Project Machine Learning
— Milestone 1 —

Seif Daknou, Patrick Lindemann, Nguyen Pham

November 24, 2023

1 Introduction

This report presents a prototype of a simple Denoising Diffusion Probabilistic Model (DDPM) as proposed by Ho et al. (2020). The primary focus of this milestone centers on the forward process implementation and the integration of a fundamental U-Net architecture for denoising tasks.

Throughout this milestone, we dive into of the implemented forward process using different noising techniques and testing multiple facets of the noise scheduler. Additionally, the incorporation of the U-Net architecture for denoising is explored, showcasing its adaptability and performance in handling the denoising task. The report aims to provide a comprehensive understanding of the prototype's structure and functionality laying the groundwork for the next milestone.

2 Datasets

To test our pipeline in the context of the first milestone, we used several the same datasets as the original paper.

CIFAR-10 by Krizhevsky and Hinton (2009) CIFAR-10 is a dataset of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The classes include common objects like airplanes, automobiles, birds, cats, etc. It has a balanced distribution, with an equal number of images per class. It is considered a clean dataset with no missing values and no duplicates are present in the original dataset. Although, Images in CIFAR-10 are of relatively low resolution they are suitable for training small-scale models. The dataset is widely used for benchmarking and research due to its simplicity and diversity.

CelebA (CelebFaces Attributes Dataset) by Liu et al. (2015) contains images of celebrities. Targets include various attributes such as gender, age, and presence of accessories like sunglasses. CelebA has a large number of images, around 200,000. The dataset is well-distributed with a diverse set of celebrities, ensuring a broad representation of different attributes. CelebA is known for having near-duplicate images, often from similar poses or lighting conditions. It's therefore crucial to check for near-duplicates to avoid introducing bias.

LSUN by Yu et al. (2015), or Large-scale Scene Understanding, stands out as a comprehensive collection tailored for scene recognition tasks in computer vision. Featuring millions of labeled images, LSUN covers a broad spectrum of scenes, including bedrooms, kitchens, living rooms, and outdoor environments. Researchers leverage this dataset to train and evaluate models in tasks related to scene understanding, object detection, and other applications that require a nuanced understanding of visual contexts.

The impact of noise schedulers on different datasets is a crucial consideration in the context of denoising process. The size and resolution of images in datasets such as CelebA, which contains larger and more detailed images compared to images from CIFAR-10, can indeed influence the noise requirements, as shown by Chen (2023). Larger images might demand more intricate noise patterns to effectively capture and model the variability in the data.

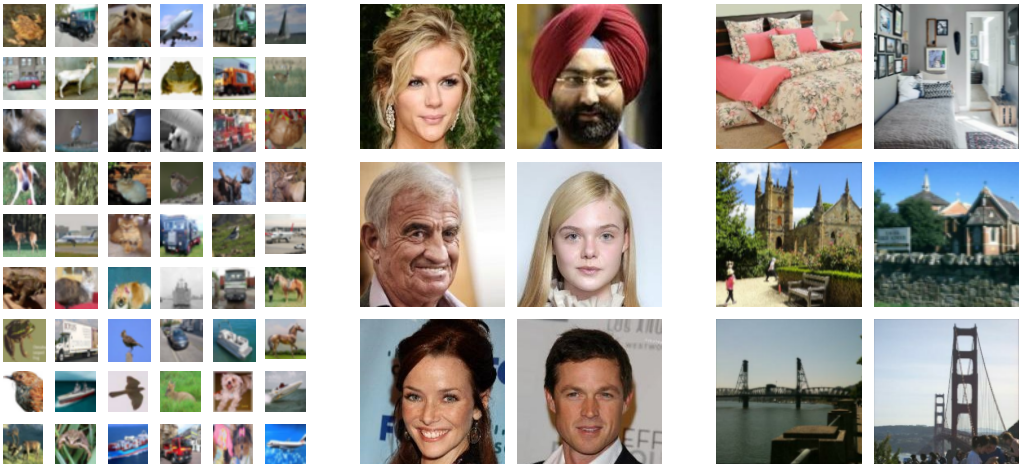


Figure 1: Example images from the CIFAR-10 (left), CelebA (middle) and LSUN (right) datasets.

