|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Academic Year: | 2022/2023 | Term: | Spring 2023 |  |
| Course Code: | ELC 4028 | Course Title: | Artificial Neural Networks and its Applications |

Cairo University

Faculty of Engineering

Electronics and Communications Engineering Department – 4th Year

Neural Networks Applications

- Text-to-Image synthesis using Stable Diffusion -

*Submitted to: Dr. Mona Riad*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | BN | Sec | ID | رقم الجلوس |
| احمد محمود حسيني عطية | 22 | 1 | 9180178 | 34022 |
| علي ماهر عبدالسلام نبيه | 5 | 3 | 9190067 | 34117 |
| محمد احمد طه السيد | 30 | 3 | 9191043 | 34142 |
| محمد حسام عثمان يسن | 35 | 3 | 9191083 | 34147 |
| محمد عاطف ربيع | 43 | 3 | 9190924 | 34155 |

Table of Contents

[Abstract 1](#_Toc132225721)

[1. Introduction 2](#_Toc132225722)

[1.1. Motivation for Stable Diffusion 3](#_Toc132225723)

[2. Overview of Stable Diffusion Main Components 3](#_Toc132225724)

[2.1. Text Encoder 3](#_Toc132225725)

[2.2. Image Information Creator 4](#_Toc132225726)

[2.2.1. Diffusion 4](#_Toc132225727)

[2.3. Image Decoder 5](#_Toc132225728)

[3. Diffusion Model Training 6](#_Toc132225729)

[4. Diffusion Model Testing 7](#_Toc132225730)

[5. Why Latent Space Instead of Pixel Space? 7](#_Toc132225731)

[6. The Text Encoder 8](#_Toc132225732)

[6.1. CLIP Training 8](#_Toc132225733)

[6.2. Merging Text with Diffusion Process 9](#_Toc132225734)

[7. Results 10](#_Toc132225735)

[8. Conclusion 10](#_Toc132225736)

[9. References 11](#_Toc132225737)

**List of Figures**

Figure 1: text to image examples by Lexica 2

Figure 2: image to image block diagram example 3

Figure 3: Stable Diffusion Main Components 4

Figure 4: diffusion example from a noise pattern then slowly morphs into a desired output 4

Figure 5: visualizing outputs by a sweep over steps parameter 5

Figure 6: full cycle of SD 5

Figure 7: creating a noise predictor model from dataset 6

Figure 8: denoising or testing process of diffusion model 7

Figure 9: block diagram of full SD process 8

Figure 10: CLIP dataset 9

Figure 11: CLIP training cycle 9

Figure 12: U-Net with cross attention to include text 10

**List of Tables**

[Table 1: summary of Stable Diffusion main components 6](#_Toc132223984)

**List of Abbreviations**

A

Artificial Intelligence

(AI) 2

D

Diffusion Models

(DM) 1

L

Latent Diffusion Model

(LDM) 1

S

Stable Diffusion

(SD) 1

# Abstract

Text to image synthesis is a process of generating an image from a given text description. It is a form of natural language processing that involves the use of deep learning algorithms to generate an image from a text description. The goal of text to image synthesis is to create an image that accurately reflects the content of the text. In this report, we will discuss a paper called “High-Resolution Image Synthesis with Latent Diffusion Models” [1] which uses Stable Diffusion (SD) also called Latent Diffusion Model (LDM) to achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while signiﬁcantly reducing computational requirements compared to the previous pixel-based Diffusion Models (DM).

# 1. Introduction

Artificial Intelligence (AI) art has become extremely popular in recent years as technology advances and more people became interested in exploring the possibilities of using AI to create artwork. The Public release of SD in August 22, 2022 [2] has massively impacted this field as it’s not only a high-performance model competitive with present AI image generation models such as DALL-E by OpenAI or Imagen by Google but also its model weights and source code are fully open to anyone which allows anyone to download the model and tinker with it and adjust the internal parameters in a way that they can’t do with the closed solutions as DALL-E and Imagen. [3]. In addition, as we will discuss in this report, the latent space approach allowed relatively low resources and memory requirements [1].

the most common way of generation AI art from text is by typing a prompt to the model and it will generate the image, The best platform for finding examples and the prompts used to generate images is Lexica [4], which archives over 10 million sample artworks. Each artwork includes its full prompt and the seed number, some examples of images and their prompt are shown below

A realistic cute adorable baby owl made of crystal ball with low poly eye's surrounded by glowing aura highly detailed intricated concept art with vivid beautiful colors trending artstation 8k

A cute wizard elven, blue long hair, golden eyes, in a forest, simple dress, intricate, elegant, highly detailed, digital painting, hyper realistic, fantasy, dungeons and dragons, art by artgerm and greg rutkowski



streets of 1980s pripyat with neon lights in a foggy rainy night



Shabby Chic Teal and Pink Flowers with Fluffy Realistic Rabbit

Figure 1: text to image examples by Lexica

In addition, painting images from text isn’t the only feature of SD but also image to image and object removal is also possible. Also, various SD examples can be found in twitter social platform with the hashtag #StableDiffusion.

## 1.1. Motivation for Stable Diffusion

Before we dive into the model, we have to know the disadvantages of previous models.

DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data. Leading to 2 consequences for the research community and users in general which are: [1]

1. Training such a model requires massive computational resources only available to a small fraction of the ﬁeld. [1]
2. Evaluating an already trained model is also expensive in time and memory, since the same model architecture must run sequentially for a large number of steps. [1]

Hence, for these reasons a method is needed to increase the accessibility of this powerful model class and reduce its significant resource consumption without impairing their performance. [1]

They decided to work completely in the image information space (or latent space) which makes it faster than previous DMs that worked in pixel space. [1]

# 2. Overview of Stable Diffusion Main Components

The operation of SD is either text-to-image or can take both as an input to help it figure out the output image [5]



Figure 2: image to image block diagram example

This SD block consists of several components and models which we will look into.

## 2.1. Text Encoder

It is a text understanding component that translates the text prompt into a numeric representation so that the image generator can process. It is a special Transformer language model; It takes the input text and outputs a list of numbers representing each word/token in the text. [5]

## 2.2. Image Information Creator

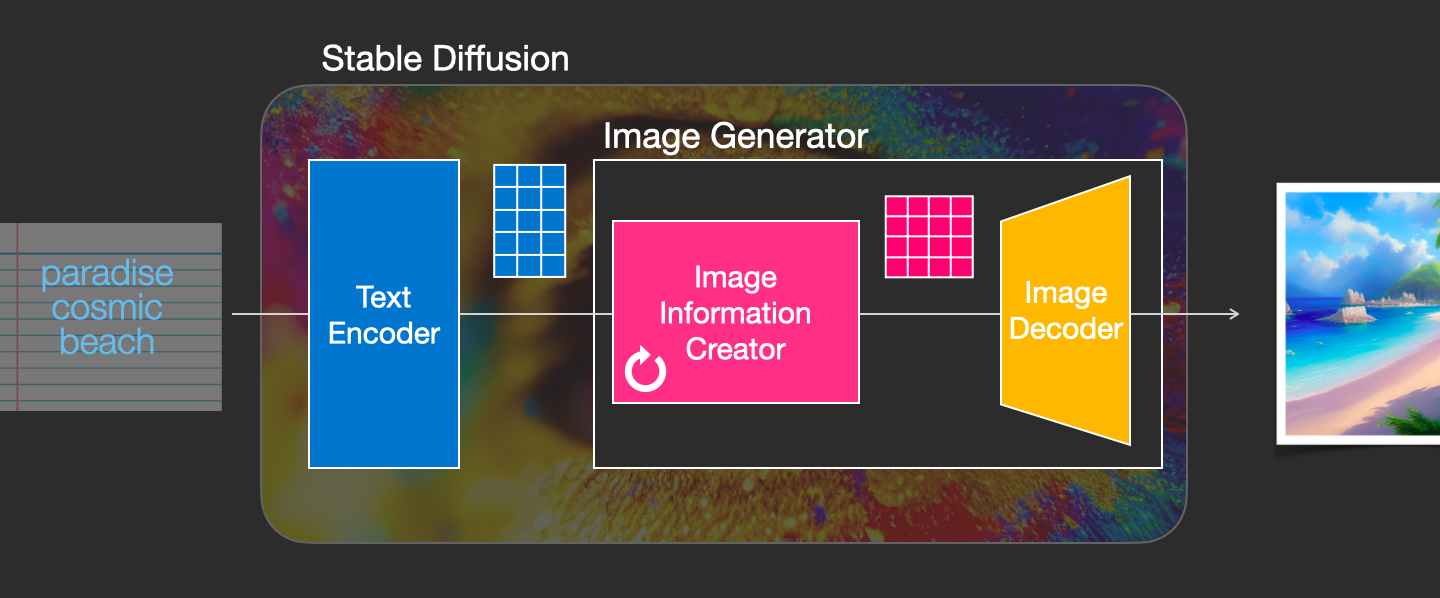


Figure 3: Stable Diffusion Main Components

Basically, what this component does from its name is generation image information from the data received from text encoder. It’s where a lot of the performance gain over previous models is achieved. This component runs for multiple steps to generate image information. This is the steps parameter in SD interfaces and libraries which often defaults to 50 or 100. This component is made up of a UNet-NN and a scheduling algorithm. [5]

### 2.2.1. Diffusion

The term diffusion means the step-by-step processing of information, from just a noise shaped pattern to a high-quality image with a meaning. [3]

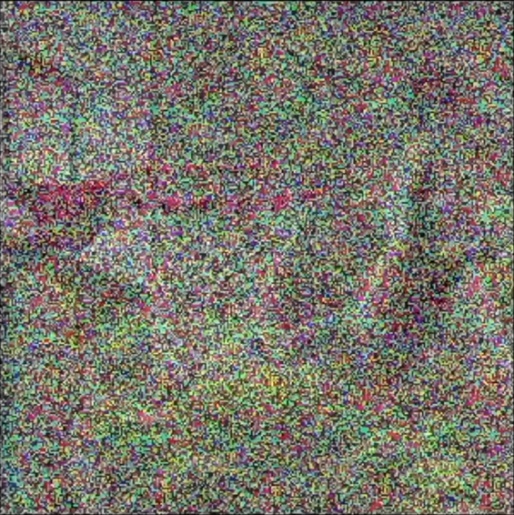


Figure 4: diffusion example from a noise pattern then slowly morphs into a desired output

