An Implementation Report on Diffusion Transformers: Guidance Effects, Attention Efficiency, and Advanced Evaluation Metrics

Aditya Nagarsekar

Abstract

Diffusion models represent the state-of-the-art in generative modeling, with Diffusion Transformers (DiTs) emerging as a scalable alternative to traditional U-Net architectures [16]. This report details our implementation and evaluation of key aspects of DiT models. First, we analyze the qualitative effects of Classifier-Free Guidance (CFG) scale and the number of sampling steps on image generation using a pre-trained DiT. Second, we address computational efficiency by integrating optimized attention mechanisms from the xformers library into our setup, quantifying the observed inference speedup. We further explore attention variants by training a DiT model employing Sliding Window Attention (SWA) from scratch on a landscape dataset, comparing its generative quality (FID) and sample characteristics against a baseline DiT with standard full attention trained under similar conditions. Finally, recognizing the limitations of standard metrics like FID, we implement and evaluate CLIP Mean Maximum Discrepancy (CMMD) [14] as an alternative evaluation protocol, using the squared MMD (MMD^2) statistic. We compare the calculated $CMMD^2$ scores with FID for the SWA-trained models and extend the analysis by incorporating SigLIP [22] and ALIGN [10] embeddings into the MMD^2 framework using readily available implementations. Our findings provide practical insights based on these experiments into DiT sampling behavior, demonstrate achievable efficiency gains, characterize the observed trade-offs of SWA in our setup, and offer a comparative analysis of contemporary evaluation metrics based on different vision-language model embeddings.

1 Introduction

Generative models capable of synthesizing high-fidelity data, particularly images, have witnessed remarkable progress, largely driven by the success of diffusion probabilistic models [9, 18]. These models learn to reverse a diffusion process that gradually corrupts data with noise, enabling the generation of complex samples starting from pure noise. Architectures like Stable Diffusion [15], built upon U-Net backbones [16], have demonstrated impressive capabilities but face inherent scalability challenges associated with CNNs.

The Diffusion Transformer (DiT) [13] was proposed to overcome these limitations by replacing the U-Net with a standard Transformer architecture [21] (which are easier to parallelise, increasing scalability). Operating on latent representations of images, DiT leverages the known scalability and success of Transformers in domains like computer vision [5], offering a promising path towards larger and potentially more powerful generative models. DiT's design demonstrated that Transformers can effectively learn the denoising function required by diffusion models and benefit from increased model size and compute.

Developing a practical understanding of using a Transformer backbone involves exploring several key aspects influencing both performance and usability. This report focuses on our exploration of:

(a) **Sampling Control:** Parameters governing the reverse diffusion process, such as Classifier-Free Guidance (CFG) scale [8] and the number of denoising steps, significantly impact

output quality, diversity, and adherence to conditioning signals. We investigated these effects qualitatively.

(b) **Computational Efficiency:** The standard self-attention mechanism in Transformers has a computational cost that scales quadratically ($O(N^2)$) with the sequence length N (number of image patches). This motivates exploring more efficient attention implementations. We specifically evaluated the integration of the xformers library [11] and implemented an architectural variant, Sliding Window Attention (SWA) [1], to measure their impact on inference speed and training/generation trade-offs in our setup.

(c) **Model Evaluation:** Evaluating the quality of generated samples remains challenging. While Fréchet Inception Distance (FID) [7] is widely used, its reliance on Inception features and underlying Gaussian assumption warrant investigating alternative metrics. We implemented and applied metrics leveraging powerful vision-language model embeddings (e.g., CLIP [14], SigLIP [22], ALIGN [10]) combined with robust statistical distances like Mean Maximum Discrepancy (MMD) [6] to provide potentially more semantically grounded evaluation alongside FID.

This report documents these explorations through a series of targeted implementations and experiments, structured as follows:

- **Task 1: Sampling Parameter Analysis.** We report a qualitative analysis of the effects of varying CFG scale and the number of sampling steps on image generation from a pre-trained DiT model (Section 5.1).
- **Task 2: Attention Efficiency Evaluation.** We first report the quantified inference speedup achieved by replacing the standard DiT attention block with an optimized implementation from the xformers library (Section 5.2.1). We then detail a comparative study involving training two DiT models (standard full attention vs. Sliding Window Attention) from scratch on a landscape dataset and evaluating their performance using FID and qualitative sample inspection (Section 5.2.2).
- **Task 3: Advanced Evaluation Metrics.** We describe the implementation and application of CLIP Mean Maximum Discrepancy (CMMD), using the MMD^2 statistic, as an alternative evaluation metric. We compare its results with FID for the models trained in Task 2b and detail the extension of this analysis using SigLIP and ALIGN embeddings within the MMD^2 framework (Section 5.3).

