Text-to-Image Generation using CLIP and Diffusion Models - SSY340 - Project Group 24

Daniel González Muela

Sotiris Koutsoftas

***Abstract*—This project explores the generation of high-quality images using diffusion models, while building a comprehensive understanding of the processes involved in text-to-image synthesis. The work is structured into two main parts. The first part focuses on the foundational steps of diffusion models, specifically Denoising Diffusion Probabilistic Models (DDPM) and Denoising Diffusion Implicit Models (DDIM), highlighting their application in generating high-quality outputs and experimenting with scheduler variations for efficiency improvements. Initial experiments include fine-tuning models on diverse datasets, such as celebrity faces [6] and butterflies [7]. In addition to that, the first part proceeds with a DDPM model trained on a subset of bedrooms from the WikiArt dataset [8], demonstrating the capabilities of these models in adapting to different image domains. The second part extends these foundations by exploring advanced guidance techniques and bigger pre-trained diffusion models, while incorporating CLIP and BERT-based conditioning, to enhance the fidelity and alignment of generated images to specific prompts. This approach provides insights into the interplay between text and image embeddings and their impact on generative quality. The report concludes with a comparison of guidance mechanisms and their effectiveness in creating diverse, visually coherent outputs.**

I. Part 1: Diffusion Models and Initial Experiments [3]

A. Traditional Generative Models

Traditional generative models, such as Variational Autoencoders (VAEs) and autoregressive models, have been widely used to model data distributions. However, they each come with inherent challenges.

VAEs operate by encoding data into a lower-dimensional latent space and then decoding it back, with a probabilistic framework that allows for the generation of new samples. While VAEs offer stability in training, they often struggle with generating sharp, high-quality images due to the smoothing effect of their probabilistic sampling.

Autoregressive models generate data sequentially, predicting each element conditioned on the previous ones. These models can capture complex dependencies and generate high-quality samples but are often limited by slow sampling speeds due to their step-by-step nature.

Diffusion models address some of these challenges by providing a more stable training process without adversarial components. They can generate high-quality images by leveraging the understanding of data transformations through noise, making them a promising alternative to traditional methods.

B. Diffusion Models Overview

Diffusion models have gained attention for their ability to generate high-quality samples by learning to reverse a noising process. At the core of these models are two primary processes: forward diffusion and reverse diffusion.

The **forward diffusion process** involves gradually adding noise to data, effectively transforming a structured input into pure noise. This transformation allows the model to learn how data deteriorates under noise, facilitating its understanding of how to reverse this process.

Conversely, the **reverse diffusion process** works to recover the original data from the noisy distribution. By learning how to denoise or revert the noising process, the model can generate realistic samples starting from random noise.

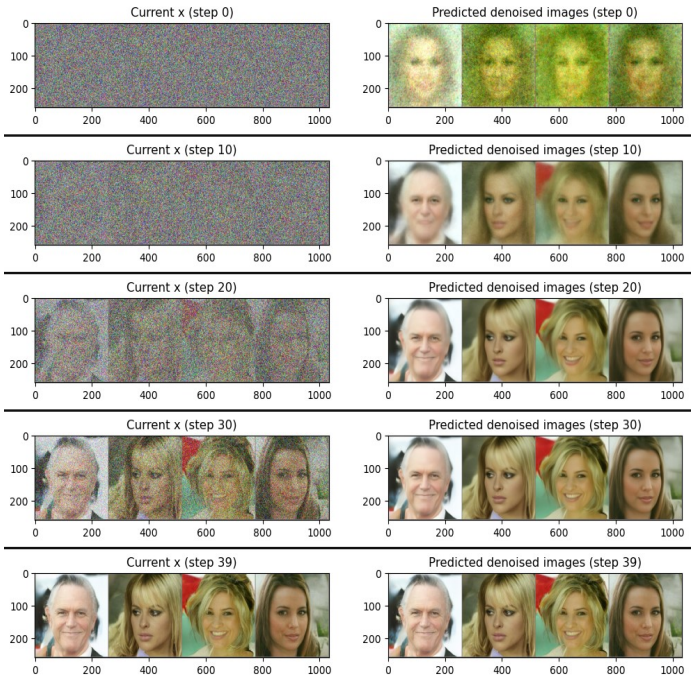


Fig. 1: Denoising process using the DDIM Scheduler over multiple steps. The figure shows the transformation from noisy input (left) to the denoised image (right) as steps progress.

C. Denoising Diffusion Probabilistic Models (DDPM) and Denoising Diffusion Implicit Models (DDIM)

Denoising Diffusion Probabilistic Models (DDPM) is one of the simplest types of diffusion models, widely recognized for its effectiveness in generating high-quality images. DDPM

introduces a forward process where noise is added to images until they become random noise. The model is trained to reverse this process, progressively denoising random noise step-by-step.

