Project Machine Learning
— Milestone 3 —

Seif Daknou, Patrick Lindemann, Nguyen Pham

February 2, 2024

1 Introduction

In the project's final phase, our primary focus is on refining the implementation of the Denoising Diffusion Probabilistic Model (DDPM) framework to achieve image quality comparable to the original DDPM paper by Ho et al. (2020). We begin by refining the model architecture and improving both the diffusion process and training pipeline. Further, we experiment with the schedulers introduced in Milestone 1 and fine-tune the model- and training-related hyperparameters to identify the best configuration for our setup. Ultimately, we compare our results based on the CIFAR-10 dataset using the *Inception Score* Szegedy et al. (2015), which we implemented in the last milestone, and present our results based on other datasets.

2 Improvements

In this section, we outline the modifications made to our model and the improvements which we integrated into the training process.

Revised Model Architecture. Moving away from the U-Net structure from milestone 2, which consisted of six down- and upward blocks comprising five residual blocks and one self-attention block similar to *PixelCNN++*[1] used by Ho et al. (2020), we transition to a more streamlined configuration with four down- and upward blocks, consisting of one residual block and three self-attention blocks (see Figure 1). This adjustment is motivated by the specific constraints of our project, focusing on 64×64 and 32×32 images, dictated by cost considerations associated with the available hardware for training. The revised architecture yields improved the quality of the generated images, which we attribute to the fact that images with resolutions lower than 64×64 are reduced by the U-Net to a one-dimensional tensor in the bottleneck, resulting in a loss of meaningful information in the convoluted image.

Addition of an EMA model. During the iterative training process, the denoising model performs ongoing parameter updates with the primary goal of improving its ability to generate realistic images. Alternatively, instead of utilizing the most recent parameters directly, it is possible to use the Exponential Moving Average (EMA) to maintain a dynamically diminishing average of the generator's parameters as the training progresses Efimov (2024). Thus, we implement a second model variant with the EMA in order to encapsulate a refined and stabilized iteration of the generator's parameters, which leads to a more consistent and reliable image generation process.

[1] `https://github.com/openai/pixel-cnn`

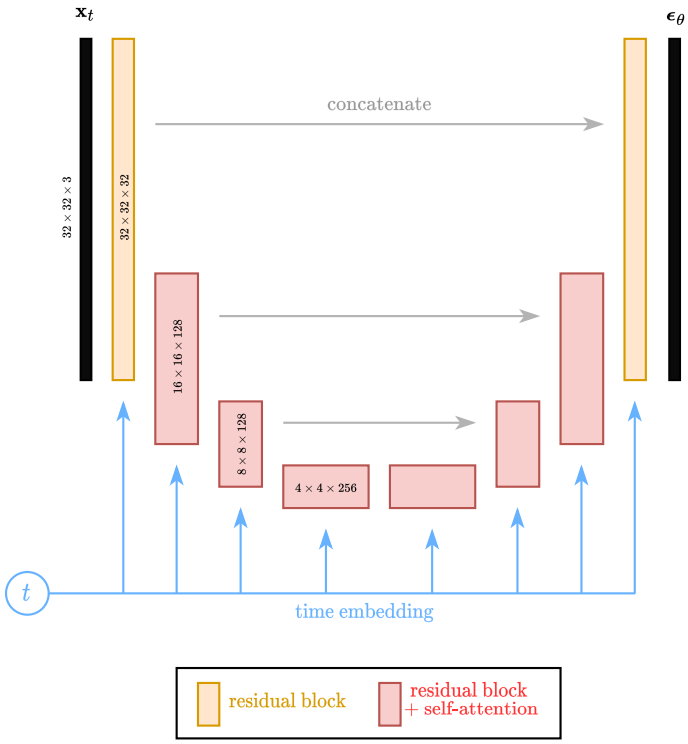

Figure 1: The updated architecture of the 32×32 noise prediction U-Net model. The same model is used for all image sizes.