This report aims to share practical insights gained from these implementations and evaluations regarding DiT behavior, achievable efficiency optimization strategies, and the nuances observed in generative model evaluation using different metrics in the context of Transformer-based diffusion.

2 Related Work

This section briefly reviews prior work relevant to the techniques implemented and evaluated in this report, providing context for our choices and comparisons.

2.1 Diffusion Models

Diffusion models [18] formulate generation as learning the reverse of a process that gradually adds noise to data. Denoising Diffusion Probabilistic Models (DDPMs) [9] provided a simplified and effective training objective based on predicting the noise added at each step. Key developments include accelerated sampling algorithms like Denoising Diffusion Implicit Models (DDIM) [19], methods for conditioning the generation process [4], and Classifier-Free Guidance (CFG) [8],

which allows controlling the trade-off between sample quality and diversity without needing an explicit classifier. While highly successful models like GLIDE [12], Imagen [17], and Stable Diffusion [15] primarily utilize U-Net architectures [16], often within a latent space for efficiency (Latent Diffusion Models, LDM [15]), the scalability limitations of CNNs motivated alternatives like DiT.

2.2 Transformers in Vision and Generative Modeling

The Vision Transformer (ViT) [5] established Transformers as a powerful architecture for computer vision by treating images as sequences of patches. This success led to widespread adoption in various vision tasks. The Diffusion Transformer (DiT) [13] represents a key advancement, demonstrating that a standard Transformer architecture can effectively replace the U-Net backbone in an LDM framework. DiT showed strong performance and, crucially, exhibited favorable scaling properties with increased model size and compute, aligning with observations from Transformers in other domains. Our work investigates practical aspects of using this architecture.

2.3 Efficient Attention Mechanisms

The primary bottleneck in scaling Transformers is the $O(N^2)$ complexity of standard self-attention [21]. Numerous approaches aim to mitigate this. *Efficient implementation* methods focus on optimizing the standard attention calculation without changing the output, primarily by reducing memory access costs. Libraries like xformers [11], which we evaluated in Task 2a, provide such implementations, often leveraging techniques like FlashAttention [3], which uses tiling and recomputation in fast on-chip memory. *Efficient attention variants* modify the attention mechanism itself to reduce complexity. Examples include local attention mechanisms like Sliding Window Attention (SWA) [1], which we implemented and compared in Task 2b. SWA computes attention only within a fixed-size local window around each token, achieving linear complexity $O(N \cdot W)$ at the cost of a limited receptive field.

2.4 Generative Model Evaluation

Evaluating generative models remains an open challenge. Fréchet Inception Distance (FID) [7], comparing statistics of InceptionV3 [20] features between real and generated samples, is the most common metric but has known limitations: reliance on classification-biased features and a potentially inaccurate Gaussian assumption [2]. Kernel-based distances like Maximum Mean Discrepancy (MMD) [6] offer a non-parametric alternative. Recently, leveraging embeddings from vision-language models like CLIP [14], SigLIP [22], and ALIGN [10] has gained popularity. These embeddings capture rich semantic information. Using them within an MMD framework (as implemented in Task 3) aims to provide a potentially more semantically meaningful comparison of distributions than FID, without relying on the Gaussian assumption.

3 Methodology

This section details the conceptual underpinnings and specific choices made for the models, techniques, and evaluation metrics implemented and employed in this study.

3.1 Diffusion Transformer (DiT) Architecture

Our work utilized the DiT model [13], which replaces the convolutional U-Net backbone common in many diffusion models with a Transformer architecture, operating within a Latent Diffusion

Model (LDM) framework [15]. The core components, as implemented in our setup based on [13], include:

1. **Latent Space Operation:** Input images are first encoded into a lower-dimensional latent representation z_0 using a pre-trained Variational Autoencoder (VAE) encoder. The diffusion process operates entirely within this latent space, significantly reducing computational cost. The VAE decoder is used only at the end to transform the final denoised latent z_0 back into an image. VAE weights were kept frozen during our DiT training.

2. **Patchification:** The noisy latent z_t at timestep t is divided into a grid of non-overlapping patches (e.g., 2×2 or 4×4 in the latent space).

