Team 6: Implementation and Evaluation of Diffusion Models (DDPM, DDIM, Latent Diffusion Models) in Image Generation

Aditya Aayush
Carnegie Mellon University
Pittsburgh, PA 15213
aaayush@andrew.cmu.edu

Yash Jaiswal
Carnegie Mellon University
Pittsburgh, PA 15213
yjaiswal@andrew.cmu.edu

Priya Lalwani
Carnegie Mellon University
Pittsburgh, PA 15213
plalwani@andrew.cmu.edu

Abstract

Diffusion models have gained significant attention in generative modeling for their ability to synthesize high-quality data. This report addresses the problem of efficiently generating high-fidelity images by implementing and improving Denoising Diffusion Probabilistic Models (DDPM), DDIM, and Latent Diffusion Models (LDM). The primary focus is on the design and evaluation of key components, including U-Net and Variational Autoencoders (VAE), which act as the foundation of these models. Inputs to the models involve progressively corrupted data, and outputs are restored through a learned denoising process. This study is motivated by the increasing demand for more efficient and flexible generative models, which can impact fields ranging from image synthesis to natural language processing. Experimental results highlight improvements in image quality and computational efficiency, evaluated using metrics such as FID and Inception Score. The abstract articulates a concise overview of the methods, motivation, and outcomes of this study, emphasizing its relevance and contributions to generative modeling research. Our work can be accessed here : https://github.com/aditya0520/DiffusionModels

1 Introduction

Diffusion models have emerged as a powerful approach in generative modeling, enabling the synthesis of high-quality images and other complex data structures. This project focuses on implementing and evaluating key diffusion-based methods, including Denoising Diffusion Probabilistic Models (DDPM), Denoising Diffusion Implicit Models (DDIM), and Latent Diffusion Models (LDM). These models will be analyzed to understand their architectural innovations, training configurations, and performance trade-offs.

To evaluate the models comprehensively, we utilize well-established datasets such as CIFAR-10 and ImageNet-128. The performance of these models will be assessed using quantitative metrics, including Frechet Inception Distance (FID) and Inception Score (IS), and qualitative analysis through generated image outputs.

The primary objectives of this project are:

- To implement state-of-the-art diffusion models, including DDPM, DDIM, and LDM.
- To explore the impact of different training configurations, such as Classifier-Free Guidance (CFG) and beta scheduling.
- To evaluate the models using both quantitative metrics and visual results, identifying key strengths and areas for improvement.

Preprint. Under review.

This project aims to provide a comprehensive understanding of diffusion models, highlighting their strengths and limitations while contributing to the ongoing research in generative modeling.

2 Literature Review

Generative modeling has witnessed significant advancements over the years, with diffusion models emerging as a leading framework for generating high-quality data. This section reviews the technical foundations and key advancements in diffusion models.

2.1 Diffusion Models: Forward and Reverse Process

Diffusion models rely on two main phases: the forward process and the reverse process. In the forward process, Gaussian noise is gradually added to the data over a series of timesteps, eventually transforming the data into pure noise. This process is modeled as a Markov chain, ensuring that each timestep depends only on the previous one. The reverse process involves a learned denoising mechanism that progressively removes noise, reconstructing data from noise step by step. This probabilistic approach enables accurate modeling of complex data distributions (1).

2.2 Denoising Diffusion Probabilistic Models (DDPM)

Denoising Diffusion Probabilistic Models (DDPM), introduced by Ho et al. (1), represent a significant milestone in generative modeling. By learning the reverse diffusion process, DDPMs generate high-quality data while addressing limitations of earlier generative models, such as instability in GANs. However, DDPMs suffer from slow inference due to the iterative nature of the reverse process.

2.3 Denoising Diffusion Implicit Models (DDIM)

To address the slow inference of DDPMs, Song et al. (2) introduced Denoising Diffusion Implicit Models (DDIM). DDIM modifies the reverse diffusion process to follow a deterministic path, significantly reducing the number of required timesteps during inference. This improvement makes DDIM a practical alternative for scenarios requiring fast generation without compromising image quality.