2.1 Transformations

In the preprocessing pipeline, each dataset is converted into tensors using the PyTorch library. Images originating from the mentioned datasets notably the CIFAR10 dataset have pixel values within the range of [0, 255]. To establish a consistent range for the data, a normalization process is applied, scaling the pixel values to [-1, 1]. This enforces the pixel values into a fixed range that is zero centered and proves advantageous when dealing with Rectified Linear Unit (ReLU) activations. Other transformations such as center cropping and image resizing are also applied. Conversely, reverse transform pipelines were designed to revert the processed data to a format compatible with visual representation.

3 Forward Process

The forward process shown in figure 2 involves the distribution $q(x_t, x_{t-1})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$$q(x_t, x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I})$$

with

$$q(x_t, x_0) = \prod_{t=1}^{T} P(x_t, x_{t-1})$$

where $(\beta_0, \ldots, \beta_T)$ with $0 \leq \beta_0 \leq \cdots \leq \beta_T \leq 1$ are fixed constants which are determined by a *schedule*.

After reparameterization of β_t with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ we obtain:

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

This trick allows us to sample x_t at any time step t as well as a more stable training when using gradient based optimisation techniques.

3.1 Schedules

The original paper by Ho et al. (2020) fixes the forward process variances β_t to be constant and generated by a *linear schedule* with $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Yet, alternative schedules such as the *cosine*-based [Nichol and Dhariwal (2021)] or *sigmoid*-based schedules [Jabri et al. (2022)] have been shown to yield better results when used within the diffusion process. The importance of the noise schedule has been studied further in recent work by Chen (2023), which concludes that choosing the right scheduler and it's

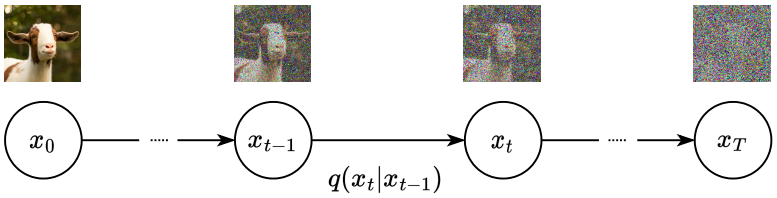


Figure 2: Through the forward process, noise is added incrementally to the image x_0 until it becomes random noise.

parameters is crucial for the model performance and should be selected dependent on the task and image size.

It is mentioned by the original authors Ho et al. (2020) that the variances β_t can be learned by reparameterization, however, they leave that to future work. Kingma et al. (2021) do this through a sigmoid parameterization of the variances $\beta_t = \sigma(\gamma_\theta(t))$ where $\gamma_\theta(t)$ is a monotonic neural network with learned parameters θ. In this report, we exclusively focus on constant schedulers, but leave open the possibility of implementing trained schedulers for the next milestones.

In the following, we use a similar notation as Chen (2023). We define a schedule as a one-dimensional function $\gamma : [0, 1] \to [0, 1]$ that maps a time step t to the variance β_t. In our implementation, we vectorize this operation to allow for efficient calculations for vectors of randomly chosen time steps $\boldsymbol{t} \in \mathcal{U}(0, 1)$ during the training process.

3.1.1 Linear Schedule

With the linear schedule $\gamma(t) = t$, noise is added to the image at a constant rate over time. Additionally, the parameters $s, e \in [0, 1]$ with $s \leq e$ are introduced to control the starting and ending values of the schedule such that $\gamma(t_0) = s$ and $\gamma(t_T) = e$. While this is straightforward to implement, the noteable drawback is that the pertient image information is destroyed too fast during the forward process. Furthermore, since the rate of change for the variances is not adjustable, the scheduler is inflexible in comparison to other schedules presented in this report.