Updates to the diffuser. In Milestone 1, we introduced four types of constant *schedules* that determine the variances β_t of noises during the forward diffusion process at each timestep $1 < t < T$, namely

- *linear schedules*, which were used in the original DDPM framework by Ho et al. (2020),
- even-degree *polynomial schedules* that introduce either a "warm-up" and "cool-off" phase to the variances depending on the exponent 2τ,
- *cosine schedules* proposed by Nichol and Dhariwal (2021) which combine the "warm-up" and "cool-off" phases into one schedule, and
- *sigmoid schedules* introduced by Chen (2023) which have trajectories similar to their cosine counterparts.

We incorporate these schedules into the diffuser and evaluate their impact on the in section 3.

Refinement of training parameters. In addition to the *ADAM* optimizer (Kingma and Ba (2015)) introduced in milestone 2, we integrate *StepLR* as a dynamic learning rate scheduler into the training process. This scheduler adjusts the learning rate by a factor of 0.85 every five steps, as illustrated in Figure 2, providing an adaptive mechanism that facilitates optimal convergence and prevents overshooting. Further, we introduce a dropout rate of 0.01 into the model for regularization purposes. This dropout technique randomly deactivates a fraction of neurons during each training iteration, enhancing generalization and reducing the risk of overfitting. These refinements collectively contribute to a more resilient and efficient training process.

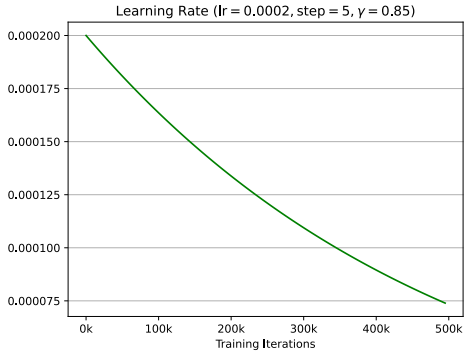

Figure 2: Adaptive learning rate used during training over 100 epochs on the CIFAR-10 dataset.

3 Evaluation

In this section, we conduct a model selection for different kinds of schedulers, evaluate the images generated by our implementation based on different datasets and discuss our results.

3.1 Model Selection

In our model selection process, we employ the (unconditional) CIFAR-10 dataset[2], containing 60,000 32×32 images. From this dataset, we randomly select a subset of $N = 6,250$ images, on which we perform an 80/20 train/test split. This allocates 5,000 images for training and 1,250 images for validation. The model undergoes training for 100 epochs, resulting in 500,000 training iterations in total, and is tested after each epoch on the . Subsequently, we generate 500 images for each model and calculate their Inception Score. This methodology closely follows the approach of the study by Chen (2023), ensuring the comparability of our obtained scores with the results thereof. We document the common parameters of all runs in Table 1.

Parameter	Value
Time Steps T	1000
Subset Size	6,250
Training Set Size	5,000
Validation Set Size	1,250
Training Epochs	100
Batch Size	16
Loss Function	MSE
Optimizer	ADAM
Initial Learning Rate	$2 \cdot 10^{-4}$
Dropout Probability	0.1

Table 1: The training hyper-parameters chosen for the model selection process.

Our focus during model selection involves experimenting with various schedulers and their parameterizations. The linear scheduler is parameterized identically to Ho et al. (2020). The parameters for the cosine and sigmoid schedules are determined based on the optimal results reported by Nichol and Dhariwal (2021) and Chen (2023). This comprehensive approach to model selection ensures a thorough exploration of scheduler types and their parameter variations. We provide the schedule plots in Figure 3 and list the resulting Inception Scores in Table 2.

