

[SQUEAKING]

[RUSTLING]

[CLICKING]

SAYAK PAUL: Thanks for having me here today. And also, thanks for accepting my request for presenting on this topic. So just for some context, sort of shamelessly reintroducing myself, I work as a research engineer at Hugging Face. I try to focus on different facets of diffusion models for image and video generation.

Part of my time at Hugging Face is focused on maintaining and growing the Diffusers Library, but this talk is not going to be about the Diffusers Library. And the rest of my time at Hugging Face is also spent around researching diffusion models from many different perspectives. And I'm going to be covering some of that in this talk.

As this talk suggests, it's going to be about optimizing the full stack for image and video generation models. In particular, we will see how the optimization aspects of these models are a little non-trivial and also kind of atypical in nature. And also, some perspectives into how the whole optimization landscape can transcend beyond latency and speed. So with that, I'll start.

I'll try to also give you a very hand-wavy introduction to diffusion models because diffusion model itself will take another lecture to cover. So I'll try to give you just enough context so that we can step through the rest of the discussion for this talk. And hopefully by the end of this talk, there will be time for some Q&A. In case there isn't, you can always reach out to me via email and we can always discuss things offline.

I want to get the excitement rolling by giving you some outputs from some of the recent text-to-image models. This is from PixArt-Alpha. This is from back in the days in 2024-- I think 2023, not even 2024. So this is quite old, but you can see, the quality is not bad.

This is from DALL-E 3, OpenAI. And this is from 2024 September, if I remember correctly. This is from Flux. And this example is by far my most favorite because even imagining a tiny astronaut, let alone trying to imagine it hatching from an egg, is wild. It's simply wild, but somehow, these models are expressive enough to be able to come up with something as real as this. So yeah.

There's an ongoing emergence around text-to-video models and the whole moniker around wire models as well. This is one of those. And then we have got this poor cat. And then you have got very cinematic light, landscape frames. All of these are open models except for DALL-E 3 that I've shown in the previous slide.

As a cautionary note, I'm going to be interchangeably using diffusion and flow in this talk, which also means that the points and the discussions that we'll do in this talk, they will apply to both diffusion and flow models. Flow model, as I like to explain them, is a generalization of diffusion models, but they are not exactly the same, but for the purposes of this talk, I'll interchangeably use them.

Now let me try to give you some context as to how diffusion models work. I like to think of diffusion models as the following. What happens when we start from a random noise drawn from a Gaussian? And we then try to slowly denoise it over a period of time unless we get something realistic. It can be any image, it can be a video, or it can be even a piece of audio as well.

But the point is-- the crucial point is we are starting from a pure random noise, and then we are iterating over a period of time, denoising it so that it becomes a clean and realistic representation of what we want. So it can be considered as some of an iterative denoising process.

And we can also condition this denoising process. When we try to condition with text descriptions, we can do tasks like text to image like we are seeing here.

Now the thing is, diffusion models can have different variants. Like when we are operating directly on the pixel space, that family of model is typically referred to as pixel space diffusion model, but pixel space diffusion is pretty intensive from both memory and compute standpoints.

So which is exactly why operating on the latent space is almost the de facto for diffusion models. And in this diagram, we can get a sense of how latent space diffusion works. Now we need to have some of an encoder and a decoder in order to be able to encode the pixel representation of an image into its latent representation. And then when we are converting that latent space into the pixel space, we need to have some decoder.

Typically for the case of latent space diffusion models, we usually use a VAE, which has an encoder variant, as well as decoder variant. So, yeah. Just note that this talk is mostly going to be about latent space diffusion models, but the approaches are fairly general enough, they will also apply equally-- they should apply equally to pixel space diffusion models as well.

Now let's try to also see how different components within a diffusion model are connected with one another. Because unlike language models or vision language models, diffusion-- any modern state-of-the-art diffusion model is not a single model. That's a very important distinction to be aware of.

Now let's say we want to take this text prompt and we want to generate a video out of it. Now let's see what are the typical components that are involved and how they are connected.

Now to be able to have salient representations of the textual description, we need to have text encoder, which is fairly intuitive. And then once we have the text embeddings out, we start the-- this is the inference workflow, by the way. Now with the text embeddings out, we start off with a pure random noise, as I was mentioning. These are our noisy latents as we are operating on the latent space.

And we also need to have access to a component-free term as Scheduler, which is a nonparametric component, which takes care of-- which basically makes the model aware of the position it is in in the entire denoising trajectory.

And this is the beast that we will try to optimize, but more on that later. It can be a typical unit-like architecture or a transformer-like architecture. And broadly, this is referred to as the diffusion network, which is conditioned on the time step, like the position in the denoising iterations that it is in, the text embeddings in this case, and also the noisy latents.

And then we invoke it over a period of time, hence the loop. Now once we get our refined latents out the diffusion network, it's passed off to a decoder, and we finally get our frames. In case of an image, it will just be a single frame or just the single final image.

Now that we have some very basic understanding of the different components that are involved in a standard text-to-image or text-to-video diffusion model and how those components are connected with one another, how they draw the chronology in the whole inference workflow. I'll try to now motivate why we might have to optimize them to be able to--