3.1.2 Polynomial Schedule

The polynomial schedule $\gamma(t) = t^\eta$ with $\eta \in \mathbb{R}^+ \setminus \{0\}$ incorporates a parameterized exponent that delineates both the slope and the trajectory of the curve. When employing the quadratic schedule with $\eta = 2$ (as seen in figure 3), a prolonged "warm-up" phase is observed in comparison to the linear schedule, which results in a more gradual introduction of noise at the outset, accelerating as the forward process progresses. On the other hand, using square root schedule with $\eta = \frac{1}{2}$ results in the opposite by introducing more noise at the beginning and containing a "cool-off" phase during the end.

3.1.3 Cosine Schedule

The cosine schedule $\gamma(t) = \cos\left(\frac{\pi}{2}t\right)^{2\eta}$ combines the warm-up and cool-off phases of the polynomial schedule with a constant rate of change in between. The exponent is parameterized by $\eta \in \mathbb{R}^+ \setminus \{0\}$, which again governs the degree of steepness in the curve. To prevent negative values, the exponent is made even with the factor 2. In addition, the parameters $s, e \in [0, 1]$ contribute to the slope as well, which can be observed in figure 4.

3.1.4 Sigmoid Schedule

Finally, the sigmoid schedule $\gamma(t) = \frac{1}{1+\exp\left(-\frac{1}{\eta}t\right)}$ delineates a trajectory characterized by an escalating rate of change from the initial time step t_0 to its midpoint at $t = \frac{1}{2}$, followed by a subsequent decrease in change until the end at t_T. Unlike the cosine schedule, the sigmoid schedule lacks a phase of constant change within its temporal span. The parameters $s, e \in \mathbb{R}$ determine the direction of the trajectory, while $\eta \in \mathbb{R}^+ \setminus \{0\}$ serves as a determinant of the curve's steepness, which is depicted in figure 4.

Figure 3: The forward process for schedules with default parameters $s = 0$ and $e = 1$: linear (first), quadratic (second), cosine (third) and sigmoid (last)

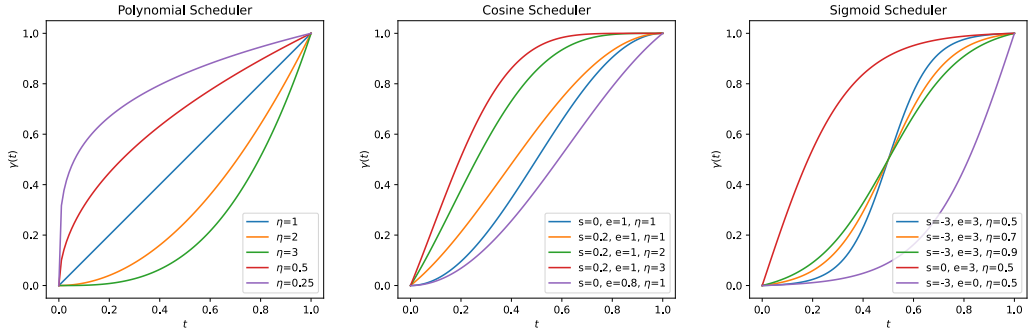


Figure 4: The variances $\gamma(t)$ generated by the polynomial (left), cosine (middle) and sigmoid (right) schedulers for different values for s, e and η.

3.2 Discussion

The exploration of different schedulers has unveiled distinct behaviors and influences on the injected noise during the forward process. The comparison among various schedulers, each defined by a unique set of betas, provides valuable insights into the temporal evolution of the noise. Our preliminary observations indicate that the choice of schedulers significantly impacts the amplitude and structure of the noise added to the images. For instance, certain schedulers with slower decay rates seem to introduce noise more gradually, allowing the model to capture finer details over time. On the other hand, faster-decaying schedulers tend to produce noisier images earlier in the diffusion process. This finding emphasizes the importance of scheduler selection in shaping the noise characteristics, and consequently, the quality of denoising achieved by the model. As we progress we will further delve into the quantitative assessment of different schedulers, revealing their specific contributions and trade-offs in the DDPM framework.

4 Denoising

The denoiser model discussed in this study was constructed utilizing the PixelCNN++ framework, itself rooted in the U-Net architecture Ho et al. (2020). Therefore, for milestone 1, we implemented a conventional U-Net as our denoising model, utilizing solely the noisy image as input to predict the characteristics of the introduced noise. The architectural representation of this approach is depicted in Figure 5.