[2] `https://www.cs.toronto.edu/~kriz/cifar.html`

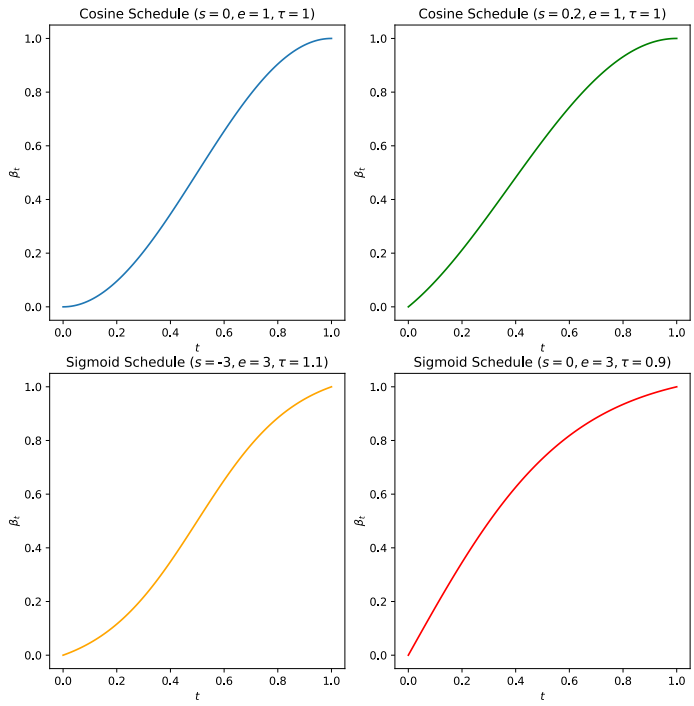

Figure 3: The cosine and sigmoid schedules used for the model selection process.

Schedule	Inception Score
linear $(s = 0.0001, e = 0.02)$	5.5970
cosine $(s = 0.0, e = 1.0, \tau = 1.0)$	7.854
cosine $(s = 0.2, e = 1.0, \tau = 1.0)$	X
sigmoid $(s = -3.0, e = 3.0, \tau = 1.1)$	X
sigmoid $(s = 3.0, e = 3.0, \tau = 0.9)$	X

Table 2: Inceptions Scores based on CIFAR-10 for 500k training iterations.

.

3.2 Results

We present the results of our CIFAR-10 experiments using the cosine schedule, comparing the performance of the standard U-Net model (Figure 4) with its EMA-enhanced counterpart (Figure 5). Sampling 10,000 images, the non-EMA model achieved an inception score of 7.854, equivalent to 83.02% of the original paper's score of 9.46. In contrast, the EMA-enhanced model yielding an inception score of 7.932, corresponding to 84.39% of the benchmark set by the original paper and a marginal improvement of approximately 1% compared the non-EMA model.

The diffusion model was employed for training on two additional datasets: FGVC-Aircraft[3] and Flowers102[4]. The Flowers102 images (Figure 8) yielded an inception score of 2.1, while the FGVC-Aircraft images (Figure 7) achieved a score of 3.5. Notably, the latter demonstrated a substantial improvement, approximately 60% better than the score of 2.076 of the images generated for our last report.

Given our utilization of the same algorithm as the original paper, coupled with the cosine schedule, EMA, and other enhancements, we posit that further increments in the inception score could be achieved through larger image sizes, increased training iterations, and a higher number of generated images for score calculation. This suggests potential avenues for future improvements in our generative model.

[3] `https://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/`
[4] `https://www.robots.ox.ac.uk/~vgg/data/flowers/102/`