In contrast, Denoising Diffusion Implicit Models (DDIM) offers an approach that allows for the generation of images with similar quality to DDPM but with fewer iterations. DDIM estimates what the final image would look like based on fewer but larger steps, enhancing efficiency.

D. Initial Model Implementation

Setting Up the Diffusion Framework: We begin our project by importing a pretrained DDPM model on a face dataset of high-quality celebrity images. While it generates very high-quality images, it takes around 2 minutes to generate a single image using DDPM due to its high number of iterations. Implementing a DDIM Scheduler substantially improved the generation time to around 20 seconds, as DDIM only requires 40 steps to achieve comparable quality.

DDDPM

DDIM

E. Fine-Tuning on Different Datasets

Fine-tuning a model with new training data begins with a pre-trained model, which provides the advantage of leveraging weights that already represent useful features, rather than starting from scratch. We fine-tuned the model using a butterfly dataset [7] and observed the evolution of the generated images over several epochs. Initially, the images appeared noisy and unrecognizable, but as training progressed, distinctive butterfly patterns started to emerge.

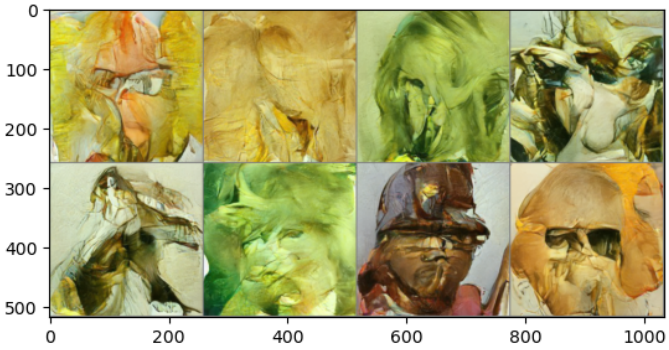


Fig. 2: Butterflies after 1 Epoch

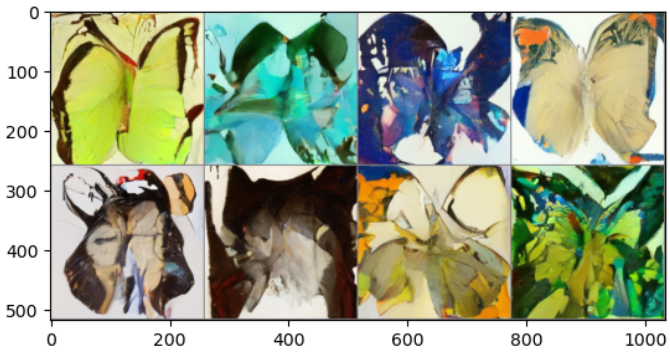


Fig. 3: Butterflies after 10 Epochs

To optimize training, we adjusted various hyperparameters, including noise levels, gradient accumulation, learning rate, and batch size.

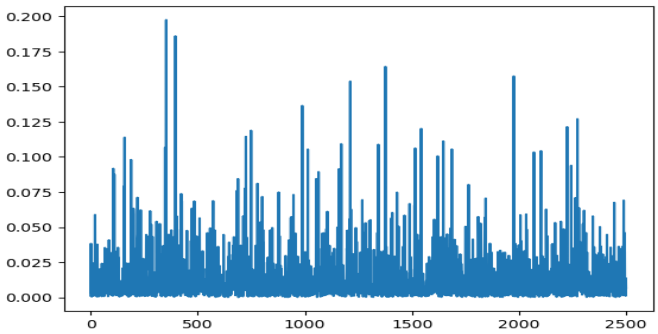


Fig. 4: Loss Curve During Fine-tuning

F. Guidance Techniques

To explore guidance techniques, we applied them to a more diverse dataset, specifically using a DDPM model trained on a subset of bedrooms from the WikiArt dataset. By defining a loss function that compares the pixels of an image to a target color, we guided the diffusion process to produce images closer to the specified condition.

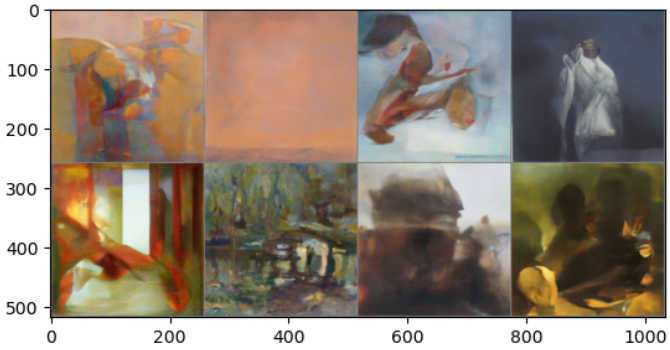


Fig. 5: Original Room Image

Fig. 6: Guided Room Image with Green Color Condition

G. Zero-shot Guidance with CLIP [10]

CLIP is a model pre-trained on the LAION-5B dataset [9] designed to align text and image representations [4] in a shared latent space, making it capable to align text with diverse images. We leverage this by conditioning our diffusion models using CLIP's text embeddings as a guidance mechanism to generate images that align with the input text prompts. For instance, conditioning a fine-tuned model trained on WikiArt-bedrooms [8] dataset using a CLIP loss with the prompt "Red Rose (still life), red flower painting", effectively results in generation of a realistic representation aligned with the input text.

