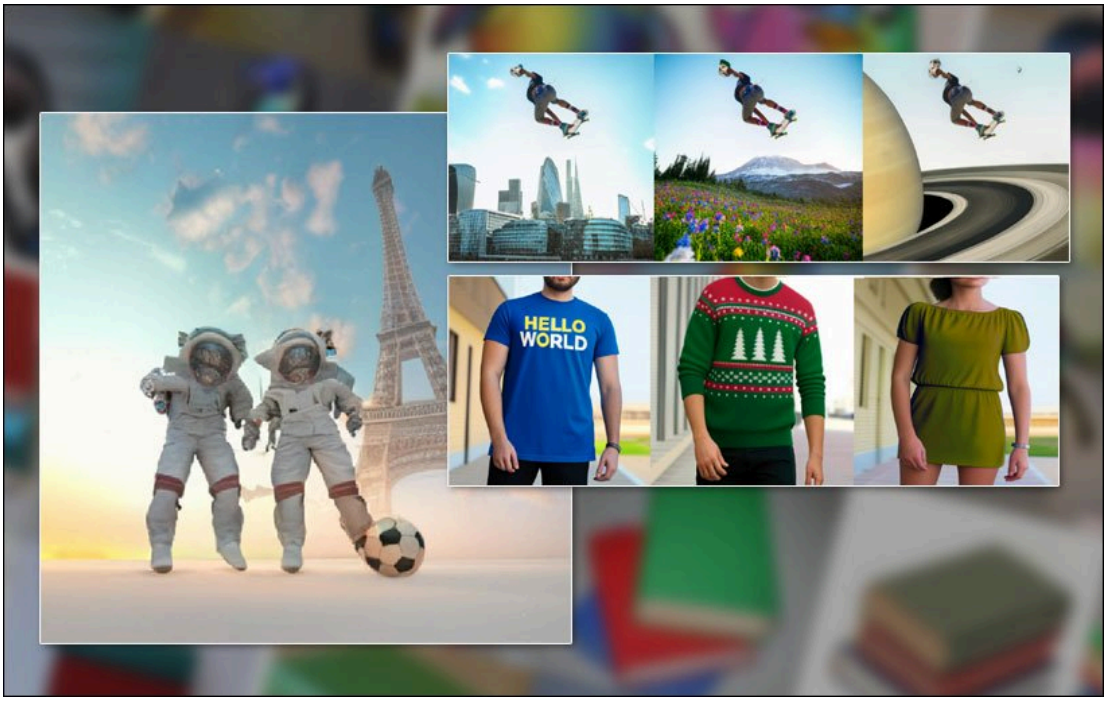
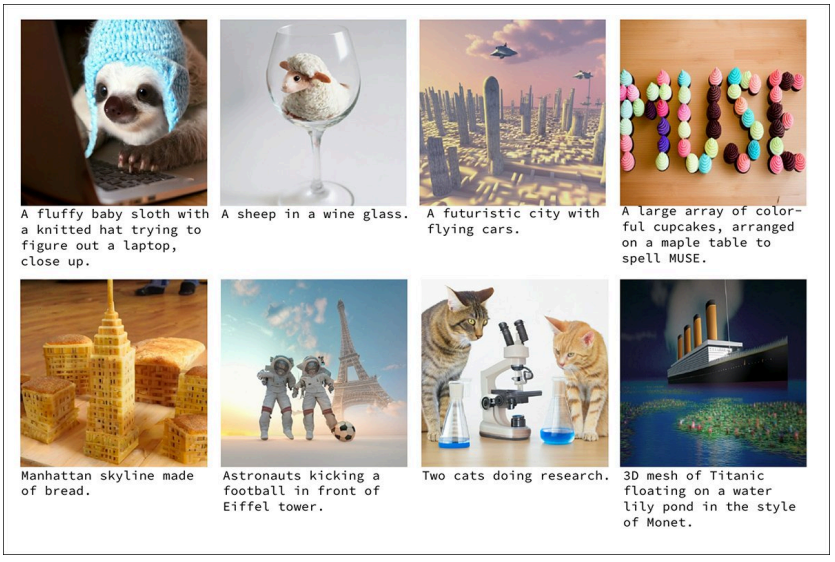
Muse: Google's Super-Fast Text-To-Image Model Abandons Latent Diffusion for Transformers

Originally published at https://arquivo.pt/wayback/20240203192022oe_/https://blog.metaphysic.ai/muse-googles-super-fast-text-to-image-model-abandons-latent-diffusion-for-transformers/

January 5, 2023



Google Research has revealed a new type of framework for text-to-image synthesis, based on the Transformers architecture, rather than latent diffusion.



