

# Everything you need to know about GLIDE and DALL-E 2



Zain ul Abideen · [Follow](#)

8 min read · Oct 3, 2023



Learn about Guidance of Diffusion models and the application of CLIP.

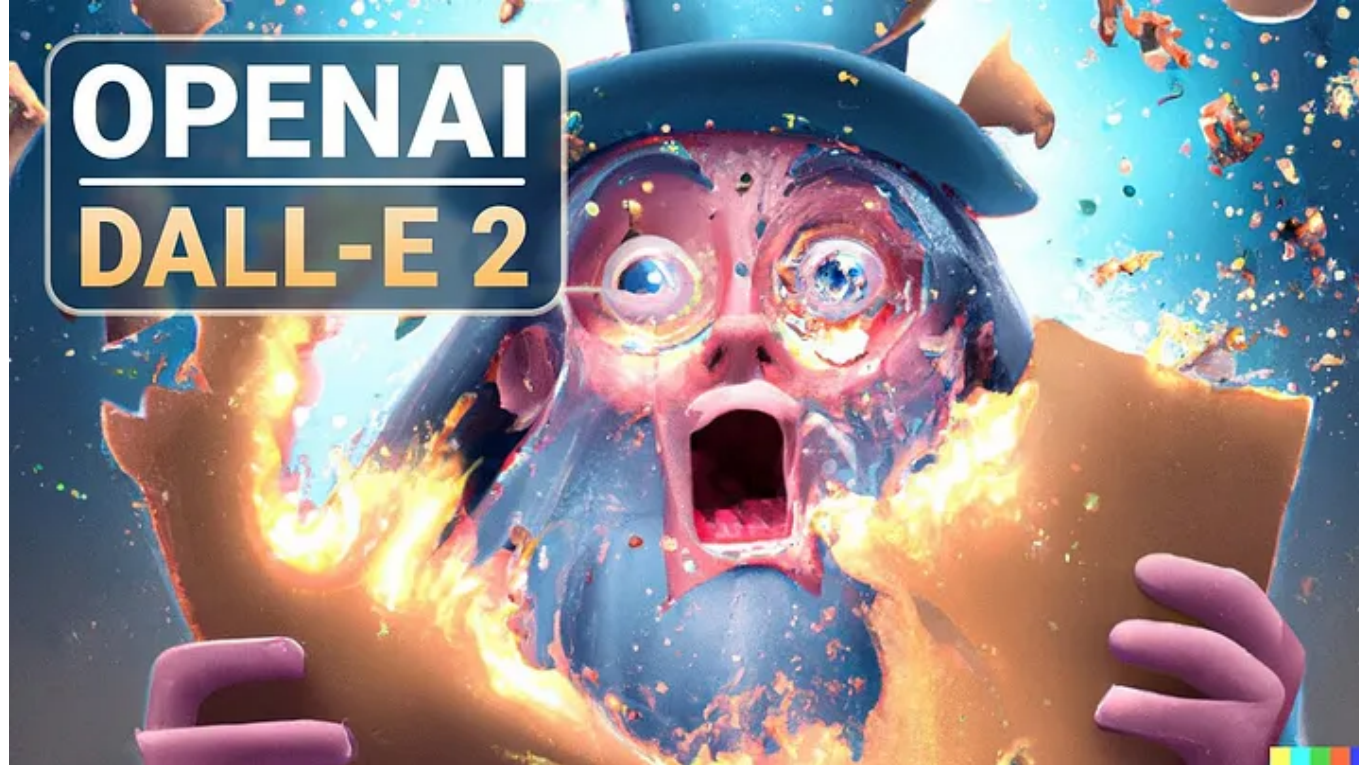


Photo from [Two Minute Papers](#)

## GLIDE

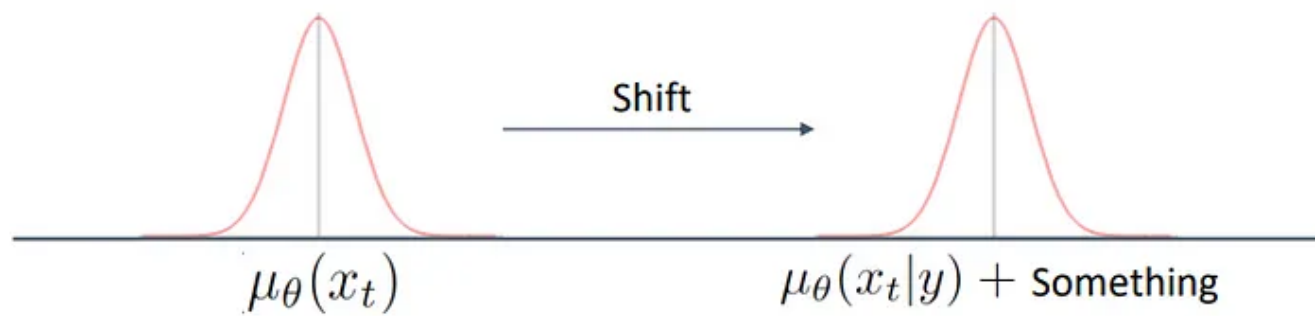
The “GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models” paper was published in March, 2022. The main contribution of this paper is that it developed a guided diffusion model to generate photorealistic images given textual prompts and ability to perform image inpainting. The overall architecture of the diffusion model is based on the UNET model. The diffusion model can be guided using two approaches: CLIP guidance or Classifier free guidance. Rather than jumping straight into the working of these guidance methods. Let’s discuss why we need guidance. Traditional diffusion models have the ability to generate new images from

