ISSN 2007-9737

A Comparative Study of Noise Schedules in Denoising Diffusion Probabilistic Models

Emmanuel Martínez-Guerrero, GuoHua Sun*

Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Mexico

{emartinezg2025, gsun}@cic.ipn.mx

Abstract. Noise scheduling plays a crucial role in the performance of denoising diffusion probabilistic models (DDPMs), affecting both training dynamics and sample quality. Although various noise schedules have been proposed, a comprehensive comparative analysis remains limited. In this work, we evaluate five widely used noise schedules: linear, cosine, quadratic, sigmoid and exponential across datasets with increasing complexity: MNIST, Fashion-MNIST and CIFAR-10. We analyze their impact on training performance and generative quality using metrics such as Fréchet Inception Distance (FID), Kernel Inception Distance (KID) and Inception Score (IS). Our quantitative results show that the linear schedule offers the most rapid training convergence, whereas the exponential schedule shows the lowest performance. In contrast, cosine, quadratic, and sigmoid schedules tend to produce higher-quality samples, depending on the complexity of the dataset. Qualitative analysis reveals that nonlinear schedules like cosine and exponential accelerate the formation of structured and recognizable images in early training stages, suggesting greater efficiency in producing better samples with less training. Our findings indicate that nonlinear schedules may be preferable when early sample quality is critical, while linear schedules offer advantages in training speed.

Keywords. Generative models, diffusion models, denoising diffusion probabilistic models, noise schedule.

1 Introduction

Over the past few years, generative artificial intelligence has rapidly advanced, becoming a central area of machine learning research and application. Its influence now extends from everyday interactions such as conversational agents and automated text generation to the growing demand for AI-powered image creation, which has captured substantial interest from both academia and industry.

Alongside this progress, several foundational modeling approaches have shaped the field, most notably Generative Adversarial Networks (GANs) [12], Variational Autoencoders (VAEs) [25] and flow-based models [37]. GANs employ a competitive framework between a generator and a discriminator, resulting in highly realistic outputs but often face instability during training and issues such as mode collapse, where diversity in generated samples is limited [3, 52]. VAEs, in contrast, use probabilistic encodings to map data into a continuous latent space, offering stable training and efficient sampling, though typically at the expense of sharpness and fidelity, leading to blurrier results compared to GANs [44]. Flow-based models construct a series of invertible transformations to explicitly model data distributions, enabling exact likelihood computation and reversible sampling, but are often constrained by architectural requirements that limit flexibility and can be computationally intensive, especially for high-dimensional data [15].

Despite the notable successes of these methods, each presents inherent trade-offs related to fidelity, diversity, efficiency, and scalability. Addressing these challenges, diffusion models have emerged as a leading approach in generative modeling, valued for their greater flexibility and

reduced constraints [8]. In particular, Diffusion Probabilistic Models (DDPMs) [16, 45] are able to accurately learn complex data distributions and generate high-quality samples. These models have made a substantial impact across a wide range of application domains, most notably in image generation tasks such as text-to-image synthesis, super-resolution and inpainting [38, 9, 42, 31]. In addition, DDPMs have shown effectiveness in other areas, including video generation [51, 19], the generation of complex molecular structures for materials science [34] and drug discovery [7], as well as audio generation [26, 36], texture synthesis [11], and medical imaging [48, 22].

The DDPM framework relies on a two-step process grounded in probabilistic modeling. Forward diffusion gradually corrupts data by adding noise in a series of steps, transforming structured data into a nearly random distribution. This progression is controlled by a noise schedule that determines the amount of noise introduced at each stage. The reverse process, learned by a neural network, iteratively denoises the corrupted data, reconstructing samples from noise back to the original distribution. The network is trained to predict the noise component at each step, enabling it to generate new high-quality samples by reversing the diffusion process starting from pure noise. The noise schedule is a critical factor in this mechanism, as it influences how noise is injected during training and sampling, thus affecting the model's capacity to recover fine details and produce diverse outputs.

Related work: Much of the existing research has focused primarily on improving the architecture of diffusion models or exploring novel applications [50, 39, 46, 17, 35, 1, 5, 29]. However, the impact of different noise schedules on these models remains underexplored. Recent studies have examined common noise schedules such as linear and cosine, revealing that these standard approaches can introduce limitations in sample quality and training efficiency [33, 35]. For instance, the cosine schedule was introduced to better align the noise variance with pure Gaussian noise at the terminal steps, improving the image sharpness compared to linear schedules. Other work has proposed adaptive noise scheduling techniques that learn optimal noise injection strategies during training, demonstrating gains in convergence speed and sample quality [47]. Complementary research has focused on architectural simplicity and training stability to enhance high-resolution image generation, highlighting the interplay between noise scheduling and model design [20].

In this work, we present a systematic comparison of several noise schedules within the framework of DDPMs, focusing on linear, cosine, quadratic, sigmoid, and exponential schemes. These schedules are selected for their prevalence in the literature and their theoretical diversity in controlling noise injection throughout the diffusion process. Our evaluation is carried out through extensive experiments on the MNIST, Fashion MNIST and CIFAR-10 datasets that span a range of complexity, from simple grayscale digits and fashion items to more complex, colored natural images, allowing a robust assessment of the performance of each schedule.

Through controlled experiments, we isolate and examine the specific influence of each noise schedule on key performance dimensions, including sample quality, diversity, and training performance. By rigorously analyzing how different noise injection strategies affect model behavior, this study aims to provide comprehensive insight into the role of noise scheduling in diffusion-based generative modeling.

Our main contributions are as follows.

— A systematic comparison of five widely used and theoretically distinct noise schedules: linear, cosine, quadratic, sigmoid, and exponential within the DDPMs framework.

— Extensive empirical evaluations on MNIST, Fashion MNIST, and CIFAR-10 to investigate the impact of noise schedules across datasets with varying levels of visual complexity.

— Detailed analysis of the effects of each schedule on sample quality, diversity, and training performance using standard metrics including Fréchet Inception Distance (FID),

Kernel Inception Distance (KID) and Inception Score (IS).

— Practical insights into how different noise scheduling strategies interact with dataset characteristics and application-specific priorities.

The remainder of this paper is organized as follows. Section 2 introduces preliminary concepts and foundational background relevant to this work, including an overview of DDPMs and noise schedules. Section 3 details the methodology, describing the datasets, model architecture, compared noise schedules, training procedures, and evaluation metrics. The results are presented in Section 4, where we analyze the impact of different noise schedules in multiple datasets. Finally, Section 5 concludes the article with a summary of the findings and directions for future research.

2 Background

In this section, we review the essential components of DDPMs, alongside an examination of noise schedules that dictate the temporal progression of noise throughout the diffusion process.

2.1 Denoising Diffusion Probabilistic Models

Diffusion models [16, 45] generate samples by modeling a sequence of probabilistic transitions, where each state depends solely on the previous one, thus forming a Markov chain. To effectively approximate complex data distributions, these models utilize variational inference, which frames the problem as an optimization task over approximate distributions.