Familiar functionality in Google's new system - but the internals and inference times are notably different from the competition. Source: https://arxiv.org/pdf/2301.00704.pdf

The project is titled *Muse*, and is essentially a patchwork assembly of multiple prior works (many also from Google Research) – though in itself it constitutes the first major generative system to leverage transformers so directly, and to abandon latent diffusion, despite the current huge popularity of that architecture.

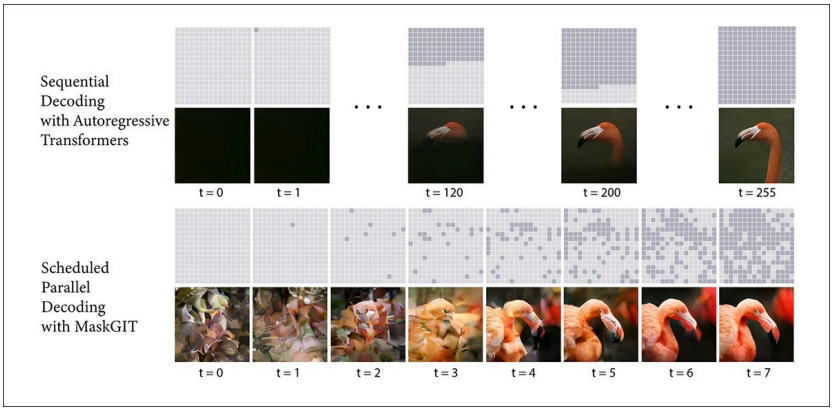
The Transformers architecture itself was debuted by Google Brain (amongst others) in mid-2017, and has since formed only sections of some implementations of popular image synthesis frameworks, rather than being the central enabling technology – as is the case with the new work.

The project's most notable breakthrough is in how quickly it can perform inference (i.e., how long you have to wait for your *'photo of a bear drinking coffee'*). In this respect, Muse definitively beats Stable Diffusion, as well as Google's own prior Imagen text-to-image model:

Model	Resolution	Time
Imagen	256×256	9.1s
Imagen	1024×1024	13.3s
LDM (50 steps)	512×512	3.7s
LDM (250 steps)	512×512	18.5s
Parti-3B	256×256	6.4s
Muse-3B	256×256	0.5s
Muse-3B	512×512	1.3s

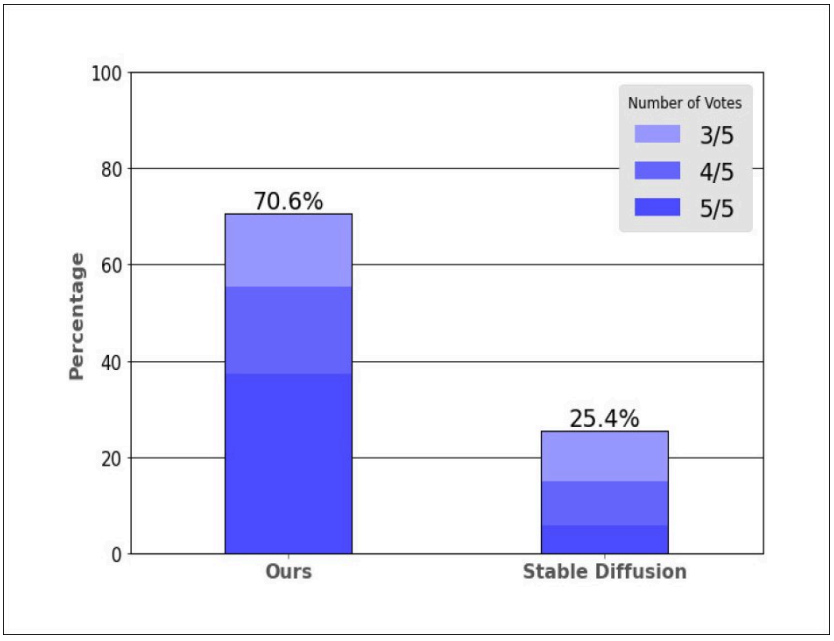
With lower numbers better, Muse is able to produce images faster than Stable Diffusion (V1.4 checkpoint) or general Latent Diffusion Models (LDMs), and comfortably beats Google's own Imagen text-to-image framework in this respect. All tests were conducted on A100 GPUs.

