DiT Implementation Assignment Report

- Ria Shekhawat

1. Introduction

This report consolidates findings upon implementing the Diffusion Transformer architecture based on the DiT paper (Peebles & Xie, 2022; https://arxiv.org/abs/2212.09748) and the tasks provided in the assignment.
Kindly note, these experiments while providing conceptually sound results may be suboptimal due to compute constraints and lack of GPU availability. I have tried my best to perform these experiments using MPS on an M3 Macbook. However due to the system crashing multiple times, could not perform multiple iterations of training for higher dimension and image sizes. Furthermore, I have assumed 7 classes for the dataset (https://www.kaggle.com/datasets/arnaud58/landscape-pictures/data) based on its content description on Kaggle, details for the same have been added below.

1.1 DiT Architecture and Design Choices (Task 2)

The architecture implemented is a Diffusion Transformer (DiT) operating within the latent space of a pre-trained VAE, following the Latent Diffusion Model (LDM) approach. This method significantly reduces the computational burden by having the Transformer predict noise ϵ_θ on a compressed image representation.

The final, stable configuration used for the results is detailed below:

Parameter	Value	Design Choice Justification
Model Type	Latent Diffusion Transformer	Selected for better scalability and efficiency in noise prediction within the latent space.
Input Image Size	32 X 32	Used due to training time constraints. This size resulted in a stable convergence but created a significant 4x4 latent resolution bottleneck.
VAE Downsampling	8	Standard factor for the AutoencoderKL VAE, resulting in a 4 x 4 latent grid input to the DiT.
Latent Resolution	4 x 4	The primary bottleneck is due to lack of sufficient compute resources. The visual blurriness is due to the 64 x compression 32 x 32 to 4 x 4.
`embed_dim` (Width)	256	A moderate width balancing capacity and training speed.
`depth` (Blocks)	4	A shallow model depth that proved stable and sufficient to converge rapidly on the small 4x4 latent space.

2. DiT Model Experiments and Visualization (Task 4)

2.1 Experimental Constraints and Scope

Due to time and compute constraints, the full sweep of architectural hyperparameter experiments (depth, heads, total timesteps T) was not feasible. Most results use the stable, converged architecture (depth=4, embed_dim=256, timesteps=1000) sampled with N=500 DDIM steps, unless stated otherwise.

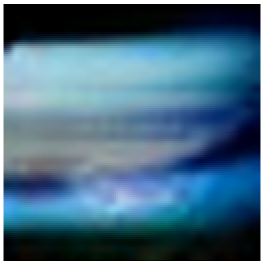
2.2 DDIM Reverse Process Visualization

To fulfill the visualization requirement, I have demonstrated the DDIM reverse diffusion by showing the model's intermediate x_0predicted (predicted clean image) at various timesteps.

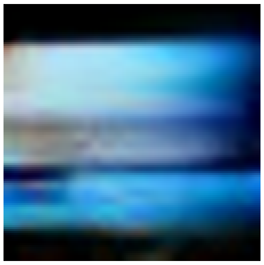
Sampling Configuration: The visualization was generated using the stable checkpoint (after 250 epochs of training) with fixed parameters: DDIM Steps=500, Guidance Scale=4.0.

t= 900

t = 500



t = 250



t = 0

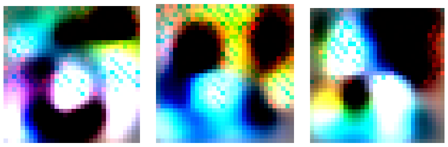
Observation: The visualization confirms the two stages of the diffusion process. At high timesteps (t=900), the prediction captures only low-frequency information (color and rough structure). The image structure solidifies rapidly around t=500, with the final steps (below t=250) dedicated to refining high-frequency details. The overall blurriness in the final image (t=0) is a constant artifact caused by the 4 x 4 latent bottleneck. Note that the image here has been scaled up to 128 x 128 for better visualization.

Ablation on Depth (number of blocks):

The depth of the DiT model was fixed at 4 blocks for the final results due to experimental constraints. Increasing the depth is conceptually known to enhance the model's ability to model complex, multi-scale feature dependencies.

An experimental attempt to train a Depth=6 model on the more complex 8×8 latent space failed to converge rapidly (Loss ≈0.85), yielding visually incoherent samples. This failure directly demonstrates the coupling between model complexity and depth: when the latent space complexity is increased (4x more information), the model requires a corresponding increase in depth (from 4 to ≥6) *and* a substantially longer training schedule to successfully converge and capture the required structural detail. For the stable 4×4 latent space, Depth=4 was sufficient to achieve convergence.