The core mechanism involves a forward process that incrementally corrupts the data by adding Gaussian noise at each step. Specifically, given a data sample $\boldsymbol{x}_0$ drawn from an unknown distribution $q(\boldsymbol{x}_0)$, noise is progressively added over a finite number of time steps T. At each step $t \in [T]$, the forward diffusion process $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ adds Gaussian noise according to a predefined variance schedule $0 < \beta_1 < \cdots < \beta_T < 1$, described by:

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right) \sim \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\right), \quad (1)$$

$$q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right) = \prod_{t=1}^{T} q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right). \quad (2)$$

From Equation (2), it follows that as t increases, the sample $\boldsymbol{x}_t$ progressively loses its original structure, and in the limit as $T \to \infty$, $\boldsymbol{x}_T$ converges to an isotropic Gaussian distribution. According to Equation (1), each step involves sampling from a conditional Gaussian distribution with mean $\boldsymbol{\mu}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}$ and covariance $\boldsymbol{\Sigma}_t^2 = \beta_t \mathbf{I}$. Thus, the forward diffusion at step t can be expressed as sampling noise $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I})$ and setting:

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1}. \quad (3)$$

Using the reparameterization trick commonly used in machine learning [24, 30], it becomes possible to directly sample noisy data $\boldsymbol{x}_t$ at any arbitrary time step t using a closed-form expression. This relationship can be expressed as:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0, \quad (4)$$

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) \sim \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, \left(1-\bar{\alpha}_t\right)\boldsymbol{I}\right), \quad (5)$$

where the cumulative product $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\alpha_i)$, with $\alpha_t = 1 - \beta_t$, defines the effective noise schedule up to time t. This relationship implies that for any step within the diffusion process, the noisy sample $\boldsymbol{x}_t$ can be generated directly from the clean data $\boldsymbol{x}_0$ by blending it with appropriately scaled noise.

Reconstructing the original data from a corrupted, noisy state involves tracing the process backward in time. Specifically, this requires sampling from the reverse conditional distribution $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, which would, in principle, allow one to iteratively transform a noise sample back into structured data. However, this reverse transition is generally not tractable, as it depends on the full unknown data distribution and thus cannot be computed directly from available information.

To overcome this, the reverse conditional distribution is approximated by a parameterized

probabilistic model, denoted $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, where $\boldsymbol{\theta}$ represents a set of parameters to be learned. This approximation is commonly modeled as a Gaussian distribution:

$$p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right) \sim \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)\right). \tag{6}$$

The main task reduces to estimating the mean vector $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ for the probabilistic model $p_{\boldsymbol{\theta}}(\cdot)$, enabling an effective approximation of the true reverse transitions. In practice, these parameters $\boldsymbol{\theta}$ are learned by training a neural network to approximate the reverse diffusion process. When we have $\boldsymbol{x}_0$ as reference, the reverse step can be described more precisely by the conditional distribution $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$. Using the Bayes rule and the known forward conditionals $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ and $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$, this distribution can be derived as:

$$q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0\right) \sim \mathcal{N}\left(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}\left(\boldsymbol{x}_t, \boldsymbol{x}_0, t\right), \tilde{\beta}_t \boldsymbol{I}\right), \tag{7}$$

$$\tilde{\boldsymbol{\mu}}\left(\boldsymbol{x}_t, \boldsymbol{x}_0, t\right) = \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0, \tag{8}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{9}$$

From Equation (9), we can infer that the covariance matrix does not depend on the learnable parameters, so the focus is on estimating the mean vector $\tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0, t)$. Using the reparameterization trick again, $\boldsymbol{x}_0$ can be expressed in terms of the noisy sample $\boldsymbol{x}_t$ and noise $\boldsymbol{\epsilon}_t$ as:

$$\boldsymbol{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t\right). \tag{10}$$

Substituting $\boldsymbol{x}_0$ into Equation (8) yields the following:

$$\tilde{\boldsymbol{\mu}}\left(\boldsymbol{x}_t, \boldsymbol{x}_0, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right). \tag{11}$$

Now we can learn the conditioned probability distribution $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$ of the reverse diffusion process by training a neural network that approximates $\tilde{\boldsymbol{\mu}}\left(\boldsymbol{x}_t, \boldsymbol{x}_0, t\right)$. Therefore, we simply need to set the approximated mean vector $\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)$ to have the same mathematical format as the target mean vector $\tilde{\boldsymbol{\mu}}\left(\boldsymbol{x}_t, \boldsymbol{x}_0, t\right)$. Hence, we have:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)\right), \tag{12}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, t\right)$ denotes the model for predicting $\boldsymbol{\epsilon}_t$.

Training diffusion models involves optimizing a variational approximation of the negative log-likelihood, where the model distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)$ approximates the true posterior $q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)$. The similarity between these distributions is measured using the Kullback-Leibler (KL) divergence [16], which yields the bound:

$$-\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) \leq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T})}\right]. \tag{13}$$

Since $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$ is intractable, the Evidence Lower Bound (ELBO) is optimized instead, decomposing into KL terms between forward and reverse conditionals. Minimizing ELBO thus maximizes a lower bound on the likelihood of the data [45]. For efficiency, Gaussian diffusion models often use a simplified loss that removes weighting factors [16, 18]:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]. \tag{14}$$

This loss emphasizes denoising at higher noise levels, improving the ability of the model to generate high-quality samples during synthesis [16].

2.2 Noise Schedules

As discussed previously, the noise schedule controls the incremental addition of noise during the forward diffusion process and its removal during the reverse process. This sequence of noise levels $\{\beta_t\}_{t=1}^T$ fundamentally shapes the diffusion trajectory and influences the model's training and generation behavior [41, 16].

Several noise schedules have been proposed, each prescribing a distinct pattern for noise variance progression over time. Commonly used schedules include linear, cosine, quadratic,

sigmoid and exponential functions. These schedules differ in how they distribute noise increments across timesteps, which can affect the difficulty of the denoising tasks and the overall model performance. The selection of these schedules for comparison is motivated by their prevalence in literature and their diverse mathematical characteristics, providing a broad spectrum of noise injection strategies [46, 17, 35, 6, 13, 32].

Linear Schedule: The linear schedule increases noise variance at a constant rate from an initial value β_1 to a final value β_T over T timesteps:

$$\beta_t = \beta_1 + \frac{t-1}{T-1}(\beta_T - \beta_1). \tag{15}$$

This straightforward approach results in uniform increments of noise variance at each step, producing a steady and predictable increase in noise magnitude throughout the diffusion process. Its simplicity makes it a common baseline in diffusion model implementations [16].

Cosine Schedule: The cosine schedule defines noise progression using a smooth, non-linear function inspired by the squared cosine curve. It is expressed through the cumulative noise term $\bar{\alpha}_t$ as:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad \text{where} \quad f(t) = \cos\left(\frac{\frac{t}{T}+s}{1+s}\cdot\frac{\pi}{2}\right)^2, \tag{16}$$

where s is a small offset to prevent near-zero noise at the start [35]. The incremental noise variance is then derived as:

$$\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}. \tag{17}$$

This schedule produces a noise progression that is slower near the beginning and end of the diffusion timeline, with a more rapid increase around the midpoint. The smooth transitions reduce abrupt changes in noise levels, potentially enabling a gradual shift in denoising difficulty.

Quadratic Schedule: The quadratic schedule increases noise variance following a squared progression:

$$\beta_t = \beta_1 + \left(\frac{t-1}{T-1}\right)^2(\beta_T - \beta_1). \tag{18}$$

This schedule begins with relatively small noise increments that increase more rapidly towards the later timesteps, reflecting a nonlinear acceleration in noise addition [41].