U-Net is a convolutional neural network architecture primarily designed for biomedical image segmentation tasks, renowned for its effectiveness in precise pixel-wise segmentation. Developed by researchers at the Computer Science Department of the University of Freiburg, it features a unique symmetric and contracting-expanding pathway structure. U-Net has proven instrumental in various medical image analysis tasks, showcasing exceptional performance and robustness in segmenting structures like cells, organs, or anomalies from medical scans Banerjee (2020).

The architecture consists of a contracting path responsible for capturing context and

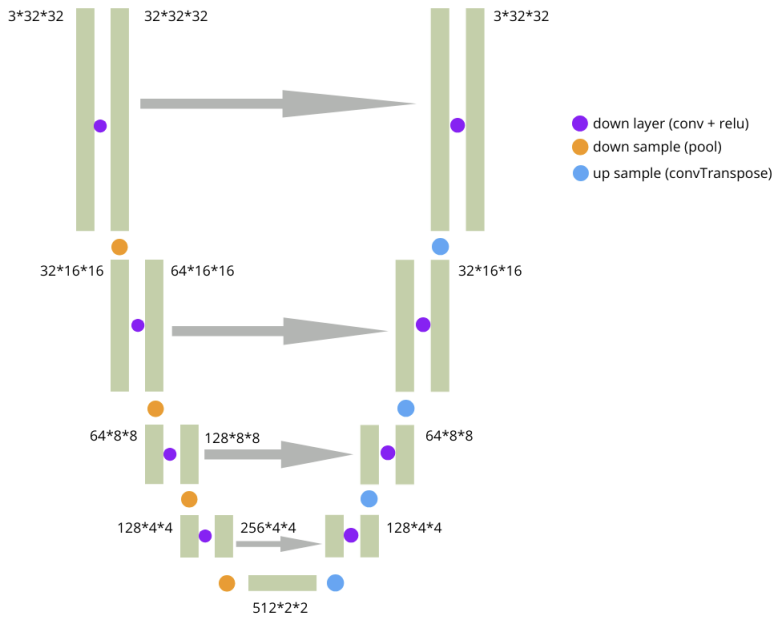


Figure 5: deployed U-Net architecture

a symmetric expanding path aiding in precise localization. Its skip connections between mirrored layers facilitate the fusion of low-level features with higher-level semantic information, enabling the network to maintain fine-grained details during the upsampling process Ronneberger (2017).

4.1 Training

We've configured two training scenarios focusing on maximizing the precision of predicted noise against the true noise:

4.1.1 Single Image Learning

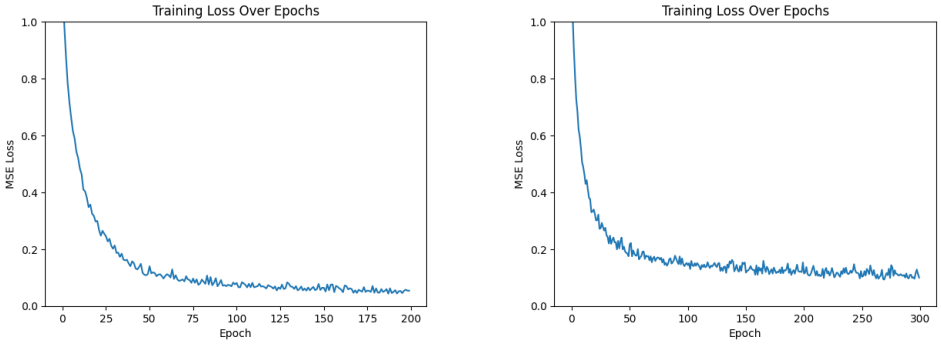
- Data: Prepare pairs of clean and noisy images.
- Model: Use U-Net, MSE loss, and an optimizer. Implement a linear learning rate scheduler with 300 time steps.
- Training: Use batch size 128, train for 200 epochs, aiming for accurate noise prediction on one image.