Fig. 7: Image Generated with CLIP Guidance using the prompt: "Red Rose (still life), red flower painting"

Fig. 8: CLIP Guidance applied to the WikiArt Dataset

Fig. 9: CLIP Guidance applied to the Butterfly Dataset

II. Part 2: More Advanced Diffusion Techniques and Guided Image Generation [2]

A. Introduction

This section focuses on the implementation and detailed analysis of the Stable Diffusion (SD) model using the CompVis/stable-diffusion-v1-4 configuration [5]. The SD model combines a Variational Autoencoder (VAE) and latent diffusion mechanisms, which allows it to generate images efficiently by compressing the high-dimensional image data into a smaller latent space. The first step involves setting up the model and generating images based on textual prompts, and then we explore the deeper aspects of the architecture.

B. Image Generation Process

The image generation begins with the setup of the SD model and the integration of the CLIP tokenizer and text encoder. The initial noise seed used to start the generation process influences the randomness of the output images. When the noise seed is fixed, the model consistently generates the same image for a given prompt. This reproducibility can be valuable in controlled experiments but limits the diversity of the outputs. To explore multiple visual interpretations of a prompt, we set the noise seed to a random value in each generation, allowing the model to produce varied images for the same text input.

C. Model and Tokenizer Setup

The CLIP tokenizer and text encoder are loaded to process the input prompt into a format compatible with the diffusion model. The text is tokenized and encoded into a latent representation that the model uses to guide the image generation. The encoded text embeddings serve as the conditioning input, guiding the UNet component in generating an image that corresponds to the prompt.

D. Image Generation from Prompt

For generating an image we pass a text prompt ("A beautiful sunset above sea and mountains") through the tokenizer and the text encoder to produce text embeddings, which are then used as input to the UNet model which processes the latent noise to generate the image. Figure 10 illustrates an example output image generated using the above input.

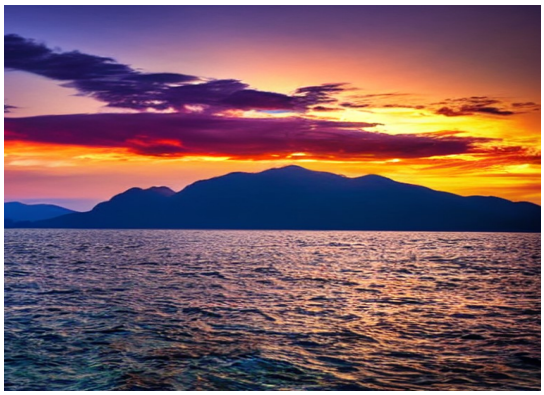


Fig. 10: Image generated using the Stable Diffusion model conditioned on the prompt "A beautiful sunset above the sea."

III. Latent Diffusion and VAE Compression

After understanding the image generation setup, we delve into the latent diffusion process which is crucial for handling high-resolution images. Latent diffusion minimizes computational costs by compressing the image into a latent space using a Variational Autoencoder (VAE). The VAE encodes the high-dimensional pixel space (x) into a lower-dimensional latent representation ($z = \mathcal{E}(x)$), which allows the model to perform diffusion in a computationally efficient manner. The process retains essential information while reducing the image dimensions significantly.

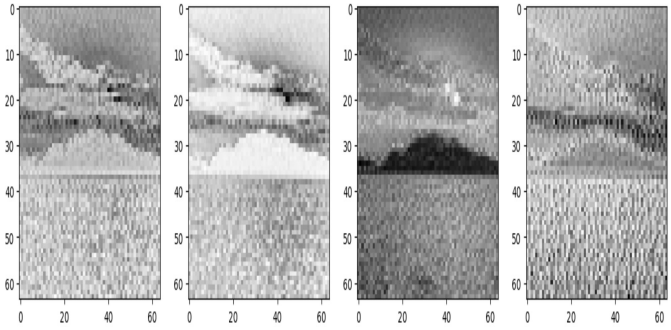


Fig. 11: The latent diffusion process in SD where high-resolution images are compressed into a smaller latent space for efficient processing.

A. Theoretical Considerations and Regularization Techniques

Latent Diffusion Models (LDMs) use the VAE to encode the image into a latent space, reducing its spatial dimensions by a factor of $F = \frac{H}{h} = \frac{W}{w}$. This downscaling significantly reduces memory usage, allowing the model to generate high-resolution images efficiently. To avoid high-variance latent spaces, the model uses two types of regularizations: KL-regularization (KL-reg.) and vector quantization (VQ-reg.).