Sigmoid Schedule: The sigmoid schedule uses an S-shaped curve to control noise variance, characterized by gradual transitions at the start and end of the diffusion process:

$$\bar{\alpha}_t = \frac{-\text{sigmoid}\left(\frac{t(e-s)+s}{\tau}\right) + \text{sigmoid}\left(\frac{e}{\tau}\right)}{\text{sigmoid}\left(\frac{e}{\tau}\right) - \text{sigmoid}\left(\frac{s}{\tau}\right)}, \tag{19}$$

where s and e specify the input range of the sigmoid function, and τ controls the slope steepness [21, 32]. This schedule features a smooth and continuous increase in noise variance, with slower changes at the extremes of the diffusion process, which may contribute to stability in noise injection.

Exponential Schedule: The exponential schedule increases noise variance following an exponential curve:

$$\beta_t = \beta_{\min}\left(\frac{\beta_{\max}}{\beta_{\min}}\right)^{\frac{t-1}{T-1}}, \tag{20}$$

where $\beta_{\min} = \beta_1$ and $\beta_{\max} = \beta_T$. This results in noise increments that grow more rapidly as the diffusion process progresses [13].

To complement the mathematical definitions of the noise schedules presented above, Fig. 1, Fig. 2, and Fig. 3 visualize their distinct characteristics on the full diffusion timeline of 1000 time steps. Fig. 1 plots the incremental noise variance β_t for each schedule, illustrating how noise is added at each step and highlighting differences in noise injection rates. Fig. 2 shows the cumulative signal retention $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$, representing how quickly the original data signal diminishes as noise accumulates. This cumulative perspective provides insight into the overall corruption trajectory induced by each schedule.

ISSN 2007-9737

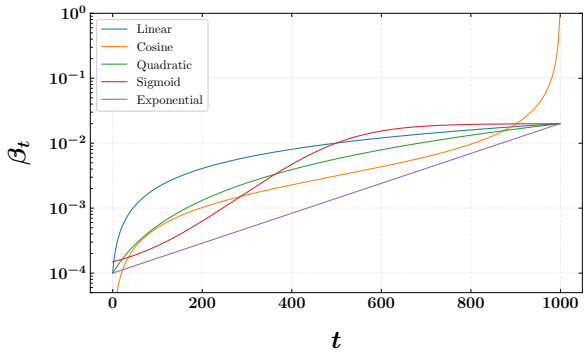


Fig. 1. Noise variance β_t (log scale) plotted against timestep t for different noise schedules. Each curve illustrates how noise is incrementally added during the forward diffusion process over 1000 timesteps.

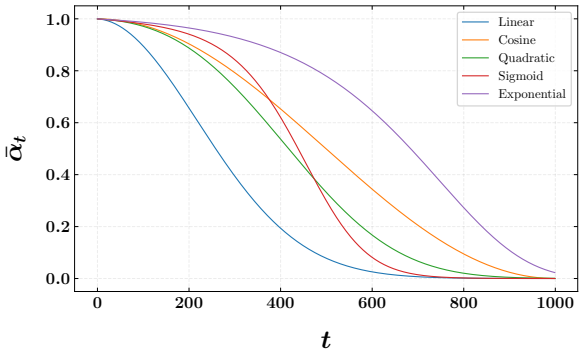


Fig. 2. Cumulative signal retention $\bar{\alpha}_t$ across timesteps for various noise schedules. This plot reflects how quickly the original data signal diminishes as noise accumulates.

Finally, Fig. 3 presents sequence snapshots of a sample image progressively corrupted at selected time steps under each noise schedule, offering an intuitive visual understanding of how the varying noise progressions affect the degradation of the data.

3 Methodology

This section describes the detailed experimental procedures employed to systematically compare different noise schedules within DDPMs. Our goal is to isolate the effect of noise scheduling on model training and generation quality by maintaining consistent conditions across all experiments.

Datasets and Preprocessing: To comprehensively assess the impact of different noise schedules on DDPMs, we conducted experiments on three widely adopted datasets. MNIST [28], Fashion-MNIST [49] and CIFAR-10 [27]. MNIST consists of 70,000 grayscale images of handwritten digits, each sized 28×28, serving as a canonical dataset for generative modeling with relatively low complexity. Fashion-MNIST, also containing 70,000 grayscale images of the same resolution, offers a more challenging task by including diverse clothing items, thus providing a step up in complexity while maintaining manageable computational requirements. CIFAR-10 presents a further increase in complexity with 60,000 color images of size 32×32 in 10 classes, encompassing natural scenes and objects with significant variability in texture and color. This selection ensures that our analysis covers a broad spectrum of data distributions and complexities, allowing us to observe how noise schedules perform under varying conditions. Before training, all images are normalized to the $[-1, 1]$ range, a standard preprocessing step that facilitates stable gradient flow and helps the model learn effectively by centering the data distribution.

Model Architecture and Implementation: Our experiments utilize a U-Net architecture tailored for diffusion models, consistent with the design introduced in [16, 40]. The model takes as input images with a number of channels matching the dataset: grayscale datasets (MNIST, Fashion-MNIST) use a single input channel, while CIFAR-10 uses three channels for RGB images. The U-Net base number of channels is set at 32, balancing the capacity and computational efficiency; this base is expanded over three resolution levels with channel multipliers of $(1, 2, 4)$, allowing the network to capture features at multiple scales. To enhance representational power, attention mechanisms are applied selectively in the deeper layers, specifically enabled only in the highest-resolution block, which can help the model focus on important spatial regions

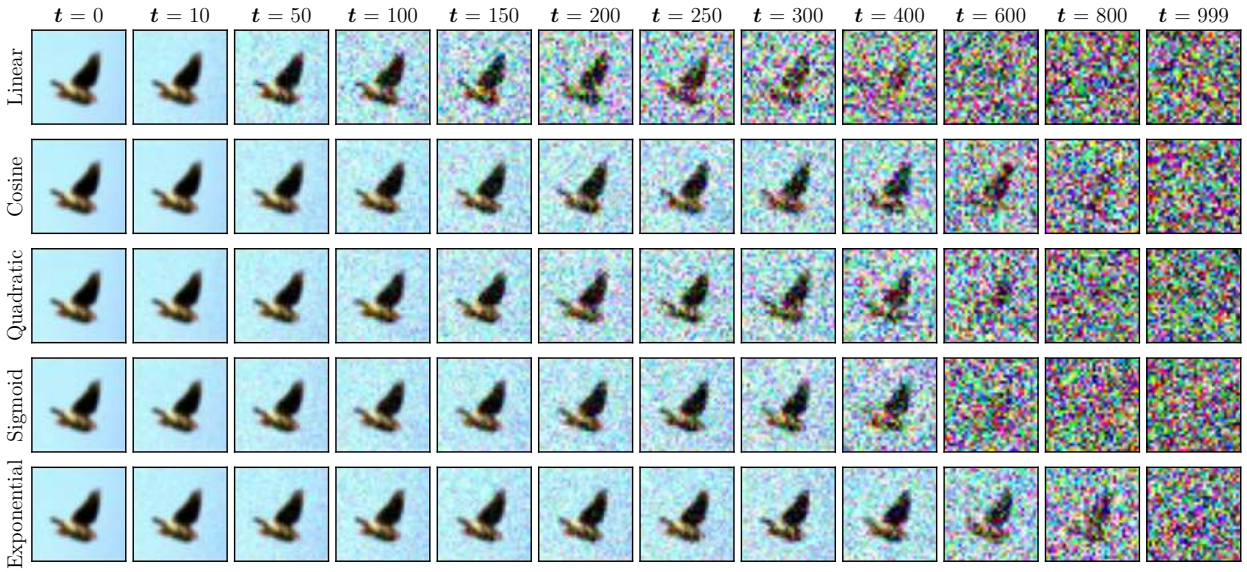


Fig. 3. Progressive corruption of a sample image at selected timesteps under different noise schedules. These snapshots visualize the impact of each schedule on data degradation during the forward diffusion process.

without excessive computational overhead. Time conditioning is incorporated through sinusoidal embeddings with a multiplier of 2, which modulates the timestep representation fed into the network, enabling the model to adapt its denoising behavior dynamically across the 1000 diffusion steps. Each resolution block contains two residual units, a configuration that has been shown to effectively balance depth and training stability. Additionally, a dropout rate of 0.1 is applied throughout the network to mitigate overfitting and improve generalization, particularly important given the diversity of datasets and the complexity of the denoising task. These architectural choices provide a strong and consistent backbone for all experiments, ensuring that differences in performance can be attributed to noise-schedule variations rather than model capacity or design.