It takes place inside the image information creator component, with the inputs as the token embeddings that represents input text and a random starting image information array or can be called random latents array. We will look into the process of a trained DM. [5]

The main parameter of this component is the steps parameter. This process happens in a step-by-step fashion. Each step adds more relevant information. at the end of all the steps we get the image in the pixel space by the image decoder. [5]

each step operates on an input latents array, and produces another latents array that better resembles the input text and all the visual information the model picked up from all images the model was trained on. [5]

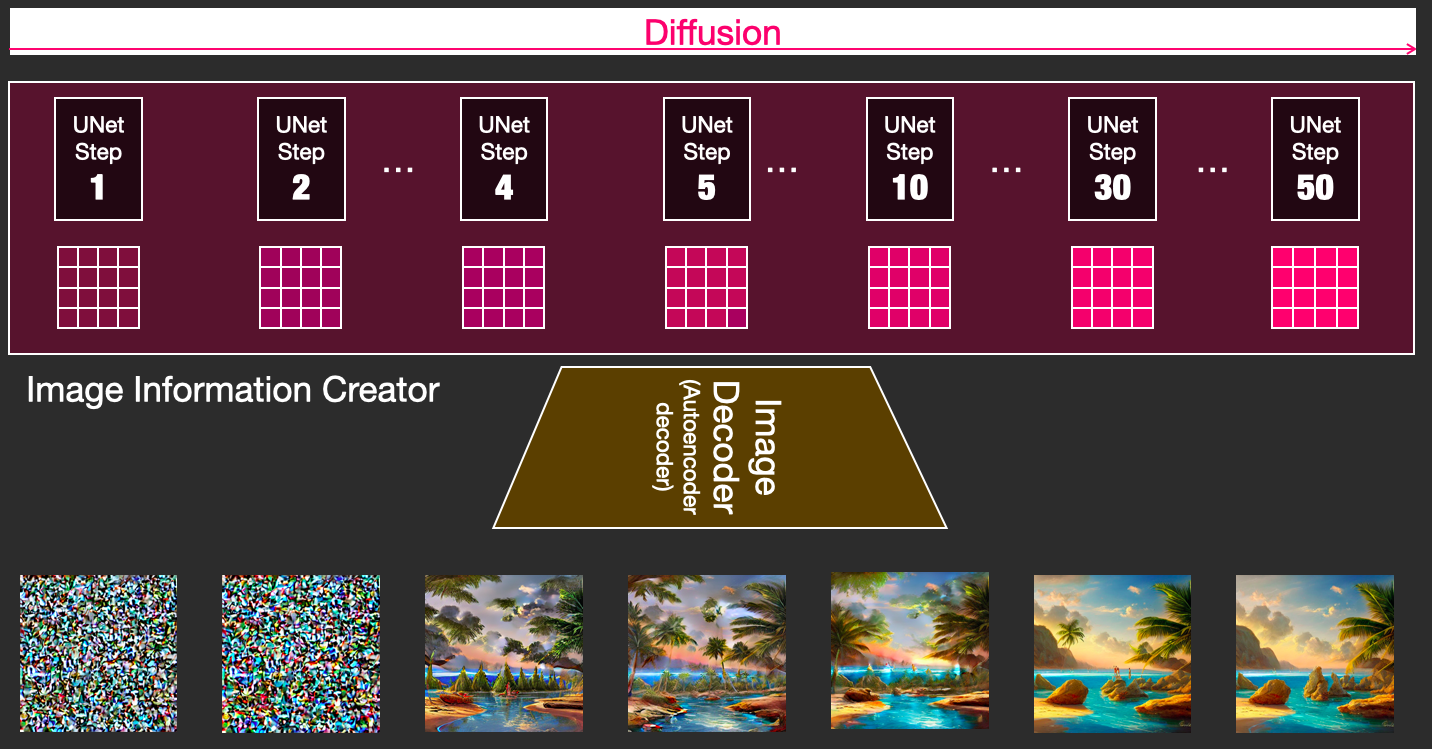


Figure 5: visualizing outputs by a sweep over steps parameter

We can notice the de-nosing process in the above figure.

## 2.3. Image Decoder

At the end of the process, it runs only once to produce the final pixel image. [5]

