# Deep Learning Project 3

Dutt Salveen Singh, Patryk Prusak

supervisor

mgr inż. Stanisław Kaźmierczak

Warsaw University of Technology

November 18, 2024

## Contents

# 1 Problem Description

The aim of this project is to test and compare different architectures of image generators, with the constraint that at least one of the examined models must be capable of generating satisfactory images. The dataset used for training and testing the models should be the 10% sample of the LSUN Bedrooms Dataset that consists of a number of images of bedrooms. Some suggested architectures are: vanilla pixel diffusion, DDPM, Improved DDPM, and Stable Diffusion. The Fréchet Inception Distance (FID) should be computed for the generated images and the results should be compared to those from literature. A number of experiments should be conducted to investigate the influence of hyperparameters on the quality of the results.

# 2 Application Instruction

The solution has been prepared in Python Jupyter Notebook. It is enough to install the required packages (visible at the beginning of the notebook) and run cells one by one. The preferable folder with the downloaded and extracted dataset is also visible at the beginning of the notebook. Note that seeds are predefined to ensure reproducible randomness.

# 3 Theoretical Introduction

The topic of this project is image generation. One of the most common ways to generate images with machine learning approaches is to utilize some kind of diffusion model. The diffusion model, as the name suggests involves diffusion, that is turning a sensible image to noise. The training images are transformed into noise and then the neural network is trained to denoise the images. As a result, we get a model that is capable of generating images from noise [1]. In our solution, we make use of a Denoising Diffusion Probabilistic Model (DDPM) [2] and Score Matching with Langevin Dynamics (SMLD) [3] that fall into the category of diffusion models. The architecture of the tested DDPM consists of a U-Net model with 3 down blocks and 3 up blocks, coupled with a DDPM Scheduler provided by PyTorch and HuggingFace [4], that takes the output of the model and returns a denoised sample. Similarly, in the case of SMLD, we utilize the same U-Net architecture but with a custom implementation of an SMLD Scheduler. All the models operate on images scaled to 64 by 64 pixels.

# 4 Experiments

A number of experiments have been conducted to investigate the influence of optimizers with different learning rates, number of epochs, scheduler's training steps, and batch sizes. The model's performance was measured in terms of Fréchet Inception Distance, and so the parameters have been chosen to minimize this metric. After splitting the dataset into train and test sets, with 10% test size the procedure was as follows:

1. Choose a starting parameter X (such as batch size) and answer the following question: given a set of basic (commonly seen in literature) parameter values (such as a hyperparameter) excluding parameter X, what value for X results in best accuracy?

2. Choose the best-found value for parameter X and replace the parameter's basic initial value.
3. Repeat 1. with the new parameter's X value and choose a different parameter X.
4. The procedure finishes after investigating all parameters of interest.

Each experiment has been repeated 5 times on 3% of the original dataset given time and computing power constraints. The following parameter values have been taken into consideration:

- Batch size: 128, 64 and 32
- Learning rate: 0.0001, 0.001, and 0.01
- Optimizers: Adam, AdaDelta, SGD (with each of the different learning rates)
- Training Steps: 200, 500, 100
- Number of epochs: 2, 5, 10

As a last step, a number of final images have been generated by each model, for which we also present the FID.

## 5 Results

The results of the experiments are as follows. Figure 1 depicts FID scores for different combinations of optimizers and learning rates for the DDPM model, note that outliers have been removed from all the figures for clarity. Having in mind that the lower the score the better, one can see that Adam and Stochastic Gradient Descent (SGD) optimizers perform best with 0.0001 and 0.001 learning rates for Adam and 0.01 for SGD. The same category of results for SMLD is presented in figure 2, here the results are different, Adam with 0.0001 and 0.001 learning rates achieves the worst results, whereas all the other combinations perform very similarly.

