CS791: Programming Assignment 2

September 3, 2025

General Instructions

1. Code plagiarism will be strictly penalized including but not limited to reporting to DADAC and zero in assignments.
2. If you use tools such as ChatGPT, Copilot, you must explicitly acknowledge their usage in your report. Code borrowed from other sources must be cited in the comments.
3. Submit a report explaining your approach, implementation details, and results. Clearly mention the contributions of each team member in the report.
4. If you use external sources (e.g., tutorials, papers, or open-source code), you must cite them properly in your report and comments in the code. Submit your code and report as a compressed `<TeamName>_<student1rollno>_<student2rollno>_<student3rollno>.zip` file. Fill a student as `NOPE` if less than 3 members.
5. Start well ahead of the deadline. Submissions up to one day late will be capped at 80% of the total marks, and no marks will be awarded beyond that.
6. Do not modify the environment provided. Any runtime errors during evaluations will result in zero marks. README.md provides instructions and tips to set up the environment and run the code.
7. For most of the assignment, you have to fill in your code in already existing files. Apart from report, do not submit any additional models and files unless explicitly asked. Any additional files should be placed in the top-level directory.
8. The python files provided are, d3pm.py, d3pm_cond.py, ddpm.py, ddpm_cond.py, models.py, utils.py and scheduler.py. You need to submit these files along with model states, generated samples and the report.
9. Submission of model states is strictly necessary. Not submitting the model states will be penalized.

1 Problem Statement

1.1 Discrete Denoising Diffusion Probabilistic Models (D3PMs)

In this part, you will implement a Discrete Denoising Diffusion Probabilistic Models (D3PMs) Austin et al. 2021 on MNIST dataset. It is strongly recommended that you go through the paper before you start the assignment. You only have to implement the **Absorbing State Discrete Diffusion** as in the paper.

1. Implement the models `D3PM` and `ConditionalD3PM` in the **models.py** file for unconditional and conditional generation, respectively.

2. Implement the training and sampling functions in **d3pm.py** and **d3pm_cond.py**.

3. Study the effect of different masking schedules, specifically *linear* and *cosine* (Zhang 2025).

4. Analyze the effect of the number of diffusion steps, considering at least three different values.

5. Report the FID scores for all of these experiments. For conditional generation report the average FID score over all the different classes.

utils.py provides the function to calculate the FID score which you need use to evaluate the quality of your samples. (**Disclaimer**: Calculating the FID score between 64 actual and generated images takes about 80 secs. So, to calculate the final FID score you can sample 64 from a larger pool images a few number of times a return their average.)

1.2 Denoising Diffusion Probabilistic Models (DDPMs)

In this part, you will implement a Denoising Diffusion Probabilistic Models (DDPMs) Ho, Jain, and Abbeel 2020 on MNIST dataset. It is strongly recommended that you go through the paper before you start the assignment.

1. Implement the models `DDPM` and `ConditionalDDPM` in the **models.py** file for unconditional and conditional generation, respectively.

2. Implement the training and sampling functions in **ddpm.py** and **ddpm_cond.py**.

3. Study the effect of different masking schedules, specifically *linear* and *cosine* (Nichol and Dhariwal 2021).

4. Analyze the effect of the number of diffusion steps, considering at least three different values.

5. Report the FID scores for all of these experiments. For conditional generation report the average FID score over all the different classes.

utils.py provides the function to calculate the FID score which you need use to evaluate the quality of your samples. (**Disclaimer**: Calculating the FID score between 64 actual and generated images takes about 80 secs. So, to calculate the final FID score you can sample 64 images from a larger pool a few number of times a return their average.)

2 Submission

1. **report.pdf**: The report should contain detailed explanations of the model architectures, training and sampling procedures. It must also include observations from the experiments along with their corresponding FID scores and hyperparameters.

2. **generate.py**: A script to load the trained models and generate samples for each of the four models. The samples should be named as **samples_ddpm.pt**, **samples_ddpm_cond_0.pt**, **samples_ddpm_cond_1.pt**, . . ., **samples_ddpm_cond_9.pt** for the DDPM models. Use similar naming conventions for the D3PM samples. Each of these files should contain 64 samples, i.e., tensors of shape $(64, 1, 28, 28)$.

3. All model weight files required for generating the samples.

4. Source code files: **ddpm.py**, **ddpm_cond.py**, **d3pm.py**, **d3pm_cond.py**, **scheduler.py**, **utils.py** and **models.py**.

References

Austin, Jacob et al. (2021). “Structured denoising diffusion models in discrete state-spaces”. In: *Advances in neural information processing systems* 34, pp. 17981–17993.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Nichol, Alex and Prafulla Dhariwal (2021). *Improved Denoising Diffusion Probabilistic Models*. arXiv: `2102.09672 [cs.LG]`. URL: `https://arxiv.org/abs/2102.09672`.

Zhang, Leo (2025). *The Cosine Schedule is Fisher-Rao-Optimal for Masked Discrete Diffusion Models*. arXiv: `2508.04884 [stat.ML]`. URL: `https://arxiv.org/abs/2508.04884`.