3. **Token Embedding:** Each patch is linearly projected into a vector (token) of dimension D, the Transformer's hidden dimension. Standard learnable positional embeddings are added to provide spatial information, analogous to ViT [5].

4. **Transformer Backbone:** The sequence of tokens is processed by a stack of standard Transformer blocks, each containing multi-head self-attention followed by an MLP, with layer normalization and residual connections.

5. **Conditioning Integration:** Information about the noise level (timestep t) and any additional conditions (class labels c) are incorporated using adaptive conditioning mechanisms. Timestep t is converted to emb_t, and class label c to emb_c, which are combined. Our implementation utilized the adaLN-Zero technique [13]: the combined conditioning embedding modulates the Transformer blocks by predicting scale (γ) and shift (β) parameters for Layer Normalization ($\gamma \cdot \text{LayerNorm}(x) + \beta$) and predicting 'gate' values to adjust signal flow from attention/MLP pathways. Initializing the final projection layer of the modulation network to zero aids training stability.

6. **Noise Prediction:** The output token sequence from the Transformer blocks is projected back to predict the noise ϵ originally added to z_0 to obtain z_t. Our DiT was trained with a mean squared error objective to minimize the difference between the predicted noise $\epsilon_\theta(z_t, t, c)$ and the actual added noise.

3.2 Sampling Techniques

3.2.1 Reverse Diffusion Process and Sampling Steps

Image generation in our experiments involved reversing the diffusion process. Starting from pure Gaussian noise z_T, the model iteratively denoises it over T' steps. In each step t (from T' down to 1), the implemented DiT predicts the noise $\epsilon_\theta(z_t, t, c)$. We used a standard DDPM sampler [9], chosen for its common use and stability, although faster samplers like DDIM [19] exist, to estimate the less noisy latent z_{t-1} based on the noise prediction. The number of steps T' directly affects generation time and quality in our setup; fewer steps are faster but can yield lower fidelity if noise removal is incomplete.

3.2.2 Classifier-Free Guidance (CFG)

To control conditional generation, we implemented CFG [8]. This leverages the diffusion model's ability to generate conditionally ($\epsilon_\theta(z_t, t, c)$) and unconditionally ($\epsilon_\theta(z_t, t, \emptyset)$). During sampling, the effective noise prediction $\hat{\epsilon}$ is calculated using Equation 1:

$$\hat{\epsilon}_\theta(z_t, t, c) = \epsilon_\theta(z_t, t, \emptyset) + w \cdot (\epsilon_\theta(z_t, t, c) - \epsilon_\theta(z_t, t, \emptyset)) \tag{1}$$

The guidance scale w controls this extrapolation. We explored different values of w in Task 1.

3.3 Attention Mechanisms: Conceptual Approaches Implemented

3.3.1 Standard Multi-Head Self-Attention

Our baseline DiT model utilized standard multi-head self-attention, computing scores between all pairs of input tokens. While enabling the capture of global dependencies, its quadratic complexity ($O(N^2)$) presents challenges for large inputs.

3.3.2 xformers Memory Efficient Attention

For Task 2a, we integrated the 'xformers' library [11] to leverage its *implementation optimization*. Specifically, we used its memory-efficient attention mechanism, which computes the exact same output as standard attention but employs techniques like FlashAttention [3] to reduce memory bandwidth and improve execution speed by optimizing computation on-chip (SRAM) without materializing the full attention matrix in HBM.

3.3.3 Sliding Window Attention (SWA)

For Task 2b, we implemented SWA, an *architectural modification*, based on [1]. In our implementation, each token only attends to a fixed number ($W = 9$) of neighboring tokens within a sliding window. This reduces complexity to linear ($O(N \cdot W)$) but means information propagates globally only through multiple layers, potentially impacting tasks requiring fine-grained global coherence. We compared this implementation against the baseline.

3.4 Evaluation Metrics: Conceptual Basis and Implementation

To assess the performance of our trained models, we implemented and utilized the following metrics:

3.4.1 Fréchet Inception Distance (FID)

We calculated FID [7] using a standard implementation (specifically, the 'pytorch-fid' library). This involved:

- **Feature Extraction:** Passing batches of real (X_r) and generated (X_g) images through a pre-trained InceptionV3 network to extract activations from the final pooling layer.
- **Gaussian Assumption:** Calculating the sample mean (μ_r, μ_g) and covariance matrix (Σ_r, Σ_g) for these activation sets, assuming they follow multivariate Gaussian distributions.
- **Distance Calculation:** Computing the Fréchet distance using Equation 2:

$$d^2((\mu_r, \Sigma_r), (\mu_g, \Sigma_g)) = ||\mu_r - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \tag{2}$$

  Lower FID suggests better fidelity and diversity. We acknowledge its known limitations (ImageNet bias, Gaussian assumption).