2.4 Latent Diffusion Models (LDM)

Rombach et al. (3) proposed Latent Diffusion Models (LDM) to improve the scalability of diffusion models. By operating in a compressed latent space instead of the pixel space, LDMs reduce computational overhead while maintaining image fidelity. This approach enables high-resolution image synthesis and efficient training on large datasets like ImageNet.

2.5 Classifier-Free Guidance (CFG)

Classifier-Free Guidance (CFG), introduced by Ho and Salimans (4), enhances conditional generation capabilities without requiring a separate classifier. By training the model on both conditional and unconditional data, CFG enables flexible control over the outputs, making it an essential extension for applications like text-to-image synthesis.

2.6 Comparison with Generative Adversarial Networks (GANs)

While GANs have been the dominant paradigm for generative modeling, diffusion models offer key advantages:

- **Training Stability:** Diffusion models exhibit stable training dynamics, avoiding the instability and mode collapse often seen in GANs.
- **Mode Coverage:** Diffusion models excel in capturing the diversity of the data distribution, addressing the limitations of GANs in mode representation.
- **Efficiency Improvements:** Although diffusion models typically require more timesteps, advancements like DDIM and LDM have mitigated these challenges, making them competitive with GANs.

3 Model Architecture

This section outlines the architectural details of the models implemented in this project, including the UNet architecture, Variational Autoencoder (VAE), and the overall diffusion pipelines.

3.1 UNet Architecture

The UNet architecture is central to both DDPM and DDIM models, designed for processing images in generative tasks. It adopts an encoder-decoder structure with skip connections, making it ideal for reconstructing images from noisy data (8).

- **Input Layer:** The model takes an input image (e.g., RGB) and passes it through an initial convolution layer that reduces spatial dimensions and prepares feature maps for subsequent layers.
- **Downsampling Blocks:** These blocks include residual connections (via ResBlock (**?**)) and optional attention mechanisms (**?**), reducing spatial resolution while increasing feature depth to capture abstract features.
- **Middle Blocks:** At the bottleneck, the middle blocks process high-level features using additional residual and attention layers to model complex dependencies.
- **Upsampling Blocks:** The upsampling path reconstructs the image by increasing spatial resolution and merging features from corresponding downsampling layers via skip connections.
- **Output Layer:** The final output passes through a GroupNorm layer (**?**) followed by a convolution, mapping features back to the original image dimensions.

3.2 Variational Autoencoder (VAE) Architecture

The VAE plays a crucial role in Latent Diffusion Models (LDM), compressing images into a lower-dimensional latent space for efficient diffusion (**?**).

- **Encoder:** Reduces the input image into a latent representation using convolutional layers with normalization and activation functions.
- **Latent Space Sampling:** Uses the reparameterization trick to sample latent codes from the Gaussian distribution.
- **Decoder:** Reconstructs the original image from the latent codes using upsampling and convolutional layers.
- **Latent Diffusion:** In LDM, noise diffusion operates directly in the latent space, reducing computational costs while preserving output quality (3).

3.3 Diffusion Pipelines

The pipelines orchestrate the forward and reverse diffusion processes to generate high-quality images (1).

3.3.1 DDPM Scheduler

The DDPM scheduler manages the diffusion process using parameters like timesteps, betas, and alphas. It includes:

- Initialization of timesteps, betas, and alphas for noise scheduling.
- Adding noise during the forward diffusion process.
- Gradual denoising during the reverse process using the learned UNet.

3.3.2 DDIM Scheduler

The DDIM scheduler improves upon DDPM by reducing the number of timesteps through a deterministic sampling process, enabling faster inference while maintaining similar output quality (2).

3.4 Classifier-Free Guidance (CFG)

CFG enhances the diffusion process by enabling conditional generation without requiring explicit class labels during inference. The key idea is to train the model on both conditional and unconditional data, interpolating between the two for controlled image generation (4).