The improved inference times are attributable to various factors, not the least of which is parallel decoding, not a mainstay of existing text-to-image systems, and which optimizes the handling of tokens in the generative procedure (tokens are pieces of data which have been split up – usually, a token is simply a word, or a term). To quote the source paper from which this aspect of Muse is derived, *'at each iteration, the model predicts all tokens simultaneously in parallel but only keeps the most confident ones.'*



Predicting tokens at speed: the MaskGIT code used in Muse is at the decoding finish line while latent diffusion methods are barely out of the paddocks. Though parallel decoding is not the only reason for Muse's speed, it's a notable innovation that's quite specific to Muse.

Among a complex slew of tests, output from Muse was subject to human evaluation. In an evaluation round requiring a 3-user consensus, results from text prompts for Muse were preferred by raters in 70.6% of cases, which the authors attribute to Muse's superior prompt fidelity.



Graph showing the percentage of prompts preferred for Muse Vs. Stable Diffusion.

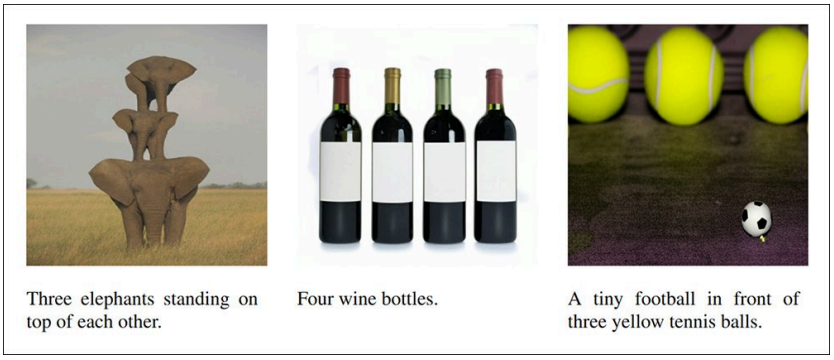
The authors estimate that Muse is over *ten times* faster at inference time than Google's prior Imagen-3B or Parti-3B models, and three times faster than Stable Diffusion (v1.4). They attribute the latter improvement to the *'significantly higher number of iterations'* that Stable Diffusion requires for inference.

The new Google paper notes that the improved speed does not impose any loss of quality, in comparison to recent and popular systems, and further observes that Muse is capable of out-of-the-box maskless editing, outpainting and inpainting.

Mask-free editing is, in particular, a 'Holy Grail' for Stable Diffusion, which struggles to make selective prompt-based changes to an existing image, largely due to the sector-wide issue of entanglement.

Examples of Muse's mask-free editing, where the semantic compositionality of the input image is preserved even in the face of massive changes requested by the editing-prompt. In these cases, the 'masking' is accomplished at the semantic level, without the need to impose manual masks, alpha channels, or any of the other workarounds that tend to dominate inverted image editing across the popular architectures.

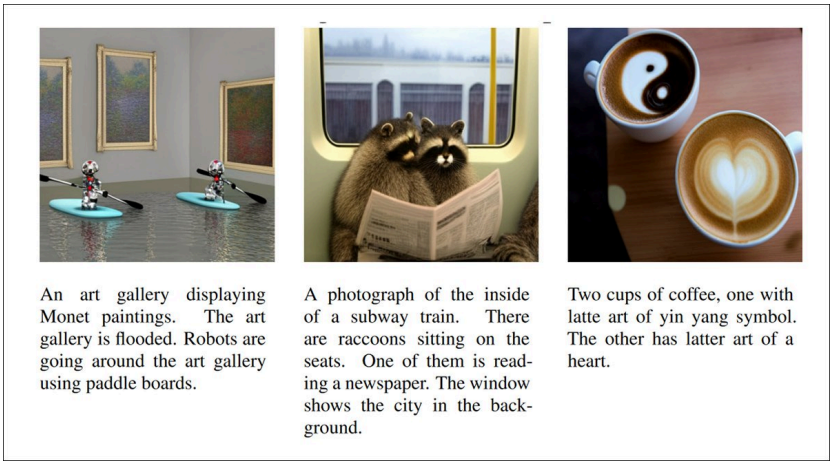
Muse addresses a frequent bugbear of systems such as Stable Diffusion and DALL-E 2 – *cardinality*: to what extent should the output image prioritize *one part of the prompt* over another, without resorting to hacks, workarounds and other types of third-party instrumentality that aren't conducive to an easy user experience?



Fidelity to the intent of a prompt, sidestepping 'literal' or ambiguous possible interpretations, is a priority for Muse. For many generative systems, even getting the number of wine bottles right (center image) can be a challenge, while words such as 'right' or 'left', among many others, can lead the system astray with respect to user intent.