Now the three main components of SD are mentioned

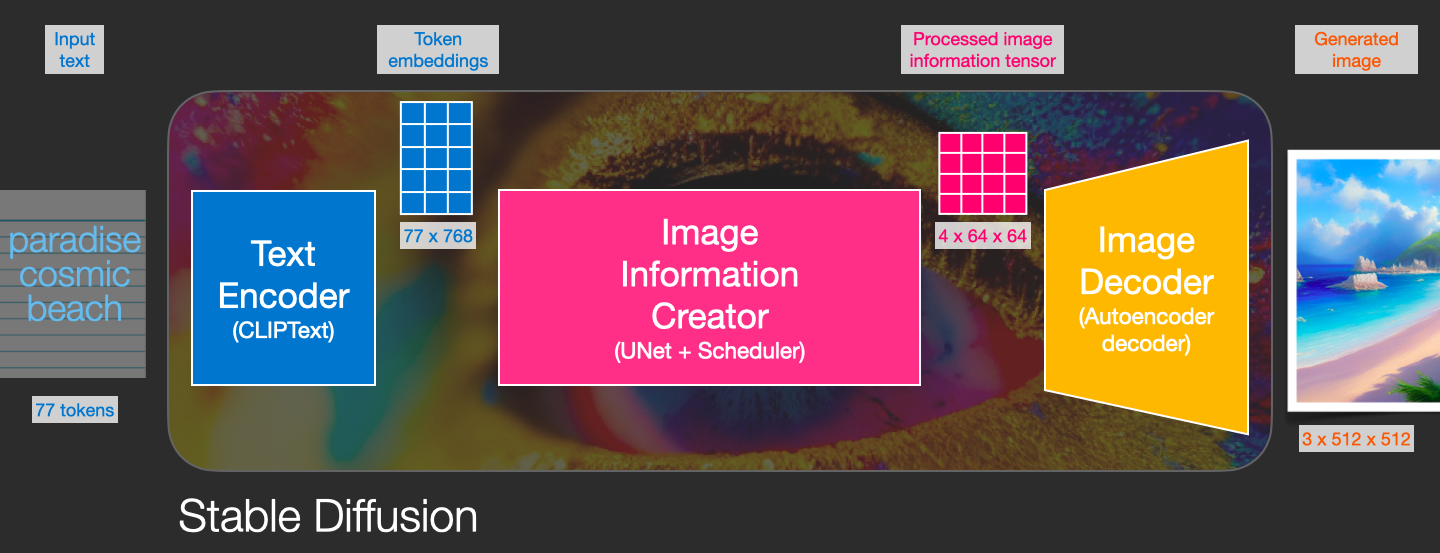


Figure 6: full cycle of SD

Each component consists of its own neural network model

Table 1: summary of Stable Diffusion main components

|  |  |  |  |
| --- | --- | --- | --- |
|  | ClipText | UNet + Scheduler | Autoencoder Decoder |
| Description | For text encoding | Gradually process the info in the latent space. | paints the final image using the processed info array. |
| Input | Text prompt | Text array made up of noise patterns | The processed info |
| Output | 77 token embeddings vectors, each in 768 dimensions. [5] | A processed info array with dimensions (4, 64, 64) | The resulting image, dimensions: (3, 512, 512) (RGB, width, height) [5] |

The 77 tokens are the max limit of the stable diffusion v2.1 release which is about 50 words. [6]

# 3. Diffusion Model Training

The training process is basically a supervised learning technique and it is called forward diffusion. Starting from an image, generate random noise and add it to the image. The more training examples the better performance we get from the model. [5]

we can easily control how much noise to add to the image, and so we can spread it over tens of steps, creating tens of training examples per image for all the images in a training dataset. [5]

With this dataset, we can train the noise predictor and end up with a great noise predictor that actually creates images when run in a certain configuration. [5]

we know the inputs and outputs. That’s why it is supervised.