Noise Schedules Configuration: We compare five noise schedules: linear, cosine, quadratic, sigmoid, and exponential, each defining a distinct sequence of noise variances $\{\beta_t\}_{t=1}^{T}$ used in the forward diffusion process. The mathematical formulations of these schedules are detailed in Section 2.2. For all schedules, we fix $\beta_1 = 0.0001$ and $\beta_T = 0.02$, values commonly adopted in the literature that balance gradual noise addition with sufficient corruption by the final timestep [16, 46].

For the cosine schedule, we use the offset parameter $s = 0.008$ as originally proposed in [35] to shape the noise variance curve smoothly, avoiding singularities at the boundaries. The sigmoid schedule is parameterized by linearly spacing inputs from $s = -6$ to $e = 6$ before applying the sigmoid function with temperature $\tau = 1$, controlling the noise increase rate [21].

Training Procedure: Training is conducted over $T = 1000$ diffusion steps, a standard choice that offers a fine-grained noise injection schedule consistent with prior work [16, 50]. At each iteration, a timestep t is sampled uniformly from 1 to T, and the model receives a noisy image generated by adding Gaussian noise with variance β_t according to the selected schedule. The model is trained to predict the noise component using the mean squared error (MSE) loss between the predicted and true noise vectors, a loss function widely adopted for its theoretical grounding and empirical effectiveness in diffusion models [16, 46]. We use the Adam optimizer [23] with

a learning rate of 1×10^{-4} and a batch size of 64, settings chosen based on preliminary experiments to balance convergence stability and computational efficiency. For the MNIST and Fashion-MNIST datasets, training is performed for 20 epochs, reflecting their relatively smaller size and lower complexity, which allows the model to converge within fewer passes over the data. For CIFAR-10, which is more complex and diverse, training extends to 50 epochs to provide sufficient exposure for the model to learn the richer data distribution. These epoch counts were selected to ensure adequate training while maintaining reasonable computational costs. All experiments use fixed random seeds to ensure reproducibility, and all other hyperparameters remain constant across noise schedules to isolate their effect on model performance.

Experimental Design: Our experimental design strictly controls for all variables except the noise schedule. By maintaining consistent model architectures, training hyperparameters and data preprocessing across all runs, we attribute observed differences in performance directly to the noise scheduling strategy. Each schedule is independently trained and evaluated on all three datasets using the same pipeline. Sampling is performed by starting from pure Gaussian noise and iteratively applying the learned reverse diffusion process for $T = 1000$ steps, mirroring the forward noise addition schedule. This approach allows us to observe how each noise schedule influences both the training dynamics and the quality of generated samples under identical conditions.

Evaluation Protocol: To quantitatively assess the performance of each noise schedule, we employ three widely recognized metrics: Fréchet Inception Distance (FID), Kernel Inception Distance (KID) and Inception Score (IS). These metrics are chosen for their ability to capture both the quality and diversity of generated images, providing a comprehensive evaluation of generative model performance [14, 2, 43, 4, 10].

The *Fréchet Inception Distance (FID)* measures the similarity between the distribution of generated images and that of real images by modeling both as multivariate Gaussians in the feature space of a pre-trained Inception-v3 network. Specifically, FID computes the Fréchet distance between the mean and covariance of the activations for real and generated images:

$$\mathrm{FID} = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}(C_r + C_g - 2(C_r C_g)^{1/2}), \quad (21)$$

where μ_r, C_r and μ_g, C_g denote the mean and covariance of the real and generated image features, respectively. Lower FID values indicate that the generated images are more similar to the real images in terms of high-level features, with a score of 0 representing perfect alignment [14]. FID is widely adopted due to its strong correlation with human judgment of image quality and its sensitivity to both image fidelity and diversity [10].

The *Kernel Inception Distance (KID)* is another metric based on the Inception-v3 feature space, but instead of assuming Gaussian distributions, it estimates the squared Maximum Mean Discrepancy (MMD) between the feature distributions of real and generated images using a polynomial kernel:

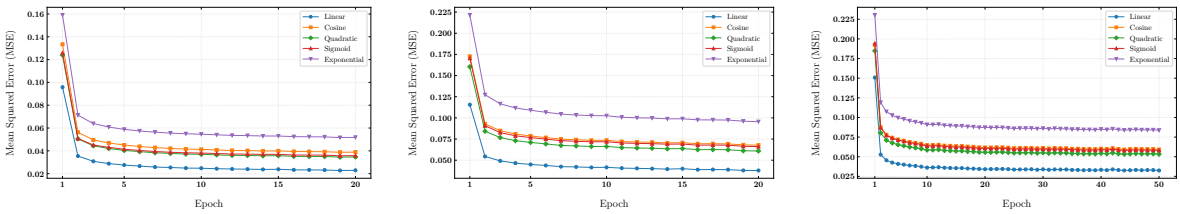
$$\mathrm{KID} = \mathrm{MMD}^2(\mathcal{F}_r, \mathcal{F}_g), \quad (22)$$

where $\mathcal{F}_r$ and $\mathcal{F}_g$ are the feature sets of real and generated images, respectively. KID is unbiased even for small sample sizes and is less sensitive to outliers than FID, making it a robust complement for evaluating generative models [2].

The *Inception Score (IS)* evaluates both the quality and diversity of generated images by analyzing the conditional label distribution $p(y|x)$ predicted by the Inception network for each generated image x, as well as the marginal distribution $p(y)$. It is defined as:

$$\mathrm{IS} = \exp\left(\mathbb{E}_x \left[\mathrm{KL}(p(y|x) \| p(y))\right]\right), \quad (23)$$

where KL denotes the Kullback-Leibler divergence. High IS values are achieved when individual images are easily classifiable (low entropy in $p(y|x)$) and the set of images is diverse (high entropy in $p(y)$) [43]. While IS is less sensitive to mode collapse within classes, it remains a standard metric for generative image evaluation.



(a) (b) (c)

Fig. 4. MSE training loss curves for DDPMs with different noise schedules on (a) MNIST, (b) Fashion-MNIST and (c) CIFAR-10. The linear schedule converges fastest with the lowest loss, exponential performs worst and cosine, quadratic and sigmoid schedules show intermediate behavior.

For all metrics, we compute the relevant statistics using 7,000 images from both the original test set and the generated samples to ensure a robust and fair comparison. The reference dataset for FID, KID, and IS calculations is the test split of the corresponding dataset (MNIST, Fashion-MNIST, or CIFAR-10), following standard evaluation protocols. This approach enables us to systematically compare the impact of different noise schedules on both the fidelity and diversity of generated images, as well as on training dynamics.