3.4.2 Mean Maximum Discrepancy (MMD) and CMMD

As a non-parametric alternative, we implemented calculations for MMD [6], specifically CMMD using embeddings from vision-language models (CLIP [14], SigLIP [22], ALIGN [10]). Our calculation process involved:

- **Feature Extraction:** Processing real ($\sim P_X$) and generated ($\sim P_Y$) images using the image encoders of the selected pre-trained vision-language models (obtained via the 'transformers' library) to yield embedding sets $X = \{x_i\}_{i=1}^n$ and $Y = \{y_k\}_{k=1}^m$.

- **Normalization:** Utilizing the inherent L2 normalization ($||x_i||_2 = 1, ||y_k||_2 = 1$) provided by these models' standard outputs, focusing the comparison on semantic direction.
- **Kernel Function:** Employing the standard RBF kernel (Equation 3):

$$K(x, y) = \exp(-||x - y||_2^2/(2\sigma^2)) \tag{3}$$

- **Bandwidth Selection:** Adaptively estimating the squared bandwidth σ^2 using the median heuristic (Equation 4):

$$\sigma^2 = \texttt{median}(\{||a - b||_2^2 | \dots\})/\log(n + m) \tag{4}$$

  calculated over the aggregated set $Z = X \cup Y$.
- **MMD^2 Calculation:** Computing the squared MMD using the *unbiased U-statistic estimator*(MMD$_u^2$, Equation 5):

$$\text{MMD}_u^2(X, Y) = \frac{1}{n(n-1)} \sum_{i \neq j}^{n} K(x_i, x_j) + \frac{1}{m(m-1)} \sum_{k \neq l}^{m} K(y_k, y_l) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{k=1}^{m} K(x_i, y_k) \tag{5}$$

  Our implementation calculated full kernel matrices (K_{XX}, K_{YY}, K_{XY}) and adjusted sums to exclude diagonal elements.
- **Advantages over FID:** This implemented metric avoids FID's Gaussian assumption and leverages potentially richer semantic features. We note its interpretability depends on the chosen feature extractor and kernel parameters.

4 Experimental Setup

This section outlines the specific configurations, datasets, and procedures used in our implementation and evaluation tasks.

4.1 Models and Codebase

- **DiT Architecture:** Our experiments used DiT models conceptually similar to [13], implemented using Python scripts (primarily leveraging PyTorch). Key parameters are available in the scripts. Note that Task 2a (xformers) used a DiT-XL/2 configuration, while Task 2b (SWA) used DiT-S/4 due to computational constraints for training from scratch.
- **Attention Mechanisms:** Baseline models used standard PyTorch multi-head self-attention. The xformers variant explicitly called `xformers.ops.memory_efficient_attention`. Our SWA implementation involved modifying the standard attention block logic to enforce the window constraint ($W = 9$).
- **Embedding Models:** CMMD calculations used image encoders accessed via the Hugging Face `transformers` library (versions TBD). Specific models included CLIP (`openai/clip-vit-base-patch3`), SigLIP (`google/siglip-base-patch16-224`), and an ALIGN implementation available on Hugging Face [https://huggingface.co/docs/transformers/en/model_doc/align].

4.2 Datasets

- **Task 1 (CFG/Steps Analysis):** We used a pre-trained ImageNet-conditional DiT model, generating samples for class [281: tabby cat].

- **Tasks 2b & 3 (SWA Training & Evaluation):** We used a publicly available landscape dataset [https://www.kaggle.com/datasets/arnaud58/landscape-pictures/data] for unconditional training and evaluation. Real images for metric calculation were drawn from the dataset's test split.

4.3 Procedures

- **Task 1 (CFG/Steps):** Samples were generated using a standard DDPM diffusion sampler, varying either the CFG scale w ($w = 0$ vs. $w = 10$) while keeping steps constant, or varying the number of steps (e.g., 50, 250, 500) while keeping $w = 7$ constant. A qualitative assessment of the output images was performed.