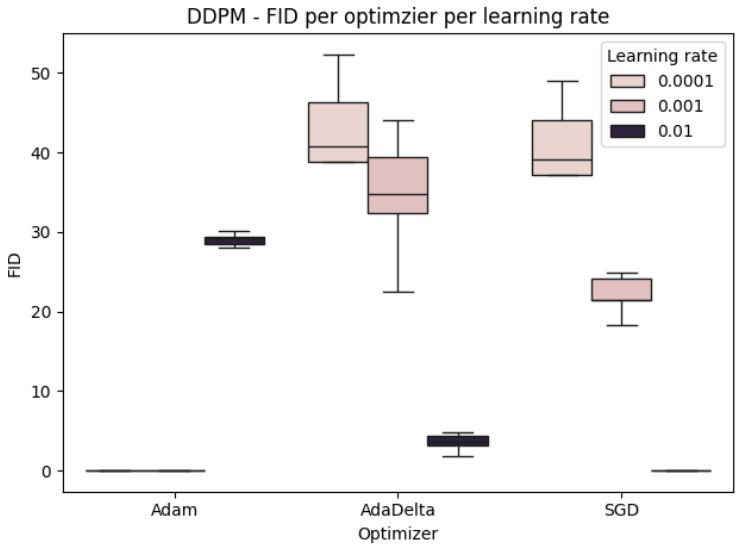


Figure 1: FID per optimizer for DDPM

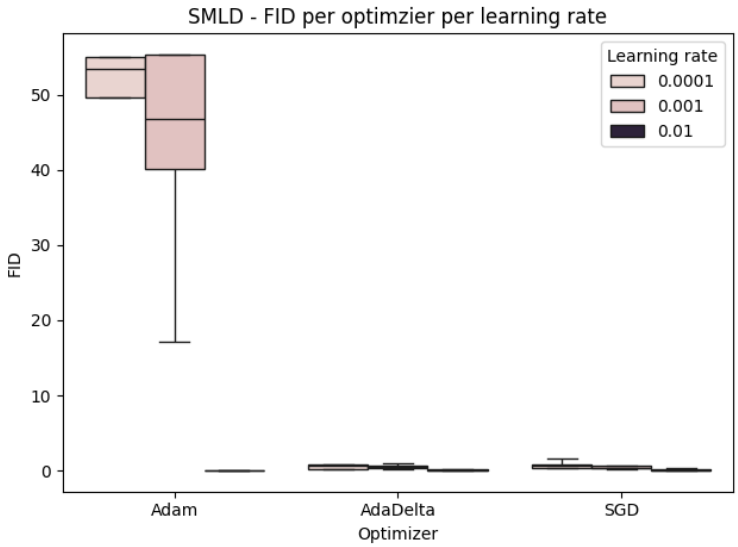


Figure 2: FID per optimizer for SMLD

In terms of the scheduler's training steps, it seems to be the case that with the increase of steps the FID lowers, which means the performance of the model increases, that seems to be the case, especially for DDPM architecture as visible in figure 3.

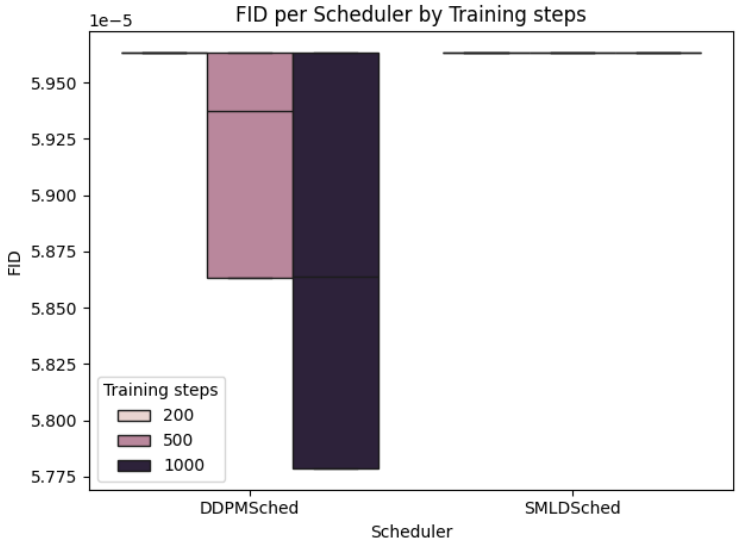


Figure 3: FID per training steps

In the case of a number of training epochs, the situation is not as clear. Naturally one would assume that with the increase of the number of epochs (up to some point) the performance of the models should also increase, here (figure 4) however the scores remain quite similar and it is not clear what would be the optimal number of epochs from the obtained results. It's worth noting that with the increase in the number of epochs the variance also increased, exact values for all means and variances are available in the attached solution notebook in the section "Combine results".

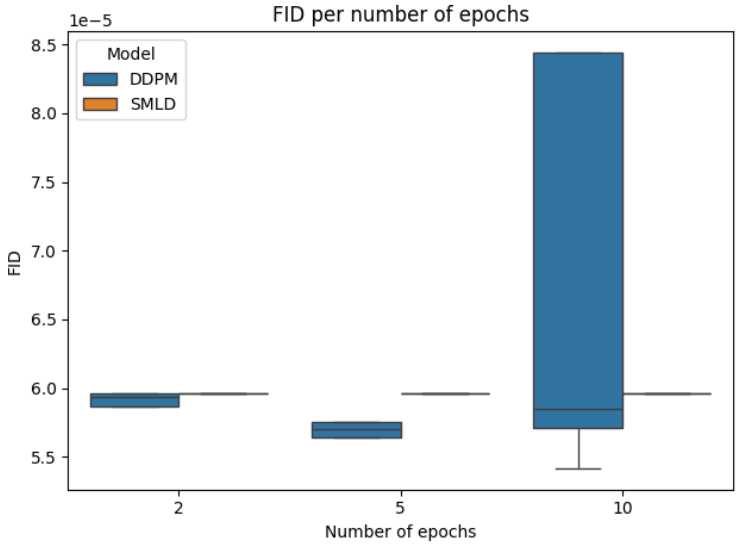


Figure 4: FID per number of epochs

The effects of change in batch size are not surprising. Figure 5 clearly shows that the lower the batch size the better the results in general, especially for DDPM architecture, for SMLD it doesn't seem to have as strong of an effect.

