



Generating other media

AI for Media, Art & Design, Spring 2024

Prof. Perttu Hämäläinen

Aalto University

Contents

- General principles
- Multimodal LLMs
- Video
- Audio
- 3D animation
- 3D models
- Behavior
- Real-world production pipelines

General principles

Token-by-token generation using Transformers

- Parti and DALL-E 1 generated images by modeling sequences of both text and image tokens
- In the same way, **anything** can be tokenized for Transformer training
- Problems:
 - Expensive and slow to train
 - Training a large model from scratch requires a lot of data
- Solution: Repurpose pretrained LLMs

Finetuning a pretrained text transformer using custom data

Lu et al. successfully converted a GPT-2 text generator into an image classifier

Kevin Lu,^{1,2} Aditya Grover,^{2,3} Pieter Abbeel,¹ Igor Mordatch⁴

¹ UC Berkeley, ² Facebook AI Research, ³ UCLA, ⁴ Google Brain
kzl@fb.com

Abstract

We investigate the capability of a transformer pretrained on natural language to generalize to other modalities with minimal finetuning – in particular, without finetuning of the self-attention and feedforward layers of the residual blocks. We consider such a model, which we call a Frozen Pretrained Transformer (FPT), and study finetuning it on a variety of sequence classification tasks spanning numerical computation, vision, and protein fold prediction. In contrast to prior works which investigate finetuning on the same modality as the pretraining dataset, we show that pretraining on natural language can improve performance and compute efficiency on non-language downstream tasks. Additionally, we perform an analysis of the architecture, comparing the performance of a random initialized transformer to a random LSTM. Combining the two insights, we find language-pretrained transformers can obtain strong performance on a variety of non-language tasks.

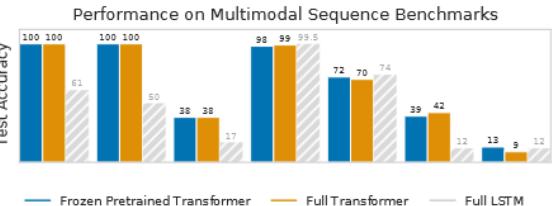


Figure 1: A *frozen* language-pretrained transformer (FPT) – without finetuning the self-attention and feedforward layers – can achieve strong performance compared to a transformer fully trained from scratch on a downstream *modality* on literature benchmarks (Tay et al. 2020; Rao et al. 2019). We show results on diverse classification tasks (see Section 2.1): numerical computation (Bit Memory/XOR, ListOps), image classification (MNIST, CIFAR-10, LRA), and protein fold prediction (Homology). We also show results for a fully-trained from-scratch LSTM as a baseline. Our code is available at: github.com/kzl/universal-computation

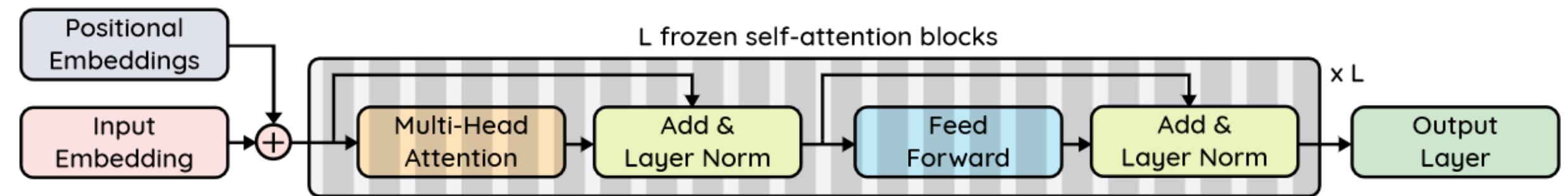
1 Introduction

The transformer architecture (Vaswani et al. 2017) has shown broad successes in deep learning, serving as the backbone of large models for tasks such as modeling natural language (Brown et al. 2020), images (Dosovitskiy et al. 2020), proteins (Jumper et al. 2021), and multimodal tasks comprising of both images and text (Lu et al. 2019; Radford et al. 2021). Inspired by these successes, we seek to explore the generalization capabilities of a transformer in transferring from one modality to another.

Classical approaches to sequence processing used recurrent neural network (RNN) approaches (Rumelhart, Hinton, and Williams 1985; Hochreiter and Schmidhuber 1997). In contrast, transformers utilize self-attention layers to extract features across tokens of a sequence, such as words (Vaswani et al. 2017) or image patches (Dosovitskiy et al. 2020). Furthermore, it has become common practice to train large models on unsupervised objectives before finetuning or evaluating zero-shot generalization on a downstream task. However, the downstream tasks that have been studied are generally restricted to the same modality as the original training set: for example, train GPT (Radford et al. 2018) on a large language corpus, and finetune on a small task-specific dataset. Our goal in this work is to investigate finetuning on modalities distinct from the training modality.

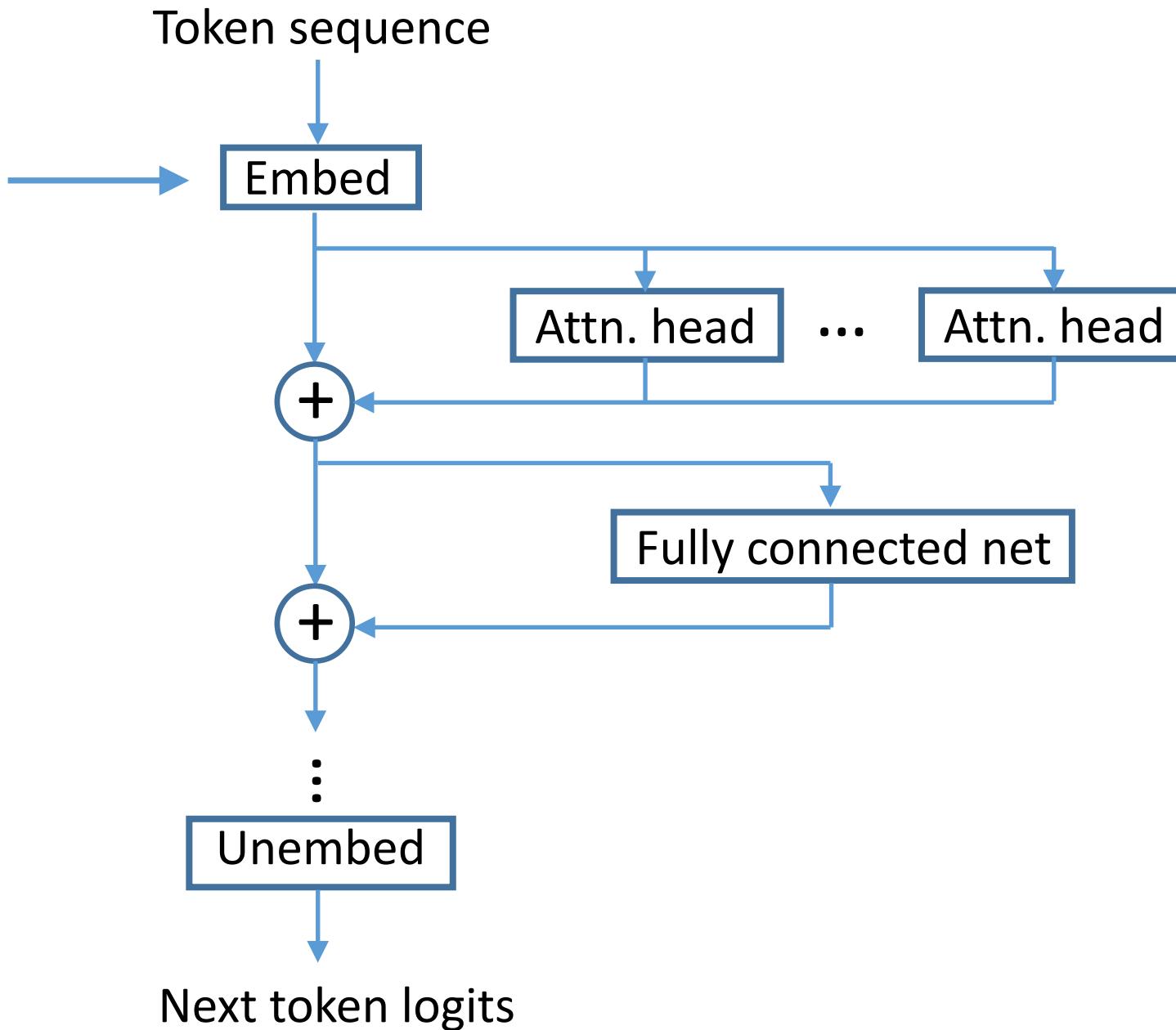
We hypothesize that transformers – namely the self-attention layers – can be pretrained on a data-rich modality (i.e. where data is plentiful, such as a language corpus) and identify feature representations that are useful for *arbitrary* data sequences, enabling downstream transfer to different modalities. In particular, we seek to investigate what pretrained language models (LMs) are capable of in terms of generalizing to other modalities with sequential structure.