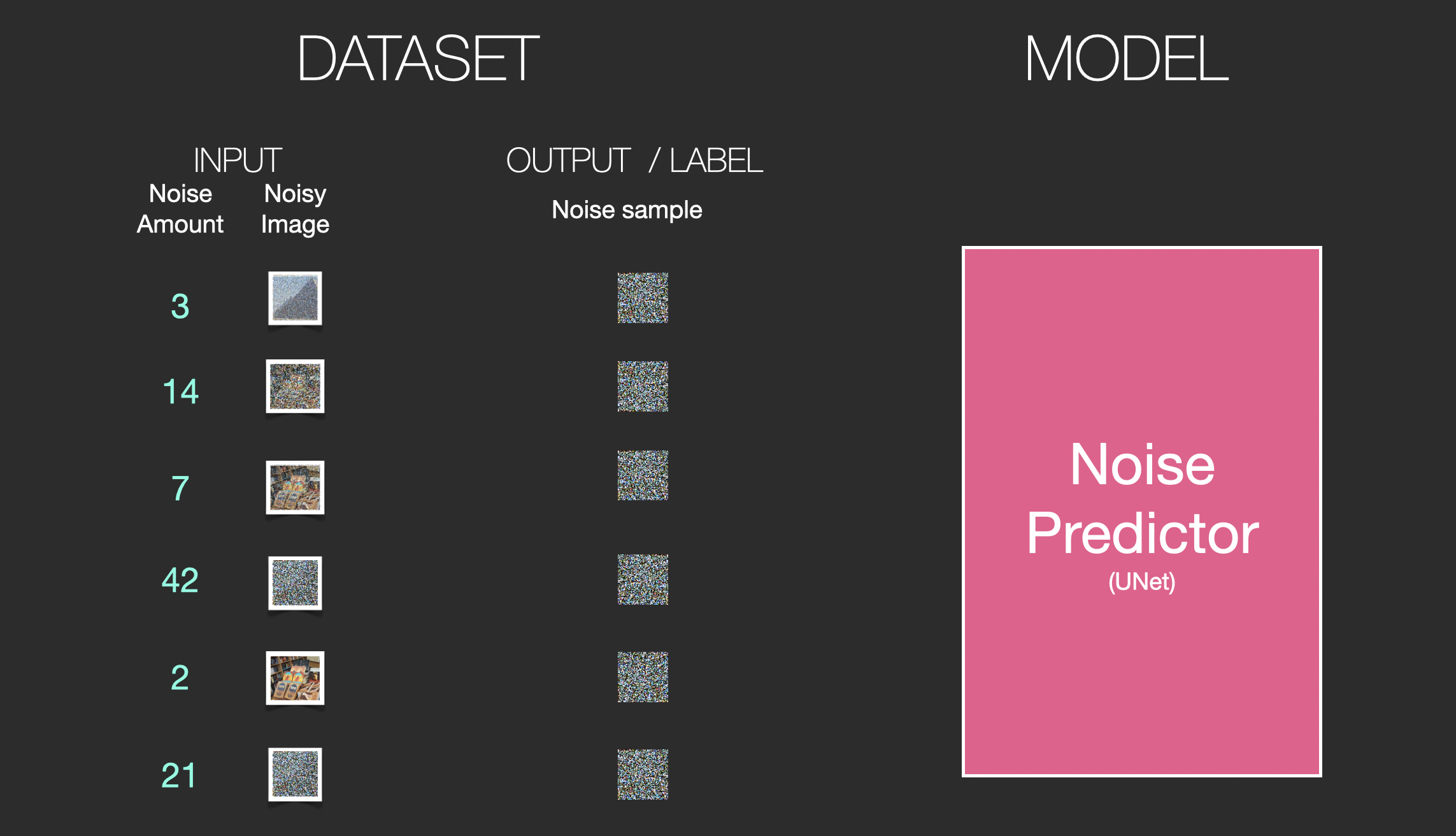


Figure 7: creating a noise predictor model from dataset

# 4. Diffusion Model Testing

Now we have a trained noise predictor, it takes a noisy image and the number of de-noising steps or the noise amount as in training and it will be able to predict a slice of noise. [5]

Then generated a predicted noise pattern which will be subtracted from the input then the result will be an input for the next step and so on till all steps are done. [5]

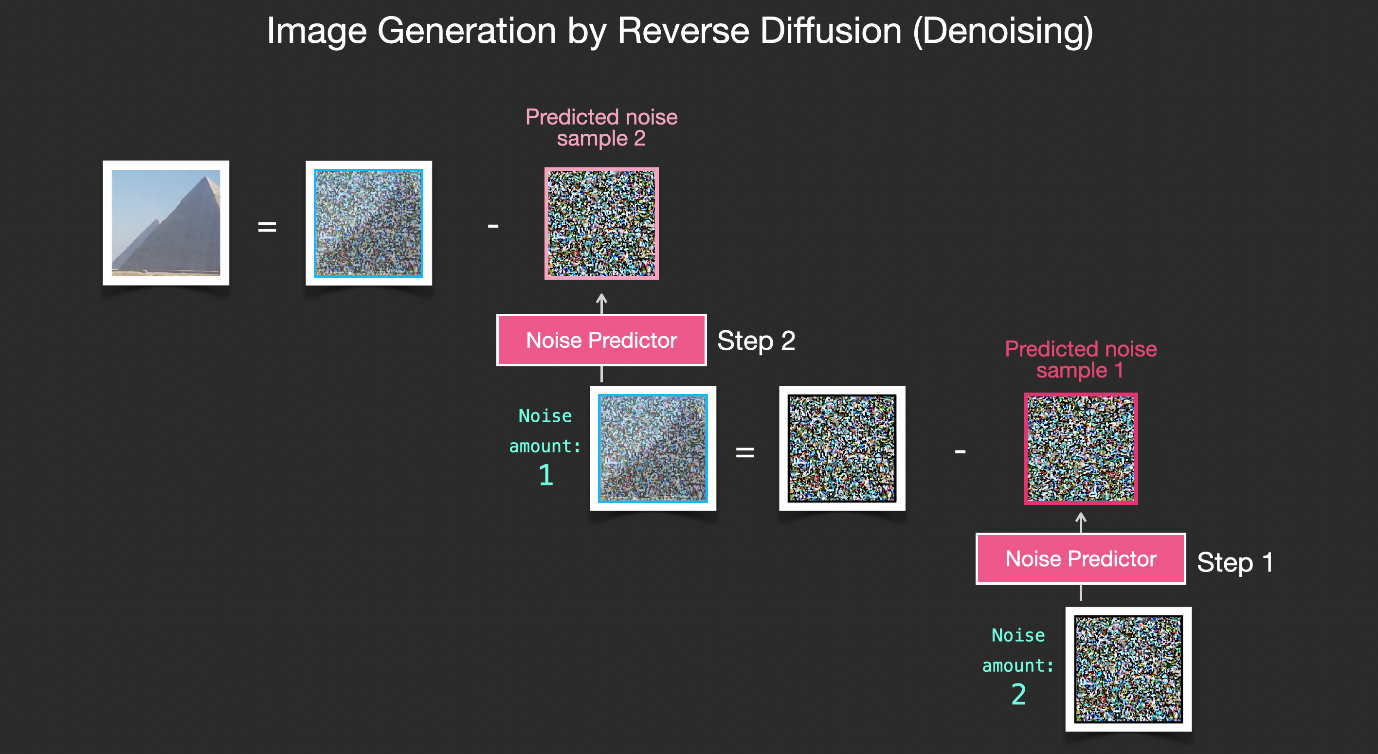


Figure 8: denoising or testing process of diffusion model

# 5. Why Latent Space Instead of Pixel Space?

To speed up the image generation process, the SD paper [1] runs the diffusion process not on the pixel images themselves, but on a compressed version of the image. This compression is done using an autoencoder. It compresses the image into the latent space using its encoder, then reconstructs it using only the compressed information using the decoder. [5]

The autoencoder is simply trained using a supervised technique as we manually give it the same image as input and output.

