
	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 نبیه	5	3	9190067	34117
محمد احمد طه السيد	30	3	9191043	34142
محمد حسام عثمان یسن	35	3	9191083	34147
محمد عاطف ربیع	43	3	9190924	34155

Table of Contents

Abstract ..... 1

1. Introduction ..... 1

    1.1. Motivation for Stable Diffusion ..... 2

2. Overview of Stable Diffusion Main Components ..... 2

    2.1. Text Encoder ..... 2

    2.2. Image Information Creator ..... 2

        2.2.1. Diffusion ..... 3

    2.3. Image Decoder ..... 4

3. Diffusion Model Training ..... 4

4. Diffusion Model Testing ..... 4

5. Why Latent Space Instead of Pixel Space?..... 5

6. The Text Encoder..... 6

    6.1. CLIP Training ..... 6

    6.2. Merging Text with Diffusion Process ..... 7

7. Results ..... 8

    7.1. Compression Trade-off ..... 8

    7.2. Image Generation with Latent Diffusion ..... 8

    7.3. Conditional Latent Diffusion ..... 9

    7.4. Super-Resolution with Latent Diffusion ..... 10

    7.5. Inpainting with Latent Diffusion..... 11

8. Social Impact of AI Art..... 12

9. Conclusion ..... 12

10. References ..... 13

## List of Figures

Figure 1: text to image examples by Lexica .....	1
Figure 2: internal architecture of SD .....	2
Figure 3: diffusion example from a noise pattern then slowly morphs into a desired output.....	3
Figure 4: visualizing outputs by a sweep over steps parameter.....	3
Figure 5: creating a noise predictor model from dataset.....	4
Figure 6: denoising or testing process of diffusion model.....	5
Figure 7: block diagram of full SD process .....	5
Figure 8: CLIP dataset .....	6
Figure 9: CLIP training cycle .....	7
Figure 10: U-Net with cross attention to include text.....	7
Figure 11: samples of datasets used for LDMs training .....	8
Figure 12: sample quality as a function of training progress for 2M steps on the ImageNet dataset.....	8
Figure 13: Super Resolution experiment .....	10
Figure 14: inpainting examples.....	11

## List of Tables

Table 1: summary of Stable Diffusion main components .....	4
Table 2: summary of training of unconditional models.....	9
Table 3: Evaluation of text-conditional image synthesis on the 256x256-sized MS-COCO dataset .....	9
Table 4: additional comparisons with recent state-of-the art models with LDMs trained on COCO and OpenImages finetuned on COCO .....	10
Table 5: FID and LPIP scores of LDM and LaMa .....	11
Table 6: User Study for SR and Inpainting.....	11

# List of Abbreviations

## A

Artificial Intelligence  
(AI)

## D

Diffusion Models  
(DM)

## F

Fréchet Inception Distance  
(FID)

## G

Generative Adversarial Networks  
(GANs)

## I

Inception Score  
(IS)

## L

Latent Diffusion Model  
(LDM)

## S

Stable Diffusion  
(SD)