- **Task 2a (xformers):** Inference time was measured for the DiT-XL/2 model generating 50 images (batch size 50) over 50 dummy diffusion steps. We used PyTorch's 'torch.cuda.Event' for accurate GPU timing after a warmup run. The comparison was between the baseline DiT attention and our implementation using xformers.

- **Task 2b (SWA Training):** Two DiT-S/4 models (baseline full attention vs. our SWA implementation with $W = 9$) were trained from scratch unconditionally on the landscape dataset. We employed standard training procedures: AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$), and a learning rate schedule with linear warmup to 1×10^{-4} followed by cosine annealing decay to 1×10^{-6} (using PyTorch's `SequentialLR`, `LinearLR`, `CosineAnnealingLR`). Total training steps depended on configuration (`warmup_steps`, dataset size, batch size, epochs = 120). Performance was evaluated using FID (via 'pytorch-fid') and qualitative inspection of generated samples.

- **Task 3 (CMMD Evaluation):** We calculated the CMMD metric to compare the distributions of real images and images generated by the models from Task 2b.

  1. **Feature Extraction:** Extracted image embeddings using pre-trained CLIP, SigLIP, and ALIGN encoders from the `transformers` library.
  2. **Normalization:** Utilized the inherent L2 normalized outputs from these models.
  3. **MMD^2 Calculation:** Calculated the unbiased squared MMD statistic (MMD_u^2) using our implementation. This involved defining the RBF kernel, selecting σ^2 via the median heuristic, and applying Equation 5 using calculated kernel matrices.
  4. **Comparison:** Compared the resulting CMMD scores across models (baseline vs. SWA) and embedding types (CLIP, SigLIP, ALIGN), and contrasted them with the FID scores from Task 2b.

5 Results

This section presents the findings observed from the described experiments.

5.1 Task 1: Effects of CFG and Sampling Steps

5.1.1 Qualitative Effects of CFG Scale

Visual inspection of generated samples showed the expected impact of the CFG scale w:

- **Zero Guidance ($w = 0$):** Samples displayed high diversity but lacked clear identity corresponding to the target class 281: tabby cat, appearing generic or abstract (Figure 1a). This observation confirms that CFG is necessary to steer generation towards the condition in this setup.

- **High Guidance ($w = 10$):** Samples strongly adhered to the target class, exhibiting high fidelity (Figure 1b). However, artifacts or feature exaggeration, characteristic of strong guidance, were also observed.

(a) CFG = 0.0

(b) CFG = 10

Figure 1: Illustration of the effect of varying CFG scale (w) on samples for class 281: tabby cat using a pre-trained DiT model.

5.1.2 Qualitative Effects of Sampling Steps

The number of denoising steps affected generation time and visual quality in our tests:

- **Few Steps (50):** Resulted in the fastest generation, but the image appears slightly uncanny and retains some noise (as expected from fewer steps) (Figure 2a).
- **Moderate Steps (250):** Offered a compromise, yielding an image with reasonable detail, although subjectively appearing somewhat sketch-like compared to the 50-step version in this instance (Figure 2b).
- **Many Steps (500):** Showed visual improvement over moderate steps, primarily in fine detail refinement and realism, at the cost of increased computation time (Figure 2c).

5.2 Task 2: Attention Efficiency and Variants

5.2.1 Task 2a: xformers Integration Speedup

Integrating xformers' optimized attention resulted in a measurable inference speedup in our test. Generating 50 images took 8.0981 seconds with xformers compared to 9.7122 seconds with the baseline attention, resulting in a 1.20× speedup (Table 1). This demonstrates the practical benefit obtainable from using such optimized implementations for standard attention calculations in DiT inference.

5.2.2 Task 2b: Sliding Window Attention (SWA) Training Comparison

Comparing the DiT trained with full attention against the one we trained with SWA (Window $W = 9$) on the landscape dataset showed performance differences in our setup.