4.1.2 Multiple Images Learning

- Data: Gather multiple images of one class, each with corresponding noisy versions.
- Model: Employ the same U-Net architecture, loss, optimizer, and learning rate scheduler with the same 300 time steps.
- Training: Utilize batch size 128, train for 300 epochs, aiming for generalized noise prediction within one class of images.

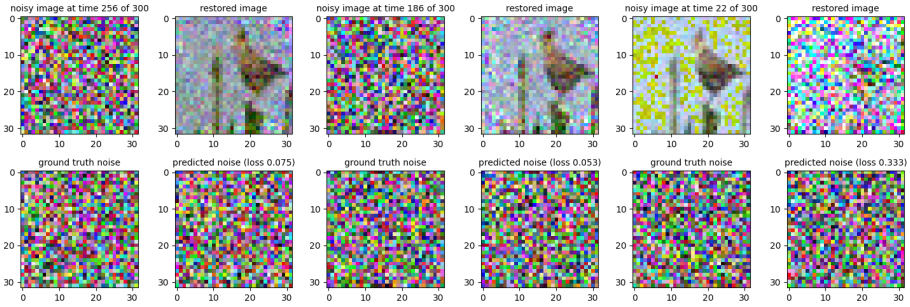
4.1.3 Comparison

The utilization of a simple U-Net model for denoising has yielded promising outcomes. In the initial scenario, convergence was achieved with approximately half the loss compared to the second scenario, owing to its lower complexity. Specifically, the first scenario exhibited a loss of approximately 0.05, whereas the second scenario incurred a loss of around 0.1. 6



(a) One image scenario (b) One class scenario

Figure 6: Training loss per epoch of the two scenarios.



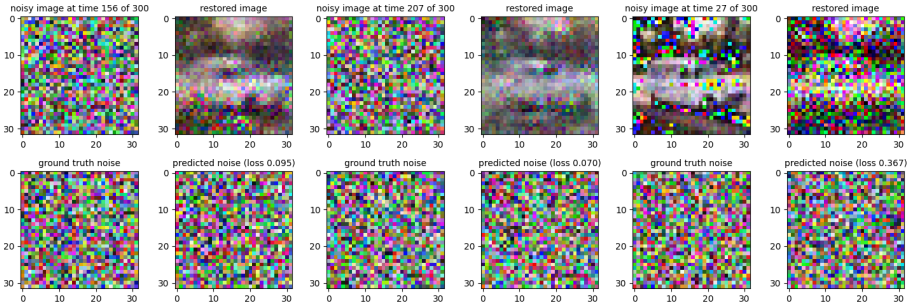
(a) step 256 - loss 0.075 (b) step 186 - loss 0.053 (c) step 22 - loss 0.333

Figure 7: Test results at different time steps for scenario 1

4.2 Test Results

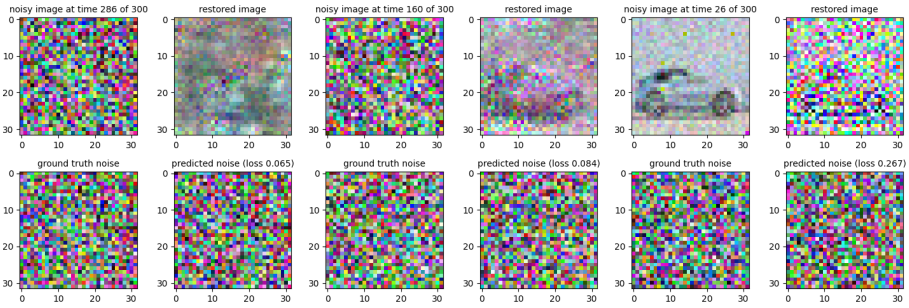
The outcomes aligned with our predictions: in scenario 1, the model showcased a remarkable ability to restore the image to a state of considerable recognizability. This restoration process not only brought back the overall structure but also revealed finer details that were otherwise obscured by noise. The resulting image presented a level of clarity and fidelity that allowed for the perception of intricate elements within the scene, as can be seen in figure 7.