noise by backward denoising. What if we want to generate a very specific type of cat? To do that, we need control of diffusion. To deal with that, we can start conditioning the diffusion model on the label  $y$  of the image during training. What if we want to generate something complex like a cat doing something ludicrous? We need even more control of the diffusion process to do that. The naive text conditional models result in incoherent samples. The solution to this is Guidance.

Now, we have understood that we need guidance to produce desired images. But what does it actually do? As we know that diffusion model takes the noised image as input and outputs mean and variance. This output forms a normal distribution from which we sample a less noisier image. This process continues for a specific number of steps and we generate a new image. Mathematically:

$$p_{\theta}(x_{t-1}|x_t) = \mathcal{N}(\mu_{\theta}(x_t), \Sigma_{\theta}(x_t))$$

Let's see how guidance influences this process. Guidance shifts the mean of the normal distribution by a specific value.



To find the value of this “Something”, we have two methods:

## **Classifier-Based Guidance or CLIP Based Guidance**

First of all let's discuss simple classifier-based guidance. This process can be divided into 2 steps. In the first step, we train the classifier model on noisy samples of the image  $x(t)$  with label  $y$ . Why train on noisy images? The answer to this is that we will be using this classifier during the diffusion steps and as we already know diffusion steps mostly have noisy samples of the labels rather than perfect crisp images. In the second step, during the diffusion process we will use this classifier. We will pass a noisy image  $x(t)$  to the classifier model and get  $y$ . We will compute the gradient of log probability of label  $y$  by  $x(t)$ . Mathematical form of the guiding process with gradient of log probability scaled by guidance scale  $s$  is shown below. Increasing guidance scale  $s$  improves sample quality at the cost of diversity.

$$\hat{\mu}_{\theta}(x_t|y) = \underbrace{\mu_{\theta}(x_t|y)}_{\text{Mean}} + \underbrace{s \cdot \Sigma_{\theta}(x_t|y) \nabla_{x_t} \log p_{\phi}(y|x_t)}_{\substack{\text{Guidance Scale} \\ \text{gradient of log probability of label } y \text{ by } x_t}}$$

Now, let's discuss CLIP based guidance. The same way as mentioned earlier, we will train a CLIP classifier model on noisy images. Then during the diffusion process, we will input  $x(t)$  and  $y$  to image encoder and text encoder respectively. The product of the outputs from image and text encoder is called Score. We will compute the gradient of the score by  $x(t)$ . Mathematically:

$$\nabla_{x_t}(f(x_t) \cdot g(y))$$

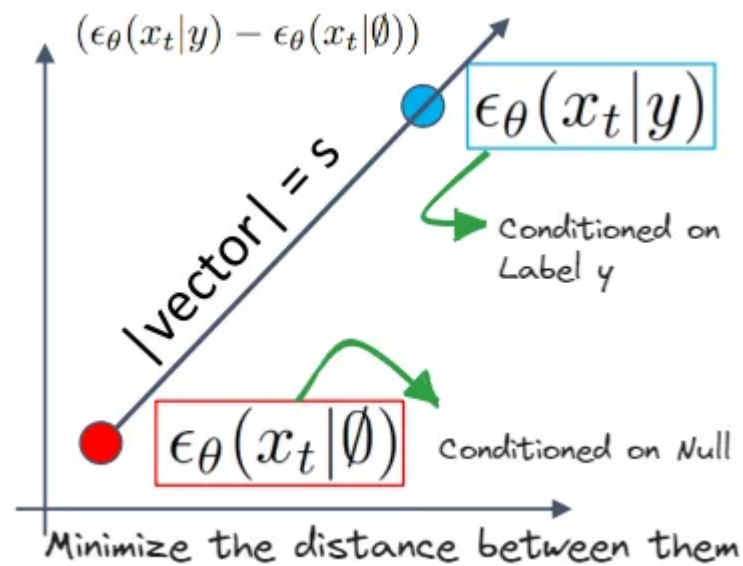
To guide of the diffusion process we perturb the reverse-process mean with the gradient of the dot product of the image and caption encodings with respect to the image:

$$\hat{\mu}_{\theta}(x_t|c) = \mu_{\theta}(x_t|c) + s \cdot \Sigma_{\theta}(x_t|c) \nabla_{x_t} (f(x_t) \cdot g(c))$$

The problem with using classifier based guidance is that the bigger generative model is based on a smaller classification model. The solution to this problem is classifier-free guidance which will be discussed in the next section.

