DEEP LEARNING

How DALL-E 2 Actually Works

How does OpenAI's groundbreaking DALL-E 2 model actually work? Check out this detailed guide to learn the ins and outs of DALL-E 2.

Ryan O'Connor
Developer Educator at AssemblyAI

Apr 19, 2022

OpenAI's groundbreaking model DALL-E 2 hit the scene at the beginning of the month, setting a new bar for image generation and manipulation. With only a short text prompt, DALL-E 2 can **generate completely new images** that combine distinct and unrelated objects in semantically plausible ways, like the images below which were generated by entering the prompt **"a bowl of**

Table of contents

Various images generated by DALL-E 2 given the above prompt (source).

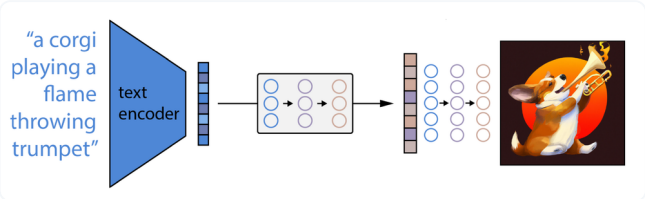
DALL-E 2 can even modify existing images, create variations of images that maintain their salient features, and interpolate between two input images. DALL-E 2's impressive results have many wondering exactly how such a powerful model works under the hood.

In this article, **we will take an in-depth look at how DALL-E 2 manages to create such astounding images** like those above. Plenty of background information will be given and the explanation levels will run the gamut,

experience. Let's dive in!

How DALL-E 2 Works: A Bird's-Eye View

Before diving into the details of how DALL-E 2 works, let's orient ourselves with a high-level overview of how DALL-E 2 generates images. While DALL-E 2 can perform a variety of tasks, including image manipulation and interpolation as mentioned above, **we will focus on the task of image generation** in this article.



A birds-eye view of the DALL-E 2 image generation process (modified from source).

At the highest level, DALL-E 2's works very simply:

1. First, a text prompt is input into a **text encoder** that is trained to

2. Next, a model called the **prior** maps the text encoding to a corresponding **image encoding** that captures the semantic information of the prompt contained in the text encoding.
3. Finally, an **image decoder** stochastically generates an image which is a visual manifestation of this semantic information.

From a bird's eye-view, that's all there is to it! Of course, there are plenty of interesting implementation specifics to discuss, which we will get into below. If you want a bit more detail without getting into the nitty-gritty, or you prefer to watch your content rather than read it, feel free to check out our video breakdown of DALL-E 2 here:

How DALL-E 2 Works: A Detailed Look

Now it's time to dive into each of the above steps separately. Let's get started by looking at how DALL-E 2 learns to link related textual and visual abstractions.

Step 1 - Linking Textual and Visual Semantics

After inputting **"a teddy bear riding a skateboard in Times Square"**, DALL-E

source

How does DALL-E 2 know how a textual concept like "teddy bear" is manifested in the visual space? The **link between textual semantics and their visual representations** in DALL-E 2 is learned by another OpenAI model called **CLIP** (**C**ontrastive **L**anguage-**I**mage **P**re-training).

CLIP is trained on hundreds of millions of images and their associated captions, learning *how much* a given text snippet relates to an image. That is, rather than trying to *predict* a caption given an image, CLIP instead

contrastive rather than **predictive** objective allows CLIP to learn the link between textual and visual representations of the same abstract object. The entire DALL-E 2 model hinges on CLIP's ability to learn semantics from natural language, so let's take a look at how CLIP is trained to understand its inner workings.

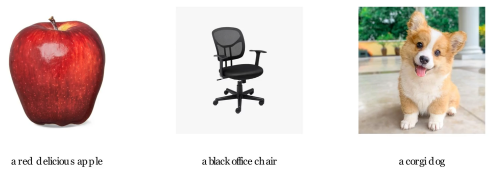
CLIP Training

The fundamental principles of training CLIP are quite simple:

1. First, all images and their associated captions are passed through their respective encoders, mapping all objects into an *m*-dimensional space.
2. Then, the cosine similarity of each *(image, text)* pair is computed.
3. The training objective is to simultaneously **maximize the cosine similarity** between N **correct** encoded image/caption pairs and **minimize the cosine**

pairs.

This training process is visualized below:



Overview of the CLIP training process

Additional Training Details

Significance of CLIP to DALL-E 2

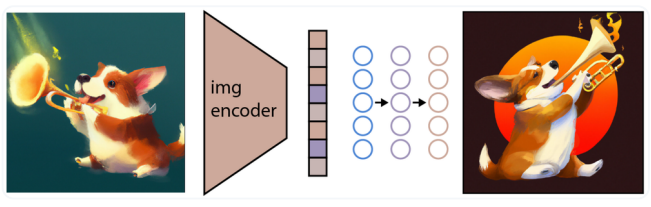
CLIP is important to DALL-E 2 because **it is what ultimately determines how semantically-related** a natural language snippet is to a visual concept, which is critical for *text-conditional* image generation.