KL-reg. applies a slight penalty to ensure the latent space follows a normal Gaussian distribution, which stabilizes the reconstructions. In contrast, VQ-reg. maps the latent representations to discrete values, improving stability and performance when integrated with the diffusion process.

B. Latent Space Compression and Efficiency

VAE compresses 512x512 pixel images into a latent space of dimensions 4x64x64, achieving a compression factor of 48. This compact representation reduces computational overhead while maintaining high-fidelity image reconstructions. The reduced dimensionality enables efficient processing during the diffusion steps, allowing the SD model to handle high-resolution images using manageable computational resources. The diagram in Figure 12 further illustrates the overall architecture of the latent diffusion process, detailing the integration of conditioning information like text and semantic maps via the cross-attention mechanism within the UNet.

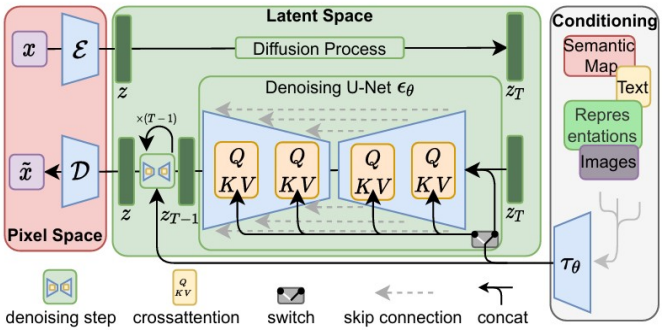


Fig. 12: Detailed diagram of the latent diffusion process in Stable Diffusion [1]

C. Denoising and Cross-Attention Mechanism

The denoising process in the latent space leverages a denoising UNet, ϵ_θ, which iteratively refines the noisy latent representations, z_t, to recover the clean version. The UNet uses cross-attention layers to incorporate different types of conditioning inputs, such as text embeddings from CLIP or semantic maps, enhancing the fidelity of the generated images.

The cross-attention mechanism, shown in the diagram, uses query (Q), key (K), and value (V) components to match relevant parts of the conditioning information with the latent representation. This approach ensures that each spatial location in the latent space is informed by the contextual information provided by the conditioning input, leading to more accurate and contextually relevant image generation.

D. Optimization Objective

The optimization objective for the latent diffusion model (LDM) is defined as:

$$L_{LDM} = E_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|^2\right] \quad (1)$$

Here, z_t represents the noisy latent at timestep t, and ϵ is noise sampled from a standard normal distribution. The model optimizes this objective by minimizing the difference between the predicted and actual noise, progressively refining the image through each denoising step.

E. Generative Modeling of Latent Representations

In the latent space, the generative process begins with a noisy latent z_T, which is gradually denoised back to a clean latent representation, z_0, using the conditioned UNet. This approach allows the model to operate in a low-dimensional space, making high-resolution image synthesis feasible without excessive computational demands. The VAE ensures that the compressed latent representations retain essential details for accurate image reconstruction, while the latent diffusion mechanism adds robustness and flexibility, making the overall process efficient and effective for generating high-fidelity outputs.

By combining latent space compression with an efficient denoising process, Stable Diffusion achieves a balance between computational efficiency and high-quality image synthesis, suitable for generating complex, high-resolution images efficiently.

IV. Text Conditioning and CLIP Integration

The Stable Diffusion (SD) model supports text conditioning through the CLIP text encoder, which transforms text prompts into embeddings that the UNet uses to guide the generation process. This conditioning mechanism provides the model with additional information about the image to be generated, allowing it to produce outputs that align closely with the text description provided.

A. Text Encoding Process

The CLIP model leverages a pre-trained transformer architecture designed to process image captions and compare images with text. When a text prompt is provided, CLIP tokenizes the input using a large vocabulary, assigning a specific token to each word. These tokens are then passed through the CLIP text encoder, which generates a high-dimensional vector for each token. In Stable Diffusion v1.4 each token is converted into a 768-dimensional vector, and these vectors are padded or truncated to a consistent length of 77 tokens. The final representation, which is used as conditioning input is a tensor of shape (77x768). This encoded representation (encoder_hidden_states) are then fed into the UNet as a conditioning input during denoising. The following figure shows the text encoding process in SD where the input tokenized, embedded and transformed through the stack of transformer blocks to generate embeddings.