## Classifier-Free Guidance

As the name suggests, classifier-free guidance does not use a separate classifier model for guiding the diffusion model. We will train a naive text conditional model but with one technique. We will sometimes condition the model on label  $y$  and sometimes we will omit the label and pass null  $\emptyset$ . During sampling, the output of the model is extrapolated further in the direction of  $\theta(x_t|y)$  and away from  $\theta(x_t|\emptyset)$ . Extrapolation will look something like this:



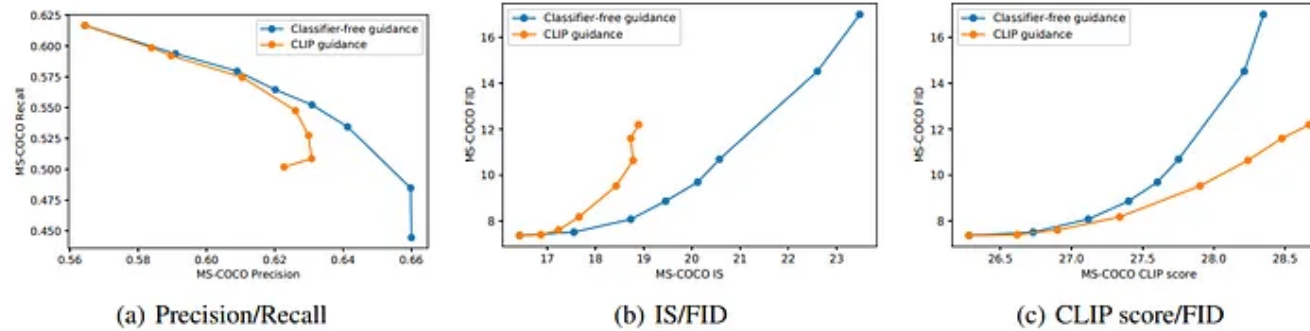
Mathematically the classifier-free guidance can be shown as:

$$\hat{\epsilon}_{\theta}(x_t|y) = \epsilon_{\theta}(x_t|\emptyset) + s \cdot (\epsilon_{\theta}(x_t|y) - \epsilon_{\theta}(x_t|\emptyset))$$

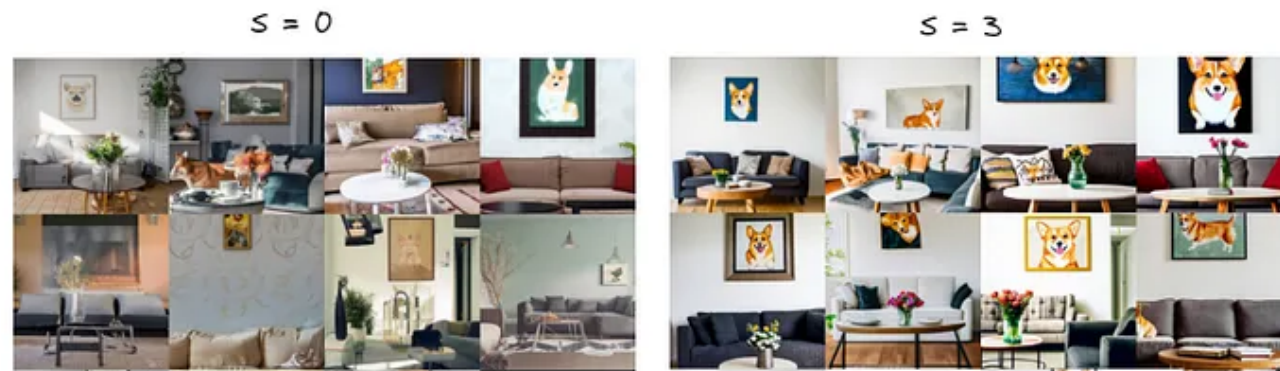
Classifier-free guidance has two appealing properties. First, it allows a single model to leverage its own knowledge during guidance, rather than relying on the knowledge of a separate (and sometimes smaller) classification model. Second, it simplifies guidance when conditioning on information that is difficult to predict with a classifier (such as text). Classifier-free guidance is preferred by human evaluators for both photorealism and caption similarity, and often produces photorealistic



samples. Comparison of classifier-free guidance and CLIP guidance is shown below with different evaluation metrics.



Let us see the effect of guidance scale  $s$  on the generations by GLIDE.