Additional Information

from Visual Semantics

After training, the CLIP model is frozen and DALL-E 2 moves onto its next task - learning to *reverse* the image encoding mapping that CLIP just learned. CLIP learns a representation space in which it is easy to determine the relatedness of textual and visual encodings, but our interest is in image **generation**. We must therefore learn how to exploit the representation space to accomplish this task.

In particular, OpenAI employs a modified version of another one of its previous models, GLIDE, to perform this image generation. The GLIDE model learns to *invert* the image encoding process in order to stochastically decode CLIP image embeddings.



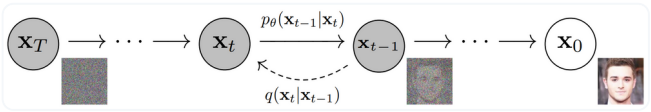
An image of a Corgi playing a flamethrowing trumpet passed through CLIP's image encoder. GLIDE then

As depicted in the image above, it should be noted that the goal is **not** to build an autoencoder and *exactly* reconstruct an image given its embedding, but to instead generate an image which **maintains the salient features of the original image** given its embedding. In order perform this image generation, GLIDE uses a **Diffusion Model**.

What is a Diffusion Model?

Diffusion Models are a thermodynamics-inspired invention that have significantly grown in popularity in recent years[1][2]. Diffusion Models learn to generate data by *reversing a gradual noising process*. Depicted in the figure below, the noising process is viewed as a parameterized Markov chain that gradually adds noise to an image to corrupt it, eventually (asymptotically) resulting in pure Gaussian noise. The Diffusion Model learns to navigate

timesteps to reverse this process.



Diffusion Model schematic (source).

If the Diffusion Model is then "cut in half" after training, it can be used to *generate* an image by randomly sampling Gaussian noise and then de-noising it to generate a photorealistic image. Some may recognize that this technique is highly reminiscent of generating data with Autoencoders, and Diffusion Models and Autoencoders are, in fact, related.

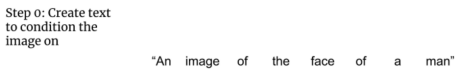
Want to learn more about Diffusion Models?

Check out our *Introduction to Diffusion Models for Machine Learning* article!

GLIDE Training

While GLIDE was not the first Diffusion Model, its important contribution was in modifying them to allow for **text-conditional image generation**. In particular, one will notice that Diffusion Models *start* from randomly sampled Gaussian noise. It at first unclear how to tailor this process to generate *specific* images. If a Diffusion Model is trained on a human face dataset, it will reliably generate photorealistic images of human faces; but what if someone wants to generate a face with a *specific* feature, like brown eyes or blonde hair?

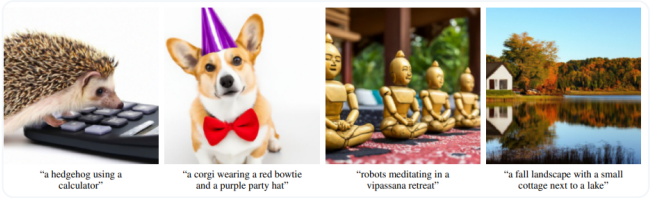
GLIDE extends the core concept of Diffusion Models by **augmenting the training process with additional textual information**, ultimately resulting in text-conditional image generation. Let's take a look at the training process for GLIDE:



GLIDE training process.

Additional Training Details

Here are some examples of images generated with GLIDE. The authors note that GLIDE performs better than DALL-E (1) for photorealism and caption similarity.



Examples of images generated by GLIDE (source).

DALL-E 2 uses a modified GLIDE model that incorporates projected CLIP text embeddings in two ways. The first way is by adding the CLIP text embeddings to GLIDE's existing timestep embedding, and the second way is by creating four extra tokens of

encoder.

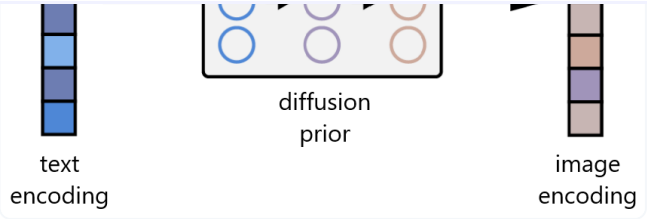
Significance of GLIDE to DALL-E 2

GLIDE is important to DALL-E 2 because it allowed the authors to easily port over GLIDE's text-conditional photorealistic image generation capabilities to DALL-E 2 by instead conditioning on **image encodings** in the representation space. Therefore, DALL-E 2's modified GLIDE learns to **generate semantically consistent images conditioned on CLIP image encodings**. It is also important to note that the reverse-Diffusion process is stochastic, and therefore variations can easily be generated by inputting the *same* image encoding vectors through the modified GLIDE model multiple times.