# 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].

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

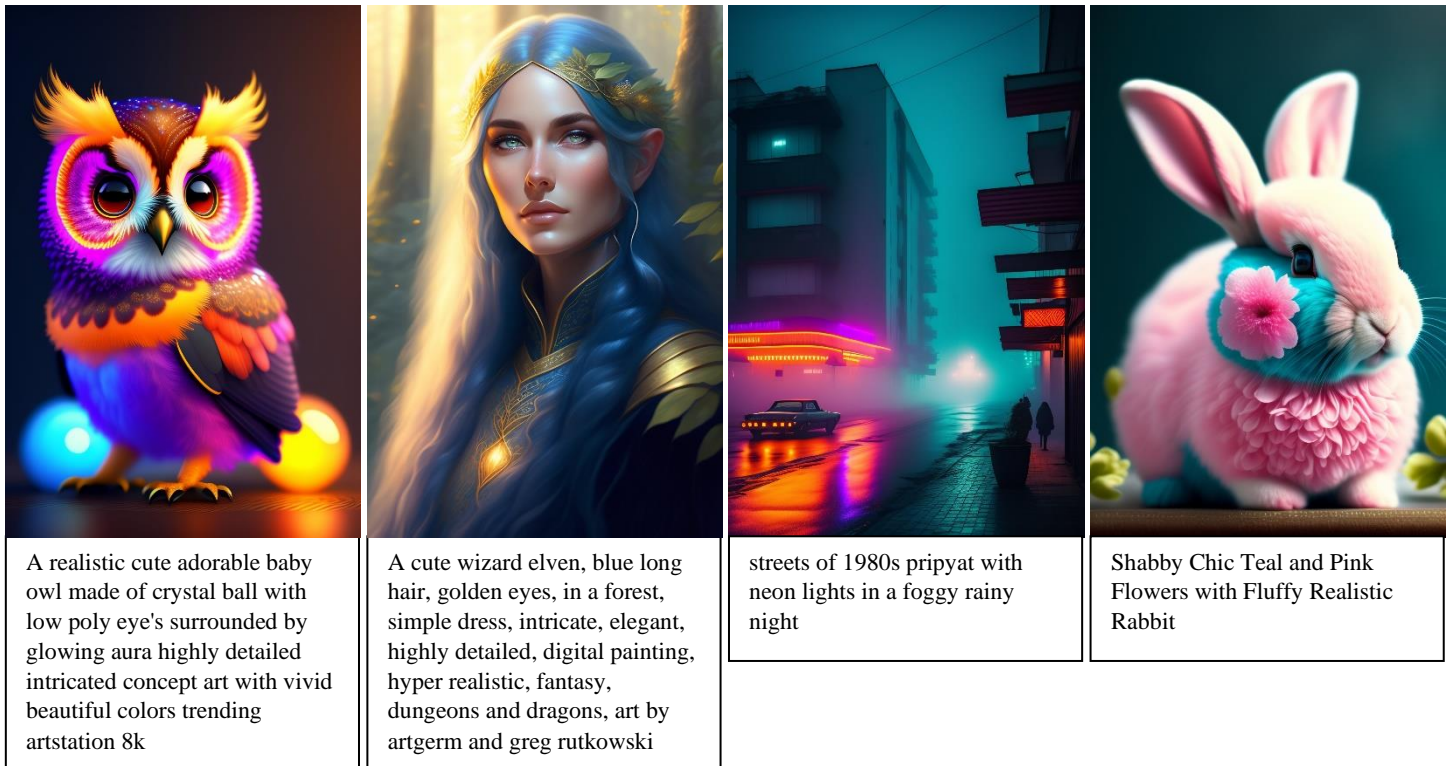


Figure 1: text to image examples by Lexica

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 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. This SD block consists of several components and models which we will look into. [5]

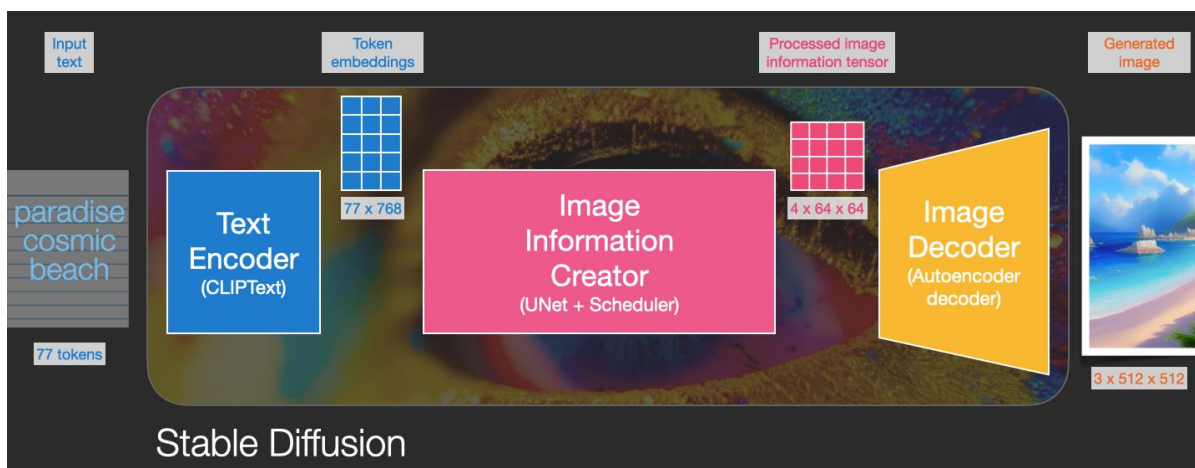


Figure 2: internal architecture of SD

### 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

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]

