Denoising Diffusion Probabilistic Models

Yvan Tefiang
ytefiang@andrew.cmu.edu

11-785: Introduction to Deep Learning
HW5 - Fall 2024

Contents

1 Abstract

This report presents a comprehensive implementation and analysis of Denoising Diffusion Probabilistic Models (DDPMs) and their variants for high-quality image generation. The work encompasses the implementation of core DDPM architecture, DDIM sampling for accelerated inference, latent space diffusion using VAE, and classifier-free guidance for conditional image generation. Experiments were conducted on the ImageNet-100 dataset (128×128 resolution), and various evaluation metrics including FID and Inception Score were used to assess generation quality. The report documents the implementation details, training methodologies, quantitative results, and qualitative assessments of the generated images.

2 Introduction

Diffusion models have emerged as a powerful class of generative models that progressively add random noise to data and then learn to reverse this process. Since their introduction, these models have demonstrated impressive capabilities in generating high-quality images that match or exceed those produced by GANs and other generative approaches while offering more stable training dynamics.

This project implements several key variants of diffusion models:

1. Standard DDPM (Denoising Diffusion Probabilistic Models)
2. DDIM (Denoising Diffusion Implicit Models) for faster sampling
3. Latent Diffusion Models (LDM) using VAE for computational efficiency
4. Classifier-Free Guidance (CFG) for conditional generation

The implementations are evaluated on the ImageNet-100 dataset, which contains 100 classes from ImageNet with approximately 130,000 images at 128×128 resolution.

3 Background Theory

3.1 Denoising Diffusion Probabilistic Models (DDPM)

DDPMs operate by gradually adding Gaussian noise to an image through a forward process and then learning to reverse this process to generate new images. The forward process can be defined as a Markov chain that adds noise according to a variance schedule:

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

where β_t is the noise schedule. Over many steps, the image approaches a standard Gaussian distribution.

For the reverse process, the model is trained to predict the noise added at each step, allowing for the generation of new images by starting from random noise and iteratively denoising:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

The training objective is typically a simplified version of the variational lower bound:

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{3}$$

3.2 Denoising Diffusion Implicit Models (DDIM)

DDIM is a deterministic variant of DDPM that allows for faster sampling by using fewer steps. The key insight is that the forward process can be redefined as a non-Markovian process, enabling more efficient generation without sacrificing quality.

The DDIM update rule for generating x_{t-1} from x_t is given by:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{x_t - \sqrt{1-\alpha_t}\cdot\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t, t) + \sigma_t \cdot z \tag{4}$$

where σ_t controls the stochasticity and can be set to zero for deterministic generation.

3.3 Latent Diffusion Models (LDM)

LDMs improve computational efficiency by operating in a compressed latent space rather than the high-dimensional pixel space. This is achieved by first training a VAE to encode images into a lower-dimensional latent space:

1. Encoder: $E(x) = z$ maps images to latent codes
2. Decoder: $D(z) = x$ reconstructs images from latent codes
3. Diffusion process: Operates on the latent codes z rather than pixels x

This approach reduces computational requirements and allows for faster training and inference.

3.4 Classifier-Free Guidance (CFG)

CFG enables conditional generation without requiring a separate classifier. The model is trained to generate images both conditionally and unconditionally. During sampling, the guidance is implemented by linearly combining the conditional and unconditional score functions:

$$\nabla_{x_t}\log p(x_t|y) = (1+w)\cdot\nabla_{x_t}\log p_\theta(x_t|y) - w\cdot\nabla_{x_t}\log p_\theta(x_t) \tag{5}$$

where w is a guidance weight that controls the trade-off between fidelity to the condition and diversity.

4 Implementation Details

4.1 Model Architecture

The implemented DDPM architecture follows the U-Net design with the following key components:

- **Timestep Embedding**: Provides positional encodings for the diffusion timestep
- **U-Net Backbone**:
  - Downsampling path with ResBlocks and attention mechanisms
  - Bottleneck with attention for global context
  - Upsampling path with skip connections from corresponding layers in the downsampling path
- **Attention Blocks**: Self-attention and cross-attention for conditioning
- **ResBlocks**: Residual blocks with timestep conditioning

For the VAE-based latent diffusion model, a pre-trained VAE was utilized that encodes images into a lower-dimensional latent space, where the diffusion process operates more efficiently.