Step 3 - Mapping from Textual Semantics to Corresponding Visual Semantics

reflect the semantics captured by image encodings, how do we go about actually go about *finding* these encoded representations? In other words, how do we go about injecting the text conditioning information from our prompt into the image generation process?

Recall that, in addition to our *image* encoder, CLIP also learns a *text* encoder. DALL-E 2 uses another model, which the authors call the **prior**, in order to map **from the text encodings** of image captions **to the image encodings** of their corresponding images. The DALL-E 2 authors experiment with both Autoregressive Models and Diffusion Models for the prior, but ultimately find that they yield comparable performance. Given that the Diffusion Model is much more computationally efficient, it is selected as the prior for DALL-E 2.



Prior mapping from a text encoding to its corresponding image encoding (modified from source).

Prior Training

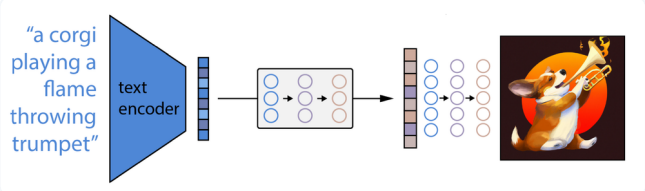
The Diffusion Prior in DALL-E 2 consists of a decoder-only Transformer. It operates, with a causal attention mask, on an ordered sequence of

1. The tokenized text/caption.
2. The CLIP text encodings of these tokens.
3. An encoding for the diffusion timestep.
4. The noised image passed through the CLIP image encoder.
5. Final encoding whose output from Transformer is used to predict the unnoised CLIP image encoding.

Step 4 - Putting It All Together

At this point, we have all of DALL-E 2's functional components and need only to chain them together for text-conditional image generation:

1. First the CLIP text encoder maps the image description into the **representation space**.
2. Then the diffusion prior maps from the CLIP text encoding to a **corresponding CLIP image encoding**.
3. Finally, the modified-GLIDE generation model maps from the representation space into the image space via reverse-Diffusion, **generating one of many possible images that conveys the semantic information** within the input caption.



Summary

In this article we covered how the world's premier textually-conditioned image generation model works under the hood. DALL-E 2 can generate semantically plausible photorealistic images given a text prompt, can produce images with specific artistic styles, can produce variations of the same salient features represented in different ways, and can modify existing images.

While there is a lot of discussion to be had about DALL-E 2 and its importance to both Deep Learning and the world at large, we draw your

1. First, DALL-E 2 demonstrates the **power of Diffusion Models** in Deep Learning, with both the prior *and* image generation sub-models in DALL-E 2 being Diffusion-based. While only rising to popular use in the past few years, Diffusion Models have already proven their worth, and those tuned-in to Deep Learning research should expect to see more of them in the future.

2. The second point is to highlight both the need and **power of using natural language as a means to train State-of-the-Art Deep Learning models**. This point does not originate with DALL-E 2 (in particular, CLIP demonstrated it previously), but nevertheless it is important to appreciate that the power of DALL-E 2 stems ultimately from the absolutely *massive* amount of paired natural language/image data that is

developmental bottleneck associated with the laborious and painstaking process of manually labelling datasets; but the noisy, uncurated nature of such data better reflects real-world data that Deep Learning models must be robust to.

3. Finally, DALL-E 2 **reaffirms the position of Transformers** as supreme for models trained on web-scale datasets given their impressive parallelizability.

References

1. Deep Unsupervised Learning using Nonequilibrium Thermodynamics
2. Generative Modeling by Estimating Gradients of the Data Distribution
3. Hierarchical Text-Conditional Image Generation with CLIP Latents

5. Denoising Diffusion Probabilistic Models
6. Learning Transferable Visual Models From Natural Language Supervision
7. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Popular posts

Aug 3, 2023

How RLHF Preferenc Model Tuning Works (And How

Aug 1, 2023

How Reinforcer Learning from AI Feedback works

Jun 27, 2023

Recent developm in Generativ AI for Audio

May 17, 2023

Introducti to Large Language Models for Generativ AI

Wrong)


Marco Ramponi
Developer Educator at AssemblyAI


Ryan O'Connor
Developer Educator at AssemblyAI


Marco Ramponi
Developer Educator at AssemblyAI


Ryan O'Connor
Developer Educator at AssemblyAI