The parts of a text-prompt that get 'preference' can greatly affect the quality of the output in a generative system. In Stable Diffusion, for instance, the earliest words in a long text-prompt will be prioritized, and later or ancillary parts of the prompt may be ignored entirely as the architecture's ability to coherently synthesize the prompt is challenged by complex requests.

The authors of the new paper note that Muse is well-disposed to include the *entire content* of even quite lengthy prompts – a shortcoming in rival systems.

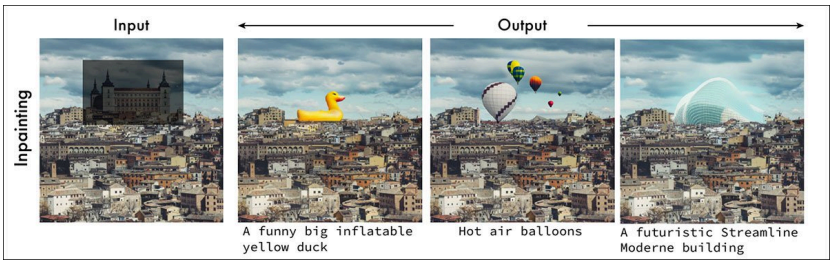


Muse performs well in acceding to the entirety of a text-prompt's request.

The authors note that Muse's approach to image editing is less laborious than many recent attempts at the challenge, stating*:

'This method works directly on the (tokenized) image and does not require "inverting" the full generative process, in contrast with recent zero-shot image editing techniques leveraging generative models.'

Inpainting, by contrast, is a more stratified procedure, where the user explicitly masks out a section of a picture and applies changes only to that area, similar to Photoshop. In that area, a new or revised text-prompt can alter the content of the picture through modified or alternative text-prompts while retaining broad context:



On the left, the defined area will be inpainted. To the right, we see the result of various text prompts in Muse, which change the content while retaining the broader context of the image, through CLIP.

Muse's sampling procedure facilitates inpainting 'for free' (as the authors describe it). The input image (which may be a real or previously synthesized image) is converted into a set of text/image tokens which are conditioned on unmasked tokens and a text-prompt (supplied by the user). The amended image parameters are downsampled and re-upsampled (256px/512px), with both images converted to high and low resolution tokens, before the system masks out the apposite region for this group of tokens. After this, parallel sampling provides the actual inpainting.

Outpainting is a broadly similar procedure, except that it usually involves extending the image beyond its current borders. Nonetheless, token-based context is still the key to generating image-consistent imagery.

In these example from the new paper, outpainting is represented as an inverse inner section. In fact, the process is also capable of extending the picture's boundaries, so that it has larger and altered dimensions, with extra content added to any number of sides of the original image.

Further, Muse, the authors assert, achieves comparable or higher realism scores, as assessed by Fréchet inception distance (FID, which measures image quality) and Contrastive Language-Image Pre-training (CLIP, which measures how closely the output image and its text prompt are related).

Approach	Model Type	Params	FID-30K	Zero-shot FID-30K
AttnGAN (Xu et al., 2017)	GAN		35.49	-
DM-GAN (Zhu et al., 2019)	GAN		32.64	-
DF-GAN (Tao et al., 2020)	GAN		21.42	-
DM-GAN + CL (Ye et al., 2021)	GAN		20.79	-
XMC-GAN (Zhang et al., 2021)	GAN		9.33	-
LAFITE (Zhou et al., 2021)	GAN		8.12	-
Make-A-Scene (Gafni et al., 2022)	Autoregressive		7.55	-
DALL-E (Ramesh et al., 2021)	Autoregressive		-	17.89
LAFITE (Zhou et al., 2021)	GAN		-	26.94
LDM (Rombach et al., 2022)	Diffusion		-	12.63
GLIDE (Nichol et al., 2021)	Diffusion		-	12.24
DALL-E 2 (Ramesh et al., 2022)	Diffusion		-	10.39
Imagen-3.4B (Saharia et al., 2022)	Diffusion		-	7.27
Parti-3B (Yu et al., 2022)	Autoregressive		-	8.10
Parti-20B (Yu et al., 2022)	Autoregressive		3.22	7.23
Muse-3B	Non-Autoregressive		-	7.88

In a quantitative evaluation of Muse against prior frameworks (including Stable Diffusion/LDMS), Muse's non-diffusion Transformer architecture performed comparably or better, while enjoying massively reduced inference times, meaning that the image will appear much more quickly after the use submits their text prompt. The framework was additionally tested against Generative Adversarial Models (GANs).