3.5 Training Pipeline

The training pipeline consists of the following stages:

- Preprocessing datasets (e.g., CIFAR-10 (6), ImageNet-128 (7)) by resizing, normalizing, and converting to tensors.
- Sampling timesteps and Gaussian noise, followed by adding noise using the forward diffusion framework.
- Using the UNet model to predict the noise and calculating the Mean Squared Error (MSE) loss between predicted and true noise.
- Updating weights through backpropagation and optimization.

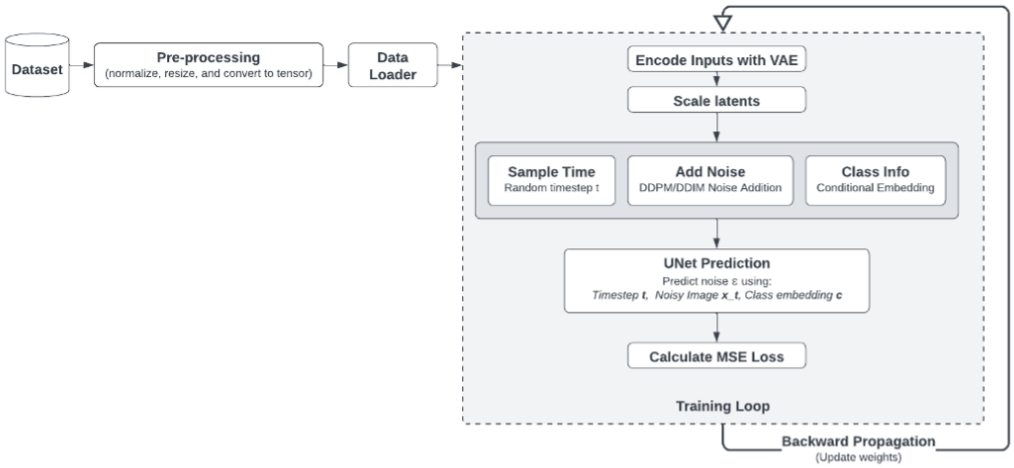


Figure 1: Overview of the training pipeline for diffusion models.

3.6 Inference Pipeline

The inference pipeline generates new images by applying the reverse diffusion process:

- Starting with random Gaussian noise and iteratively applying the reverse diffusion steps.
- Optionally incorporating conditional embeddings for guided generation.
- Decoding outputs using the VAE (for latent diffusion) to reconstruct pixel-space images (3).

4 Dataset

The dataset used for training and evaluating the diffusion models plays a crucial role in determining their performance and generalizability. This section provides details about the dataset, its preprocessing, and sampling methods used.

4.1 Data Source

For this project, we used two primary datasets: CIFAR-10 (6) and ImageNet-128 (7). CIFAR-10 is a widely used dataset for image generation tasks, consisting of 60,000 32x32 color images across 10