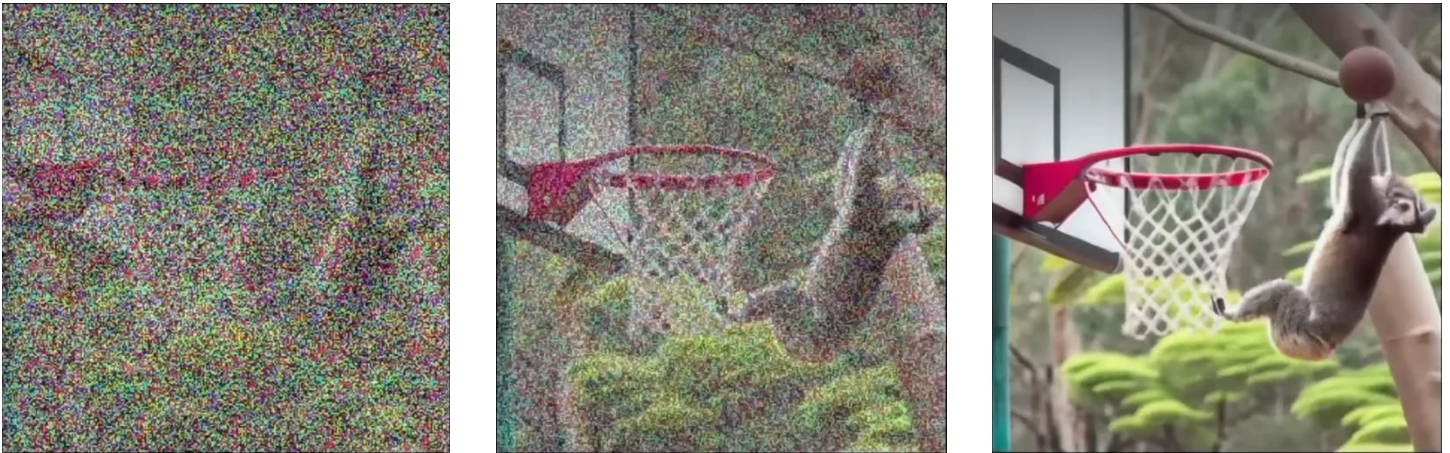


Figure 3: diffusion example from a noise pattern then slowly morphs into a desired output

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]

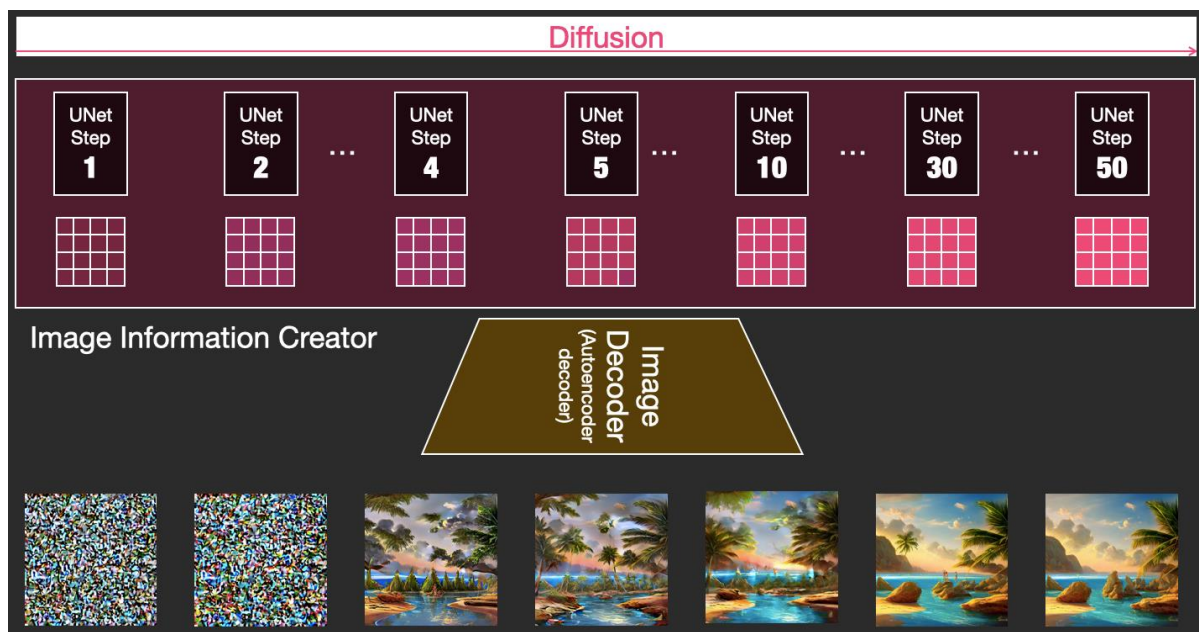


Figure 4: visualizing outputs by a sweep over steps parameter

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]

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

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]

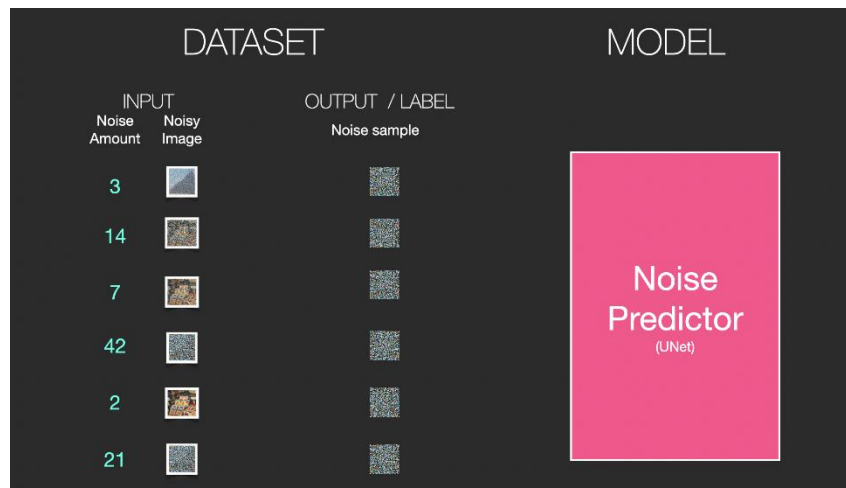


Figure 5: creating a noise predictor model from dataset

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.

