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

Sep 29, 2023

DALL-E 3

OpenAI has recently announced DALL-E 3, the successor to DALL-E 2. For information on what DALL-E 3 is, how it works, and the differences between DALL-E 3 and DALL-E 2, jump down to this section.

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 soup that is a portal to another dimension as digital art"**.

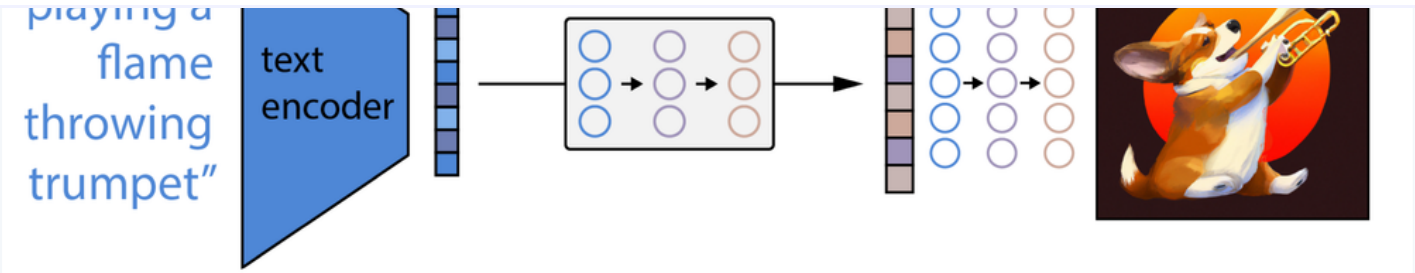
Various images generated by DALL-E 2 given the above prompt (source).

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, so this article is suitable for readers at several levels of Machine Learning 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 map the prompt to a representation space.
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

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

king at how DALL-E 2 learns to link related textual and visual abstractions.

After inputting **"a teddy bear riding a skateboard in Times Square"**, DALL-E 2 outputs the following image:

source

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 just learns how *related* any given caption is to an image. This **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.

similarity between N^2 - N **incorrect** encoded image/caption pairs.

This training process is visualized below:

Overview of the CLIP training process

Additional Training Details

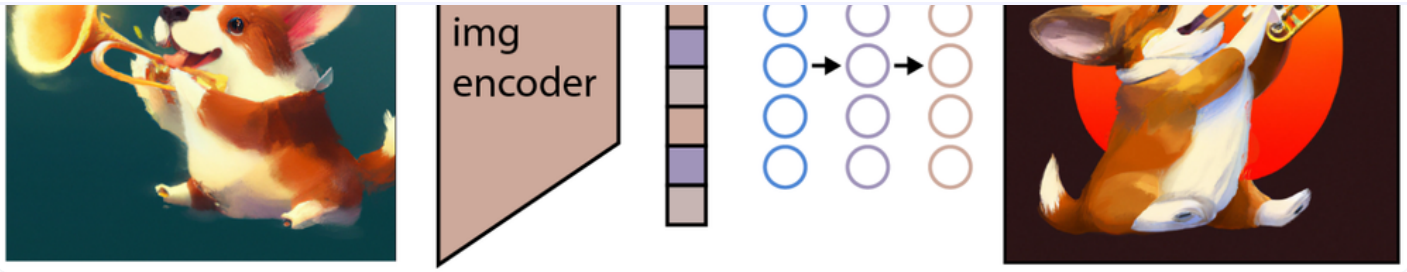
CLIP is important to DALL-E 2 because **it is what ultimately determines how semantically-related** a natural language snippet is to a visual concept, which is critical for *text-conditional* image generation.

Additional Information

Step 2 - Generating Images 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.