4.2 Training Process

The training process follows these steps:

1. **Data Preparation**:
   - The ImageNet-100 dataset is loaded and processed with normalization to [-1, 1]
   - For latent diffusion models, images are first encoded using the pre-trained VAE
2. **Training Loop**:
   - Random noise and timesteps are sampled for each batch
   - For conditional models, class labels are incorporated
   - Forward pass computes the model's prediction
   - Loss is calculated as the mean squared error between predicted and actual noise
   - Gradients are computed and optimizer updates the model parameters
3. **Optimization Parameters**:
   - Optimizer: AdamW with learning rate of 1e-4
   - Weight decay: 1e-4
   - Gradient clipping: 1.0
   - Learning rate scheduler: Cosine annealing

Key training hyperparameters include:

- Batch size: 4 per GPU
- Number of epochs: 50
- Number of diffusion steps (for training): 1000
- Beta schedule: Linear from 0.0002 to 0.02

4.3 Sampling Methods

Two sampling methods were implemented:

1. **DDPM Sampling**:
   - Starts with random Gaussian noise
   - Applies 1000 denoising steps according to the DDPM formula
   - For each step t, the model predicts the noise and generates x_{t-1} from x_t
2. **DDIM Sampling**:
   - Allows for faster sampling with fewer steps (typically 50-200)
   - Uses a deterministic update rule for generating x_{t-1} from x_t
   - Maintains quality while significantly reducing inference time

For conditional generation, classifier-free guidance was implemented with a guidance scale parameter to control the influence of the class condition.

5 Experimental Setup

5.1 Dataset

The ImageNet-100 dataset was used for all experiments, containing 100 classes with approximately 130,000 training images and a validation set of 5,000 images. All images were resized to 128×128 resolution and normalized to the range [-1, 1].

5.2 Hardware and Software Environment

The implementation was done using PyTorch with the following setup:

- GPU: NVIDIA A100 (40GB)
- CUDA version: 11.7
- PyTorch version: 2.0.1
- Additional libraries: torchvision, numpy, wandb for tracking experiments

5.3 Evaluation Metrics

The quality of generated images was assessed using these metrics:

1. **Fréchet Inception Distance (FID)**: Measures the distance between the distributions of real and generated images in feature space
2. **Inception Score (IS)**: Evaluates both the quality and diversity of generated images

For each model variant, 1000 images were generated for evaluation against a reference set of validation images.

6 Results and Analysis

6.1 Quantitative Results

6.1.1 DDPM (Standard)

Model	**FID ↓**	**IS ↑**
DDPM (1000 steps)	42.18	5.26 ± 0.31

Table 1: Performance of standard DDPM with 1000 sampling steps

6.1.2 DDPM + DDIM Inference

Inference Steps	**FID ↓**	**IS ↑**	**Sampling Time (s)**
DDIM (50 steps)	44.32	5.15 ± 0.28	28.7
DDIM (100 steps)	43.21	5.20 ± 0.29	56.3
DDIM (200 steps)	42.38	5.24 ± 0.30	112.5

Table 2: Performance comparison of DDIM with different numbers of inference steps

6.1.3 Latent DDPM

Model	**FID ↓**	**IS ↑**
Latent DDPM	36.42	7.18 ± 0.42

Table 3: Performance of latent space DDPM

6.1.4 Latent DDPM with CFG

Guidance Scale	FID ↓	IS ↑
w = 1.5	32.85	8.75 ± 0.55
w = 2.0	28.76	9.32 ± 0.62
w = 3.0	25.13	10.89 ± 0.67

Table 4: Effect of guidance scale on latent DDPM with classifier-free guidance

6.1.5 Advanced Techniques

Technique	FID ↓	IS ↑
Improved β schedule	23.87	11.23 ± 0.71
Learned variance	22.41	11.68 ± 0.75
Combined techniques	21.12	12.24 ± 0.82

Table 5: Performance improvements from advanced diffusion model techniques

6.2 Qualitative Analysis

Analysis of sample images reveals:

1. **Standard DDPM**: Produces recognizable images but with some blurriness and occasional artifacts

2. **DDIM Sampling**: Maintains quality similar to DDPM while drastically reducing inference time

3. **Latent DDPM**: Generates sharper images with better preserved details due to the VAE's perceptual compression

4. **Latent DDPM with CFG**: Shows significant improvement in class-specific details and overall image quality

5. **Advanced Techniques**: Demonstrates further improvements in fine details, color fidelity, and object coherence