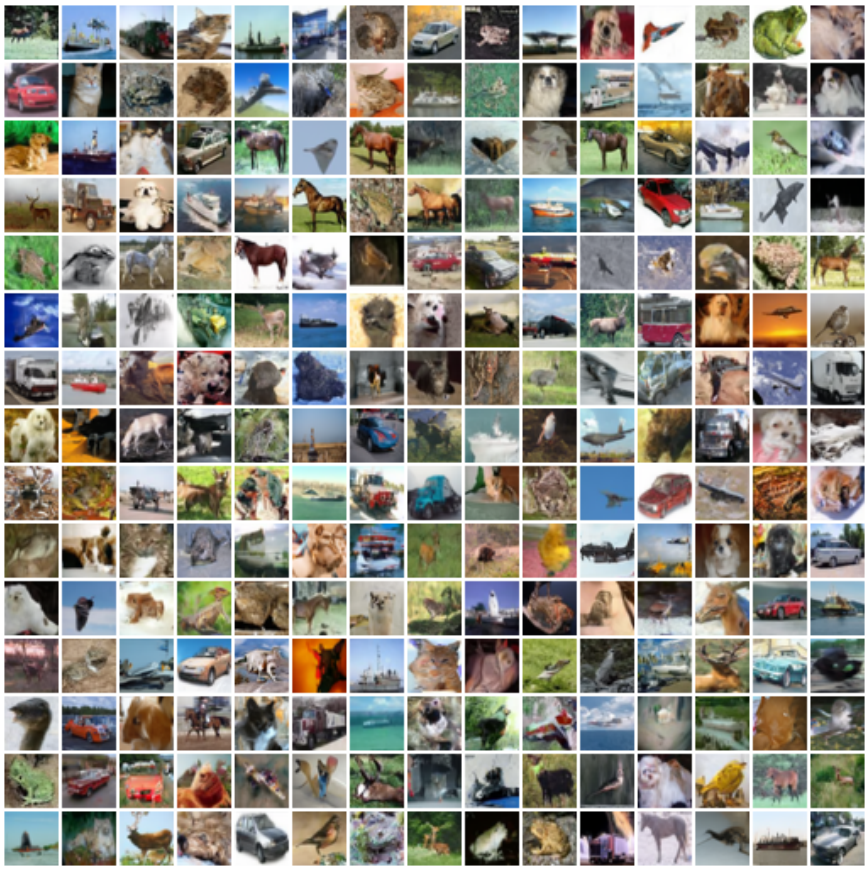

Figure 4: Sampled 32×32 images based on the CIFAR-10 dataset.

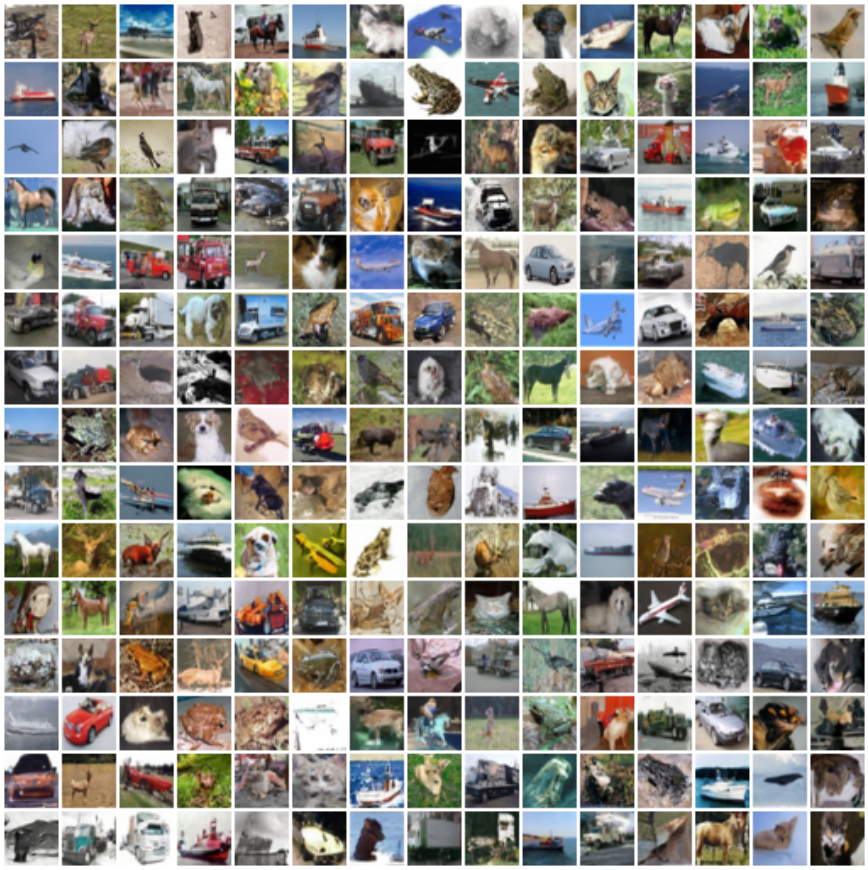

Figure 5: Sampled 32×32 images based on the CIFAR-10 dataset enhanced with EMA.