## 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]

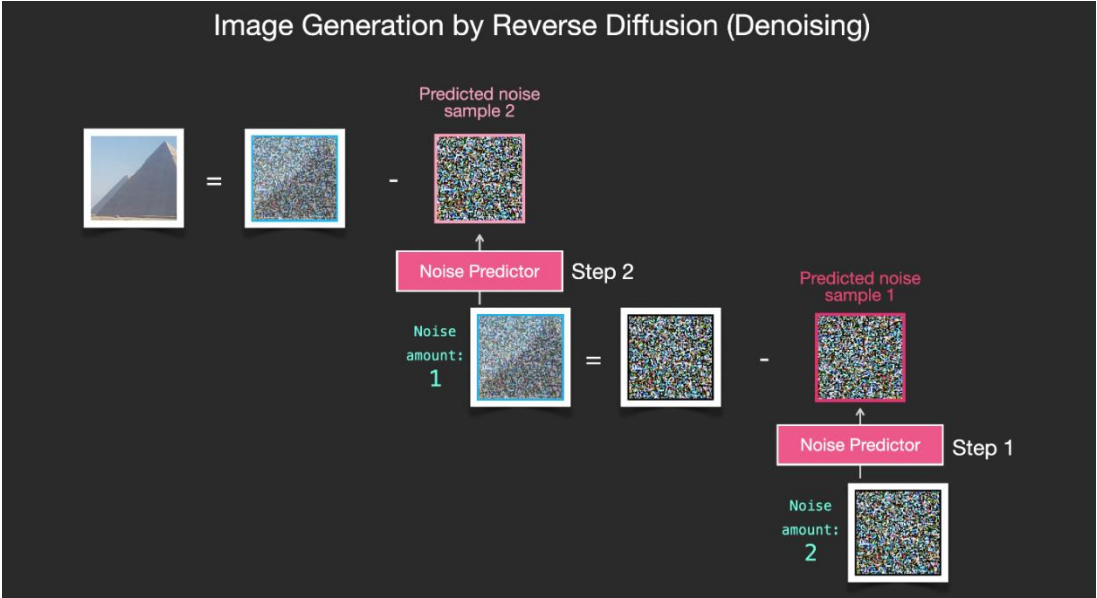


Figure 6: denoising or testing process of diffusion model

### 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.

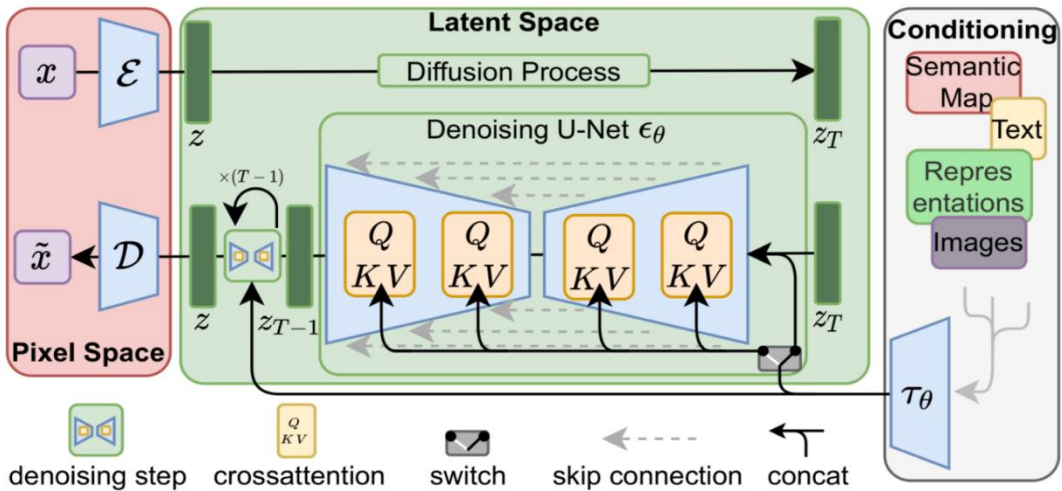


Figure 7: block diagram of full SD process

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]

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]

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]

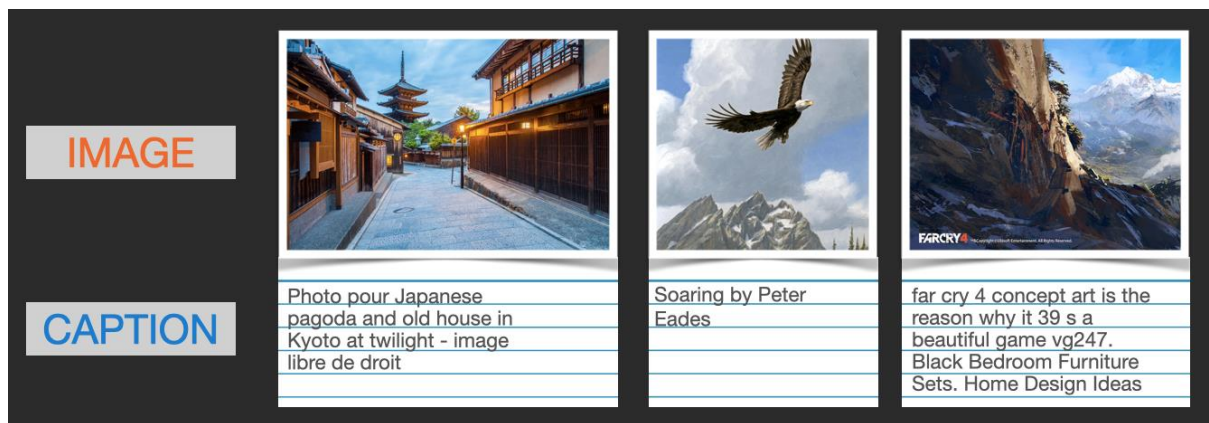


Figure 8: CLIP dataset

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]