In addition to these generative quality metrics, we monitor the mean squared error (MSE) training loss at each epoch. The MSE loss provides a direct measure of the model's ability to predict the added noise during training, offering insight into convergence behavior and training stability.

4 Results

This section presents the experimental results comparing the impact of different noise schedules on the training dynamics and generative performance of DDPMs across three datasets: MNIST, Fashion-MNIST, and CIFAR-10. We analyze both quantitative metrics and qualitative sample visualizations to provide a comprehensive evaluation.

Training Dynamics: Fig. 4 illustrates the progression of the loss of mean squared error (MSE) during training for each noise schedule in the three data sets. In MNIST (Fig. 4(a)), the linear schedule starts with an initial MSE of approximately 0.0958 and steadily decreases to a final value of 0.02291, demonstrating the fastest and most stable convergence among all schedules. In contrast, the exponential schedule begins with a notably higher initial loss of 0.1590 and converges more slowly, ending at 0.05180, indicating greater difficulty in learning the denoising function. The cosine, quadratic, and sigmoid schedules exhibit similar behaviors, starting near 0.1334 and converging to around 0.0389, positioning their performance between the linear and exponential extremes.

This trend persists in Fashion-MNIST (Fig. 4(b)), where the linear schedule again achieves the lowest final MSE of 0.03785, starting from 0.1157. The exponential schedule shows the highest initial and final losses, beginning at 0.2215 and ending at 0.09552. The cosine, quadratic and sigmoid schedules maintain intermediate

Table 1. MSE loss values for different noise schedules on MNIST, Fashion-MNIST and CIFAR-10 datasets; lowest values per dataset are highlighted.

Schedule	MSE Loss		
	MNIST	Fashion-MNIST	CIFAR-10
Linear	**0.02291**	**0.03785**	**0.03248**
Cosine	0.03886	0.06776	0.05913
Quadratic	0.03464	0.06091	0.05326
Sigmoid	0.03581	0.06607	0.05802
Exponential	0.05180	0.09552	0.08373

ISSN 2007-9737

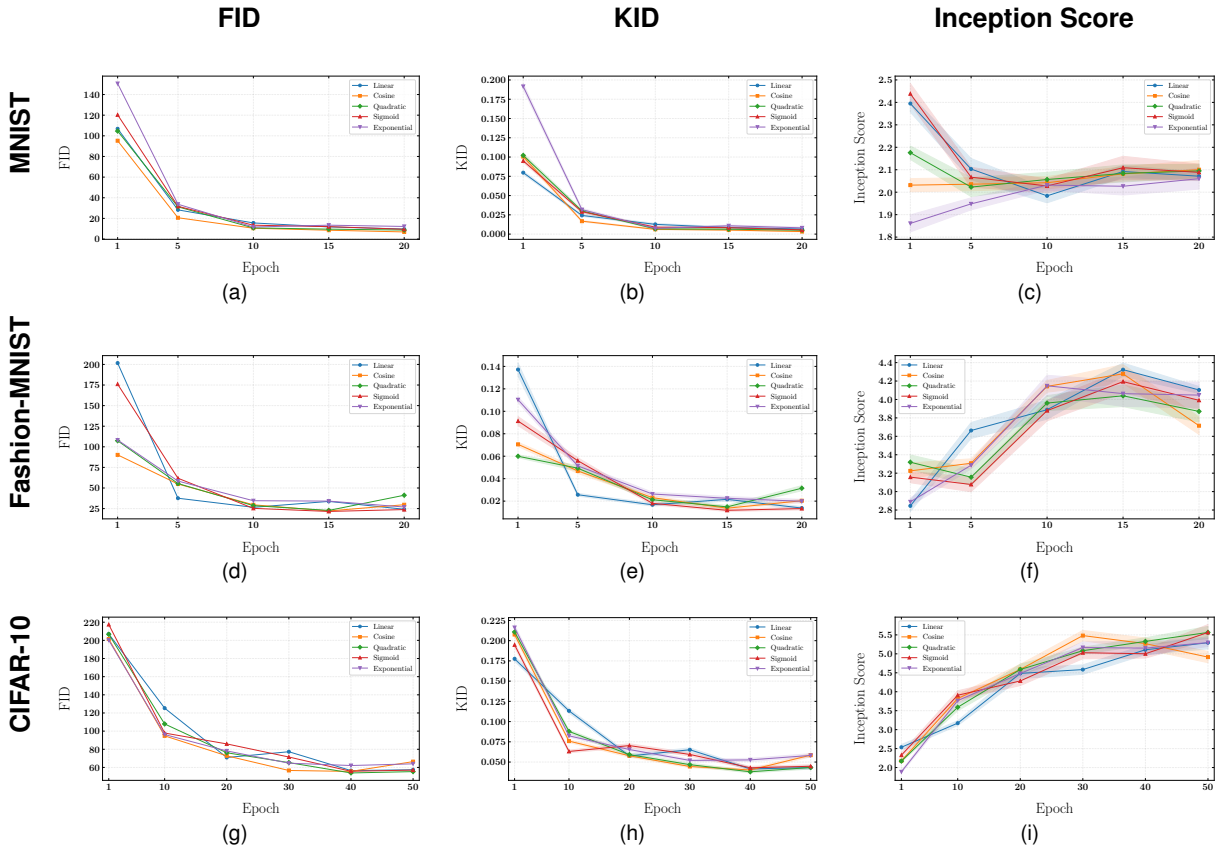


Fig. 5. Comparison of generative performance metrics across noise schedules and datasets. Columns correspond to metrics: ((a), (d) and (g)) Fréchet Inception Distance (FID), ((b), (e) and (h)) Kernel Inception Distance (KID) and ((c), (f) and (i)) Inception Score (IS). Rows correspond to datasets: (top) MNIST, (middle) Fashion-MNIST and (bottom) CIFAR-10. Each panel shows metric progression over training epochs for different noise schedules, illustrating how schedule choice impacts sample quality and diversity across datasets of varying complexity.

performance, with initial losses near 0.1726 and final losses around 0.0678.

On the more complex CIFAR-10 dataset (Fig. 4(c)), the linear schedule continues to outperform others, starting at 0.1508 and converging to 0.03248. The exponential schedule remains the least effective, with an initial loss of 0.2304 and a final loss of 0.08373. The cosine, quadratic, and sigmoid schedules again cluster together, beginning at approximately 0.1923 and finishing near 0.0591.

These results suggest that the uniform increase in noise in the linear noise schedule facilitates smoother and more effective training dynamics across datasets of variable complexity. The rapid increase in noise from the exponential schedule appears to hinder early training, leading to higher losses and slower convergence. The cosine, quadratic, and sigmoid schedules offer moderate performance, indicating that while nonlinear noise schedules can provide benefits, they do not necessarily surpass the simplicity and stability of the linear schedule in minimizing training loss. Table 1 presents the final MSE losses achieved by each noise schedule in the data sets.

Quantitative Analysis: We complement the quantitative evaluation with a detailed analysis of

Table 2. Numerical results for images generated using different noise schedules on MNIST, Fashion-MNIST, and CIFAR-10 datasets, reporting FID, KID, and Inception Score. Bold values mark the best results per metric and dataset.