(a) Few Steps (50) (b) Moderate Steps (250) (c) Many Steps (500)

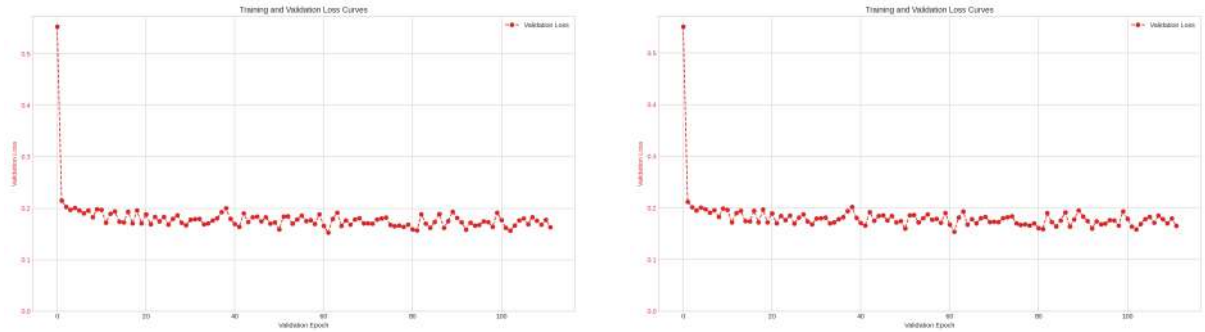
Figure 2: Illustration of the effect of varying numbers of sampling steps on samples for class 281: tabby cat using a pre-trained DiT model.

Table 1: Inference time comparison for generating 50 images using the benchmark DiT with standard vs. xformers attention. Lower time is better.

Attention Implementation	Total Time (s)	Time per Image (s)	Speedup Factor
Baseline DiT Attention	9.7122	0.1942	1.00×
xformers Attention	8.0981	0.1620	1.20×

Quantitative Results (FID): The baseline Full Attention model achieved FID = 230.7231, whereas our SWA model scored FID = 248.3587 (Table 2). The higher FID for SWA indicates a larger measured discrepancy between its generated samples and real samples in the Inception feature space in this experiment.

Training Dynamics: Validation loss curves for both models exhibited stable training dynamics, following similar downward trends until around the 70th epoch, after which a plateau was observed (Figure 3). This trend, coupled with observed noise/artifacts in sample images, suggests potential underfitting within the 120 epochs of training performed (true epochs trained is 112 due to early stopping patience of 50 epochs). The two trajectories are nearly identical, with the full attention loss curve showing a slightly lower final loss than the SWA curve.



(a) Validation Loss Curve for Full Attention (b) Validation Loss Curve for SWA

Figure 3: Validation loss curves comparing the baseline Full Attention (left) and DiT-SWA (right) models during training.

Qualitative Samples: Visually, both models produced partially recognizable landscapes (Figure 4). However, consistent with the FID difference observed, the baseline Full Attention samples exhibit slightly superior global coherence. The SWA samples show more visual artifacts, potentially related to the limited attention window.

Efficiency Considerations: The Full Attention model took 14 447 seconds for training in our setup, compared to 14 373 seconds for the SWA model. This indicates that for this specific model size (DiT-S/4) and task, the training time difference was marginally in favour of SWA over 120 epochs.

Table 2: FID scores (lower is better) for DiT models trained from scratch on the landscape dataset.

Model Configuration	FID
Baseline DiT (Full Attention)	230.7231
DiT-SWA (SWA, W=9)	248.3587

5.3 Task 3: CMMD Evaluation

5.3.1 CMMD (CLIP) vs. FID

Our evaluation using CMMD^2 with CLIP embeddings yielded results consistent with FID for the landscape-trained models (Table 3). The baseline model achieved a lower (better) CMMD^2(CLIP) score (0.3600) than the SWA model (0.4148), mirroring the FID ranking. This suggests that for this specific comparison, the distributional differences captured by FID were also reflected in the CLIP embedding space as measured by our MMD^2 implementation.

Table 3: Comparison of FID and CMMD^2 (CLIP ViT-B/32) scores (lower is better) for the landscape-trained DiT models.

Model Configuration	FID	CMMD^2 (CLIP)
Baseline DiT (Full Attention)	230.7231	0.3600
DiT-SWA (SWA, W=9)	248.3587	0.4148

5.3.2 CMMD with SigLIP and ALIGN Embeddings

Extending the evaluation to SigLIP and ALIGN embeddings also showed consistent ranking (Table 4). All MMD^2-based metrics we calculated identified the baseline (Full Attention) model as superior (lower score) to the SWA model in this experiment. However, the absolute scores varied significantly depending on the embedding space used:

- SigLIP MMD^2 scores (Baseline 0.3442, SWA 0.4792) were numerically similar to the CLIP MMD^2 scores we obtained.
- ALIGN MMD^2 scores (Baseline 0.2471, SWA 0.3368) were notably lower overall than CLIP or SigLIP scores in our calculations.