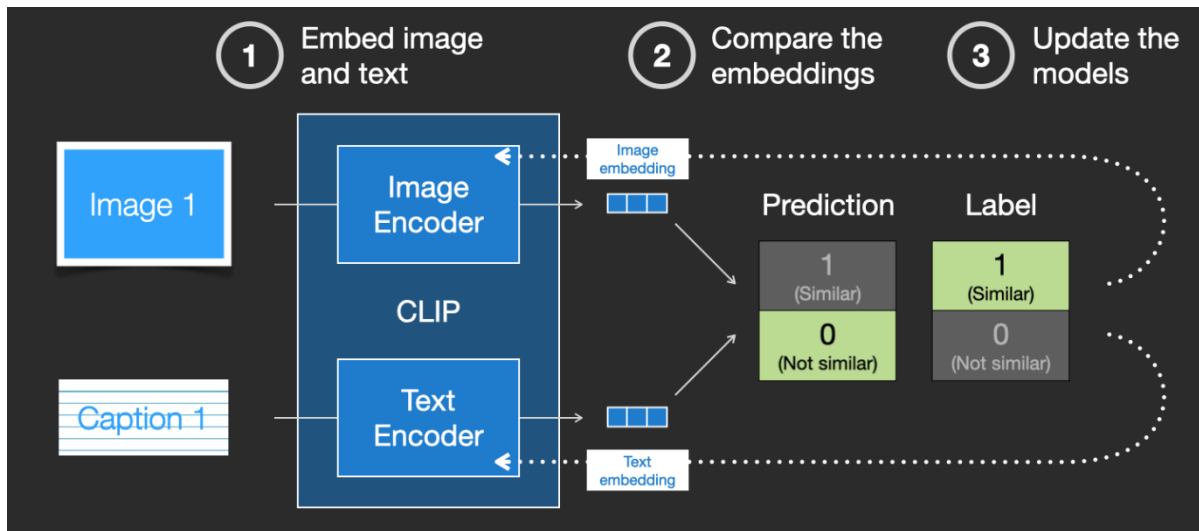


Figure 9: CLIP training cycle

## 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.

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]

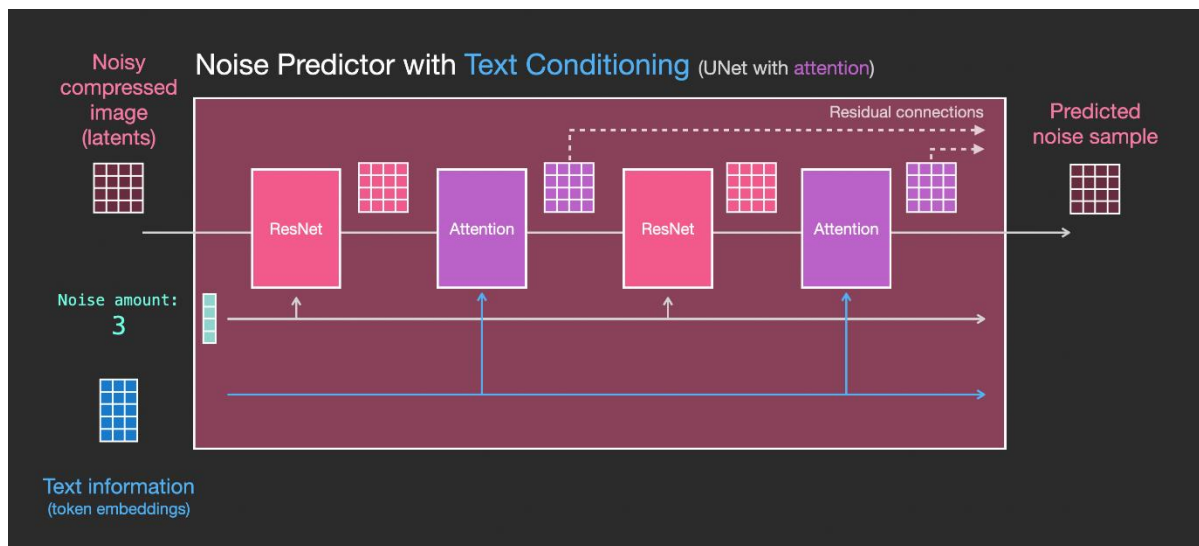


Figure 10: U-Net with cross attention to include text

## 7. Results

In this section, we present the results of LDMs compared to pixel-based DMs in both training and inference. [1]

Some of the generated images were evaluated using these metrics:

1. Inception Score (IS) measures the quality and diversity of generated images
2. Fréchet Inception Distance (FID) measures the similarity between generated images and real images.