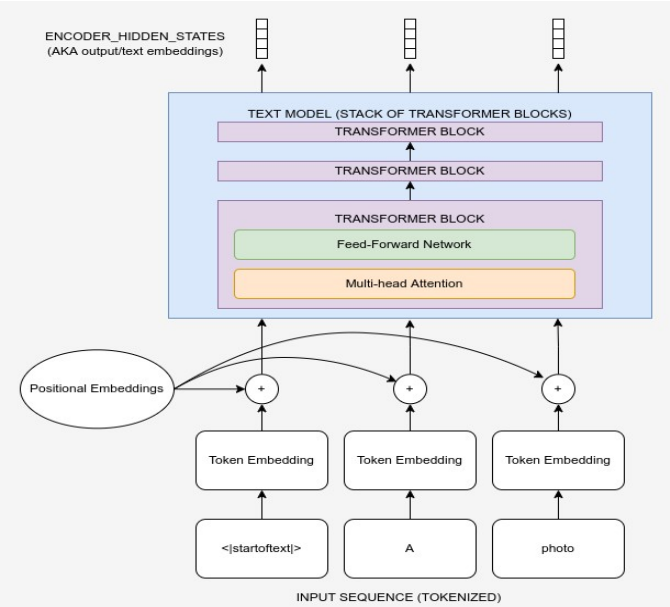


Fig. 13: CLIP text encoding process used in Stable Diffusion [2]

B. Cross-Attention Mechanism in the UNet

The encoded text embeddings generated by the CLIP model are incorporated into the UNet using cross-attention layers. These cross-attention mechanisms are scattered throughout the UNet's architecture, allowing the model to attend to different parts of the text embeddings at various spatial locations. By doing so, the UNet aligns the noisy latent representations with the details from the prompt, ensuring that the generated image accurately reflects the textual description.

This process works by matching each spatial location in the UNet with relevant tokens from the text embeddings. The cross-attention layers compute attention scores using query (Q), key (K), and value (V) vectors, bringing in information from the text prompt at each level of the UNet. As shown in Figure 14, the text conditioning information (along with timestep-based conditioning) is fed into different stages of the UNet, enhancing the coherence and quality of the generated image.

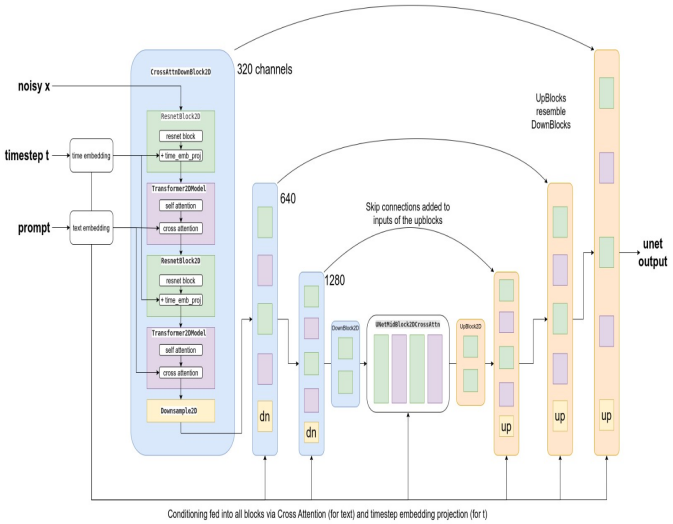


Fig. 14: Conditioning process in the UNet, where cross-attention layers integrate text embeddings and timestep information at various stages. This enables the model to generate images that accurately align with the input prompt [1].

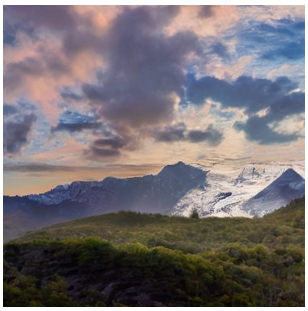
By integrating text conditioning via cross-attention, Stable Diffusion allows for detailed and contextually relevant image synthesis, making use of the extensive information embedded in the text prompt at every level of the generation process.

V. Experiments and Analysis

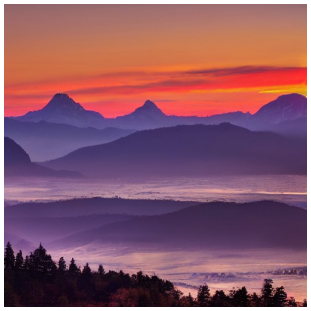
Various tests were conducted to evaluate the performance and sensitivity of the SD model to different parameters. These experiments explore the effects of guidance scale, noise scheduling, and latent space manipulation.

A. Guidance Scale Analysis

The guidance scale in SD adjusts reliance on the prompt, thereby influencing the alignment between the generated image, input text and improving image quality. A lower guidance scale allows more random elements in the image, making it less tightly aligned to the prompt. In contrast, a higher guidance scale emphasizes the prompt more strongly, resulting in images that are both clearer and more vibrant. We tested guidance scales ranging from 1 to 20 and observed the effects on image quality and prompt alignment, as shown in Figure 15.