This variation observed in our results highlights that while these metrics agreed on the ranking in this case, the perceived distance between distributions is highly dependent on the properties of the chosen embedding space, influenced by the models' training data and objectives.

(a) Baseline (Full Attention)
Best Validation Loss

(b) SWA (W=9)
Best Validation Loss

(c) Baseline (Full Attention)
Final Epoch

(d) SWA (W=9)
Final Epoch

Figure 4: Comparison of generated landscape samples from Baseline (Full Attention) and SWA (W=9) models. Top row shows samples from the epoch with the best validation loss; bottom row shows samples from the final training epoch.

Table 4: Comparison of Evaluation Metrics using different embedding backbones (lower is better) for the landscape-trained DiT models.

Model Configuration	FID	CMMD^2	SigLIP MMD^2	ALIGN MMD^2
Baseline DiT (Full Attention)	230.7231	0.3600	0.3442	0.2471
DiT-SWA (SWA Attention, W=9)	248.3587	0.4148	0.4792	0.3368

6 Analysis and Discussion

This section interprets the results observed in our experiments, discussing their practical implications and connections to existing knowledge.

6.1 Sampling Parameters (CFG and Steps)

The qualitative experiments (Section 5.1) were consistent with the well-established roles of CFG and sampling steps. CFG (Equation 1) acts as a control knob trading diversity for fidelity to the condition, as observed in our generated samples. Zero guidance removed conditional influence, while high guidance strongly enforced it, sometimes introducing artifacts [8]. The number of sampling steps controlled the trade-off between computational cost and sample quality in our tests. Fewer steps were faster but risked incomplete noise removal, while more steps improved fidelity, aligning with expectations [19]. Optimal settings in practice would depend on the specific application, model configuration, and desired balance.

6.2 Attention Efficiency (xformers and SWA)

Our experiments evaluated two distinct approaches to improving attention efficiency in DiT:

- **Optimized Implementation (xformers):** The measured 1.20× inference speedup (Section 5.2.1) demonstrates the practical value achievable by using specialized libraries like xformers. These libraries offer kernel-level optimizations (e.g., FlashAttention [3]) that reduce memory bandwidth usage for standard attention without altering mathematical behavior, providing a relatively straightforward performance gain for inference, as observed in our test.
- **Architectural Modification (SWA):** Our implementation of SWA traded expressive power for efficiency by limiting the attention scope (Section 5.2.2). Its linear complexity ($O(N \cdot W)$) holds advantages for scaling, although the training time difference was marginal in our specific DiT-S/4 experiment over 120 epochs. However, the observed increase in FID (17.6 points) and increased visual artifacts suggest that the reduced receptive field hindered performance on this landscape generation task, possibly because capturing long-range dependencies is beneficial. This illustrates the practical trade-off: SWA can be computationally cheaper (especially for larger N or W), but may yield lower quality compared to full attention if global context is critical, as observed in our limited training run. Suitability depends on the task and resource constraints.

6.3 Evaluation Metrics (FID vs. CMMD Variants)

The evaluation using our CMMD^2 implementation (Section 5.3) provided valuable context for interpreting generative model metrics alongside the traditional FID in our specific comparison.

FID Assumptions and CMMD Implementation Benefits: FID relies on InceptionV3 features and a Gaussian assumption [7]. Our CMMD implementation aimed to mitigate these by leveraging embeddings from vision-language models (CLIP, SigLIP, ALIGN) designed to capture richer semantic content, and by using the non-parametric MMD^2 statistic which avoids the Gaussian assumption [6].

Consistent Ranking Across Metrics (in this study): In our comparison between the baseline (Full Attention) and SWA models, a notable consistency emerged. FID and all tested CMMD^2 variants (using CLIP, SigLIP, ALIGN) ranked the baseline model as superior (lower score). This concordance suggests that, for this specific task and the extent of training performed,

the performance difference manifested as a distributional shift detectable across multiple feature representations.

Embedding Space Sensitivity Observed: Crucially, while the *ranking* remained consistent in our tests, the *absolute* CMMD^2 scores and the *magnitude* of the perceived difference were highly sensitive to the choice of the underlying embedding model. Our results showed distinct numerical ranges for MMD^2 depending on whether CLIP, SigLIP, or ALIGN embeddings were used, with ALIGN yielding lower scores overall. This observation aligns with the understanding that embedding space geometry, shaped by training data and objectives (e.g., ALIGN's noisy web data [10]), influences MMD^2 measurements. The relative gap between baseline and SWA MMD^2 scores also varied (appearing larger with SigLIP). This underscores that while CMMD offers advantages, interpreting specific scores requires careful consideration of the embedding model used. No single embedding space appears universally optimal based on these results; therefore, leveraging multiple CMMD variants alongside metrics like FID likely provides a more comprehensive and nuanced evaluation, as demonstrated in our report.