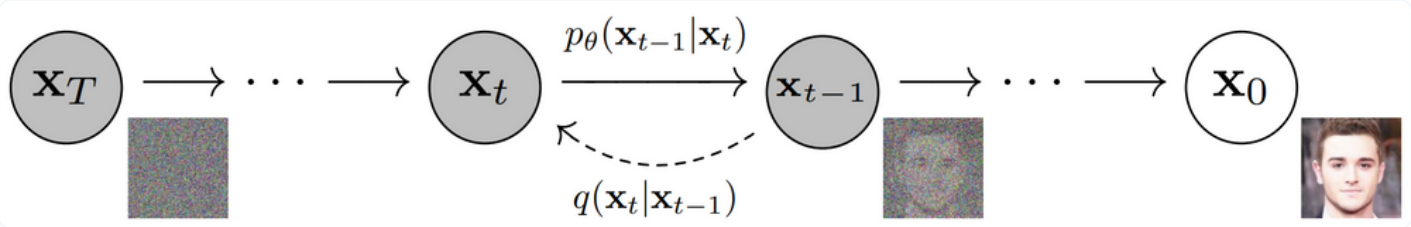


An image of a Corgi playing a flamethrowing trumpet passed through CLIP's image encoder. GLIDE then uses this encoding to generate a new image that maintains the salient features of the original. (modified from source)

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 ge to corrupt it, eventually (asymptotically) resulting in pure Gaussian noise. The



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.

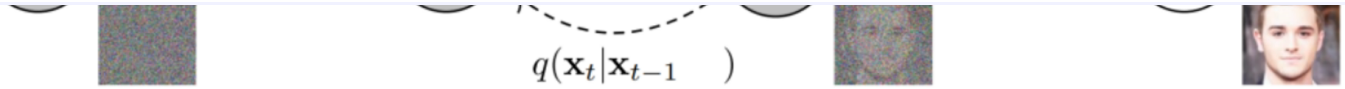
Want to learn more about Diffusion Models?

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

Check it Out

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:



Adding textual conditioning to the diffusion process

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 context, which are concatenated to the output sequence of the GLIDE text encoder.

Significance of GLIDE to DALL-E 2

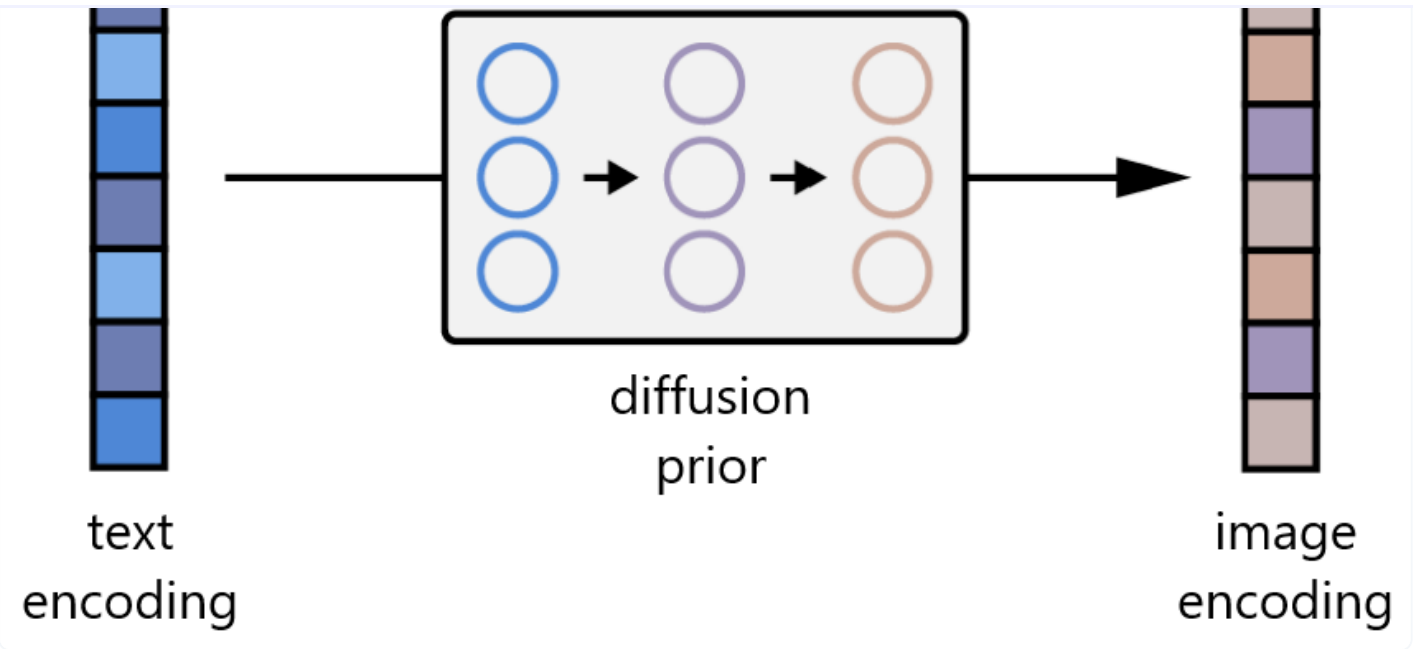
GLIDE is important to DALL-E 2 because it allowed the authors to easily port over GLIDE's text-conditional photorealistic image generation capabilities to DALL-E 2 by instead conditioning on **image encodings** in the representation space. Therefore, DALL-E 2's modified GLIDE learns to **generate semantically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