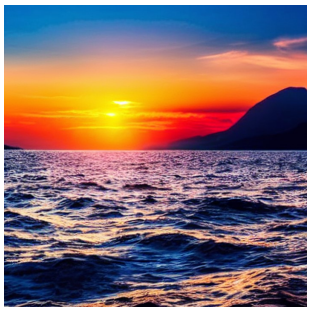


Scale 1



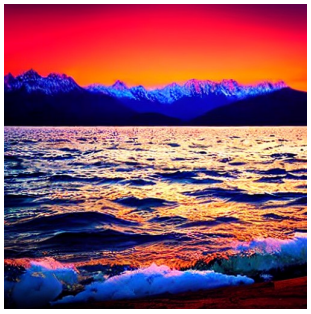
Scale 4

Scale 8



Scale 12

Scale 16



Scale 20

Fig. 15: Images generated with different Guidance scales for the prompt "A sunset over the sea with mountains."

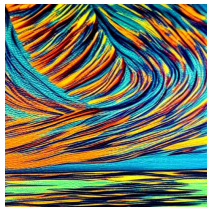
As the guidance scale increases, the images become more coherent, vibrant, and aligned with the prompt. At lower scales (1 up to 5) the generated images lack detail and deviate from the specified scene. However, as the scale is increased to 8 and 16, the images are more clear, have more vibrant colors and better alignment with the prompt's description, but also higher quality and more realistic outputs. However, higher scales come with a trade-off: increased computational load and slower generation time (3 minutes for 1 image, using Nvidia RTX3070 8Gb GPU). At scale 20 the image quality degrades as the model may overfit to the prompt's details, producing overly stylized or distorted outputs.

B. Noise Scheduling Experiments

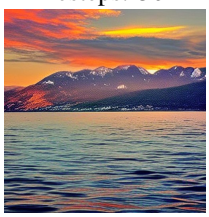
The noise schedule is managed by the LMSDiscreteScheduler. We tested variations in timesteps, ranging from 50 to 1500, to analyze the trade-off between image quality and generation time. The results shown in the following figures show how increasing the number of timesteps impacts the visual quality of the generated images.

Timesteps: 50

Timesteps: 600

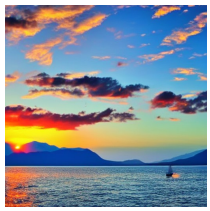


Timesteps: 675

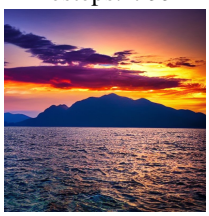


Timesteps: 700

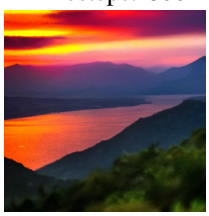
Timesteps: 800



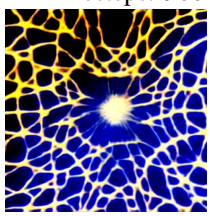
Timesteps: 900



Timesteps: 1000



Timesteps: 1200



Timesteps: 1500

As shown in the above figures, increasing the timesteps generally improves image quality up to a point. With only 50 timesteps, the image lacks structure and displays only some beige and grey patches. At 600 to 675 on can see some colourful patterns, and at 700 the first blurry image begins to form. At 800 timesteps, we observe a significant improvement with almost clear features. The best quality image is achieved at 1000 timesteps, with clear details and accurate color representation. Beyond this, at 1200 the image quality begins to slightly degrade, and by 1500 timesteps the image is over-processed and distorted.

C. Latent Space Manipulation

To understand the latent space's structure and its impact on image reconstruction, we conducted several manipulations using the latent vectors. Starting with an image generated based on the prompt "A beautiful picture of a sport car in Tokyo," we encoded it into the latent space and added various noise levels to observe how the output changes. The following figures show these steps sequentially:

Fig. 16: Generated image based on the prompt.

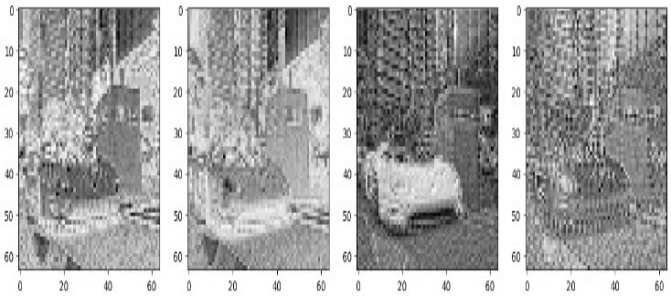


Fig. 17: Image encoded into latent space, showing its four-channel representation.

Fig. 18: Image decoded from latent space, demonstrating accurate reconstruction.