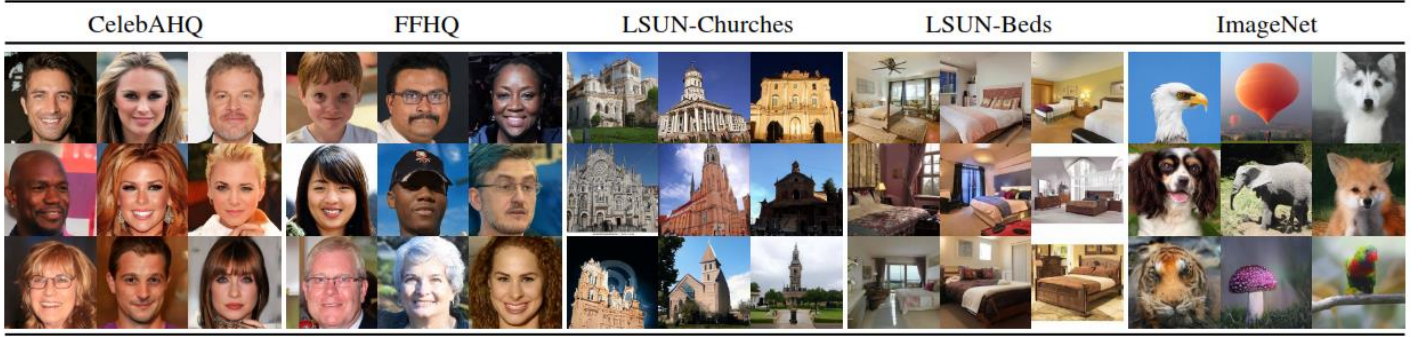


Figure 11: samples of datasets used for LDMs training

### 7.1. Compression Trade-off

The paper analyzes the behavior of LDMs with downsampling factors  $f \in \{1, 2, 4, 8, 16, 32\}$  abbreviated as LDM- $f$  where LDM-1 means the pixel-based DMs. [1]

We notice from the figure that the lower the downsampling factors result in a slow training progress. Starting from LDM-4 we get similar result and a good balance between efficiency and perceptually faithful results, which manifests in a significant FID gap of 38 between pixel-based diffusion (LDM-1) and LDM-8 after 2M training steps. [1]

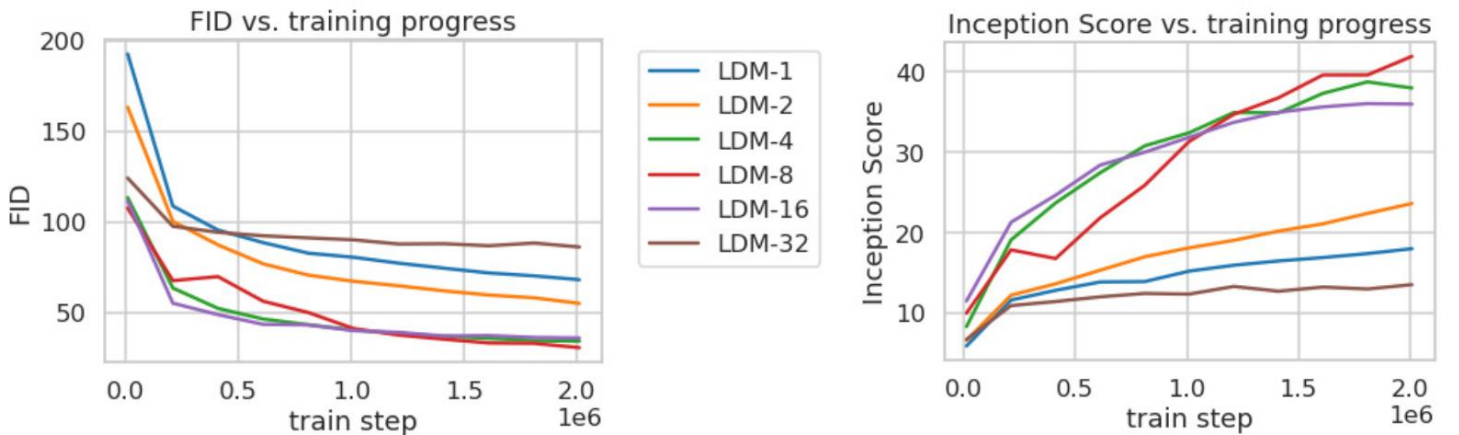


Figure 12: sample quality as a function of training progress for 2M steps on the ImageNet dataset

The LDM-4 and LDM-8 offer the best conditions to achieve high-quality synthesis results. [1]

### 7.2. Image Generation with Latent Diffusion

In this section, training of unconditional models of 256x256 images on the mentioned datasets. Then evaluate the sample quality and their coverage of the data manifold using FID, Precision and Recall.