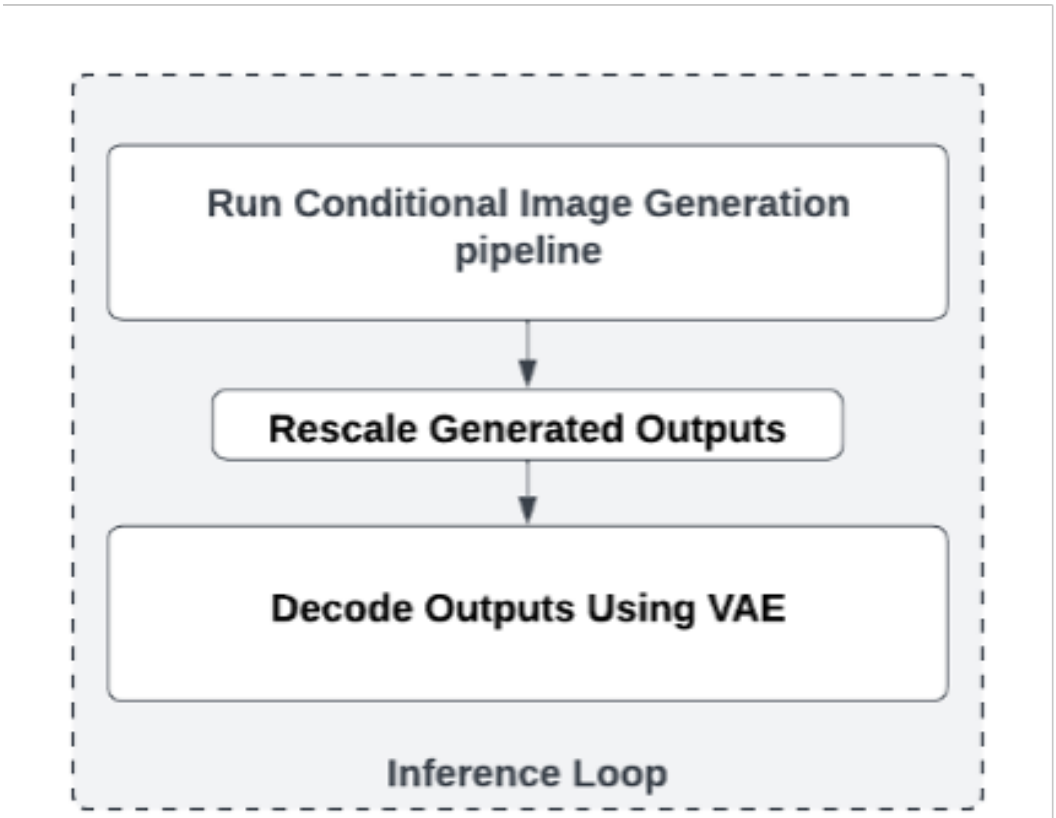


Figure 2: Overview of the inference pipeline.

classes. ImageNet-128 is a subset of the ImageNet dataset, containing images resized to 128x128 pixels, suitable for higher-resolution image generation tasks. These datasets were chosen for their diverse image content and availability in standard benchmarks for generative models.

4.2 Data Description

The datasets consist of labeled images that allow conditional image generation tasks. CIFAR-10 provides a compact dataset for experimentation and validation, while ImageNet-128 offers a more complex and high-dimensional dataset that better evaluates the scalability and fidelity of the diffusion models.

4.3 Preprocessing Steps

To make the datasets practically usable for the diffusion models, several preprocessing steps were performed:

- **Resizing**: Images from ImageNet were resized to 128x128 pixels to standardize the input size (3).
- **Normalization**: Pixel values were scaled to the range [-1, 1], a standard practice to improve model stability and training performance (**?**).
- **Transformation**: Images were converted into tensors to enable efficient processing in PyTorch-based pipelines (**?**).

4.4 Instance and Batch Sampling

The data was loaded into batches using PyTorch's `DataLoader` (**?**). This method ensures efficient instance and batch sampling with proper collation, making training feasible for large datasets like ImageNet. The following configurations were used:

- **Batch Size**: A batch size of 64 was chosen to balance memory constraints and training efficiency.
- **Shuffling**: During training, the dataset was shuffled to ensure model generalizability by avoiding biases from data order.

These steps were critical to aligning the datasets with the requirements of diffusion models and ensuring efficient training.

5 Evaluation Metrics

To evaluate the performance of the diffusion models, we used two widely adopted metrics in generative modeling: Frechet Inception Distance (FID) (**?**) and Inception Score (IS) (**?**). These metrics assess the quality and diversity of the generated samples, providing both quantitative and qualitative insights into model performance.

5.1 Frechet Inception Distance (FID)

Description: FID measures the similarity between the real and generated data distributions by computing the Frechet distance between their feature embeddings, extracted using a pre-trained Inception network (**?**). Lower FID values indicate better alignment between real and generated distributions.

Mathematical Formulation: The FID is computed as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right)$$

where:

- μ_r, μ_g: Mean vectors of the real and generated feature distributions.
- Σ_r, Σ_g: Covariance matrices of the real and generated feature distributions.
- Tr: Trace operator.