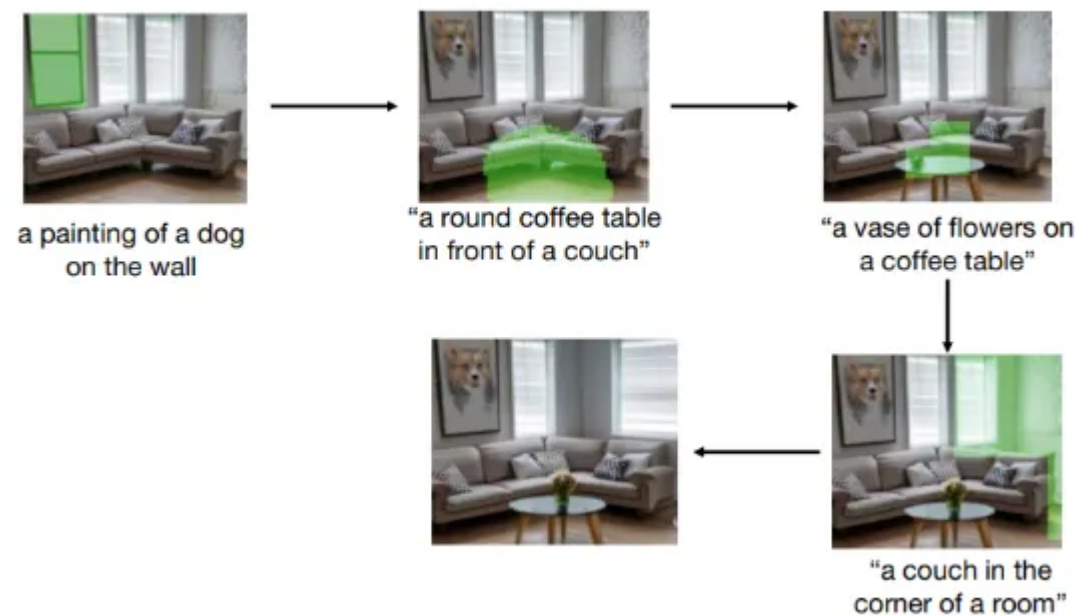
Images generated from GLIDE with prompt "A cozy living room with a painting of a corgi on the wall above a couch and a round coffee table in front of a couch and a vase of flowers on a coffee table" with different guidance scale.

We can observe from the above results that increasing the value of guidance scale results in better quality samples which fulfill all the details mentioned



in the prompt. While lower value of guidance scale results in diversity of images but quality is not the best.

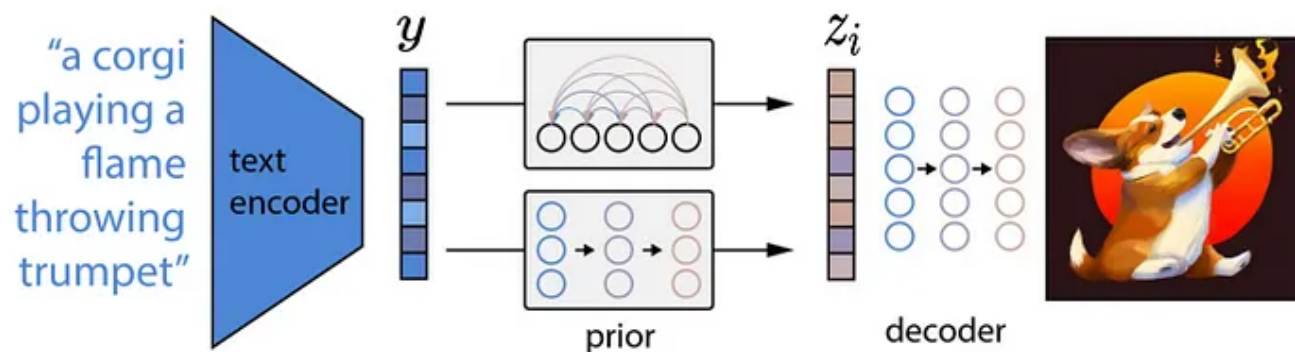
GLIDE is also capable of image inpainting. The model can generate very complex samples in multiple steps. To learn about the details of image inpainting, refer to the paper. An example of image inpainting is shown below:



Next, we cover another model from OpenAI i.e. DALL-E 2.

## DALL-E 2/UnCLIP

The “Hierarchical Text-Conditional Image Generation with CLIP Latents” paper was published in March, 2022. This paper introduced a new generative model DALL-E-2 or UnCLIP. In the Glide model, we found out that classifier-free guidance gives better results than CLIP based guidance. What if we want to use CLIP more effectively to improve generations? Contrastive models like CLIP have shown remarkable ability to learn robust representations of images that capture both semantics and style. This paper performed a number of experiments. They first tried using the text embeddings of CLIP text encoder and Glide’s decoder to generate images. This did not result in good results. Then an idea came up: what if we feed image embeddings to Glide’s decoder. The problem with this is that the text embeddings and image embeddings are not aligned. What this paper proposes is a way to use CLIP’s text embeddings and Glide’s decoder effectively by learning a Prior model which converts text embeddings to image embeddings. The architecture of DALL-E 2 is shown below:



The major parts of the above architecture are: Prior and Decoder. What prior does is that given CLIP text encoder output (text embedding)  $y$ , it generates corresponding image embedding  $z$ . The decoder produces the image from image embeddings  $z$ .