we report a new state-of-the-art FID of 5.11, outperforming previous likelihood-based models as well as Generative Adversarial Networks (GANs). We also outperform LSGM where a latent diffusion model is trained jointly together with the first stage. [1]

Moreover, LDMs consistently improve upon GAN-based methods in Precision and Recall, thus confirming the advantages of their mode-covering likelihood-based training objective over adversarial approaches. [1]

CelebA-HQ $256 \times 256$				FFHQ $256 \times 256$			
Method	FID ↓	Prec. ↑	Recall ↑	Method	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-	ImageBART [21]	9.57	-	-
VQGAN+T. [23] (k=400)	10.2	-	-	U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
PGGAN [39]	8.0	-	-	UDM [43]	5.54	-	-
LSGM [93]	7.22	-	-	StyleGAN [41]	4.16	0.71	0.46
UDM [43]	7.16	-	-	ProjectedGAN [76]	3.08	0.65	0.46
<i>LDM-4</i> (ours, 500-s <sup>†</sup> )	<b>5.11</b>	0.72	0.49	<i>LDM-4</i> (ours, 200-s)	4.98	<b>0.73</b>	<b>0.50</b>

LSUN-Churches $256 \times 256$				LSUN-Bedrooms $256 \times 256$			
Method	FID ↓	Prec. ↑	Recall ↑	Method	FID ↓	Prec. ↑	Recall ↑
DDPM [30]	7.89	-	-	ImageBART [21]	5.51	-	-
ImageBART [21]	7.32	-	-	DDPM [30]	4.9	-	-
PGGAN [39]	6.42	-	-	UDM [43]	4.57	-	-
StyleGAN [41]	4.21	-	-	StyleGAN [41]	2.35	0.59	0.48
StyleGAN2 [42]	3.86	-	-	ADM [15]	1.90	<b>0.66</b>	<b>0.51</b>
ProjectedGAN [76]	<b>1.59</b>	0.61	0.44	ProjectedGAN [76]	<b>1.52</b>	0.61	0.34
<i>LDM-8*</i> (ours, 200-s)	4.02	<b>0.64</b>	<b>0.52</b>	<i>LDM-4</i> (ours, 200-s)	2.95	<b>0.66</b>	0.48

Table 2: summary of training of unconditional models

### 7.3. Conditional Latent Diffusion

Here the paper list their results after introducing cross-attention based conditioning into LDMs for the text-to-image synthesis. Training a 1.45B parameter KL-regularized LDM conditioned on language prompts on LAION-400M and employing BERT language model. [1]

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	$N_{\text{params}}$	
CogView <sup>†</sup> [17]	27.10	18.20	4B	self-ranking, rejection rate 0.017
LAFITE <sup>†</sup> [109]	26.94	26.02	75M	
GLIDE* [59]	12.24	-	6B	277 DDIM steps, c.f.g. [32] $s = 3$
Make-A-Scene* [26]	<b>11.84</b>	-	4B	c.f.g for AR models [98] $s = 5$
<i>LDM-KL-8</i>	23.31	20.03 $\pm$ 0.33	1.45B	250 DDIM steps
<i>LDM-KL-8-G*</i>	12.63	<b>30.29<math>\pm</math>0.42</b>	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

Table 3: Evaluation of text-conditional image synthesis on the 256x256-sized MS-COCO dataset

The paper follows prior work and evaluate text-to-image generation on the MS-COCO validation set. They noted that for the last row of the table, applying classifier-free diffusion guidance greatly boosts sample quality. such that the guided LDM-KL-8-G is on par with the recent state-of-the-art AR and diffusion models for text-to-image synthesis, while substantially reducing parameter count. [1]

To further analyze the flexibility of the cross-attention based conditioning mechanism they also train models to synthesize images based on semantic layouts on OpenImages and finetune on COCO. [1]

	COCO256 × 256	OpenImages 256 × 256	OpenImages 512 × 512
Method	FID↓	FID↓	FID↓
LostGAN-V2 [87]	42.55	-	-
OC-GAN [89]	41.65	-	-
SPADE [62]	<u>41.11</u>	-	-
VQGAN+T [37]	56.58	<u>45.33</u>	<u>48.11</u>
<i>LDM-8</i> (100 steps, ours)	42.06 <sup>†</sup>	-	-
<i>LDM-4</i> (200 steps, ours)	<b>40.91*</b>	<b>32.02</b>	<b>35.80</b>

Table 4: additional comparisons with recent state-of-the art models with LDMs trained on COCO and OpenImages finetuned on COCO

In addition, A LDM trained on 256x256 resolution can generalize to larger resolution and produce synthesis of landscape images. [1]

## 7.4. Super-Resolution with Latent Diffusion

The experiment, comparing with SR3 and fix the image degradation to a bicubic interpolation with 4x-downsampling and train on ImageNet following SR3’s data processing pipeline. [1]



Figure 13: Super Resolution experiment

LDM-SR has advantages at rendering realistic textures but SR3 can synthesize more coherent fine structures. [1]