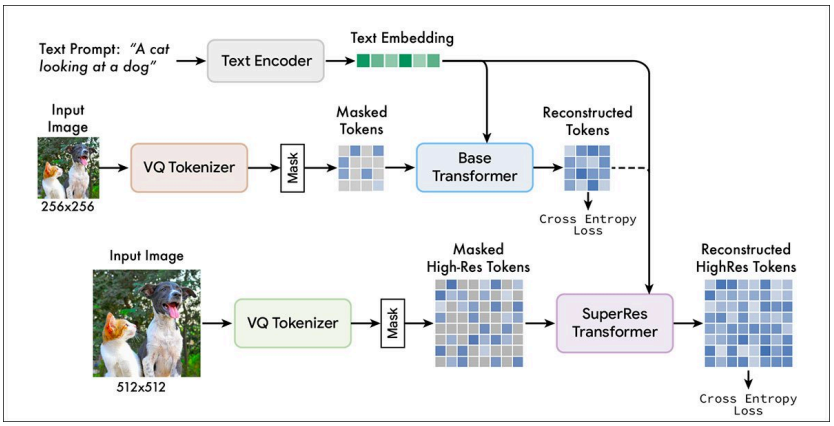
Approach

As mentioned above, the central approach of Muse is based on Google's prior work, MaskGIT masked image modeling, which debuted in February of 2022. The MaskGIT paper leveled some criticism against prior Transformer-centric image synthesis systems, broadly comparing such approaches to the linear processes of obtaining a picture with a flatbed scanner.

Instead, Muse's image decoder architecture is conditioned on embeddings obtained from a pre-trained and frozen large language model (LLM), T5-XXL. The authors observe*:

'In agreement with Imagen, we find that conditioning on a pre-trained LLM is crucial for photorealistic, high quality image generation. Our models (except for the VQGAN quantizer) are built on the Transformer architecture.'

In the image below, we can see the parallel processes in action for the conceptual architecture of Muse, with both the higher and lower-res versions of the input image being run through differing routines which extract and reconstruct semantic tokens, until high-res tokens are available for image creation and manipulation:



The conceptual workflow of Muse, which makes heavy use of internal upscaling and downscaling routines.

The VQ tokenizer for the lower-resolution model is pre-trained on 256x256px images, generating a 16×16 latent space of tokens. The resulting sequence is then masked at a variable (rather than constant) rate per sample. After this, cross-entropy loss learns to predict the masked image tokens, and the schema can be used to create higher resolution masked tokens.

To a certain extent, the researchers behind Muse do not know exactly how the T5-XXL LLM obtains the results that it's capable of within the Muse framework, and state*:

'Our hypothesis is that the Muse model learns to map these rich visual and semantic concepts in the LLM embeddings to the generated images; it has been shown in recent work that the conceptual representations learned by LLM's are roughly linearly mappable to those learned by models trained on vision tasks.

'Given an input text caption, we pass it through the frozen T5-XXL encoder, resulting in a sequence of 4096 dimensional language embedding vectors. These embedding vectors are linearly projected to the hidden size of our Transformer models (base and super-res).'

Cloistered Genius?

There are two interesting considerations here: one is that this approach is quite revolutionary, since it departs entirely from diffusion models and performs well on the home territory of Stable Diffusion (and, to a less important extent, Generative Adversarial Networks, or GANs, which have lately been eclipsed by diffusion approaches). Google Research releases papers frequently, and it can be hard to tell which are going to have a major impact on the image synthesis scene.

For instance, the release of DreamBooth late last year was almost lost in a conference-season avalanche of new releases from the research arm of the search giant. As it transpired, the relatively dry DreamBooth documentation and code was eventually spun into what may be the biggest and most controversial advent in the short history of deepfakes, since they were defined in 2017

The second consideration is that Google, once again, as stated clearly in the new paper, is apparently too afraid of repercussions to release the code. After announcing that neither the source code for Muse nor any public demo will be made available 'at this time', the new paper states:

'[we] do not recommend the use of text-to-image generation models without attention to the various use cases and an understanding of the potential for harm. We especially caution against using such models for generation of people, humans and faces.'

Barring a change of heart on this, it seems that Muse, like many other 'powerful' frameworks from Google, may end up simply being yet another benchmark for subsequent image synthesis frameworks (that may not be released either, for the same reasons); though it seems probable that the company is at least interested in developing API-only access to such systems, once the inevitable user workarounds on restrictions have been adequately nailed down.

** My conversion of the author's inline citation to hyperlinks.*