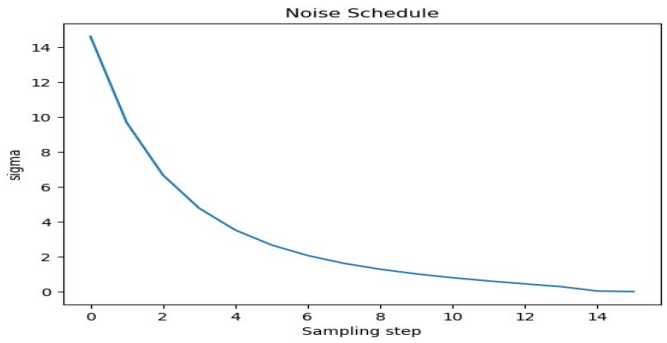


Fig. 19: Noise schedule applied using the LMSDiscreteScheduler, showing the reduction of sigma values over sampling steps.

Fig. 20: Image generated after adding noise to the latent vector and decoding it.

The results show that the encoding to latent space retains the structure and essence of the original image. When the latent vectors are decoded without additional noise, the output closely matches the input. However, as seen in Figure 20, adding noise at specific sampling steps significantly distorts the image, emphasizing the importance of precise noise scheduling during the image generation process. The noise schedule (Figure 19) controls the amount of noise applied at each step, influencing the overall quality of the generated image.

VI. CLIP-Guided Image Generation

The integration of CLIP guidance enhances control over image generation quality and adherence to prompts. CLIP guidance works by computing a similarity score between the text embeddings (from the prompt) and the visual features of the generated image. This score is used to adjust the latents in each iteration, aligning the image generation closer to the semantic content of the prompt.

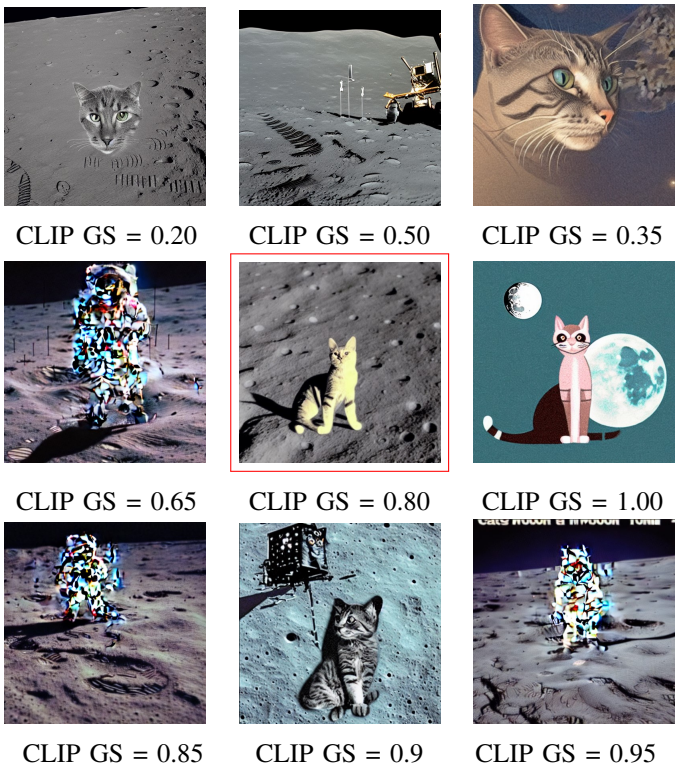


Fig. 21: Images generated with varying CLIP guidance strengths for the prompt “a cat on the moon.”

We conducted experiments using the prompt “a cat on the moon” and varied the CLIP guidance strength to evaluate the model’s response. The results, shown in Figure 21, demonstrate the impact of different CLIP Guidance Strengths (GS). At low strengths (0.2 and 0.65), the generated images fail to accurately capture the prompt, often displaying abstract or random features. As the strength increases to 0.8, the image quality and alignment improve significantly, resulting in the best visual output that also better matches the prompt description. However, further increasing the strength (values

between 0.8 and 1) results in distorted images, indicating that excessive CLIP guidance can over-penalize the generation, leading to instability. Therefore, a CLIP guidance strength of 0.8 was found to produce the best results, balancing prompt capture and image quality effectively.

VII. Novel Image Generations

To explore the creative potential of the model, we used a variety of imaginative prompts and applied CLIP guidance based on our previous findings. Below are several novel image generations:

Figure N1

Figure N2

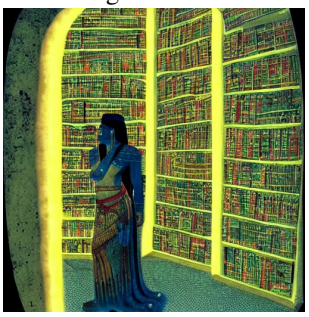


Figure N3

Figure N4

Fig. 22: Novel image generations using various imaginative prompts and CLIP guidance.

Figure N1: A spaceship docking at a crystal space station orbiting a giant blue planet.
Figure N2: A dancer made of smoke and light, performing on a stage of shattered glass.
Figure N3: Cleopatra exploring an ancient library filled with glowing hieroglyphs.
Figure N4: A floating library in the sky soaring through the clouds.