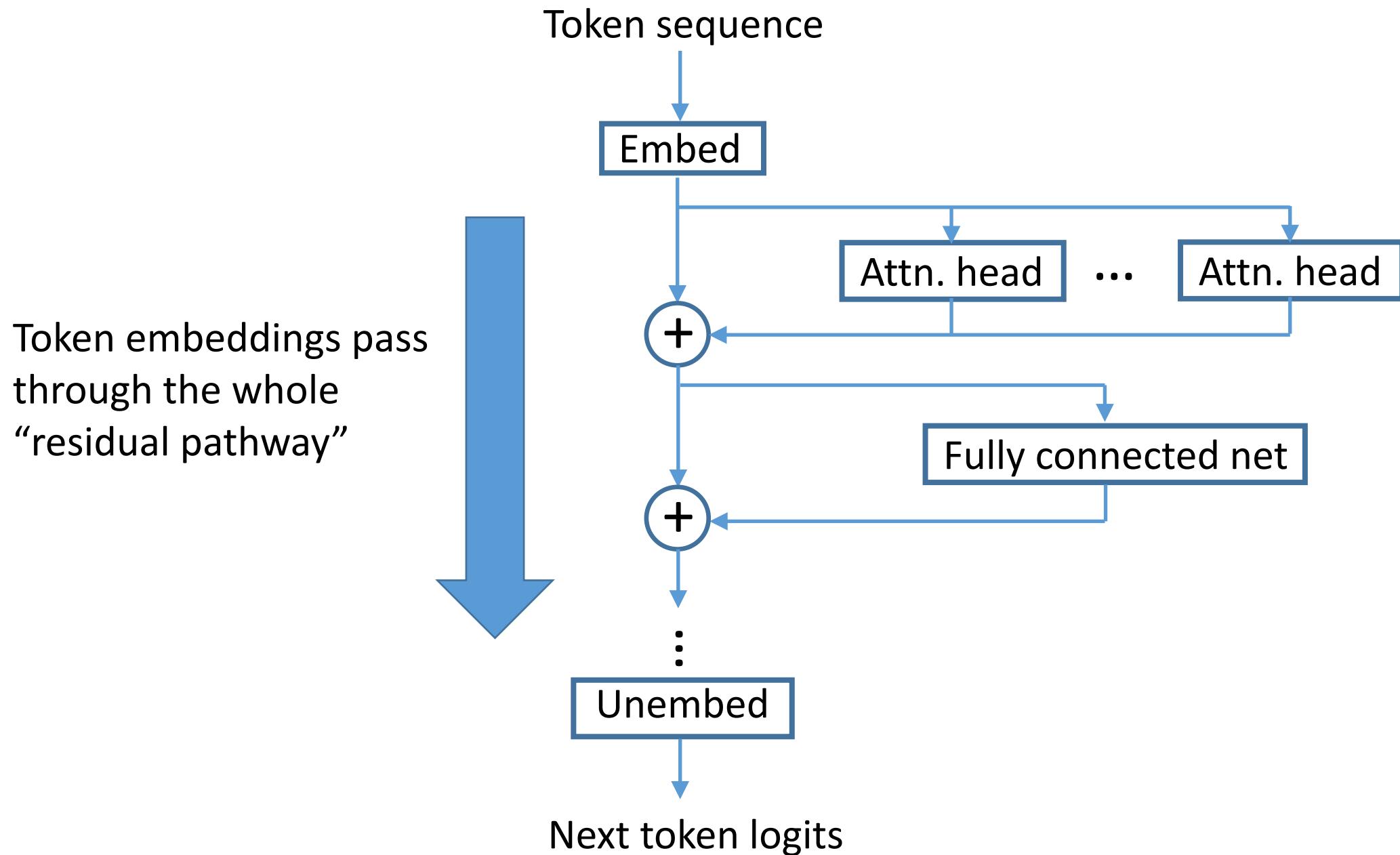
To investigate this hypothesis, we take a transformer model pretrained on natural language data, GPT-2 (Radford et al. 2019), and finetune only the linear input and output layers, as well as the positional embeddings and layer norm parameters. These decisions are made to highlight the parameters already in the language model, and not for performance purposes. We call this model a Frozen Pretrained Transformer (FPT). On a range of tasks across a variety of modalities – including numerical computation, image classification, and protein fold prediction – FPT displays comparable performance to training the entire transformer from