Variable Definitions:

- μ_r, Σ_r: Represent the mean and covariance of the real image embeddings.
- μ_g, Σ_g: Represent the mean and covariance of the generated image embeddings.
- $\|\cdot\|$: Denotes the Euclidean norm.

Relation to the Problem Statement: FID evaluates how well the diffusion models generate samples that resemble real data, which is crucial for tasks like image synthesis. A low FID score reflects higher fidelity and diversity in generated images.

5.2 Inception Score (IS)

Description: IS evaluates both the quality and diversity of generated images by analyzing the entropy of their class predictions using a pre-trained Inception network (**?**). Higher IS values indicate that the images are both realistic and varied.

Mathematical Formulation:

$$\text{IS} = \exp\left(\mathbb{E}_x\left[D_{\text{KL}}\left(p(y|x) \parallel p(y)\right)\right]\right)$$

where:

- x: A generated image.
- $p(y|x)$: The predicted label distribution for x.
- $p(y)$: The marginal distribution over all labels.
- $D_{\text{KL}}(p \parallel q)$: Kullback-Leibler divergence between distributions p and q.

Variable Definitions:

- $p(y|x)$: Indicates the confidence of the Inception model on a single generated image.
- $p(y)$: Measures the diversity across all generated samples.
- D_{KL}: Quantifies the difference between the conditional and marginal distributions.

Relation to the Problem Statement: IS is directly tied to the problem of generating high-quality and diverse images. A high IS score signifies that the model produces sharp, realistic images covering a broad range of classes.

Both FID and IS provide complementary insights into the performance of diffusion models, ensuring a rigorous and comprehensive evaluation of their ability to generate realistic and diverse samples.

6 Loss Function

The training of diffusion models relies on a carefully designed loss function that aligns with their generative objectives. The loss function ensures the accurate prediction of the noise added during the forward process and enables the reverse process to denoise the data effectively.

Definition of the Loss Function: The primary loss function used in diffusion models is the Mean Squared Error (MSE) between the predicted noise and the true noise added to the input data during the forward diffusion process. This loss encourages the model to learn the reverse diffusion process accurately (1).

Mathematical Formulation: The loss function is defined as:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta\left(\bar{\alpha}_t x_0 + (1-\bar{\alpha}_t)\epsilon, t\right)\|_2^2\right]$$

where:

- t: Randomly sampled timestep.
- x_0: Original data (image).
- ϵ: Random Gaussian noise.
- ϵ_θ: Model's predicted noise given the noisy data and timestep.
- $\bar{\alpha}_t$: Cumulative product of noise scaling terms α_t, representing the proportion of the original data preserved at timestep t.

Variable and Function Definitions:

- ϵ: Represents the true noise added to the input during the forward process.
- ϵ_θ: The predicted noise output by the model, parameterized by θ.
- $\|\cdot\|_2$: Denotes the squared L_2-norm.
- $\bar{\alpha}_t x_0 + (1-\bar{\alpha}_t)\epsilon$: Represents the noisy data at timestep t.

Relation to the Problem Statement: This loss function directly supports the goal of training diffusion models for high-quality image generation. By minimizing the error in noise prediction, the model learns to denoise inputs effectively, enabling the generation of realistic and high-fidelity samples. The choice of MSE is critical, as it provides a simple yet effective way to train the reverse diffusion process (2).

7 Baseline

7.1 Baseline Selection

This project utilizes well-established datasets such as CIFAR-10 (6) and ImageNet-128 (7), which are widely recognized benchmarks in the field of generative modeling. The use of diffusion-based approaches, including Denoising Diffusion Probabilistic Models (DDPM) (1) and Denoising Diffusion Implicit Models (DDIM) (2), provides a robust baseline for evaluating performance.

DDPM and DDIM have been extensively studied and are known for their ability to generate high-fidelity images with controllable quality through iterative noise-removal processes. These models represent state-of-the-art (SOTA) techniques in image generation, making them an ideal foundation for benchmarking and analyzing further improvements explored in this project.