Table of contents

Step 3 - Mapping from Textual Semantics to Corresponding Visual Semantics

While the modified-GLIDE model successfully generates images that 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.

5. Final encoding whose output from Transformer is used to predict the unnoised CLIP image encoding.

Additional Training Details

More information about the Prior training process can be found below.

- **Conditioning on the Caption**
  - The Diffusion Prior is conditioned not only on the CLIP text embedding of the caption, but also the caption itself. The former is a deterministic function of the latter and this dual-conditioning is therefore fully permissible.
- **Classifier-Free Guidance**
  - To improve sample quality, sampling is randomly conducted using classifier-free guidance 10% of the time by dropping the text-conditioning information.
- **Double Sample Generation**
  - To improve quality during sampling time, two image embeddings are generated with the prior and the one with the higher dot product with

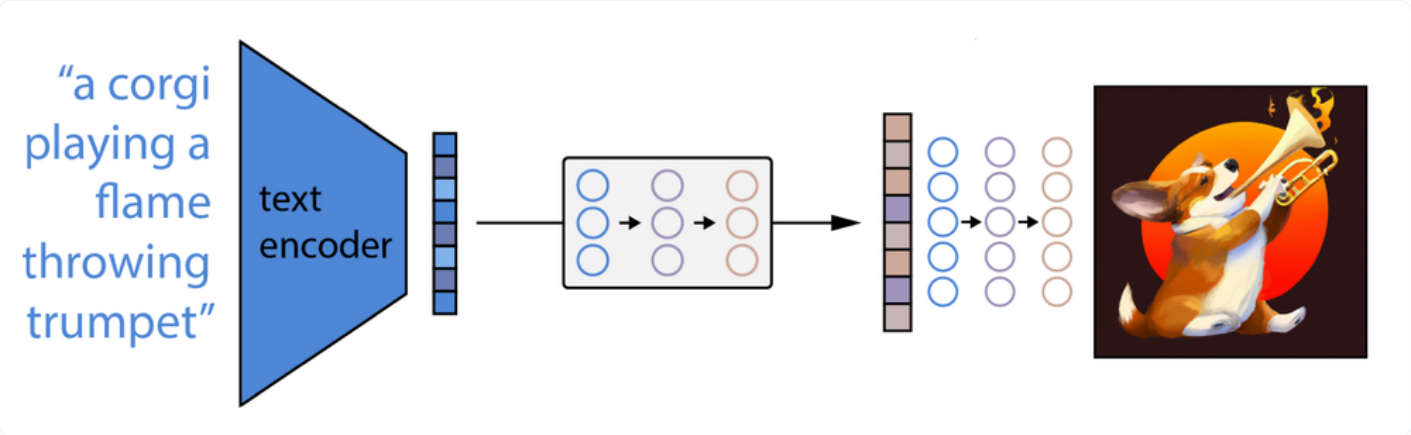
- **Why do we need the prior?**
  - The authors note that training such a prior is not strictly necessary for a caption-to-image model. One option would be to condition only on the caption itself. This would simply yield the model GLIDE, and the authors perform a thorough analysis comparing the two in the paper. Another option would be to feed into the decoder the CLIP text embedding, rather than using the prior to generate a CLIP image embedding from it and then use that. The authors found experimentally that the former produces reasonable results, although results not as good as those of the latter. Ultimately, using the prior **improves image diversity**.