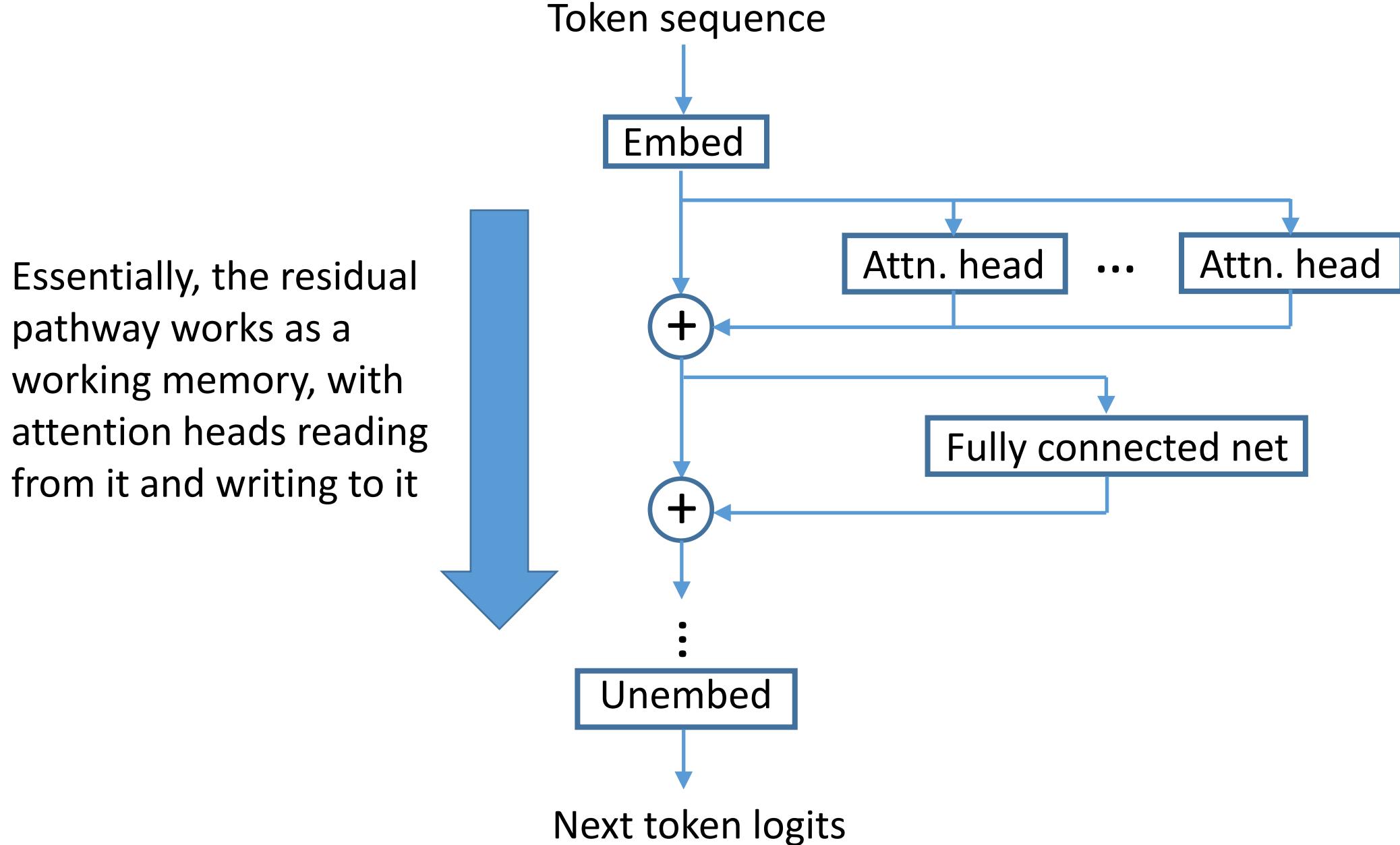


The architecture only retrains the input embedding and output layers, keeping rest of GPT-2 as is.