7.2 Baseline Implementation

The baseline implementation for this project focuses on the Denoising Diffusion Probabilistic Model (DDPM) without incorporating advanced techniques such as Classifier-Free Guidance (CFG) (4) or Variational Autoencoders (VAEs) (9). This choice ensures a straightforward evaluation of the foundational diffusion model, which serves as a starting point for further enhancements.

The DDPM implementation is based on its original formulation, where the forward process gradually adds Gaussian noise to the input data over multiple timesteps, and the reverse process learns to denoise step-by-step (1).

The U-Net architecture (8) is utilized as the backbone of the model. This encoder-decoder structure, with skip connections, facilitates efficient feature extraction and reconstruction of noisy data during the reverse diffusion process.

The noise scheduler follows a pre-defined linear or cosine beta schedule (5), which controls the noise level at each timestep. This schedule ensures a gradual and systematic transition from noisy data to a denoised sample.

7.3 Training Configuration

The model is trained using the Mean Squared Error (MSE) loss function (1), which measures the difference between the predicted noise and the true noise added during the forward diffusion process.

CIFAR-10 (6) and ImageNet-128 (7) datasets are used to train and evaluate the model, providing a diverse set of image data for testing the generative capabilities of the baseline.

7.4 Purpose of Baseline Implementation

By starting with a minimal configuration of DDPM without CFG or VAEs, the baseline provides a clear understanding of the model's core generative performance. It establishes a benchmark against which the impact of advanced techniques, such as latent-space modeling and guided generation, can be quantitatively and qualitatively assessed.

8 Experimentation

To evaluate the performance of the implemented diffusion models, several experiments were conducted using different configurations of datasets, schedulers, and enhancements. The experiments are summarized in the table below:

- **Experiment A:** Generated images for DDPM with Cosine Beta scheduler and CFG enabled on the ImageNet-128 dataset (4).
- **Experiment B:** Results for DDPM with Linear Beta scheduler and CFG enabled on ImageNet-128 (5).
- **Experiment C:** Outputs from DDPM with Linear Beta scheduler and CFG disabled on ImageNet-128.

Experiment	Model	Dataset	Scheduler	CFG	Epochs	FID
A	DDPM	ImageNet-128	Linear Beta	Yes	10	342.2
B	DDPM	ImageNet-128	Cosine Beta	Yes	17	303.1
C	DDPM	ImageNet-128	Linear Beta	No	10	
D	DDIM	CIFAR-10	Linear Beta	No	100	
E	DDPM	ImageNet-128	Cosine Beta	Yes	X (KL_VAE included)	

Table 1: Summary of experiments conducted with various configurations of models, datasets, schedulers, enhancements, and FID values.

- **Experiment D:** Images generated by DDIM with Linear Beta scheduler on CIFAR-10 (2).
- **Experiment E:** Generated images from DDPM with Cosine Beta scheduler, CFG enabled, and latent diffusion (KL_VAE included) on ImageNet-128 (3).

Purpose of Experiments

Each experiment was designed to test specific aspects of the models:

- The effect of different schedulers (cosine vs. linear beta) on model performance.
- The impact of enabling or disabling Classifier-Free Guidance (CFG) on conditional and unconditional generation.
- The advantages of integrating newer techniques such as DDIM and latent diffusion using KL_VAE.

9 Results and Observations

The following subsection showcases the generated images from the diffusion models for various experimental configurations. Each set of images highlights the results obtained under specific conditions, including variations in beta schedulers, the presence or absence of Classifier-Free Guidance (CFG), and differences in datasets.