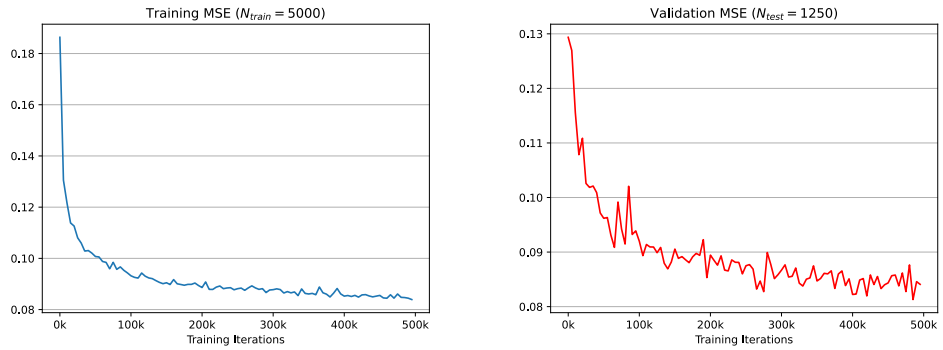

Figure 6: Train and test losses of the model with the plain cosine schedule during training over 100 epochs on the CIFAR-10 dataset.

4 Conclusion

In three milestones, we successfully implemented the DDPM framework by Ho et al. (2020), including a denoising U-Net with a customized architecture. We integrated peer-suggested modifications that significantly improved the quality of our generated images. Our experimentation spanned the usage of various datasets, schedulers, model architectures, and training hyper-parameters, allowing us to explore the model's capabilities. Successfully generating realistic images, we achieved an inception score within 83% of the original paper's CIFAR-10 benchmark. This marks the conclusion of our project and serves as a baseline for future enhancements.

References

T. Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

V. Efimov. Intuitive explanation of exponential moving average, Jan 2024. URL `https://towardsdatascience.com/intuitive-explanation-of-exponential-moving-average-2eb9693ea4dc`.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf`.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. URL `http://arxiv.org/abs/1512.00567`.

Figure 7: Sampled 64×64 images based on the FGVC-Aircraft dataset.

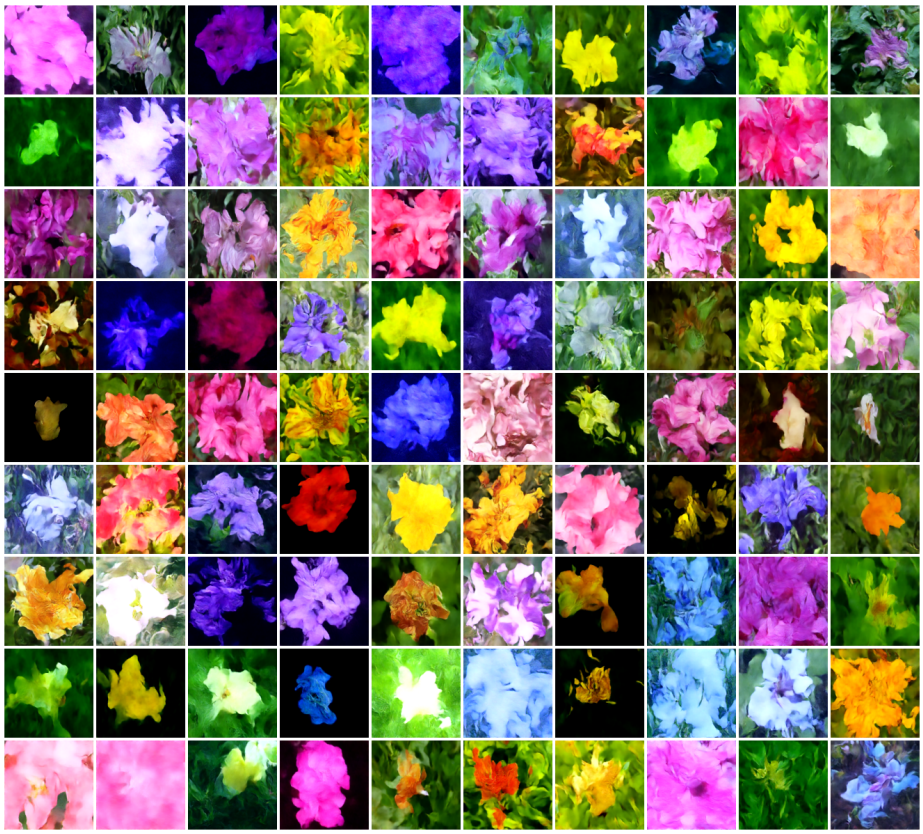

Figure 8: Sampled 128×128 images based on the Flower102 dataset.