design two distinct prompts for different perspectives. Your grade will be based on creativity and diversity. For example, if your prompts are "A photo of a wooden house" and "A photo of an upside-down wooden house," they may not be different enough to earn a full score. Additionally, your prompts must be unique and original. If they are identical to someone else's, you will receive 0 points for this section.

Part 2-2: Viewing Transformation (5%)

- We need to generate an optical illusion image that looks different through different views; therefore, the image requires a transformation to represent the view. Here, you are asked to complete IdentityView and Rotate180View. Both of them contain view() and inverse_view() that return the image after applying transformation.

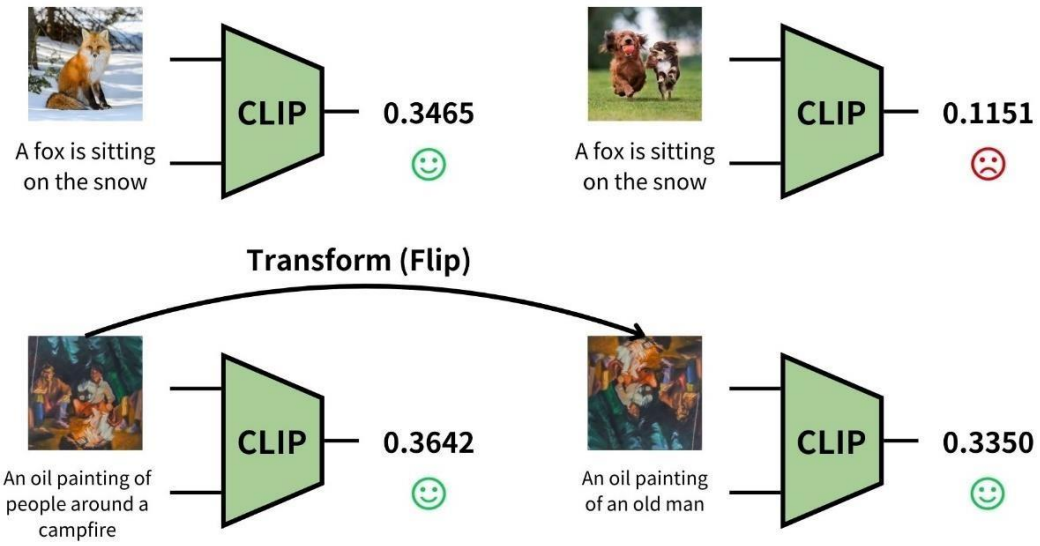
Part 2-3: Denoising Operation (10%)

- Unlike vanilla diffusion, the image should be processed through each viewing function separately, with the diffusion model predicting noise for each view independently. The predicted noises are then applied to their corresponding inverse viewing functions. Finally, the noise is removed by averaging the predicted noises. In the sample code, only the first prompt is considered for denoising. The goal is to implement these operations based on the original code to generate optical illusion images.



Part 2-4: Evaluation Baseline (5%)

- To evaluate the performance of your optical illusion image, we use the CLIP score to measure the similarity between the text and the image. A higher score indicates that the image better represents the text content. In this case, each text-image pair has their own CLIP scores, both of which must exceed 0.3 to receive full points.



Report (30%)

- A report is required.
- The report should be written in **English**.
- Please save the report as a **.pdf** file. (font size: 12)
- Answer the questions in the report template **in detail**.
- The questions are as follows:

1. In the diffusion model, we utilize the U-Net architecture as our model. Please introduce this architecture and explain the concepts of down sampling, bottleneck, and up sampling. Additionally, discuss the functionality and the importance of skip-connections. (10%)

2. In the field of image generation, various deep learning models have been developed to synthesize realistic and diverse images. Among them, Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models are widely used approaches. Please provide a brief description of each and compare their differences. (10%)

3. Since diffusion models are capable of generating diverse and high-quality images, they provide a powerful tool for data generation. Given this capability, do

you think using data augmentation to expand the dataset with diffusion models is a good idea and a beneficial practice? Please give your explanation. (10%)

QA Page

If you have any questions about this homework, please ask them on the following Notion page. We will answer them as soon as possible. Additionally, we encourage you to answer other students' questions if you can.

Link: (preparing…)

Submission

Please prepare your source code, results, and report (.pdf) in `STUDENTID_hw5.zip`, following the file structure. There should **not** be a {student_id}_hw5 folder in the zip file.

```
{student_id}_hw5.zip
├ Part1
|  ├ generated
|  |   ├ 1.jpg
|  |   ├ 2.jpg
|  |   ├ …
|  |   └ 1000.jpg
|  ├ model.pt
|  └ diffusion.ipynb
|
├ Part2
|  ├ result.jpg
|  └ multi-view.ipynb
|
└ report.pdf
```

e.g. 112550999_hw5.zip

Wrong submission format leads to -10 points.

Late Submission Policy

20% off per late day

Reference

[1] Machine Learning Material from Hung-yi Lee
[2] Denoising Diffusion Probabilistic Models (NeurIPS 2020)
[3] Denoising diffusion implicit models (ICLR 2021)
[4] DeepFloyd IF on Hugging Face
[5] Visual Anagrams (CVPR 2024 oral)

Appendix

Sample from Gaussian Distribution

when we have such equation: $p(x) = \mathcal{N}(x;\ \mu,\ \sigma^2 I)$

it can be infered like this: $x = \mu + \sigma * \epsilon \quad where\ \epsilon \sim \mathcal{N}(0, 1)$

random noise in code

Fréchet Inception Distance (FID Score)