A look-up table:
An initial
embedding
vector for each
possible token

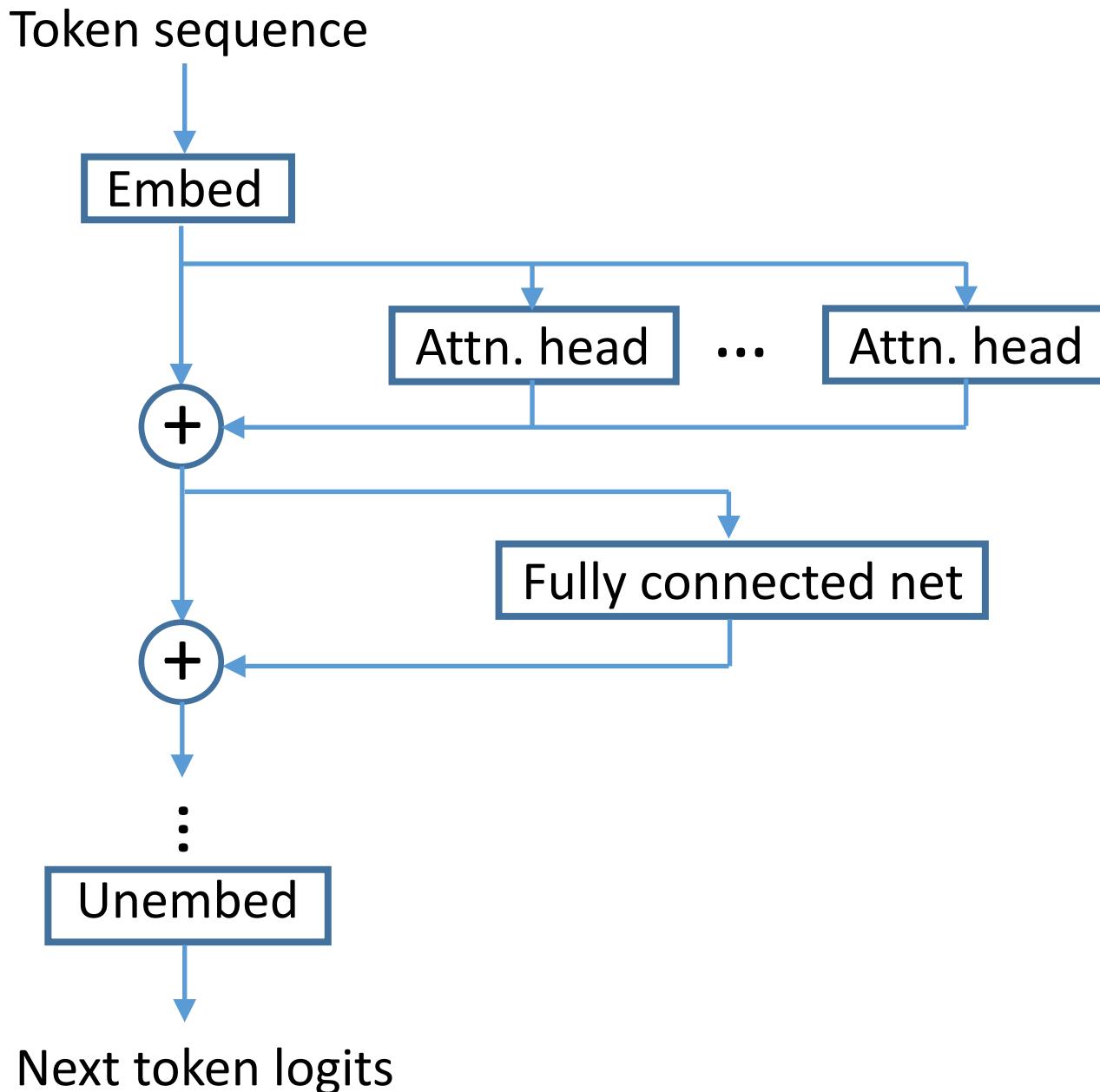


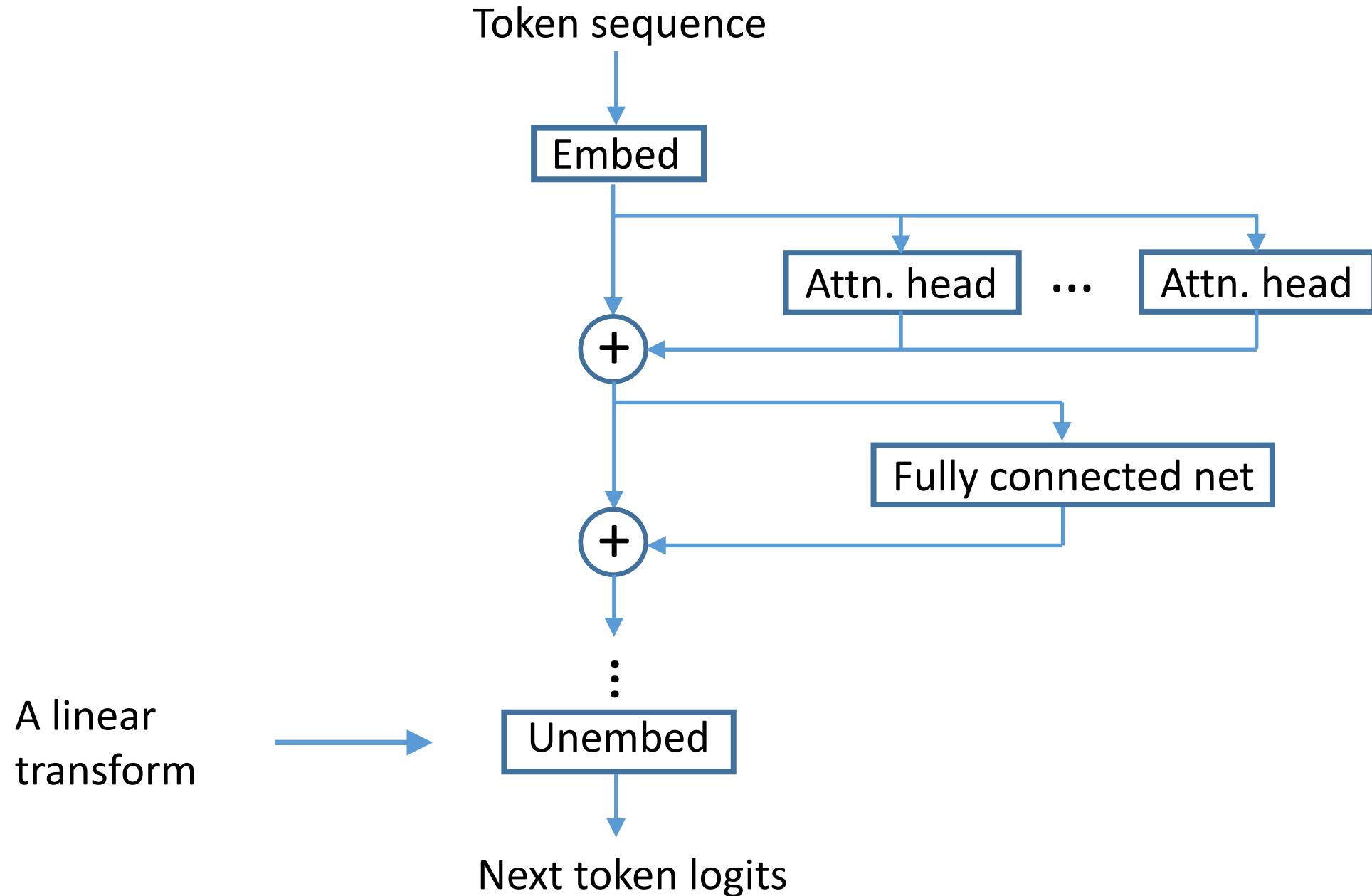




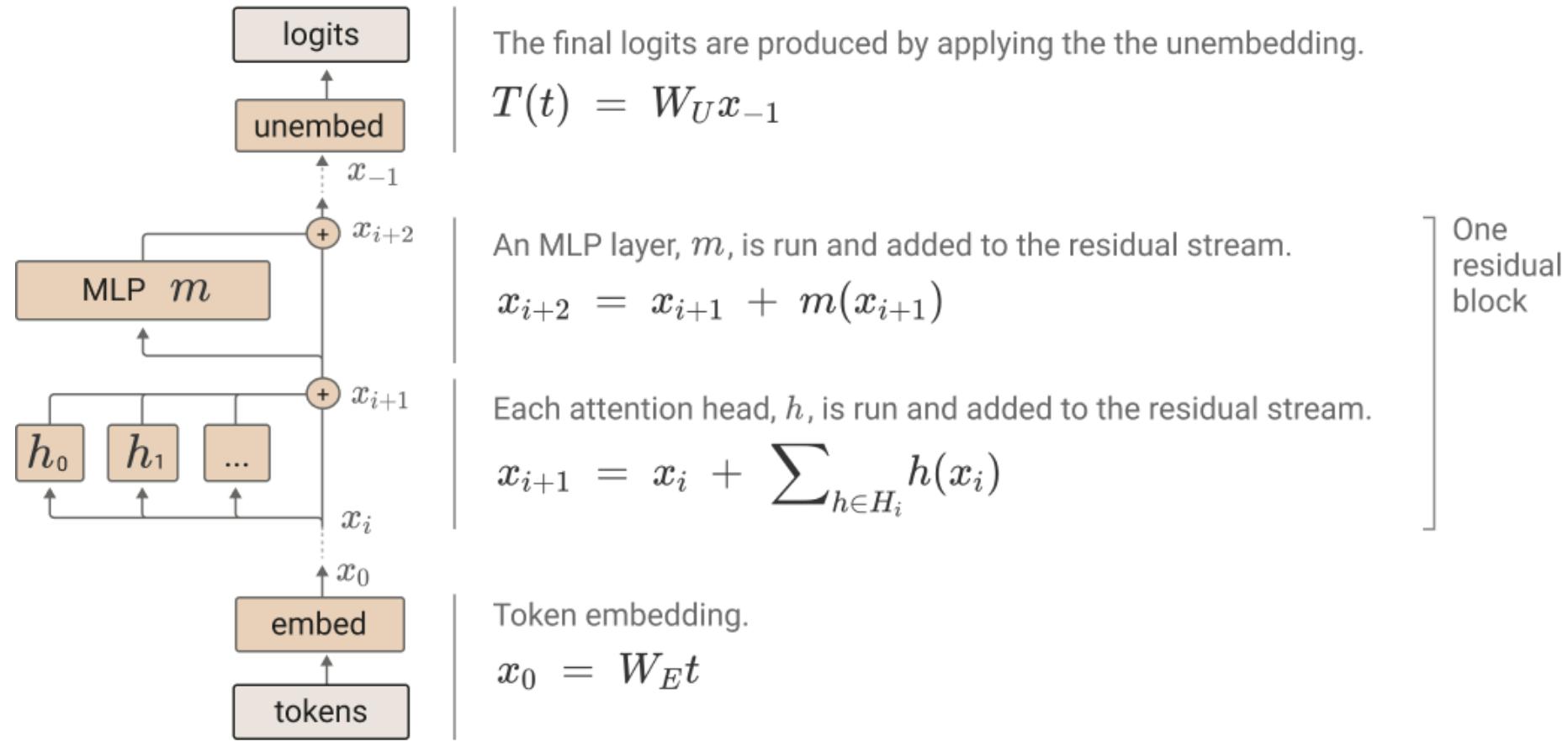
Each token's embedding represents its meaning and relation to other tokens.

The embeddings get gradually modified by each layer, e.g., annotating them with further information