However, in stark contrast, the images restored in scenario 2 exhibited a marked decline in quality. These outcomes were characterized by heightened blurriness and persistent noise artifacts. The restored images primarily retained only the general shapes and overarching color schemes present in the original, failing to recover the finer nuances or intricate details evident in the initial scene, as shown in figures 8, 9 and 10.



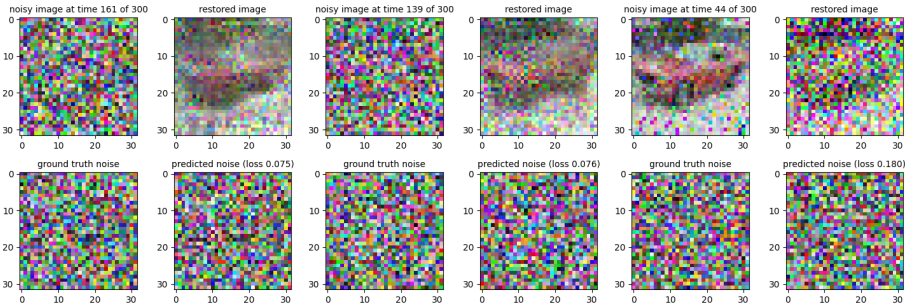
(a) step 156 - loss 0.095 (b) step 207 - loss 0.070 (c) step 27 - loss 0.367

Figure 8: Test results at different time steps for scenario 2, image 1



(a) step 286 - loss 0.065 (b) step 160 - loss 0.084 (c) step 26 - loss 0.267

Figure 9: Test results at different time steps for scenario 2, image 2



(a) step 161 - loss 0.075 (b) step 139 - loss 0.073 (c) step 44 - loss 0.180

Figure 10: Test results at different time steps for scenario 2, image 3

4.3 Discussion

In both scenarios, it's notable that the denoising process exacerbated the quality of the noisy images in earlier time steps. This deterioration occurred primarily because the U-Net architecture did not incorporate the temporal aspect, i.e., the time step information. Consequently, the denoising model struggled to account for the specific noise characteristics introduced at earlier time points.

Moreover, the noise introduced during the initial time steps was relatively subtle, leading the model to overcompensate for its intensity during the denoising process. This tendency to overshoot the noise intensity further exacerbated the degradation of image quality in those earlier time steps.

Addressing this issue is a key focus for milestone 2. Incorporating the temporal dimension into the architecture will enable the model to better understand and adapt to the evolving noise patterns across different time steps, thereby mitigating the problem of overcompensation and enhancing the overall denoising performance.

5 Conclusion

In this initial milestone our exploration has revealed intriguing insights into the behavior of various schedulers, which are a crucial component of DDPMs. We have achieved encouraging denoising, demonstrating that the model is capable of meaningful image restoration rather than generating entirely distorted outputs. These preliminary denoising results provide motivation for further refinement and improvement.

However, at this stage evaluation is constrained due to the absence of the reverse process. Looking ahead to Milestone 2, our focus will shift towards completing the DDPM framework by implementing the backward sampling process and enhancing the denoiser's quality. Specifically, we plan to integrate the temporal dimension into the UNet architecture, fostering more effective noise removal and iterative model refinement based on observations and learnings from Milestone 1.

As we set our sights on Milestone 3, our approach will pivot towards comprehensive evaluation, encompassing rigorous testing of the reverse process with different schedulers and hyperparameters across diverse datasets. Through systematic experimentation, we

aim to gain a deeper understand of the relationships between schedulers, hyperparameters, and dataset characteristics, thereby refining our model and ensuring its adaptability to various challenges. This milestone serves as the foundation for a dynamic exploration into the capabilities and potentials of DDPM.

References

R. Banerjee. A gentle introduction to u-net, Jun 2020. URL `https://medium.com/srm-mic/a-gentle-introduction-to-u-net-fc4afd12a893`.

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

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`.

A. Jabri, D. Fleet, and T. Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

D. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 1, 01 2009.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

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

O. Ronneberger. Invited talk: U-net convolutional networks for biomedical image segmentation. *Informatik aktuell*, page 3–3, 2017. doi: 10.1007/978-3-662-54345-0_3.

F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. 06 2015.