Schedule	MNIST			Fashion-MNIST			CIFAR-10		
	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑
Linear	9.717	0.007 ± 0.001	2.071 ± 0.019	24.207	0.014 ± 0.001	**4.103 ± 0.037**	57.433	0.043 ± 0.002	5.293 ± 0.162
Cosine	**6.913**	**0.003 ± 0.000**	**2.100 ± 0.041**	29.715	0.020 ± 0.001	3.715 ± 0.095	66.392	0.059 ± 0.002	4.914 ± 0.148
Quadratic	8.726	0.005 ± 0.000	2.095 ± 0.029	41.193	0.031 ± 0.002	3.870 ± 0.116	**55.393**	**0.043 ± 0.002**	**5.565 ± 0.159**
Sigmoid	9.480	0.006 ± 0.001	2.088 ± 0.037	**23.807**	**0.013 ± 0.001**	3.991 ± 0.101	57.065	0.045 ± 0.002	5.564 ± 0.208
Exponential	12.214	0.008 ± 0.001	2.060 ± 0.047	26.926	0.020 ± 0.002	4.047 ± 0.139	63.960	0.058 ± 0.002	5.297 ± 0.102

generative performance across noise schedules using the Fréchet Inception Distance (FID), Kernel Inception Distance (KID) and Inception Score (IS) metrics on each dataset. The progression of these metrics over training epochs for all noise schedules and datasets is presented in Fig. 5.

In the MNIST dataset, the cosine noise schedule consistently outperforms others in terms of FID, starting from an initial value of approximately 95.27 and converging to the lowest final score of 6.913 after 20 epochs, indicating superior sample quality and distributional similarity to real data. In contrast, the exponential schedule begins with the worst initial FID of 150.66, followed by the sigmoid schedule at 120.32. The quadratic and linear schedules start around 106.84, with all but cosine converging to slightly higher FID values by the 20th epoch. For KID, the exponential schedule performs the worst initially (0.192 ± 0.004), while the cosine, quadratic, and sigmoid schedules start near 0.101. At the end of the training, the cosine schedule achieves the best KID score, reinforcing its advantage in generating high-fidelity samples. Regarding IS, although initial values vary, all schedules converge to similar values around 2.10, with the cosine schedule slightly leading.

In the Fashion-MNIST dataset, the initial FID values vary widely. The linear schedule surprisingly starts with the worst FID at 201.67, while the cosine schedule begins with the best initial FID of 90.22. However, by the 20th epoch, the sigmoid schedule attains the best final FID of 23.807, closely followed by the linear schedule, with the quadratic schedule performing worst. For KID, the quadratic schedule starts best at 0.060 ± 0.001, whereas the linear schedule starts worst, but rapidly improves to finish second. The sigmoid schedule achieves the best final KID of 0.013 ± 0.001, while the quadratic end is the worst. The IS shows all programs improving with training, with the linear program ultimately achieving the highest final score of 4.103 ± 0.037, and the cosine schedule the lowest at 3.715 ± 0.095.

In the more complex CIFAR-10 dataset, initial FID scores are similar across schedules, around 201.44, with the quadratic schedule achieving the best final FID of 55.393. For KID, the linear schedule starts best at 0.178 ± 0.003, but the quadratic schedule surpasses others by epoch 50 with the best final KID of 0.043 ± 0.002, followed by the sigmoid and linear schedules. The cosine schedule performs worst on KID here. The IS starts similarly for all schedules (around 2.18) and improves with training, with the quadratic schedule reaching the highest final score of 5.565 ± 0.159, followed by sigmoid. The cosine schedule ends with the lowest final IS of 4.91 ± 0.148. Table 2 summarizes the numerical values of key metrics for images generated in noise schedules and data sets.

In general, these results reveal that no single noise schedule is universally dominant across all metrics and datasets. The cosine schedule excels on MNIST, particularly in FID and KID, while the sigmoid schedule shows strong performance on Fashion-MNIST. The quadratic and sigmoid

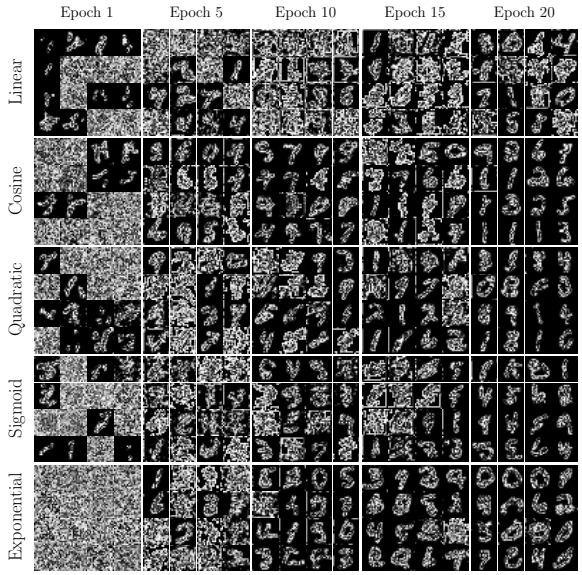


Fig. 6. Generated MNIST samples arranged in a matrix where each row corresponds to a different noise schedule and each column to selected training epochs (1, 5, 10, 15, and 20). Each cell contains 16 generated digit images illustrating the evolution of sample quality and structure over training.

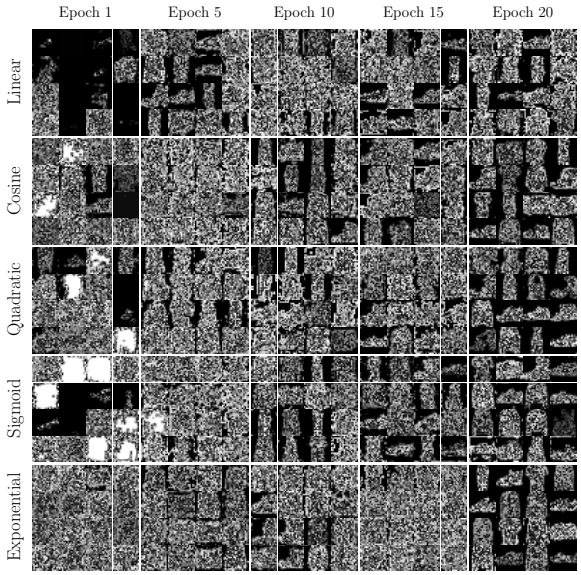


Fig. 7. Fashion-MNIST generated samples displayed in a grid format with rows representing noise schedules and columns representing selected epochs (1, 5, 10, 15, and 20). Each cell shows 16 generated clothing item images, visualizing the progressive refinement of features during training.

schedules demonstrate advantages in CIFAR-10, especially in the later training stages. The linear schedule, despite strong training loss performance, shows mixed results in sample quality metrics, performing well on Fashion-MNIST but less so on CIFAR-10. The exponential schedule consistently underperforms in both training and generative quality metrics.

These findings suggest that the effectiveness of noise schedules in DDPMs is context-dependent, varying with the complexity of the dataset and the evaluation criteria. For simpler datasets such as MNIST, characterized by low dimensionality and relatively uniform image structures, schedules such as cosine that emphasize smooth noise progression facilitate better alignment between generated and real data distributions, as reflected in superior FID and KID scores. In contrast, for more complex datasets like CIFAR-10, which feature higher variability in texture, color, and object composition, schedules such as quadratic and sigmoid that modulate noise variance more aggressively in later diffusion steps better capture intricate data characteristics, leading to improved sample diversity and quality, as indicated by IS and KID.

Qualitative Analysis: We visually inspected the samples generated from each noise schedule at selected epochs to assess the progression of sample quality and structure over training. For each data set, 16 images per schedule and epoch were sampled, arranged in matrix format that reveal the evolution of generated images.