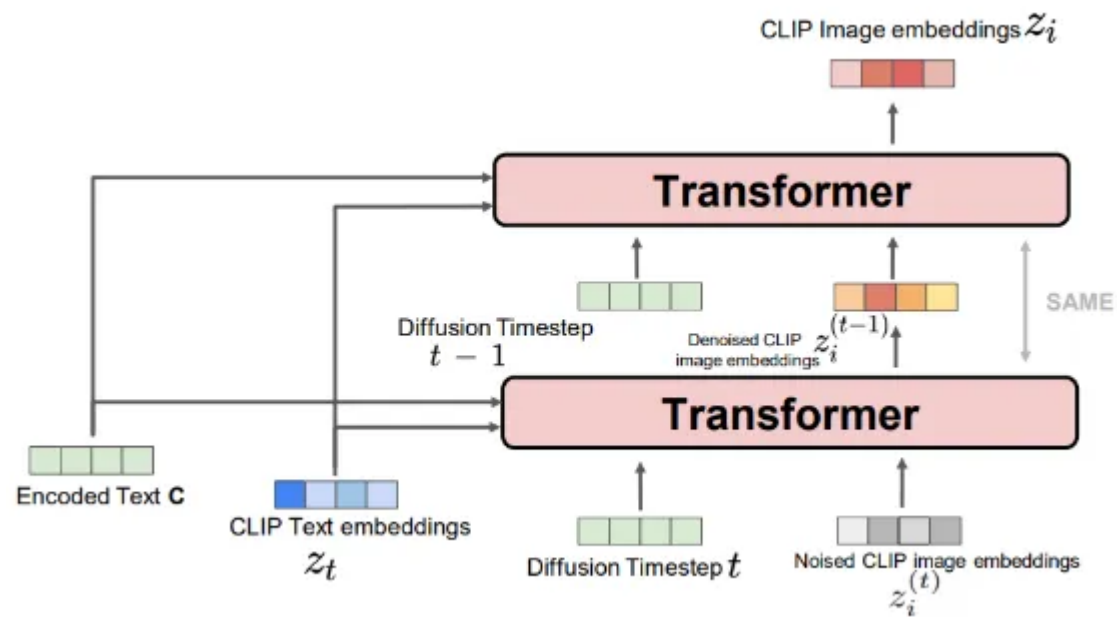
There are two different types of Prior architectures that can be used.

1. Autoregressive Prior
2. Diffusion Prior

As we already know, AR models predict a sequence of data on a previous data sequence. They use a transformer to predict the image embedding sequence from the text embedding sequence. The CLIP image embedding is converted into a sequence of discrete codes and predicted autoregressively conditioned on the caption  $y$ .

Diffusion Prior uses a diffusion model on CLIP image embedding. The input to this model is encoded text, CLIP text embedding, timestep, and noised CLIP image embedding. The continuous vector  $z$  is directly modelled using a Gaussian diffusion model conditioned on the caption  $y$ .

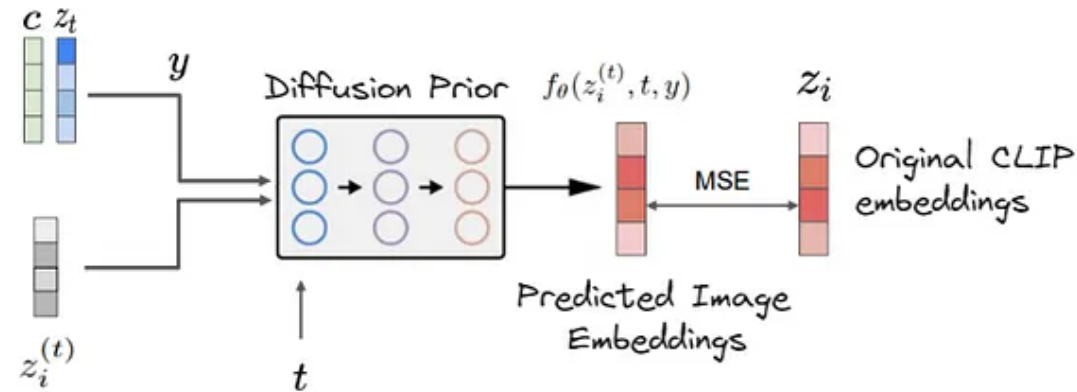
Using Diffusion model for Prior is more preferred and its architecture looks like:



Let us discuss how to train this diffusion prior model. We will use the original image embeddings of CLIP model as label. The image embeddings produced by the diffusion model are the predicted image embeddings. A simple MSE loss between the original CLIP embeddings and predicted prior embeddings is used as objective for prior.