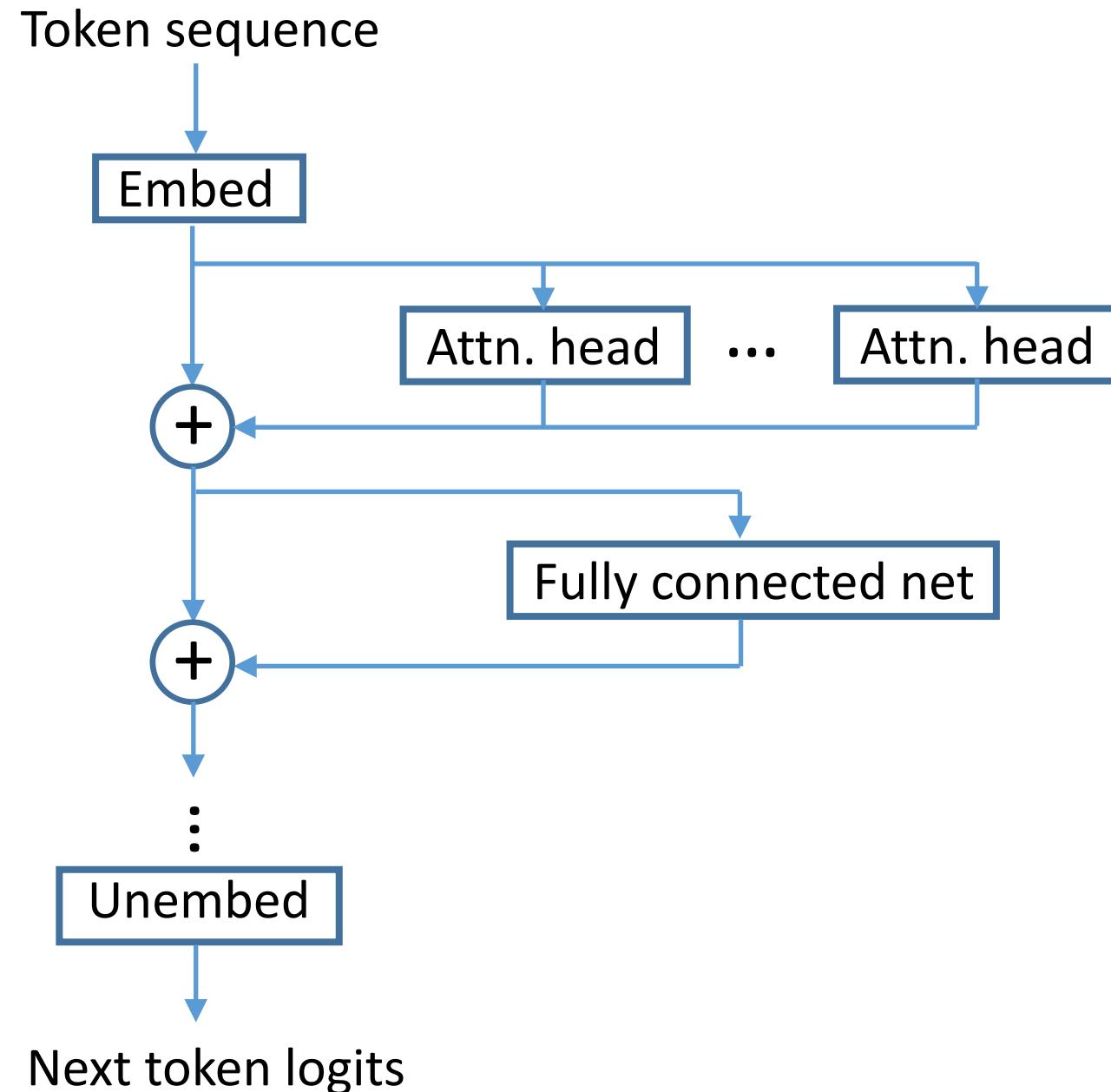


More detail



Retraining the embed and unembed layers for a new vocabulary of tokens = repurposing the complex computational machinery learned for text to other data.

Even better results can be obtained by also adding and training LoRA adapters.



How to tokenize?

- An autoencoder can compress/decompress complex data such as image patches into N-dimensional vectors
- VQ-VAE: The vectors can be converted to discrete tokens using vector quantization

Generating Diverse High-Fidelity Images with VQ-VAE-2

Ali Razavi*
DeepMind
alirazavi@google.com

Aäron van den Oord*
DeepMind
avdnoord@google.com

Oriol Vinyals
DeepMind
vinyals@google.com

Abstract

We explore the use of Vector Quantized Variational AutoEncoder (VQ-VAE) models for large scale image generation. To this end, we scale and enhance the autoregressive priors used in VQ-VAE to generate synthetic samples of much higher coherence and fidelity than possible before. We use simple feed-forward encoder and decoder networks, making our model an attractive candidate for applications where the encoding and/or decoding speed is critical. Additionally, VQ-VAE requires sampling an autoregressive model only in the compressed latent space, which is an order of magnitude faster than sampling in the pixel space, especially for large images. We demonstrate that a multi-scale hierarchical organization of VQ-VAE, augmented with powerful priors over the latent codes, is able to generate samples with quality that rivals that of state of the art Generative Adversarial Networks on multifaceted datasets such as ImageNet, while not suffering from GAN’s known shortcomings such as mode collapse and lack of diversity.

1 Introduction

Deep generative models have significantly improved in the past few years [5, 27, 25]. This is, in part, thanks to architectural innovations as well as computation advances that allows training them at larger scale in both amount of data and model size. The samples generated from these models are hard to distinguish from real data without close inspection, and their applications range from super resolution [21] to domain editing [44], artistic manipulation [36], or text-to-speech and music generation [25].



Figure 1: Class-conditional 256x256 image samples from a two-level model trained on ImageNet.

We distinguish two main types of generative models: likelihood based models, which include VAEs [16, 31], flow based [9, 30, 10, 17] and autoregressive models [20, 39]; and implicit generative

*Equal contributions.

Residual Vector Quantization (RVQ)

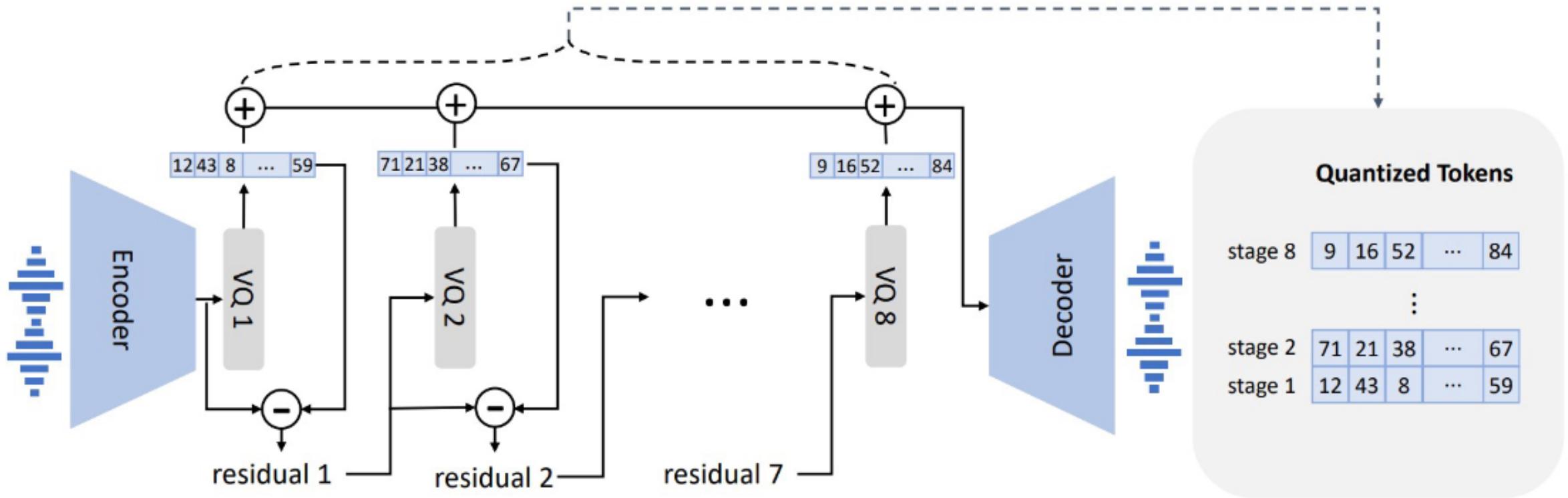


Figure 2: The neural audio codec model revisit. Because RVQ is employed, the first quantizer plays the most important role in reconstruction, and the impact from others gradually decreases.

Recap: A general and powerful recipe

- Tokenize your custom data using an autoencoder and RVQ
 - For simple data, the autoencoder may not be needed
- Finetune an LLM such as Llama or Phi
 - Custom data only: Replace the embedding layer
 - Custom data interleaved with text: Expand the vocabulary and the embedding layer with your RVQ tokens
 - Replace the unembedding layer
 - Add LoRA adapters
- This course: Colab demo provided

Another general recipe

- Diffusion models with a U-Net denoiser are a good default choice if two conditions are met:
 - The data is conveniently represented by real-valued arrays (tensors in ML lingo)
 - You can assume stronger local than global dependencies between the array values (this is what convolutional networks like U-Net are designed for)
- Video is a good example:
 - A video is an array of shape [timesteps, width, height, color channels]
 - Nearby pixels (in both space and time) typically belong to the same object

Multimodal LLMs

Multimodal LLMs

- LLMs where the token vocabulary includes text, image patches, audio segments...
- Motivation:
 - Text is high-level and abstract => can't produce precise and “grounded” understanding of the physical world
 - There's only so much text data available
 - => internet video is the next data goldmine, especially tutorials and let's play videos where a human explains what is happening
- Abilities: chat about images, revise generated images using chat, generate image sequences and video



Flamingo (2022)

Demonstrated that few-shot prompting (in-context learning) is possible with a mixture of images and words.