For the MNIST dataset (epochs 1, 5, 10, 15 and 20), as illustrated in Fig. 6, the earliest samples from the linear and sigmoid schedules appear largely unstructured, while the cosine, quadratic, and exponential schedules show early hints of digit-like shapes. By epochs 5 and 10, all schedules begin producing more defined digits, with cosine, quadratic, and linear schedules slightly ahead in clarity. By epoch 20,

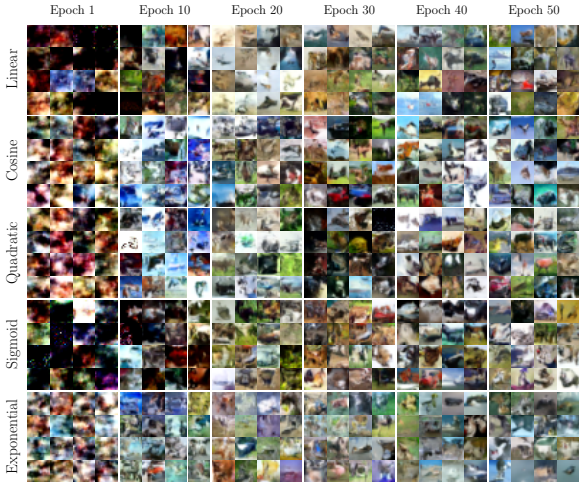


Fig. 8. CIFAR-10 generated images organized as a matrix with rows for different noise schedules and columns for selected epochs (1, 10, 20, 30, 40, and 50). Each cell contains 16 generated natural images, demonstrating the gradual development of detailed features throughout training.

recognizable digits emerge consistently across all schedules, indicating convergence toward realistic sample generation.

A comparable progression is observed for Fashion-MNIST in Fig. 7. The initial epochs show less defined and less recognizable images of linear, sigmoid, and quadratic schedules. However, by epoch 10, all schedules generate well-formed images with discernible features, and by epoch 20, the samples are visually recognizable and detailed, reflecting effective learning of clothing item characteristics.

For the CIFAR-10 dataset, Fig. 8 highlights how cosine, quadratic, and exponential schedules facilitate faster reconstruction of meaningful features starting from early epochs (1 through 20). As training advances toward epoch 50, these schedules progressively capture and refine salient image details. This suggests that certain noise schedules may better support rapid feature acquisition in complex datasets.

Together, these qualitative insights complement the quantitative metrics, illustrating how the choice of noise schedule influences the speed and quality of sample formation during training across data sets of varying complexity.

Discussion: The results presented highlight the significant impact that the choice of the noise schedule has on both the training dynamics and the quality of the generated samples in the probabilistic diffusion models. Across all datasets, the linear noise schedule consistently demonstrated the fastest and most stable training convergence, as evidenced by the lowest mean squared error (MSE) losses. This suggests that a uniform noise increment facilitates smoother optimization of the denoising function.

However, when evaluating generative quality through metrics such as FID, KID, and Inception Score, the superiority of the linear schedule is less clear-cut. For simpler datasets like MNIST, the cosine schedule outperformed others in key metrics, indicating that its noise variance progression better preserves and reconstructs the underlying data distribution. In the more complex CIFAR-10 dataset, quadratic and sigmoid schedules showed advantages, particularly in later training epochs, suggesting that more aggressive noise modulation in later diffusion steps may better capture complex image features.

The qualitative analysis of the generated samples corroborates these findings. During training, schedules such as cosine, quadratic, and exponential exhibit faster emergence of structured and recognizable images, especially on complex datasets. This indicates that while linear schedules optimize training loss efficiently, nonlinear schedules may accelerate perceptual quality improvements in generated samples.

The divergence in metric performance also underscores the importance of considering multiple evaluation criteria. Metrics such as FID and KID focus on distributional similarity, while the Inception Score balances quality and diversity, leading to differing rankings among schedules. This suggests that the optimal noise schedule may depend on specific application goals, such as whether fidelity, diversity, or training efficiency is prioritized.

In general, these findings emphasize that the effectiveness of noise schedules is context-dependent and that no single schedule consistently excels across all datasets and metrics.

5 Conclusions

In this work, we systematically evaluated the impact of various noise schedules on the training dynamics and generative performance of DDPMs across datasets of varying complexity. Our results demonstrate that while the linear noise schedule facilitates the most rapid training convergence, alternative schedules such as cosine and quadratic can yield superior sample quality and diversity depending on the dataset and evaluation metric.

These insights highlight the importance of tailoring noise schedules to the specific characteristics of the target data set and the goals of the generative task. Future research could explore adaptive or hybrid noise schedules that dynamically adjust noise variance during training to take advantage of the strengths of different schedules, potentially improving both training efficiency and sample quality.

Acknowledgments

The first author acknowledges the support provided by the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) through a scholarship that enabled the pursuit of graduate studies. The authors further thank the Instituto Politécnico Nacional, Secretaría de Investigación y Posgrado (SIP-IPN) for their valuable support of this research through projects SIP-20251087 and SIP-20251109.

References

1. **Alcaraz, J. L., Strodthoff, N. (2022).** Diffusion-based time series imputation and forecasting with structured state space models.
2. **Bińkowski, M., Sutherland, D. J., Arbel, M., Gretton, A. (2018).** Demystifying MMD GANs. International Conference on Learning Representations.
3. **Brock, A., Donahue, J., Simonyan, K. (2019).** Large scale GAN training for high fidelity natural image synthesis.
4. **Burgos, N., Svoboda, D.**, editors **(2022).** Biomedical image synthesis and simulation. The MICCAI Society book Series. Academic Press, San Diego, CA.
5. **Capel, E. H., Dumas, J. (2023).** Denoising diffusion probabilistic models for probabilistic energy forecasting. 2023 IEEE Belgrade PowerTech, pp. 1–6.
6. **Chen, T. (2023).** On the importance of noise scheduling for diffusion models.
7. **Corso, G., Stärk, H., Jing, B., Barzilay, R., Jaakkola, T. S. (2023).** Diffdock: Diffusion steps, twists, and turns for molecular docking. The Eleventh International Conference on Learning Representations.
8. **Dhariwal, P., Nichol, A. (2021).** Diffusion models beat gans on image synthesis. Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Curran Associates Inc., Red Hook, NY, USA.
9. **Dhariwal, P., Nichol, A. (2021).** Diffusion models beat gans on image synthesis. **Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J. W.**, editors, Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., pp. 8780–8794.
10. **Di Feola, F., Tronchin, L., Soda, P. (2023).** A Comparative Study Between Paired And Unpaired Image Quality Assessment In Low-Dose CT Denoising . 2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS), IEEE Computer Society, Los Alamitos, CA, USA, pp. 471–476.
11. **Foti, S., Zafeiriou, S., Birdal, T. (2024).** UV-free texture generation with denoising and geodesic heat diffusion. The Thirty-eighth Annual Conference on Neural Information Processing Systems.
12. **Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014).** Generative adversarial nets. Advances in Neural Information Processing Systems, pp. 2672–2680.
13. **Guo, Z., Lang, J., Huang, S., Gao, Y., Ding, X. (2025).** A comprehensive review on noise control of diffusion model. ArXiv, Vol. abs/2502.04669.
14. **Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. (2017).** Gans trained

by a two time-scale update rule converge to a local nash equilibrium. Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, pp. 6629–6640.