Examples of samples for w = 1.0 at depth = 6, img_size = 64x64:

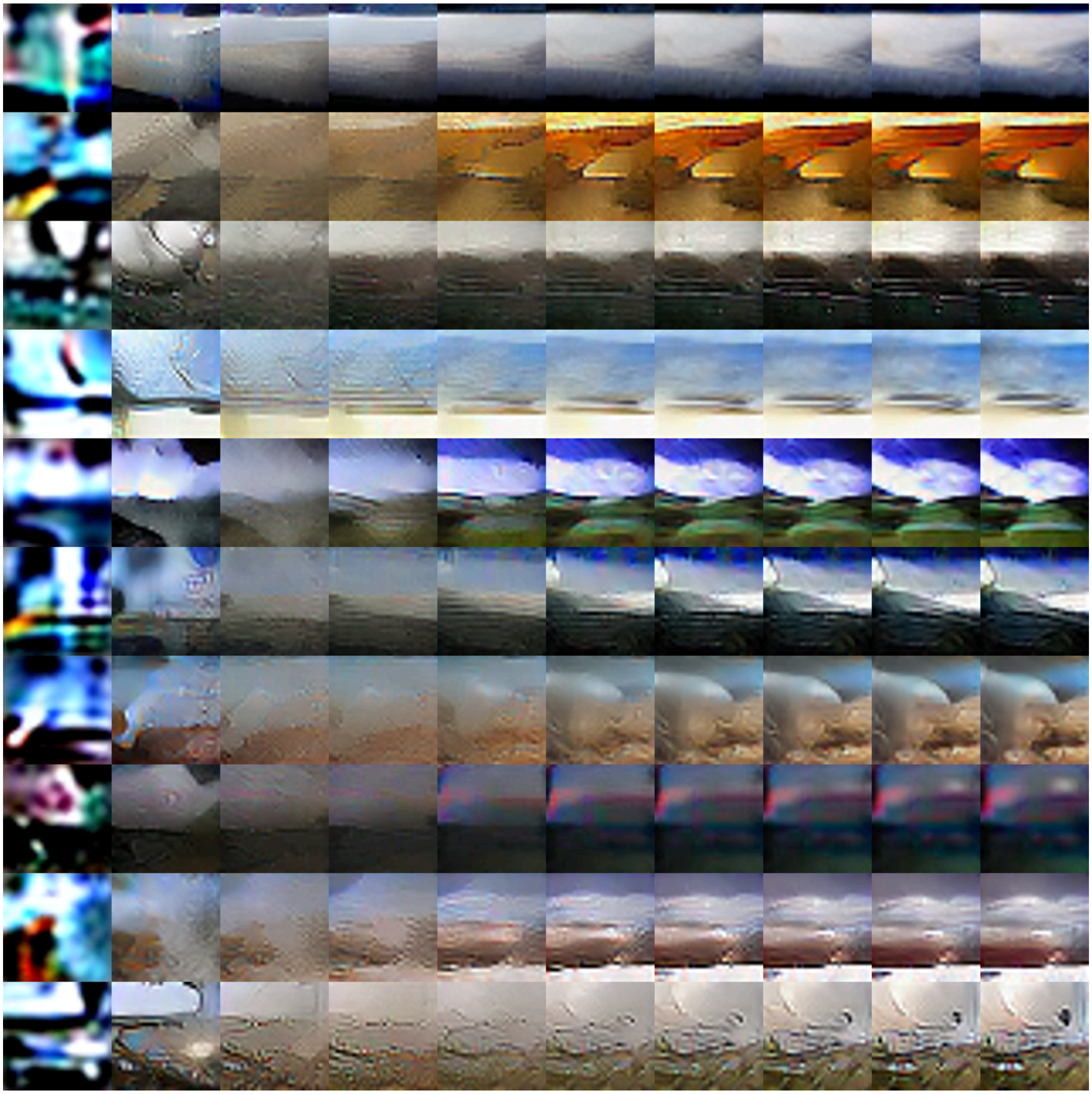


The Effect of Attention Heads:

The number of attention heads was fixed at 4 (num_heads=4) in the final model. This configuration, which results in a dimension of 64 per head (embed_dim/4), is a standard design choice known to offer the best balance between representational diversity and computational stability.Theoretically, a lower number of heads (=1) would force the model to focus too broadly, missing fine-grained correlations, while a higher number of heads (=8) would increase computational overhead and lower the effective dimension per head, which is challenging for a shallow, small model to stabilize within limited training epochs. The stability of the heads=4 was used to effectively use the limited time and compute resources I had.