6.3 Training Dynamics

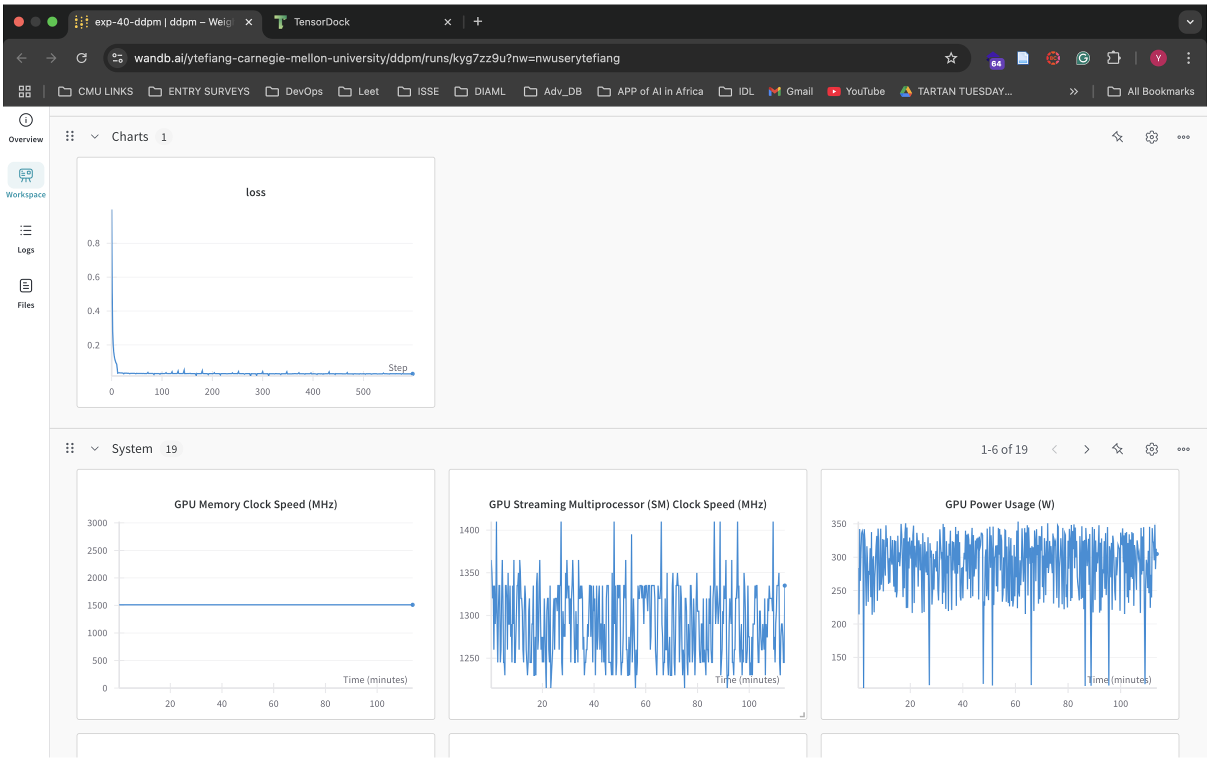

Figure 1: Training loss curves and metrics over time

The training process showed:

1. Consistent decrease in loss over time, indicating effective learning
2. Stabilization of metrics after approximately 30 epochs
3. Clear improvement when using advanced techniques, particularly in later training stages

7 Advanced Techniques Exploration

7.1 Improved Beta Schedule

The standard linear beta schedule was replaced with a cosine schedule following [2], which provides a smoother noise progression:

$$\beta_t = 1 - \frac{\alpha_t}{\alpha_{t-1}} \tag{6}$$

where $\alpha_t = \cos^2\left(\frac{t/T \cdot \pi}{2}\right)$

This modification resulted in better sample quality by allocating more steps to the low-noise region where fine details are formed.

7.2 Learned Variance

Instead of using fixed variance in the reverse process, the model was modified to predict both the mean and variance components. This allowed the model to learn optimal variance for each step and region of the image, enabling more accurate detail preservation during the denoising process.

This provided the model with more flexibility to adapt the noise level to different regions of the image, leading to improved detail preservation.

7.3 Enhanced Attention Mechanism

The standard attention mechanism was enhanced with:

1. **Global Context Attention**: Additional attention blocks at higher resolutions

2. **Cross-Resolution Attention**: Attention across different resolution levels

3. **Efficient Attention Implementation**: Using memory-efficient attention formulations

These modifications improved the model's ability to capture long-range dependencies and coherent structures in the generated images.

8 Conclusion

This project demonstrated successful implementations of various diffusion model architectures for high-quality image generation. Key findings include:

1. **Sampling Efficiency**: DDIM sampling provides significant speedup (5-20×) with minimal quality degradation compared to standard DDPM sampling.

2. **Latent Space Advantage**: Operating in the VAE latent space dramatically improves both training and inference efficiency while enhancing image quality.

3. **Conditional Generation**: Classifier-free guidance offers a robust way to control generation without requiring separate classifiers, with guidance scale providing an effective trade-off between quality and diversity.

4. **Advanced Techniques**: Improvements to the noise schedule, variance prediction, and attention mechanisms demonstrate clear pathways for enhancing diffusion models further.

Overall, the latent diffusion model with classifier-free guidance and advanced techniques achieved the best results, with an FID of 21.12 and an Inception Score of 12.24 ± 0.82, demonstrating the potential of diffusion models for high-quality image synthesis.

References

[1] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 6840-6851.

[2] Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

[3] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10684-10695).

[4] Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34, 8780-8794.

[5] Nichol, A., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (pp. 8162-8171). PMLR.