## MSE LOSS

$$L_{\text{prior}} = \mathbb{E}_{t \sim [1, T], z_i^{(t)} \sim q_t} [\|f_{\theta}(z_i^{(t)}, t, y) - z_i\|^2]$$



Now we will cover the details of the decoder in UnCLIP. Decoder is used to output generated images by taking an input of Prior's image embeddings. The decoder is almost the same as GLIDE's decoder. GLIDE uses a transformer to embedding the input text while Dall-E-2 uses CLIP embedding in this process. After the image is generated by decoder, multiple upsamplers are used to generate higher-resolution images. No conditioning and no guidance is applied during this process. UNET architecture of UnCLIP is shown below. At each step, CLIP image embeddings are added.

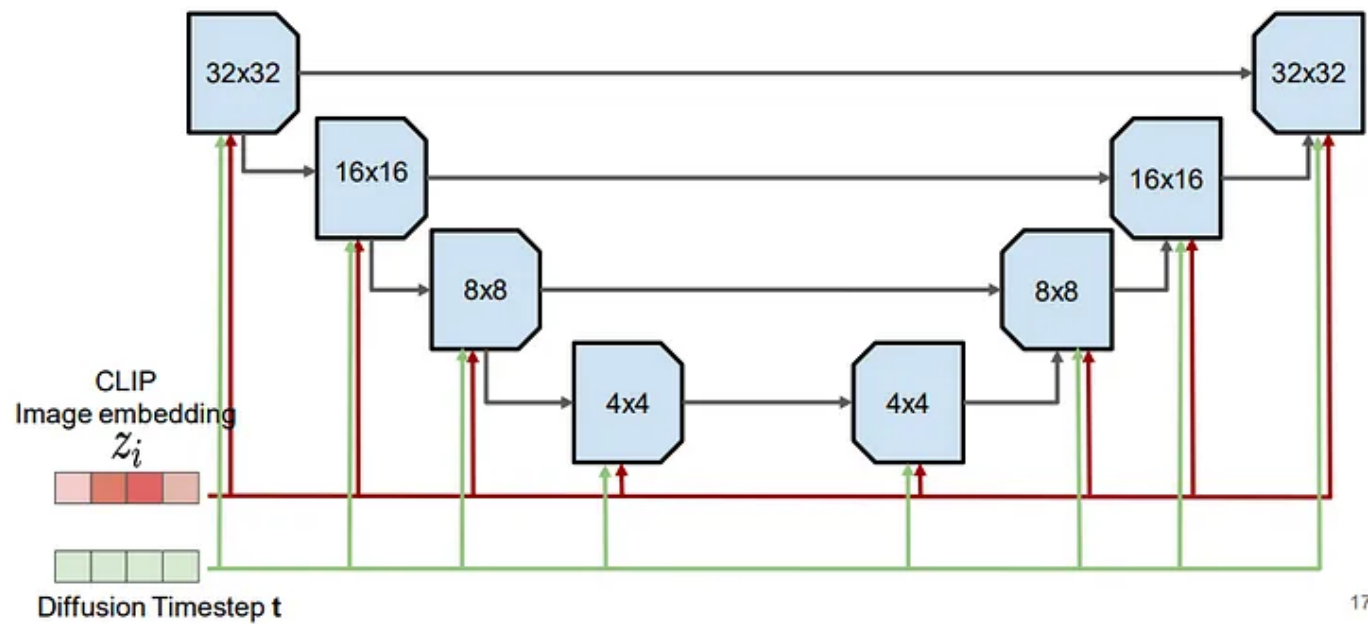


Search

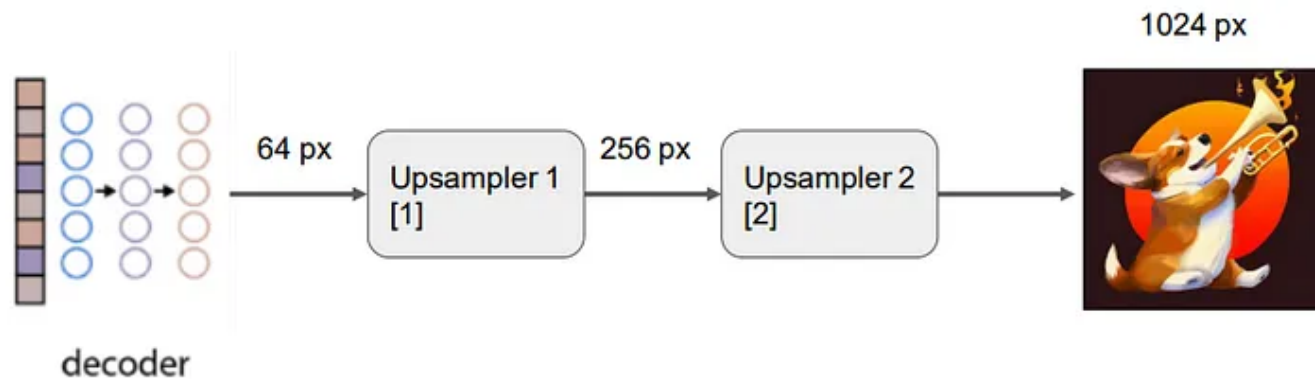
Write



R

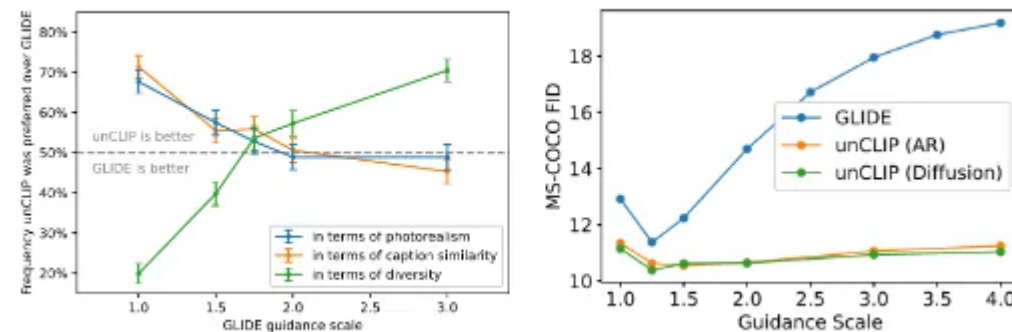


They use 2 unconditional off-the-shelf upsamplers to create images in higher resolution. The first upsampler takes in 64 pixels image and outputs 256 pixels image. The second upsampler takes 256 pixels image and outputs 1024 pixels image.



Lets see how to train this decoder model. Training the decoder requires the usage of the CLIP image encoder. We pass the original image to the CLIP image encoder which outputs image embeddings. These image embeddings are fed into the UnCLIP decoder to generate an image. The objective of decoder is a simple MSE loss between the original image and the generated decoder image. During training, the weights of the CLIP model are frozen.