6.4 Limitations and Future Work

The scope of this implementation report is subject to several limitations:

- The CFG/steps analysis was qualitative and limited to one class.
- The SWA comparison involved one dataset, only one window size ($W = 9$), and was trained for only 120 epochs, which may be insufficient for full convergence of models like DiT. Performance could vary significantly with W, dataset characteristics, and longer training durations.
- The CMMD analysis used a limited, readily available set of embedding models.

Possible directions for extending this work include:

- Performing a quantitative analysis of CFG and Sampling Steps effects.
- Investigating further optimizations beyond the standard xformers library integration, as the observed 1.20× boost, while practical, might be improvable.
- Extending the SWA comparison to more datasets (especially those requiring strong global coherence), testing different window sizes (W), and training models for significantly more epochs to assess convergence and final performance more thoroughly.
- Incorporating additional embedding models (e.g., newer models, different architectures) into the CMMD evaluation framework for broader comparison.

7 Conclusion

This report detailed our implementation and investigation of the sampling behavior, attention efficiency, and evaluation of Diffusion Transformer (DiT) models. We confirmed the standard qualitative effects of Classifier-Free Guidance (CFG) scale and sampling steps on the generation process through direct observation. We demonstrated a practical inference speedup (1.20×) achievable by leveraging the optimized attention implementation from xformers in our setup, highlighting the value of such efficient kernels. Our comparative training of standard DiT versus our DiT-SWA implementation indicated that while SWA offers computational advantages (linear complexity), it incurred a measurable performance penalty (FID increase of 17.6, more artifacts)

on the landscape generation task within our limited training regime, suggesting its restricted receptive field may be detrimental when global context is important.

Furthermore, we implemented and analyzed CLIP Mean Maximum Discrepancy (CMMD), using the MMD^2 statistic with various vision-language embeddings (CLIP, SigLIP, ALIGN). In our tests, these metrics provided rankings consistent with FID for the models studied, but also demonstrated significant sensitivity of the absolute scores to the choice of embedding backbone. This underscores the value of CMMD variants as complementary evaluation tools avoiding FID's Gaussian assumption, while also cautioning that score interpretation requires considering the properties of the chosen embedding model. Overall, this report documents our exploration of key aspects of DiTs, confirming sampling behavior in our setup, quantifying the practical speed benefits of xformers attention, and evaluating the observed quality versus efficiency trade-off of Sliding Window Attention via direct comparison. The implementation and analysis of CMMD alongside FID highlighted both the potential consistency and the critical embedding-dependent sensitivity of advanced evaluation metrics, advocating for their use as part of a comprehensive assessment toolkit, interpreted with appropriate context.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. Published in ACL 2020.

[2] Ali Borji. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019. doi: arXiv:1802.03446.

[3] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness, 2022. Published in NeurIPS 2022.

[4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. Published in NeurIPS 2021.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. Published in ICLR 2021.

[6] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. ISSN 1532-4435. URL https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 6626–6637. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[8] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. Presented at NeurIPS 2022 Workshop on Score-Based Methods.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. Published in NeurIPS 2020.

[10] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. URL https://proceedings.mlr.press/v139/jia21b.html. Published in ICML 2021.

[11] Benjamin Lefaudeux, Francisco Massa, Diana Marculescu, Jérémy Rapin, Sam Shleifer, Daniel Haziza, Patrick Labatut, Wenhan Xiong, Vincent Quenneville-Bélair, Susan Zhang, Naman Goyal, Quentin Duval, Natalia Gimelshein, Joe Spisak, and Piotr Dollar. xformers: A modular and configurable library for transformers. https://github.com/facebookresearch/xformers, 2022. Software available from GitHub.

[12] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models, 2021.

[13] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. Published in ICCV 2023.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. Published in ICML 2021.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. Published in CVPR 2022.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[17] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. Published in NeurIPS 2022.

[18] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. Published in ICML 2015.

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2020. Published in ICLR 2021.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc., 2017.

[22] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. Published in ICCV 2023.