I.e., by showing examples, one can instruct the model to perform new tasks involving both images and text.

No image generation yet.

Flamingo: a Visual Language Model for Few-Shot Learning

Jean-Baptiste Alayrac*,‡ Jeff Donahue* Pauline Luc* Antoine Miech*
Iain Barr† Yana Hasson† Karel Lenc† Arthur Mensch† Katie Millican†
Malcolm Reynolds† Roman Ring† Eliza Rutherford† Serkan Cabi Tengda Han
Zhitao Gong Sina Samangooei Marianne Monteiro Jacob Menick
Sebastian Borgeaud Andrew Brock Aida Nematzadeh Sahand Sharifzadeh
Mikolaj Binkowski Ricardo Barreira Oriol Vinyals Andrew Zisserman
Karen Simonyan*,‡

* Equal contributions, ordered alphabetically, † Equal contributions, ordered alphabetically,
‡ Equal senior contributions

DeepMind

Abstract

Building models that can be rapidly adapted to novel tasks using only a handful of annotated examples is an open challenge for multimodal machine learning research. We introduce Flamingo, a family of Visual Language Models (VLM) with this ability. We propose key architectural innovations to: (i) bridge powerful pretrained vision-only and language-only models, (ii) handle sequences of arbitrarily interleaved visual and textual data, and (iii) seamlessly ingest images or videos as inputs. Thanks to their flexibility, Flamingo models can be trained on large-scale multimodal web corpora containing arbitrarily interleaved text and images, which is key to endow them with in-context few-shot learning capabilities. We perform a thorough evaluation of our models, exploring and measuring their ability to rapidly adapt to a variety of image and video tasks. These include open-ended tasks such as visual question-answering, where the model is prompted with a question which it has to answer; captioning tasks, which evaluate the ability to describe a scene or an event; and close-ended tasks such as multiple-choice visual question-answering. For tasks lying anywhere on this spectrum, a *single* Flamingo model can achieve a new state of the art with few-shot learning, simply by prompting the model with task-specific examples. On numerous benchmarks, *Flamingo* outperforms models fine-tuned on thousands of times more task-specific data.

Corresponding authors: {jalayrac|jeffdonahue|paulineluc|miech}@deepmind.com
36th Conference on Neural Information Processing Systems (NeurIPS 2022).

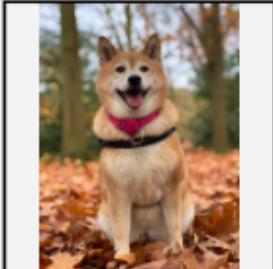
https://proceedings.neurips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html



Input Prompt



This is a chinchilla. They are mainly found in Chile.



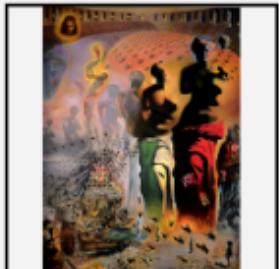
This is a shiba. They are very popular in Japan.



This is

Completion

a flamingo. They are found in the Caribbean and South America.



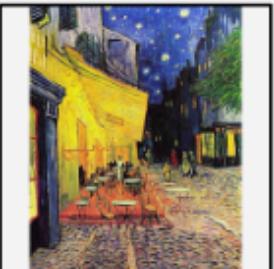
What is the title of this painting?

Answer: The Hallucinogenic Toreador.



Where is this painting displayed?

Answer: Louvres Museum, Paris.



What is the name of the city where this was painted?

Answer:

Arles.



Output:
"Underground"

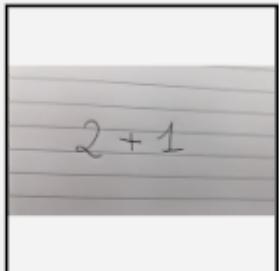


Output:
"Congress"

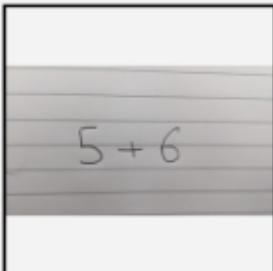


Output:

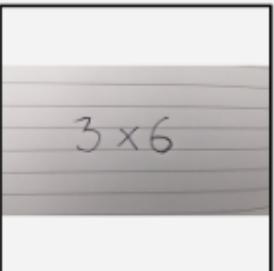
"Soulomes"



$$2+1=3$$



$$5+6=11$$



$$3 \times 6 = 18$$



P This is a picture of two teddy bears on the moon.

What are they doing?

P They are having a conversation.

What object are they using?

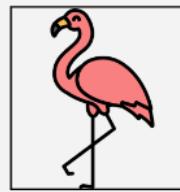
P It looks like a computer.

Is this surprising?

P Yes, it is surprising.

Why is this picture surprising to you?

P I think it is surprising because teddy bears are not usually found on the moon.



What is the common thing about these three images?

P They are all flamingos.

What is the difference between these three images?

P The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



P This is an apple with a sticker on it.

What does the sticker say?

P The sticker says "iPod".

Where is the photo taken?

P It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

P It looks like it's handwritten.

What color is the sticker?

P It's white.



P This is a cityscape. It looks like Chicago.

What makes you think this is Chicago?

P I think it's Chicago because of the Shedd Aquarium in the background.



What about this one? Which city is this and what famous landmark helped you recognise the city?

P This is Tokyo. I think it's Tokyo because of the Tokyo Tower.