VIII. BERT-Guided Image Generation

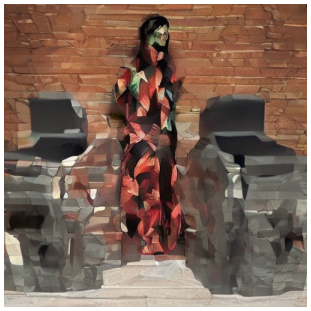
BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model designed primarily for natural language understanding tasks, such as text classification and named entity recognition. Unlike CLIP which aligns image and text representations in a shared embedding space, BERT focuses on capturing the contextual relationships between words in a sentence using a bidirectional approach. This means that BERT processes the entire input sequence at once, considering both left and right contexts simultaneously, providing a deep semantic understanding of the text.
When using BERT for image generation, we utilize the embeddings generated by the BERT model as a replacement for the traditional CLIP embeddings. The purpose is to evaluate how BERT embeddings, which are typically used for understanding textual information, influence the generation process when applied as guidance in the image synthesis pipeline.
In this experiment, we applied various guidance scales, ranging from low to high, to observe the model's behavior under different BERT-guided conditions. The images were generated using the prompt "A futuristic cityscape under a neon sky." Below, we present the generated outputs alongside their respective BERT guidance scales.

BERT GS = 0.01

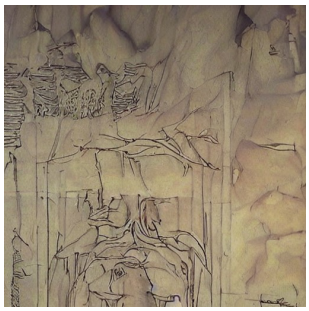
BERT GS = 2

BERT GS = 4

BERT GS = 6



BERT GS = 8



BERT GS = 10

Fig. 23: Images generated with varying BERT guidance strengths for the prompt "A futuristic cityscape under a neon sky."

As shown in Figure 23, the influence of BERT guidance on the generated images varies significantly with the guidance strength:

- **Low Guidance (0.01)**: At the lowest guidance level, the generated image appears abstract and unstructured, as the influence of BERT embeddings is minimal. The resulting image does not accurately reflect the prompt and lacks meaningful features.
- **Moderate Guidance (2-6)**: Increasing the guidance strength to values between 2 and 6 introduces more structure and patterning in the images. the images still do not capture the essence of the prompt and remain distorted and lack clarity.
- **High Guidance (8-10)**: At high guidance values, the influence of BERT becomes dominant. However, rather than improving image clarity or alignment with the prompt, the images become chaotic, even though we can see some picture for GS=8, but stil not a meaningful image. From that point on the higher the GS the more the images deviate into abstract shapes and textures that do not convey the intended neon-lit cityscape. This indicates that while BERT can provide some level of guidance, it still fails to generate meaningful images base on the input prompt, and this is due to its lack of visual understanding, and dur to that is mainly used for NLP tasks.

IX. Conclusion

Through our experimentation, we explored multiple approaches to guide image generation using CLIP and BERT embeddings. While CLIP embeddings, when adjusted with optimal guidance strengths, effectively aligned the generated images with their prompts, BERT embeddings displayed limitations due to their lack of multimodal training. As seen in the results, BERT-guided generations lacked coherence and failed to represent the intended visual elements of the prompts. This highlights the importance of using models trained specifically for aligning visual and textual information when guiding image generation. CLIP's ability to bridge the gap between text and image modalities provides significant advantages over traditional text-only models like BERT.

References

[1] **High-Resolution Image Synthesis with Latent Diffusion Models:** https://arxiv.org/pdf/2112.10752

[2] **Stable Diffusion:** https://huggingface.co/learn/diffusion-course/en/unit3/1

[3] **Fine-Tuning and Guidance:** https://huggingface.co/learn/diffusion-course/en/unit2/2what-you-will-learn

[4] **Learning Transferable Visual Models From NL Supervision (CLIP):** https://arxiv.org/pdf/2103.00020

[5] **stable-diffusion-v1-4 from Hugging face:** https://huggingface.co/CompVis/stable-diffusion-v1-4

[6] **ddpm-celebahq-256 faces:** https://huggingface.co/google/ddpm-celebahq-256

[7] **smithsonian_butterflies_subset butterflies:** https://huggingface.co/datasets/huggan/smithsonian_butterflies_subset

[8] **sd-class-wikiart-from-bedrooms wiki visual art from various artists:** https://huggingface.co/johnowhitaker/sd-class-wikiart-from-bedrooms

[9] **CLIP-ViT-B-32-xlm-roberta-base-laion5B-s13B-b90k:** https://huggingface.co/laion/CLIP-ViT-B-32-xlm-roberta-base-laion5B-s13B-b90k

[10] **Online Zero-Shot Classification with CLIP** https://arxiv.org/abs/2408.13320