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**.

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.



High-level overview of the DALL-E 2 image-generation process (modified from source).

Liking this article?

Follow our newsletter for more content like this!

What is DALL-E 3?

DALL-E 3, announced in September of 2023, is the successor to DALL-E 2. DALL-E 3 promises “significantly more nuance and detail” relative to DALL-E 2 and other previous systems.

In particular, the model seems to have a focus on capturing prompt semantics. Controlling text to image models is a difficult task, and they often may not convey visually specific concepts or details provided in the prompt. As a result, the concept of **prompt engineering** came to be, which is the study and practice of developing prompts specifically to drive tailored outputs of text-to-image models.

DALL-E 3 promises a “leap forward” on the ability to generate images that exactly adhere to provided text.

DALL-E 3 will be available to ChatGPT plus subscribers and Enterprise customers in October of 2023.

How does DALL-E 3 work?

partial information about how DALL-E 3 works based on the recent trends in text-to-image models and the field of AI more widely.

First, we can be almost assured that the actual image generation component of DALL-E 3 is a Diffusion Model. Diffusion Models are the preeminent paradigm in image generation, and it is highly unlikely that another paradigm has been developed in isolation from the wider research community for the release of DALL-E 3. Diffusion Models are physics-inspired models and, while other similar models like PFGMs are being developed, Diffusion Models are the clear frontrunner for image generation currently. DALL-E 3 may incorporate latent diffusion, seen in Stable Diffusion, and may perhaps even perform diffusion both in the latent and pixel spaces.

Next is the question of how the image generation model is conditioned on the textual prompt. That is, how does the image generation process know what to generate

Shortly after the release of DALL-E 2, another text-to-image model Imagen was released, which showed increased performance relative to DALL-E 2. The important change Imagen made was using a Large Language Model to encode the prompt, and this encoding was then **directly** used to condition the image generation process (unlike DALL-E 2, which leverages the "prior" model). The key insight of Imagen, therefore, was that LLMs, by virtue of their sheer size alone, generate representations powerful enough to beat smaller encoders purpose-built for text-image tasks. Given the very public progress of LLMs in the past year, we can be almost assured that DALL-E 3 makes more direct use of LLM encodings.

How does DALL-E 3 work with ChatGPT?

One of the flagship features of DALL-E 3 is its tight integration with ChatGPT. In fact, OpenAI says that "DALL-E 3 is built **on** ChatGPT" (emphasis added), not **into** ChatGPT - perhaps this phrasing betrays a deeper connection.

DALL-E 3 is tightly integrated into ChatGPT

LLM. To integrate DALL-E 3 with ChatGPT so seamlessly, OpenAI may have extended this process to include images.

For example, imagine a detective who describes a crime and then asks ChatGPT to "paint me a picture of a man who would commit such a crime". What is the detective asking the model to do? Is he asking the model to *metaphorically* paint a picture of such a man (e.g. his backstory, his motives, etc.), or is he asking the model to *literally* paint a picture of such a man (i.e. a literal picture of his face, perhaps using details from witness reports).

A priori, the model has no way to know, and in fact humans will interpret this request differently. Since the ChatGPT web UI now supports the output of different modalities, there may be an element of RLHF incorporated into the training to deal with this situation. For example, the prompt above may be supplied to the model and it may be asked to output a few responses, allowing them to be images or text. Humans are then used to generate preference data, e.g. by ranked preference modeling.

Then, during inference, after ChatGPT is given a prompt, there may be an intermediate step that is hidden from the user which determines whether the model should output text or an image (a kind of chain-of-thought prompting). Once

image case. ChatGPT may also incorporate lessons from Toolformer in order to decide when to generate images, which it might already be doing to support its plugins. Additionally, there may be some clever token reuse that lowers compute requirements.