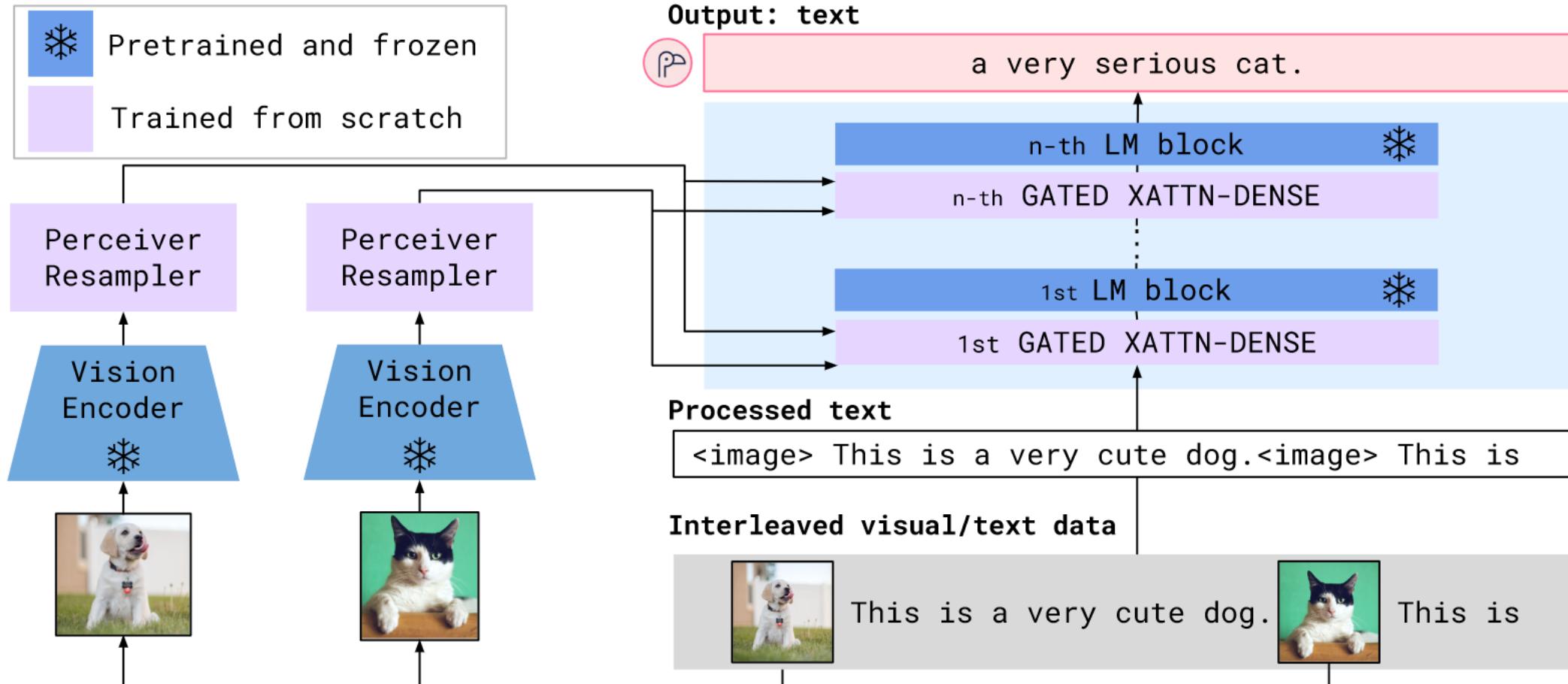


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Vision encoder: Normalizer-Free ResNet (NFNet) <https://proceedings.mlr.press/v139/brock21a.html>
 LM blocks: Chinchilla <https://arxiv.org/abs/2203.15556>

LLaVa (2023)

Like Flamingo, but more simple architecture, better quality, and fully open source.

Combines a CLIP ViT vision encoder with a Vicuna LLM (

Trained models available via Huggingface:
https://huggingface.co/docs/transformers/main/model_doc/llava

Paper: <https://arxiv.org/abs/2304.08485> (orig),
<https://arxiv.org/abs/2304.08485> (update, shown on the right)

Project page: <https://llava-vl.github.io/>

Haotian Liu¹ Chunyuan Li² Yuheng Li¹ Yong Jae Lee¹
¹University of Wisconsin–Madison ²Microsoft Research, Redmond
<https://llava-vl.github.io>

Abstract

Large multimodal models (LMM) have recently shown encouraging progress with visual instruction tuning. In this note, we show that the fully-connected vision-language cross-modal connector in LLaVA is surprisingly powerful and data-efficient. With simple modifications to LLaVA, namely, using CLIP-ViT-L-336px with an MLP projection and adding academic-task-oriented VQA data with simple response formatting prompts, we establish stronger baselines that achieve state-of-the-art across 11 benchmarks. Our final 13B checkpoint uses merely 1.2M publicly available data, and finishes full training in ~1 day on a single 8-A100 node. We hope this can make state-of-the-art LMM research more accessible. Code and model will be publicly available.

1. Introduction

Large multimodal models (LMMs) have become increasingly popular in the research community, as they are the key building blocks towards general-purpose assistants [1, 22, 35]. Recent studies on LMMs are converging on a central concept known as visual instruction tuning [28]. The results are promising, e.g. LLaVA [28] and MiniGPT-4 [49] demonstrate impressive results on natural instruction-following and visual reasoning capabilities. To better understand the capability of LMMs, multiple benchmarks [11, 20, 26, 29, 43] have been proposed. Recent works further demonstrate improved performance by scaling up the pretraining data [2, 9], instruction-following data [9, 21, 45, 46], visual encoders [2], or language models [31], respectively. The LLaVA architecture is also leveraged in different downstream tasks and domains, including region-level [6, 44] and pixel-level [19] understanding, biomedical assistants [23], image generation [3], adversarial studies [4, 47].

This note establishes stronger and more feasible baselines built upon the LLaVA framework. We report that two simple improvements, namely, an MLP cross-modal connector and incorporating academic task related data such as VQA, are orthogonal to the framework of LLaVA, and when used with

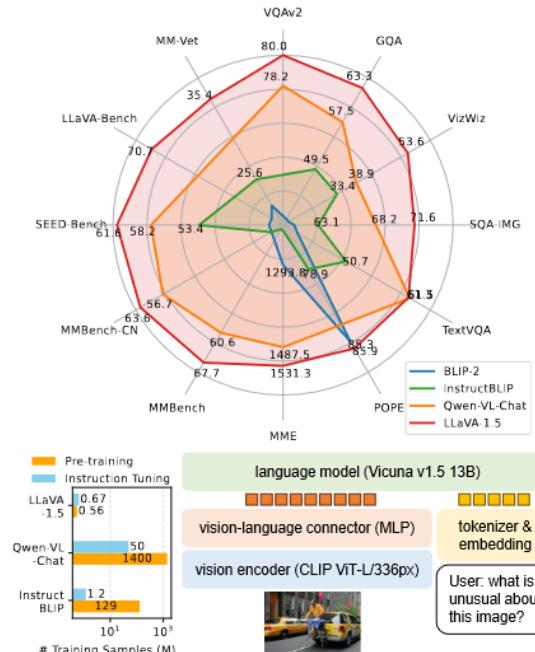
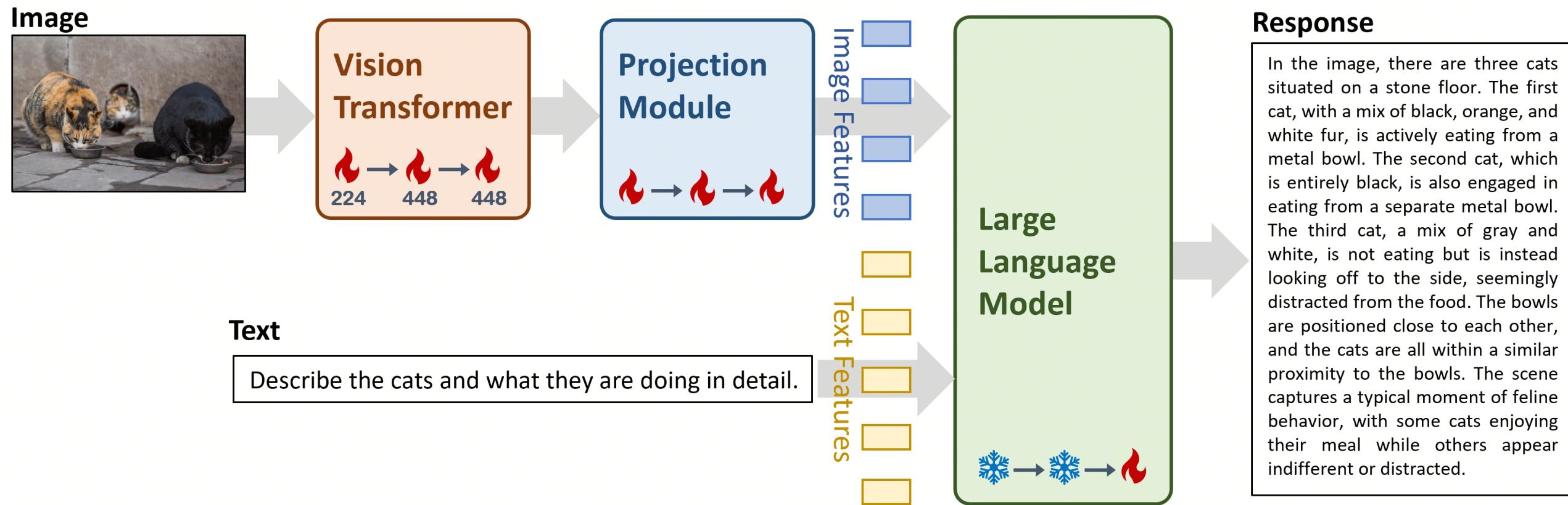


Figure 1. LLava-1.5 achieves SoTA on a broad range of 11 tasks (Top), with high training sample efficiency (Left) and simple modifications to LLava (Right): an MLP connector and including academic-task-oriented data with response formatting prompts.