Evaluation of this model reveals that UnCLIP generates more diverse images but GLIDE has better photorealism and caption similarity. Hence we can say that UnCLIP has better diversity and relatively good fidelity. unCLIP has limitations with attribute binding, text generation, and complex scenes. The paper later discusses various types of image manipulations like typographic attacks, interpolation and text diff which can be studied from the paper.





In conclusion, this article tries to explain the architectural details of Glide and DALL-E 2. These models act as the stepping stone into the field of



generative computer vision. For implementation details, view [Github](#)

**Thank you for reading!**

*Also, feel free to drop me a message or:*

1. Connect and reach me on [LinkedIn](#) and [Twitter](#)
2. Follow me on  [Medium](#)
3. Subscribe to my  weekly [AI newsletter!](#)

AI

ML

NLP

Cv

Dalle



## Written by Zain ul Abideen

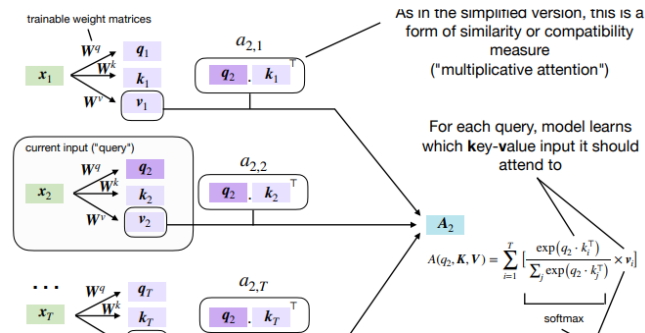
356 Followers

Machine Learning Engineer | I share what I learn.

<https://www.linkedin.com/in/zaiinulabideen/>

Follow

### More from Zain ul Abideen

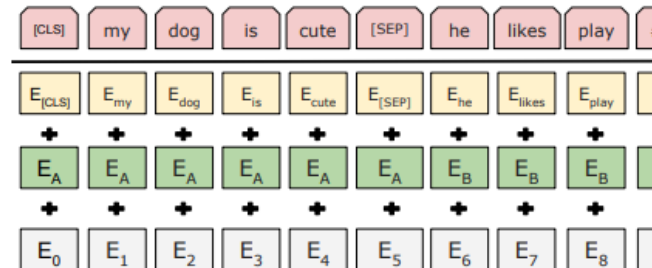


Zain ul Abideen

## Attention Is All You Need: The Core Idea of the Transformer

An overview of the Transformer model and its key components.