Now by working in this compressed latent space, the diffusion process happens in a much smaller space and works with much smaller arrays resulting in much faster generation.

The above figure from the paper. The pixel space on the left contains the original image if it’s provided by the user and its encoder and the output final image from the decoder. [1]

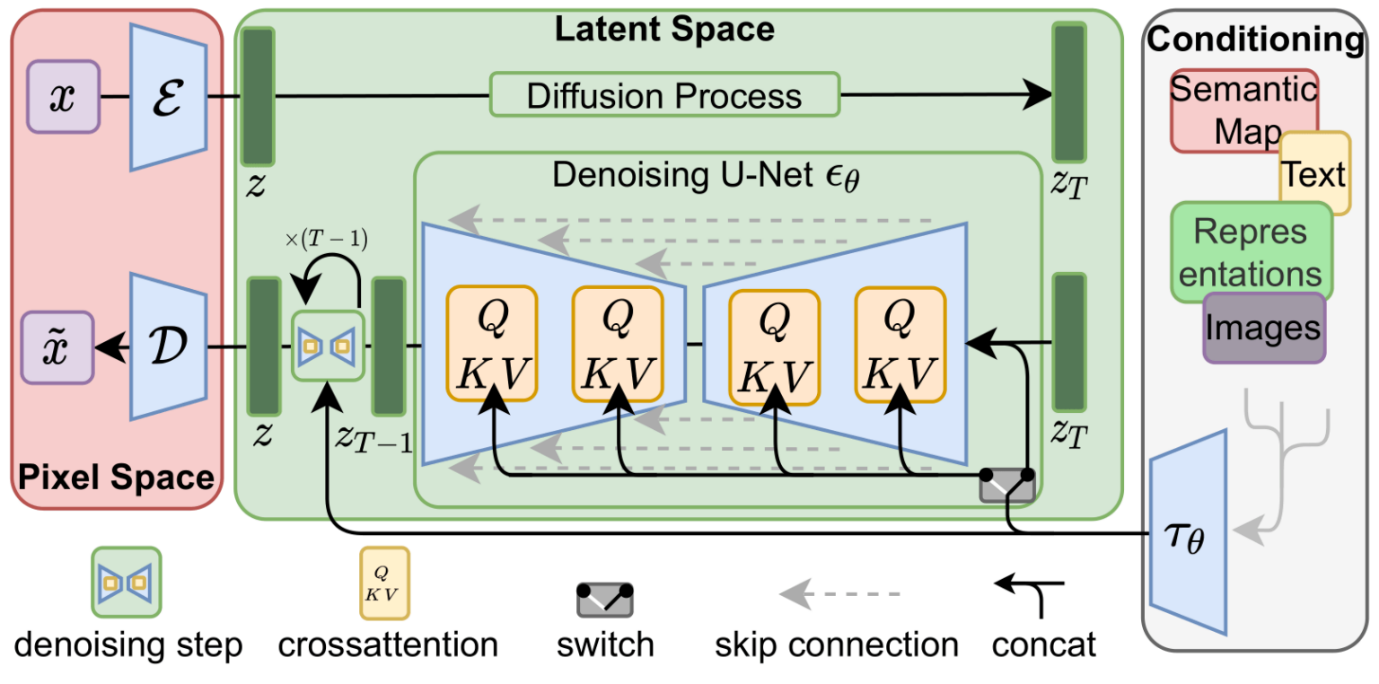


Figure 9: block diagram of full SD process

The latent space in the middle represents the diffusion de-noising process controlled with the steps parameter. The conditioning section represents the text prompts describing what image the model should generate. [1]

# 6. The Text Encoder

The diffusion process we described so far generates images without using any text data. So, if we deploy this model, it would generate great looking images, but we’d have no way of controlling if it’s an image of a pyramid or a cat or anything else. It will generate images that seem like images that it was trained on. In this section we’ll describe how text is involved in the process in order to control what type of image the model generates. Firstly, we need to know how to have some kind of understanding of the text. Secondly, how to include what it understood into the image generation process.

The text encoder is a transformer language model, the first release of SD model used the pre-trained ClipText which is release by OpenAI [2]. While the version 2 model is trained up using a brand-new text encoder (OpenCLIP), developed by LAION, that gives a deeper range of expression than version 1. [6]

## 6.1. CLIP Training

The dataset of the training are images with their corresponding caption. This dataset can be millions of images and captions. [5]

CLIP is a combination of an image encoder and a text encoder. Its training process can be simplified to thinking of taking an image and its caption. We encode them both with their respective encoder. A resulting numeric vector or array will be produced, the vectors should be similar and the model should be trained to do that. [5]