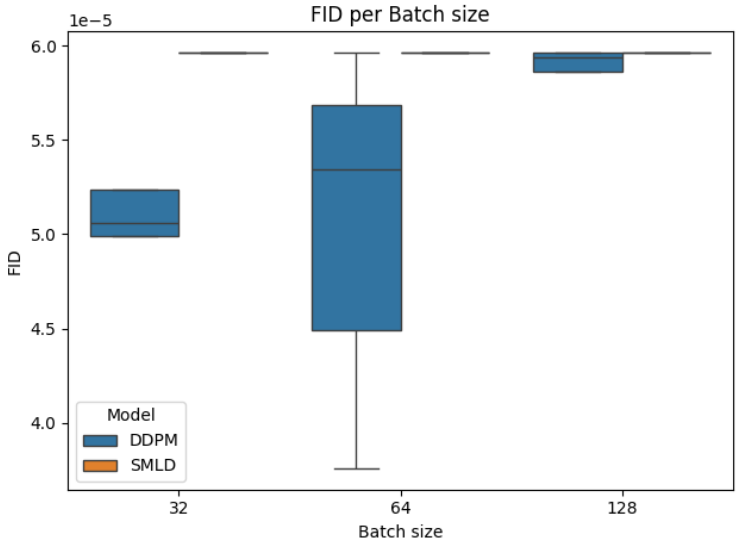


Figure 5: FID per batch size

Finally, we present satisfactory images of size 64x64 generated from the DDPM model with Adam optimizer, 0.0001 learning rate, 1000 training steps, trained for 20 epochs on the whole provided dataset in figures 6 and 7, more examples are available in files delivered with the report. Sadly, SMLD didn't produce satisfactory results. Final FID scores calculated between the 50% of the dataset (trying to compute FID on the whole dataset caused kernel crashes) and generated images for each of the models are 3.1919e-05 for DDPM and 5.58e-02 for SMLD. The score for

SMLD is significantly higher (worse) which is also reflected in the quality of generated images. One can compare these results to others available online such as the leaderboard of "Image Generation on LSUN Bedroom 256 x 256" from paperswithcode.com [5]. In such a comparison the results obtained by us seem significantly better, however, the comparison cannot be considered entirely accurate, since image scaling might affect the FID results.

Figure 6: DDPM generated image 1

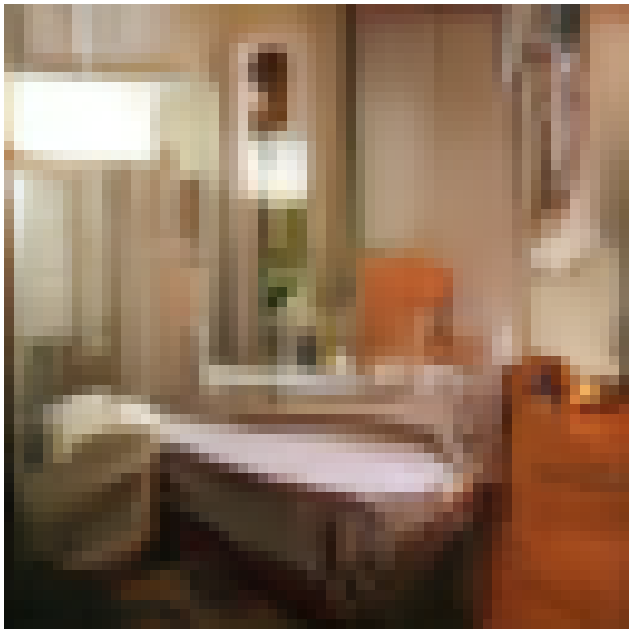

Figure 7: DDPM generated image 2

## 6 Conclusions

We have successfully investigated the effects of the following parameters: batch size, learning rate, number of epochs, optimizer type, and training steps. Many experiments have been performed to analyze their impact on the Fréchet Inception Distance of the chosen models in the task of generating images resembling bedrooms. Having processed all the data from experiments we have chosen the most promising parameters with which to train and test the final models. We have obtained satisfactory results with the DDPM model, however, we have failed to fine-tune the SMLD model to generate reasonable bedroom images. Lastly, more parameters for the SMLD could be investigated, since it is possible that a certain combination of parameters, that we have overlooked, would result in satisfactory image-generation capabilities.

## List of Figures



## References


[1] Ling Yang, Zhilong Zhang, and Yang Song. "Diffusion Models: A Comprehensive Survey of Methods and Applications". In: (2022). DOI: https://doi.org/10.48550/arXiv.2209.00796.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: (2020). DOI: https://doi.org/10.48550/arXiv.2006.11239.

[3] Ziqing Wen, Xiaoge Deng, and Ping Luo. "Score-based Generative Models with Adaptive Momentum". In: (2024). DOI: https://arxiv.org/html/2405.13726v1.

[4] *HuggingFace Diffusion models*. URL: https://huggingface.co/docs/diffusers/index.

[5] *PapersWithCode Lsun leaderboard*. URL: https://paperswithcode.com/sota/image-generation-on-lsun-bedroom-256-x-256.