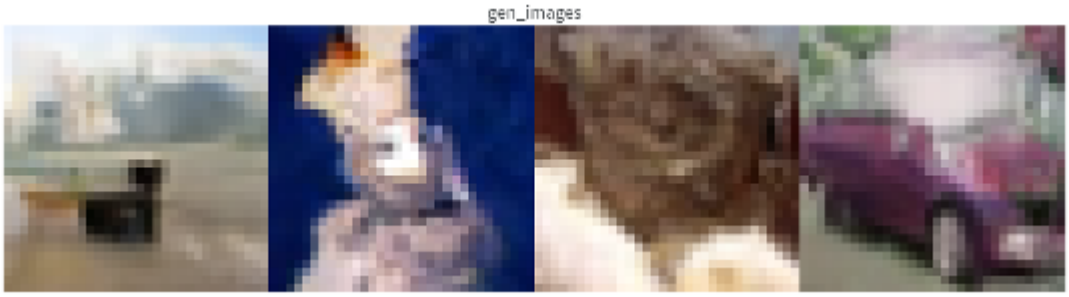
Generated Images for Each Experiment

- **Experiment A:** Generated images for DDPM with Cosine Beta scheduler and CFG enabled on the ImageNet-128 dataset.

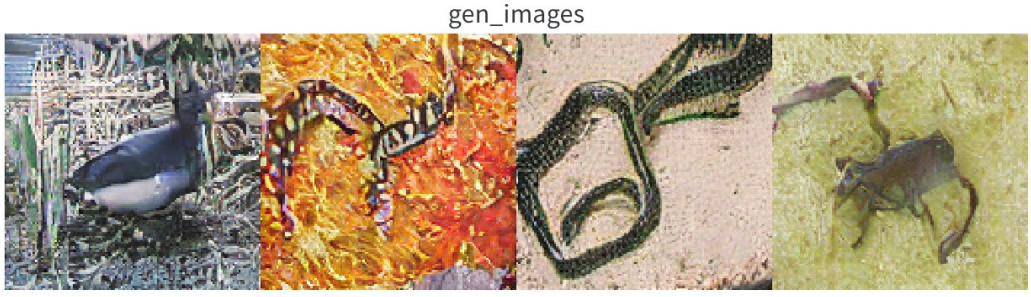
- **Experiment B:** Results for DDPM with Linear Beta scheduler and CFG enabled on ImageNet-128.

- **Experiment C:** Outputs from DDPM with Linear Beta scheduler and CFG disabled on ImageNet-128.

- **Experiment D:** Images generated by DDIM with Linear Beta scheduler on CIFAR-10, highlighting the efficiency gains in sampling.



- **Experiment E:** Generated images from DDPM with Cosine Beta scheduler, CFG enabled, and latent diffusion (KL_VAE included) on ImageNet-128.



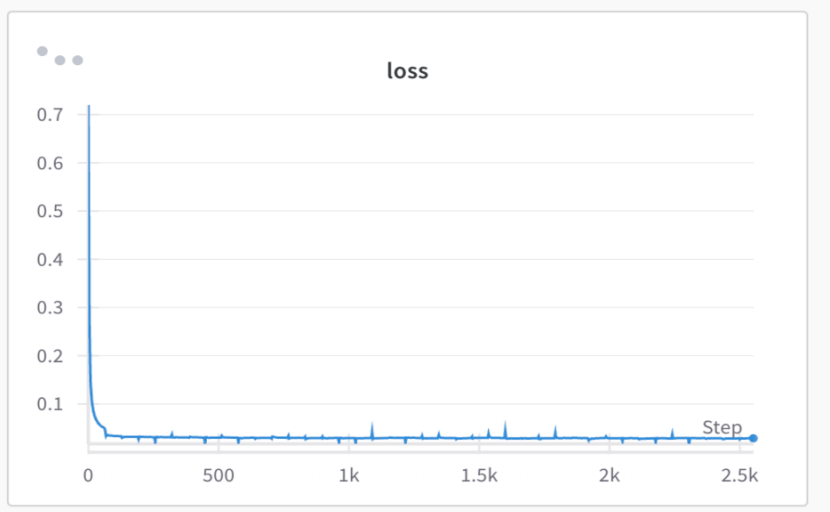
Observations

- Models trained with **Cosine Beta scheduling** demonstrated higher fidelity and smoother outputs.
- **Conditional generation with CFG** improved alignment of generated images with target classes, enhancing both diversity and fidelity.