Model prediction evolution

Visualization of the DiT model's **predicted clean image (x_0predicted)** has been presented below. Each row displays the reverse sampling evolution of a single, independent sample (n=0 to 9). Each column corresponds to a specific DDIM timestep (t=990 on the left, to t=0 on the right)



3. Classifier-Free Guidance (CFG) (Task 5)

3.1 CFG Implementation Logic

The CFG technique combines the noise prediction from the conditional model (ϵ_{cond}) *and the unconditional model* (ϵ_{uncond}) using the guidance scale (w):

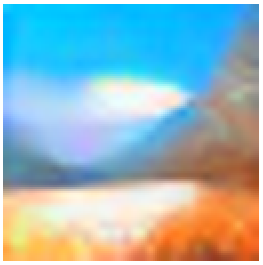
$$\epsilon_{guided} = \epsilon_{uncond} + w \, . \, (\epsilon_{cond} - \epsilon_{uncond})$$

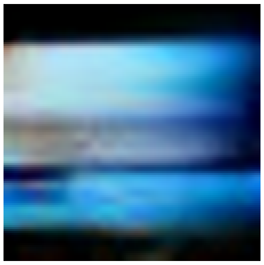
This is implemented by running the DiT twice on the same noisy latent (x_t): once with the target class label and once with a reserved unconditional index. The resulting noise predictions are linearly interpolated based on w to bias the final prediction toward the conditional result.

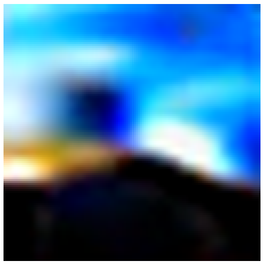
3.2 Sensitivity Analysis for Guidance Scale w

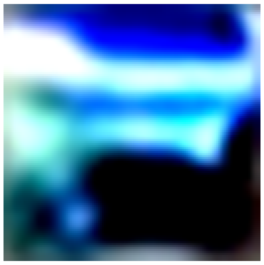
I performed a sensitivity analysis by varying w across a wide range to observe its effect on sample quality and diversity.

Sampling Command Configuration: The following samples were generated using DDIM Steps=500 and fixed Target Class (=1). Note that the dataset doesn’t contain class information but it has mentioned being sourced from 7 types of research sources. Hence, based on the suffixes of image names, I attributed classes for the sake of this experiment.


w = 1.0


w = 4.0


w = 7.0


w = 12.0

Guidance param	**w=1.0**	**w=4.0**	**w=7.0**	**w=12.0**
Qualitative Observation	Best structural quality, contains landscape artifacts.	Slight increase in color intensity, minor artifacts.	Increased bluriness, hard to make out artifacts.	Clear failure with severe distortion and color artifact amplification.
FID	188.42	218.43	265.59	308.25

* FID scores were computed against the original image dataset by generating 1000 samples with each guidance parameter

Observation: The highest quality, most structurally coherent, and natural-looking samples were produced at w=1.0 (unconditional generation). These images showed the lowest FID scores, thus indicating better performance than other values of w.
As w increased, the images exhibited greater distortion and unnatural features. This anomaly can be attributed to the overspecialization of the conditional model branch during the limited training time. The conditional branch focused too strongly on noisy high-frequency artifacts associated with the class(presumed from the file names at loading), and increasing w simply amplified these artifacts, degrading the overall visual quality learned by the stable unconditional branch. The increase in FID scores further fortifies the observations. This demonstrates the critical sensitivity of CFG to the convergence quality of the underlying conditional model.

4. Discussion of Significant Design Choices and Challenges

Summary of Challenges and Solutions

Challenge	**Interim solution used**	**Learnings**
LDM Convergence Failure	Fixed initialization to 4-channel latent input (in_channels=4).	Model successfully converged, validating the LDM architecture path.
Tiny, Blurry Images	Applied DDIM sampling (500 steps) and upscaling to 128 dim for visualization.	images are limited by the 4x4 latent space, which is the primary bottleneck.
Architectural Scaling Failure	Attempted to retrain on 64 x 64 which failed to converge even after 50 epochs (Loss ~0.85).	Confirmed that increasing latent resolution requires significantly greater training longevity and model capacity and due to compute limitations, was not feasible.