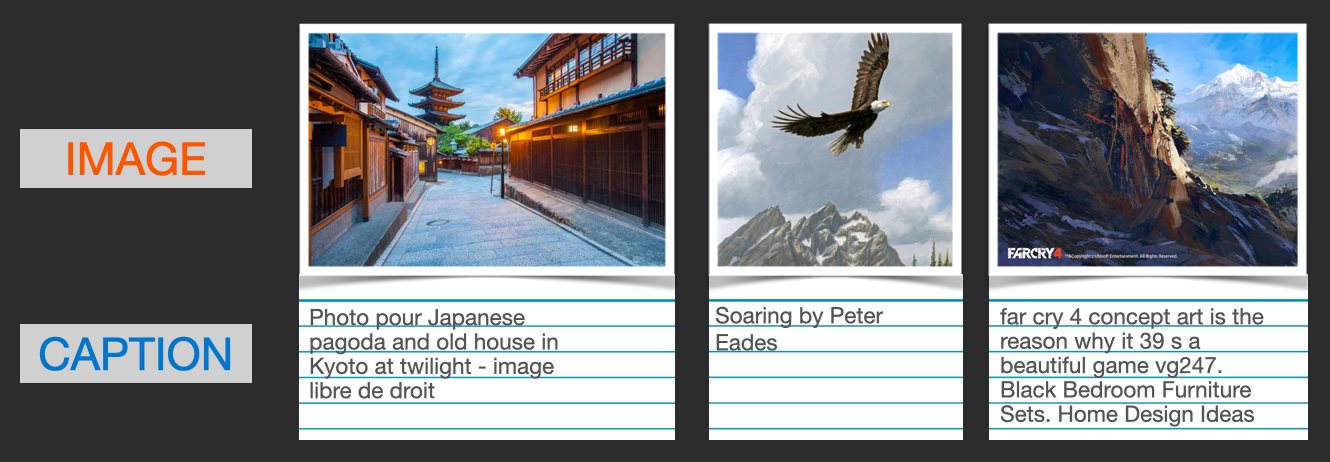


Figure 10: CLIP dataset

We compare the resulting embeddings. At the beginning of the training process, the similarity will be low, even if the text describes the image correctly. We update the two models so that the next time we embed them, the resulting embeddings are similar. [5]

By repeating this across the dataset and with large batch sizes, we end up with the encoders being able to produce embeddings where an image of a dog and the sentence “a picture of a dog” are similar. The training process also needs to include negative examples of images and captions that don’t match, and the model needs to assign them low similarity scores. [5]

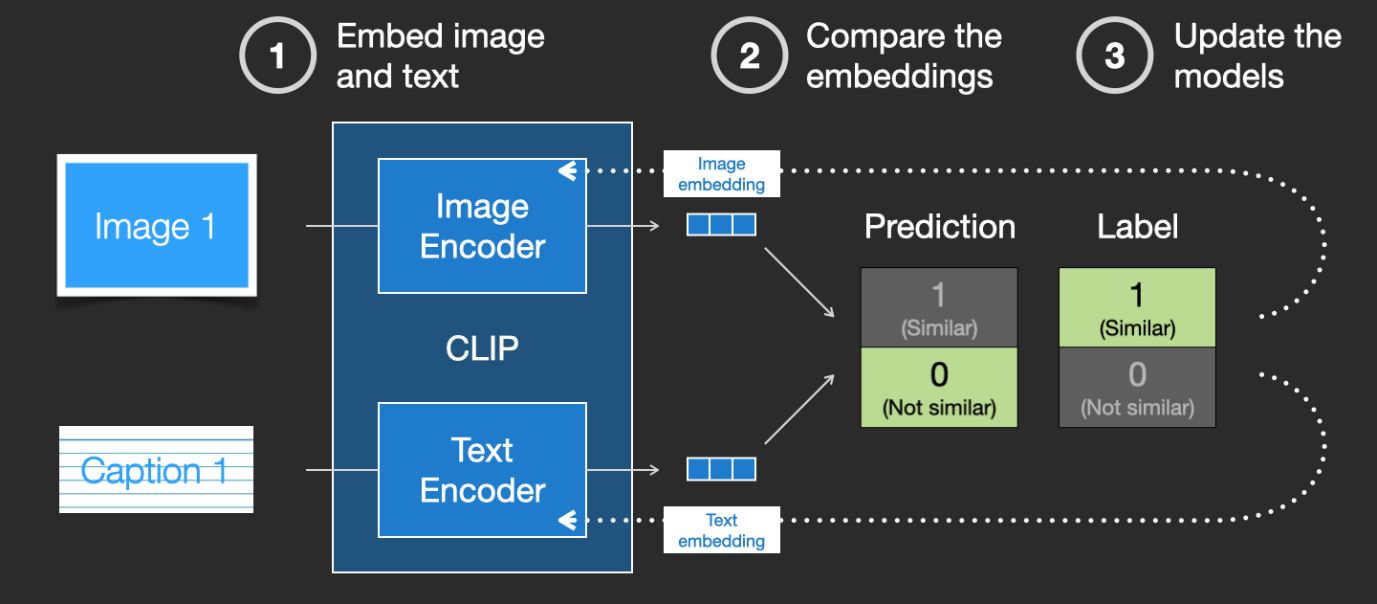


Figure 11: CLIP training cycle

## 6.2. Merging Text with Diffusion Process

To make text a part of the image generation process, we have to adjust our noise predictor to use the text as an input. Now the dataset will include noise amount (steps), images and text. [5]

The said before the noise predictor is formed using U-Net, it consists of a series of ResNet blocks each block’s input is the output of previous one.