## 7.5. Inpainting with Latent Diffusion

Inpainting is the task of filling masked regions of an image with new content either because parts of the image are corrupted or to replace existing but undesired content within the image. The paper evaluates how their general approach for conditional image generation compares to more specialized, state-of-the-art approaches for this task. [1]

The evaluation follows the protocol of LaMa, a recent inpainting model that introduces a specialized architecture relying on Fast Fourier Convolutions. [1]

Method	40-50% masked		All samples	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
<i>LDM-4</i> (ours, big, w/ ft)	<b>9.39</b>	<u>0.246</u> ± 0.042	<b>1.50</b>	<u>0.137</u> ± 0.080
<i>LDM-4</i> (ours, big, w/o ft)	12.89	0.257 ± 0.047	2.40	<u>0.142</u> ± 0.085
<i>LDM-4</i> (ours, w/ attn)	11.87	0.257 ± 0.042	2.15	<u>0.144</u> ± 0.084
<i>LDM-4</i> (ours, w/o attn)	12.60	0.259 ± 0.041	2.37	<u>0.145</u> ± 0.084
LaMa [88] <sup>†</sup>	12.31	<b>0.243</b> ± 0.038	2.23	<b>0.134</b> ± 0.080
LaMa [88]	12.0	<b>0.24</b>	2.21	<u>0.14</u>
CoModGAN [107]	<u>10.4</u>	0.26	<u>1.82</u>	0.15
RegionWise [52]	21.3	0.27	4.75	0.15
DeepFill v2 [104]	22.1	0.28	5.20	0.16
EdgeConnect [58]	30.5	0.28	8.37	0.16

Table 5: FID and LPIPS scores of LDM and LaMa



Figure 14: inpainting examples

The comparison with other inpainting approaches shows that LDM model with attention improves the overall image quality as measured by FID over that of LaMa. LPIPS between the unmasked images and our samples is slightly higher than that of LaMa. The paper attribute this to LaMa only producing a single result which tends to recover more of an average image compared to the diverse results produced by our LDM.

User Study	SR on ImageNet		Inpainting on Places	
	Pixel-DM ( <i>f</i> 1)	<i>LDM-4</i>	LAMA [88]	<i>LDM-4</i>
<b>Task 1:</b> Preference vs GT ↑	16.0%	<b>30.4%</b>	13.6%	<b>21.0%</b>
<b>Task 2:</b> Preference Score ↑	29.4%	<b>70.6%</b>	31.9%	<b>68.1%</b>

Table 6: User Study for SR and Inpainting

Additionally in a user study, human subjects favor LDM results.

## 8. Social Impact of AI Art

Generative models for media like imagery are a double-edged sword: On the one hand, they enable various creative applications, and in particular approaches like ours that reduce the cost of training and inference have the potential to facilitate access to this technology and democratize its exploration. On the other hand, it also means that it becomes easier to create and disseminate manipulated data or spread misinformation and spam such as changes a voice or a video clip to something that didn't actually happen. In particular, the deliberate manipulation of images ("deep fakes") is a common problem in this context. [1]

In summation these are the topics of concern for generative models:

- Dual use technology
- Media manipulation
- Harassment, abuse and objectification
- Unethical data sources
- Memorize and leak private data

More general, detailed discussion of the ethical considerations of deep generative models can be found in a paper called "Ethical considerations of generative ai" by Emily Denton. [7]

## 9. Conclusion

In this report, we presented a new trend took over the internet over the past year which is text to image generation using stable diffusion. It is a simple and efficient way to significantly improve both the training and sampling efficiency of denoising diffusion models without degrading their quality. We presented the top-level architecture and the main components explaining how each of them works. Then we dived into some of the main concepts of training and testing of these components to have an understanding of the major concept. Finally presented all the experiments of the paper and their results compared to previous works. An example of SD code and explanation can be found in [8] and the written code in [9].

## 10. 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 September 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] Youtube, "Emily Denton Ethical Considerations of Generative AI - YouTube," 6 August 2021. [Online]. Available: <https://youtu.be/RUA4CKiKSic>. [Accessed 2023].
- [8] Hugging Face, "Stable Diffusion with  Diffusers," 22 August 2022. [Online]. Available: [https://huggingface.co/blog/stable\\_diffusion](https://huggingface.co/blog/stable_diffusion). [Accessed 2023].
- [9] Hugging Face, "stable\_diffusion.ipynb - Colaboratory," 22 August 2022. [Online]. Available: [https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable\\_diffusion.ipynb](https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable_diffusion.ipynb). [Accessed 2023].
- [10] 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].
- [11] Stability AI, "Stable Diffusion 2-1 - a Hugging Face Space by stabilityai," [Online]. Available: <https://huggingface.co/spaces/stabilityai/stable-diffusion>. [Accessed 2023].
- [12] CompVis, "CompVis/latent-diffusion: High-Resolution Image Synthesis with Latent Diffusion Models," [Online]. Available: <https://github.com/CompVis/latent-diffusion.git>. [Accessed 2023].