6 min read · Jun 26, 2023



Zain ul Abideen

## A Comparative Analysis of LLMs like BERT, BART, and T5

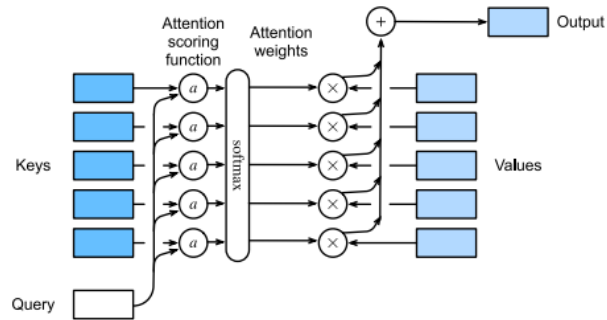
Exploring Language Models

6 min read · Jun 26, 2023

👏 235



👏 44



Zain ul Abideen

## From Seq2Seq to Attention: Revolutionizing Sequence...

Investigating the origin of Attention mechanism and Bahdanau attention

6 min read · Jun 26, 2023

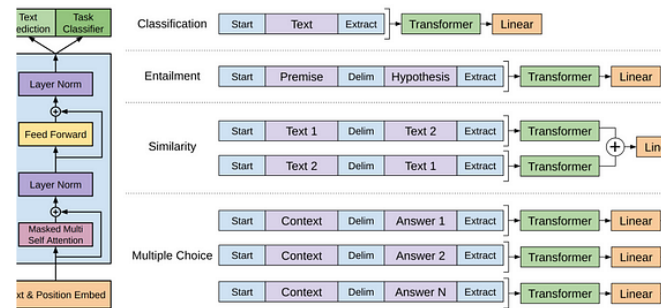
👏 42



👏 29



See all from Zain ul Abideen



Zain ul Abideen

## Autoregressive Models for Natural Language Processing

The Evolution of GPT: From GPT to GPT-2 to GPT-3

7 min read · Jun 26, 2023

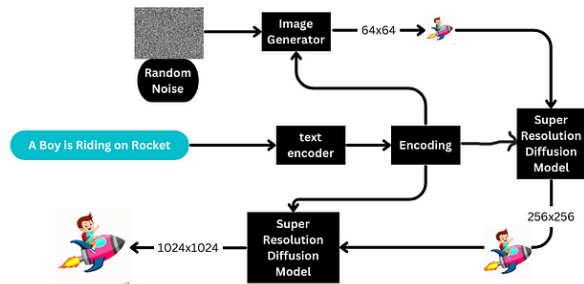
👏 42




👏 29



## Recommended from Medium



 Shyam Patel

### Introduction to Diffusion Models and IMAGEN: The Magic Behind...

Introduction to Diffusion Model

4 min read · Sep 27, 2023

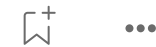


 Mario Namtao Shianti Lar... in Towards Data Scie...

### Paper Explained — High-Resolution Image Synthesis with Latent...

While OpenAI has dominated the field of natural language processing with their...

★ · 10 min read · Mar 31, 2023



## Lists



**The New Chatbots: ChatGPT, Bard, and Beyond**



**Natural Language Processing**

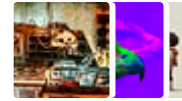
1117 stories · 589 saves

12 stories · 276 saves



## Generative AI Recommended Reading

52 stories · 641 saves



## What is ChatGPT?

9 stories · 278 saves



 Tapan Patel

## Top 5 Most Important Generative AI Trends in 2024

A major leap forward in 2024. The groundbreaking technology, capable of...

6 min read · Dec 11, 2023



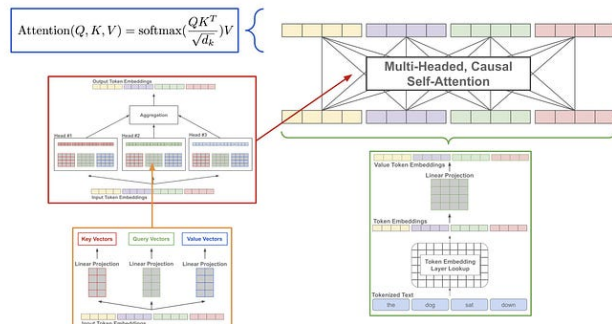
 Zain ul Abid... in Artificial Intelligence in Plain En...

## Complete Roadmap For Learning Diffusion Models (Prereqs, DDPM...

Everything you need to get started with Diffusion Models!

16 min read · Sep 26, 2023





Akash Kesrwani

## Multi-Head Self Attention: Short Understanding

Each “block” of a large language model (LLM) is comprised of self-attention and a feed-...

3 min read · Sep 8, 2023



51



1



...



Aguimar Neto

## What is Latent Diffusion in AI?

Latent diffusion models are deep learning models that have recently emerged as a...

5 min read · Oct 7, 2023



2



...

See more recommendations