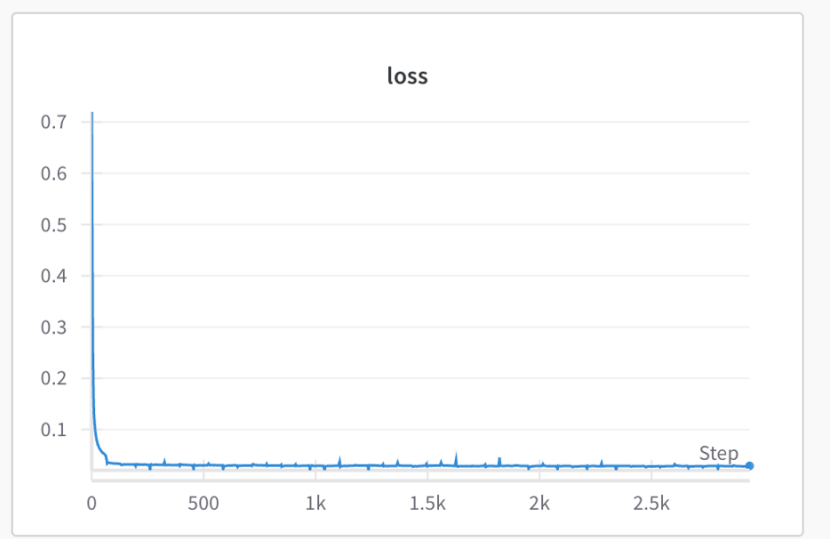
- **DDIM** offered significant speedups in sampling while maintaining comparable image quality to DDPM.

This systematic exploration provided insights into the trade-offs and benefits of different configurations, establishing a robust comparison framework for evaluating diffusion models.

9.1 DDPM Results



9.2 DDIM Results



10 Conclusion

This study demonstrates the potential of diffusion models, including DDPM, DDIM, and Latent Diffusion Models, in generating high-quality and diverse images across various configurations. By evaluating these models on CIFAR-10 and ImageNet-128 datasets, it highlights key strengths such as stability, scalability, and the ability to handle high-resolution data, while identifying limitations like slow inference and sensitivity to input configurations. These findings contribute to a deeper understanding of diffusion-based generative modeling and establish a foundation for further advancements in this domain.

11 Discussion

The experimental results show that diffusion models, particularly those with Cosine Beta scheduling and Classifier-Free Guidance (CFG), produce superior image quality and alignment with conditional inputs compared to Linear Beta scheduling. Latent Diffusion Models further improve efficiency by operating in a compressed latent space, but they require careful tuning to preserve image fidelity. The choice of noise schedules, guidance levels, and training configurations significantly impacts the outcomes, and these sensitivities must be carefully managed to ensure consistent performance across datasets.

12 Future Works

Future work could focus on optimizing diffusion models for real-world applications by developing adaptive noise scheduling, improving latent representations, and exploring multi-scale training for high-resolution datasets. Addressing limitations such as sensitivity to input configurations, computational overhead, and data bias will be critical to scaling these models effectively. Additionally, integrating diffusion models into practical applications like video synthesis or medical imaging can expand their impact beyond academic research.

13 Division of Work

The responsibilities for this project were distributed among the team members as follows: Aditya Aayush focused on the implementation of the Denoising Diffusion Probabilistic Model (DDPM) and conducted the related experimentation to evaluate its performance across various configurations. Yash Jaiswal contributed by implementing the Denoising Diffusion Implicit Model (DDIM) and conducting research to enhance the understanding and optimization of the DDPM framework. Priya Lalwani took charge of compiling the team's work, crafting the literature review, and creating the mid-term and final reports, ensuring a comprehensive and cohesive presentation of the project.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[2] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," *International Conference on Learning Representations (ICLR)*, 2021.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

[4] J. Ho and T. Salimans, "Classifier-Free Diffusion Guidance," *arXiv preprint arXiv:2107.00630*, 2021.

[5] A. Q. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," *arXiv preprint arXiv:2102.09672*, 2021.

[6] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," *Technical Report, University of Toronto*, 2009.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *International Conference on Learning Representations (ICLR)*, 2014.