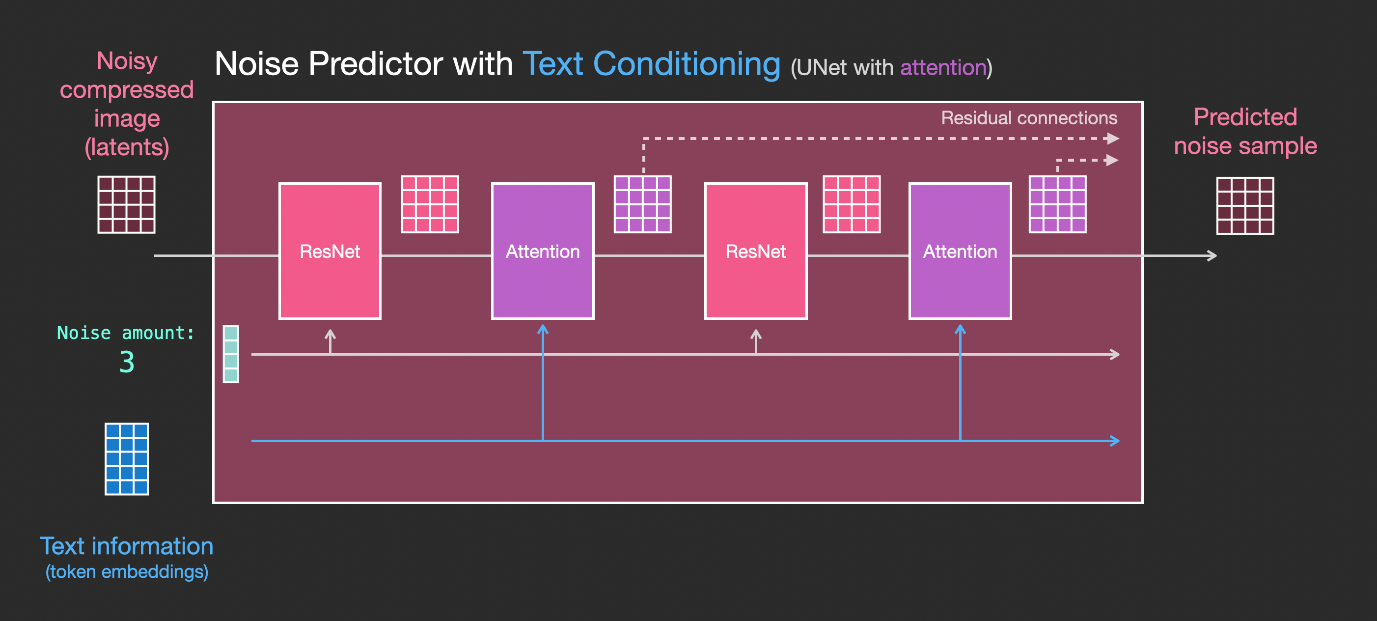


Figure 12: U-Net with cross attention to include text

The paper introduces a cross-attention conditioning mechanism [1]. The ResNet block doesn’t directly look at the text. But the attention layers merge those text representations in the latents. And now the next ResNet can utilize that incorporated text information in its processing. [5]

# 7. Results

# 8. Conclusion

# 9. References

|  |  |
| --- | --- |
| [1] | R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models | Papers With Code," 2022. [Online]. Available: https://paperswithcode.com/paper/high-resolution-image-synthesis-with-latent. [Accessed 2023]. |
| [2] | Stability AI, "Stable Diffusion Public Release — Stability AI," [Online]. Available: https://stability.ai/blog/stable-diffusion-public-release. [Accessed 2023]. |
| [3] | Two Minute Papers, "Stable Diffusion: DALL-E 2 For Free, For Everyone! - YouTube," 6 Septemper 2022. [Online]. Available: https://youtu.be/nVhmFski3vg. [Accessed 2023]. |
| [4] | "Lexica," [Online]. Available: https://lexica.art/. [Accessed 2023]. |
| [5] | J. Alammar, "The Illustrated Stable Diffusion," 2022. [Online]. Available: https://jalammar.github.io/illustrated-stable-diffusion/. [Accessed 2023]. |
| [6] | Stability AI, "Stable Diffusion v2.1 and DreamStudio Updates 7-Dec 22 — Stability AI," [Online]. Available: https://stability.ai/blog/stablediffusion2-1-release7-dec-2022. [Accessed 2023]. |
| [7] | CompVis, "High-Resolution Image Synthesis with Latent Diffusion Models - Computer Vision & Learning Group," [Online]. Available: https://ommer-lab.com/research/latent-diffusion-models/. [Accessed 2023]. |
| [8] | Stability AI, "Stable Diffusion 2-1 - a Hugging Face Space by stabilityai," [Online]. Available: https://huggingface.co/spaces/stabilityai/stable-diffusion. [Accessed 2023]. |
| [9] | CompVis, "CompVis/latent-diffusion: High-Resolution Image Synthesis with Latent Diffusion Models," [Online]. Available: https://github.com/CompVis/latent-diffusion.git. [Accessed 2023]. |