DALL-E 3 vs DALL-E 2

While DALL-E 2 was released only a year ago, DALL-E 3 has had the benefit of the fruits of that *particular* year. That is, the past year has seen AI moving forward at a breakneck pace and substantial progress has been made in text-to-image models, Large Language Models, Graph Neural Networks, audio generation models, and more.

Prompt: An expressive oil painting of a basketball player dunking, depicted as an explosion of a nebula.

DALL-E 3 surely makes use of a more sophisticated diffusion process, learning from the releases of Imagen and Stable Diffusion. In addition, the revelation of RLHF as a method crucial to seamless human-AI interaction gives DALL-E 3 the advantage over DALL-E 2. Perhaps DALL-E 3 makes use of even more cutting edge guidance techniques, like a modified version of Reinforcement Learning from AI Feedback or Constitutional AI more widely.

RLHF vs RLAIF

Check it out

If such techniques are used, a big question that may differentiate DALL-E 3 from DALL-E 2 is whether or not RLHF was somehow incorporated into the image generation model itself. In particular, RLHF uses Proximal Policy Optimization to actually train with Reinforcement Learning, and it is at first unclear how to integrate this process into current image generation techniques, although recent research may be making ground on this front.

Overall, DALL-E 3 appears to be an improvement over DALL-E 2 along every evaluation axis of note. DALL-E 3's tight integration with the ChatGPT web UI makes it significantly easier and more intuitive to use, and it will likely see widespread adoption because of this integration. After the release of DALL-E 3, there will (likely) be no reason to use DALL-E 2, unlike in the case of the release of Stable Diffusion 2, after which some users elected to continue using Stable Diffusion 1.5.

Summary

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 attention to 3 **key takeaways** from the development of DALL-E 2

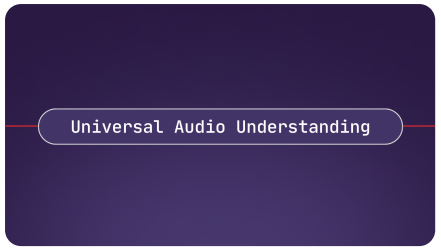
1. First, DALL-E 2 demonstrates the **power of** Diffusion Models in Deep Learning, with both the prior *and* image generation sub-models in DALL-E 2 being Diffusion-based. While only rising to popular use in the past few years, Diffusion Models have already proven their worth, and those tuned-in to 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 available on the internet. Using such data not only removes the developmental bottleneck associated with the laborious and

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
4. Diffusion Models Beat GANs on Image Synthesis
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



Dec 7, 2023

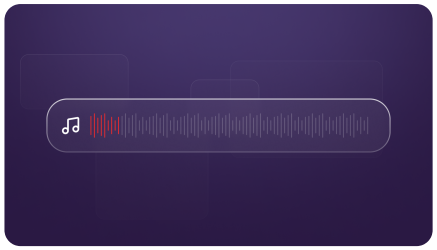
AI for Universal Audio Understanding: Qwen-Audio Explained

Marco Ramponi
Developer Educator at AssemblyAI

Oct 27, 2023

Combining Speech Recognition and Diarization in one model

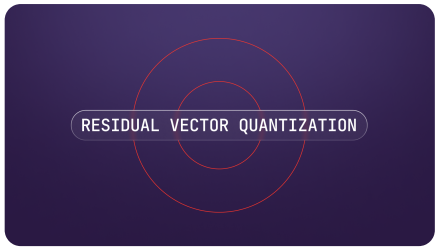
Marco Ramponi
Developer Educator at AssemblyAI



Sep 22, 2023

What AI Music Generators Can Do (And How They Do It)

Marco Ramponi
Developer Educator at AssemblyAI



Sep 4, 2023

What is Residual Vector Quantization?

Marco Ramponi
Developer Educator at AssemblyAI

Pricing

Use Cases

Telephony Services

Media Companies

Video Platforms

Remote Organisations

Industry News

Deep Learning

Engineering

FAQs

Contact Us

Terms of Service

Privacy Policy

Subprocessors