15. **Ho, J., Chen, X., Srinivas, A., Duan, Y., Abbeel, P. (2019).** Flow++: Improving flow-based generative models with variational dequantization and architecture design. **Chaudhuri, K., Salakhutdinov, R.**, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, PMLR, pp. 2722–2730.
16. **Ho, J., Jain, A., Abbeel, P. (2020).** Denoising diffusion probabilistic models. Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA.
17. **Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., Salimans, T. (2022).** Cascaded diffusion models for high fidelity image generation. Journal of Machine Learning Research, Vol. 23, No. 47, pp. 1–33.
18. **Ho, J., Salimans, T. (2021).** Classifier-free diffusion guidance.
19. **Ho, J., Salimans, T., Gritsenko, A. A., Chan, W., Norouzi, M., Fleet, D. J. (2022).** Video diffusion models. ICLR Workshop on Deep Generative Models for Highly Structured Data.
20. **Hoogeboom, E., Heek, J., Salimans, T. (2023).** Simple diffusion: end-to-end diffusion for high resolution images. Proceedings of the 40th International Conference on Machine Learning, ICML'23, JMLR.org.
21. **Jabri, A., Fleet, D. J., Chen, T. (2023).** Scalable adaptive computation for iterative generation. Proceedings of the 40th International Conference on Machine Learning, ICML'23, JMLR.org.
22. Khader, F., Müller-Franzes, G., Tayebi Arasteh, S., Han, T., Haarburger, C., Schulze-Hagen, M., Schad, P., Engelhardt, S., Baeßler, B., Foersch, S., others **(2023).** Denoising diffusion probabilistic models for 3d medical image generation. Scientific Reports, Vol. 13, No. 1, pp. 7303.
23. **Kingma, D. P., Ba, J. (2014).** Adam: A method for stochastic optimization. CoRR, Vol. abs/1412.6980.
24. **Kingma, D. P., Salimans, T., Welling, M. (2015).** Variational dropout and the local reparameterization trick. Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, MIT Press, Cambridge, MA, USA, pp. 2575–2583.
25. **Kingma, D. P., Welling, M. (2013).** Auto-encoding variational bayes. CoRR, Vol. abs/1312.6114.
26. **Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B. (2021).** Diffwave: A versatile diffusion model for audio synthesis. International Conference on Learning Representations.
27. **Krizhevsky, A., Nair, V., Hinton, G. (2009).** CIFAR-10 (Canadian Institute for Advanced Research).
28. **LeCun, Y., Cortes, C., Burges, C. (2010).** Mnist handwritten digit database. ATT Labs, Vol. 2.
29. **Letafati, M., Ali, S., Latva-aho, M. (2024).** Denoising diffusion probabilistic models for hardware-impaired communications. 2024 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6.
30. **Letafati, M., Ali, S., Latva-Aho, M. (2025).** Conditional denoising diffusion probabilistic models for data reconstruction enhancement in wireless communications. IEEE Transactions on Machine Learning in Communications and Networking, Vol. 3, pp. 133–146.
31. **Li, H., Yang, Y., Chang, M., Chen, S., Feng, H., Xu, Z., Li, Q., Chen, Y. (2022).** Srdiff: Single image super-resolution with diffusion probabilistic models. Neurocomputing, Vol. 479, pp. 47–59.
32. **Lin, H., Chen, Y., Wang, J., An, W., Wang, M., Tian, F., Liu, Y., Dai, G., Wang, J., Wang, Q. (2024).** Schedule your edit: A simple yet effective diffusion noise schedule for image editing. The Thirty-eighth Annual Conference on Neural Information Processing Systems.
33. **Lin, S., Liu, B., Li, J., Yang, X. (2024).** Common diffusion noise schedules and sample steps are flawed. 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 5392–5399.
34. **Manica, M., Born, J., Cadow, J., Christofidellis, D., Dave, A., Clarke, D., Teukam, Y. G. N., Giannone, G., Hoffman, S. C., Buchan, M., Chenthamarakshan, V., Donovan, T., Hsu, H. H., Zipoli, F., Schilter, O., Kishimoto, A., Hamada, L., Padhi, I., Wehden, K., McHugh, L., Khrabrov, A., Das, P., Takeda, S., Smith, J. R. (2023).** Accelerating material design with the generative toolkit for scientific discovery. npj Computational Materials, Vol. 9, No. 1.

35. **Nichol, A. Q., Dhariwal, P. (2021).** Improved Denoising Diffusion Probabilistic Models. Proceedings of the 38th International Conference on Machine Learning, PMLR. ISSN: 2640-3498.
36. **Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M. (2021).** Grad-tts: A diffusion probabilistic model for text-to-speech. **Meila, M., Zhang, T.**, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, PMLR, pp. 8599–8608.
37. **Rezende, D., Mohamed, S. (2015).** Variational inference with normalizing flows. PMLR, Vol. 37, pp. 1530–1538.
38. **Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2022).** High-Resolution Image Synthesis with Latent Diffusion Models . 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, pp. 10674–10685.
39. **Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2022).** High-Resolution Image Synthesis with Latent Diffusion Models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, pp. 10674–10685.
40. **Ronneberger, O., Fischer, P., Brox, T. (2015).** U-net: Convolutional networks for biomedical image segmentation. **Navab, N., Hornegger, J., Wells, W. M., Frangi, A. F.**, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, pp. 234–241.
41. **Sabour, A., Fidler, S., Kreis, K. (2024).** Align your steps: optimizing sampling schedules in diffusion models. Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org.
42. **Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M. (2022).** Palette: Image-to-image diffusion models. ACM SIGGRAPH 2022 Conference Proceedings, SIGGRAPH'22, Association for Computing Machinery, New York, NY, USA.
43. **Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X. (2016).** Improved techniques for training gans. **Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R.**, editors, Advances in Neural Information Processing Systems, volume 29, Curran Associates, Inc.
44. **Sami, M., Mobin, I. (2019).** A comparative study on variational autoencoders and generative adversarial networks. 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), pp. 1–5.
45. **Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S. (2015).** Deep unsupervised learning using nonequilibrium thermodynamics. **Bach, F., Blei, D.**, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, PMLR, Lille, France, pp. 2256–2265.
46. **Song, J., Meng, C., Ermon, S. (2021).** Denoising diffusion implicit models. 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net.
47. **Wang, Y., Wang, X., Dinh, A.-D., Du, B., Xu, C. (2023).** Learning to schedule in diffusion probabilistic models. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, Association for Computing Machinery, New York, NY, USA, pp. 2478–2488.
48. **Webber, G., Reader, A. J. (2024).** Diffusion models for medical image reconstruction. BJR—Artificial Intelligence, Vol. 1, No. 1, pp. ubae013.
49. **Xiao, H., Rasul, K., Vollgraf, R. (2017).** Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv e-prints, arXiv:1708.07747.
50. **Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.-H. (2023).** Diffusion models: A comprehensive survey of methods and applications. ACM Comput. Surv., Vol. 56, No. 4.
51. **Yang, R., Srivastava, P., Mandt, S. (2023).** Diffusion probabilistic modeling for video generation. Entropy, Vol. 25, No. 10, pp. 1469.
52. **Zhao, S., Ren, H., Yuan, A., Song, J., Goodman, N., Ermon, S. (2018).** Bias and generalization in deep generative models: An empirical study. Advances in Neural Information Processing Systems, pp. 10792–10801.

Article received on 19/05/2025; accepted on 22/09/2025.
**Corresponding author is GuoHua Sun.*