LLava, lead to better multimodal understanding capabilities. In contrast to InstructBLIP [9] or Qwen-VL [2], which trains specially designed visual resamplers on hundreds of millions or even billions of image-text paired data, LLava uses the simplest architecture design for LMMs and requires only training a simple fully-connected projection layer on merely 600K image-text pairs. Our final model can finish training in ~1 day on a single 8-A100 machine and achieves state-of-the-art results on a wide range of benchmarks. Moreover, unlike Qwen-VL [2] that includes in-house data in training, LLava utilizes only publicly available data. We hope these improved and easily-reproducible baselines will provide a reference for future research in open-source LMM.



Yi-VL (2023)



Multimodal and bilingual (English + Chinese) LLM
LLaVa architecture but with the Yi LLMs instead of Vicuna
<https://github.com/01-ai/Yi>

Emu2 (2023)

Open model like LLaVa but adds an image decoder for image tokens => can generate both images and text

<https://baavision.github.io/emu2/>

Quan Sun^{1*} Yufeng Cui^{1*} Xiaosong Zhang^{1*} Fan Zhang^{1*} Qiying Yu^{2,1*} Zhengxiong Luo¹
Yueze Wang¹ Yongming Rao¹ Jingjing Liu² Tiejun Huang^{1,3} Xinlong Wang^{1†}

¹ Beijing Academy of Artificial Intelligence ² Tsinghua University ³ Peking University

*equal contribution †project lead

code & models: <https://github.com/baaivision/Emu>

Abstract

The human ability to easily solve multimodal tasks in context (i.e., with only a few demonstrations or simple instructions), is what current multimodal systems have largely struggled to imitate. In this work, we demonstrate that the task-agnostic in-context learning capabilities of large multimodal models can be significantly enhanced by effective scaling-up. We introduce Emu2, a generative multimodal model with 37 billion parameters, trained on large-scale multimodal sequences with a unified autoregressive objective. Emu2 exhibits strong multimodal in-context learning abilities, even emerging to solve tasks that require on-the-fly reasoning, such as visual prompting and object-grounded generation. The model sets a new record on multiple multimodal understanding tasks in few-shot settings. When instruction-tuned to follow specific instructions, Emu2 further achieves new state-of-the-art on challenging tasks such as question answering benchmarks for large multimodal models and open-ended subject-driven generation. These achievements demonstrate that Emu2 can serve as a base model and general-purpose interface for a wide range of multimodal tasks. Code and models are publicly available to facilitate future research.

1. Introduction

Multimodal tasks [26, 42] encompass anything involving understanding and generation in single or multiple modalities [5, 20, 59], which can be highly diverse and long-tail. Previous multimodal systems largely rely on designing task-specific architecture and collecting a sizable supervised training set, both of which are difficult to scale, particularly when this process needs to be repeated for each new task encountered. By contrast, humans can solve a new task in context, i.e., with only a few demonstrations or simple

instructions – a capability that current multimodal models have yet to learn.

Recently, generative pretrained language models have demonstrated strong in-context learning abilities [12, 22, 74]. By training a 37-billion-parameter model Emu2 and thoroughly evaluating it on diverse multimodal tasks, we demonstrate that a scaled-up multimodal generative pretrained model can harness similar in-context learning abilities and effectively generalize to unseen multimodal tasks. Emu2 is trained with a unified autoregressive objective: predict-the-next-multimodal-element (either visual embeddings or textual tokens). In this unified generative pretraining process, large-scale multimodal sequences (e.g., text, image-text pairs, and interleaved image-text-video) are used for model training.

We measure Emu2’s capabilities of learning from a few examples or instructions on standard multimodal datasets, as well as new tasks unseen in the training set. Specifically, Emu2 is evaluated under two scenarios: (a) *few-shot setting*, where we allow as many examples as possible to fit the context window of the model; and (b) *instruction tuning*, where the model is tuned to follow specific instructions.

Emu2 achieves promising results in the few-shot setting on a wide range of vision-language tasks. For example, it demonstrates state-of-the-art few-shot performance on multiple visual question-answering datasets. We observe a performance improvement when the number of examples in context increases. Figure 1 illustrates Emu2’s strong multimodal reasoning capabilities for tasks in the wild, e.g., recognition and counting in a specific format. Emu2 also learns to follow visual prompting in context (e.g., the circles laid on the images in Figure 1), even although it struggles at a smaller scale or at zero shot.

As Emu2 is inherently equipped to handle interleaved text-image-video at both input and output, it serves as a powerful and versatile base model for diverse multimodal tasks, by following specific task instructions. For example, after instruct tuning with conversational data,

[†]Correspondence to wangxinlong@baai.ac.cn



Encoder

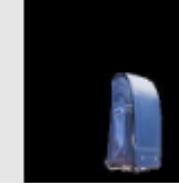
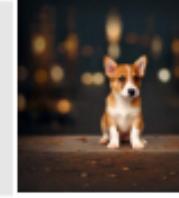
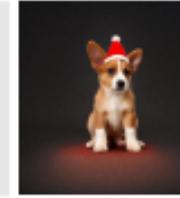
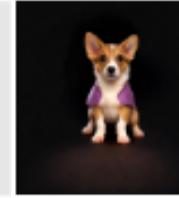
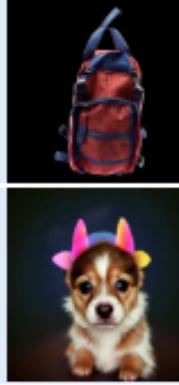
< s > [IMG] [] [] ... [/IMG] An emu egg that will hatch into a baby emu [IMG] [] [] ... [/IMG] < /s >

Generative Multimodal Model

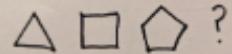


Figure 2. Overview of **Emu2** architecture. **Emu2** learns with a predict-the-next-element objective in multimodality. Each image in the multimodal sequence is tokenized into embeddings via a visual encoder, and then interleaved with text tokens for autoregressive modeling. The regressed visual embeddings will be decoded into an image or a video by a visual decoder.

In-context (few-shot) text & image completion

Input Prompt					Completion		
	[dog: 1, frisbee: 1].		[burger: 1, glass: 1, bottle: 1].		[cat: 3].		
	The text in the red circle: 'Rights'.		The text in the red circle: 'Ave'.		The text in the red circle: 'Do Not'.		
	motorcycle's wheel.		woman's feet.		car's license plate.		
a photo of a yellow backpack:		a photo of a blue backpack:		a photo of a red backpack:		a photo of a brown backpack:	
The subject A with a city in the background:		The subject A wearing a santa hat:		The subject A in a purple wizard outfit:		The subject A wearing a rainbow hat:	

Language and image understanding



Look at this sequence of three shapes. What shape should come as the fourth shape? Explain your reasoning with detailed descriptions of the first shapes.

The first shape is a triangle, the second shape is a square, and the third shape is a pentagon. To find the next shape in the sequence, we need to identify a pattern. In this case, the pattern is that each shape has one more side than the previous shape. Therefore, the next shape in the sequence should have six sides, which is a **hexagon**.



What is funny about this image?

The funny aspect of this image is that a **squirrel is holding a camera and appears to be taking a picture of a bird**. This is amusing because animals, such as squirrels and birds, do not typically use cameras or engage in human-like activities like photography. The scene is set up in a way that makes it look like the squirrel is intentionally capturing the bird's image, which adds a humorous twist to the scene. Additionally, **the bird is perched on a mushroom**, further enhancing the whimsical and playful nature of the image.

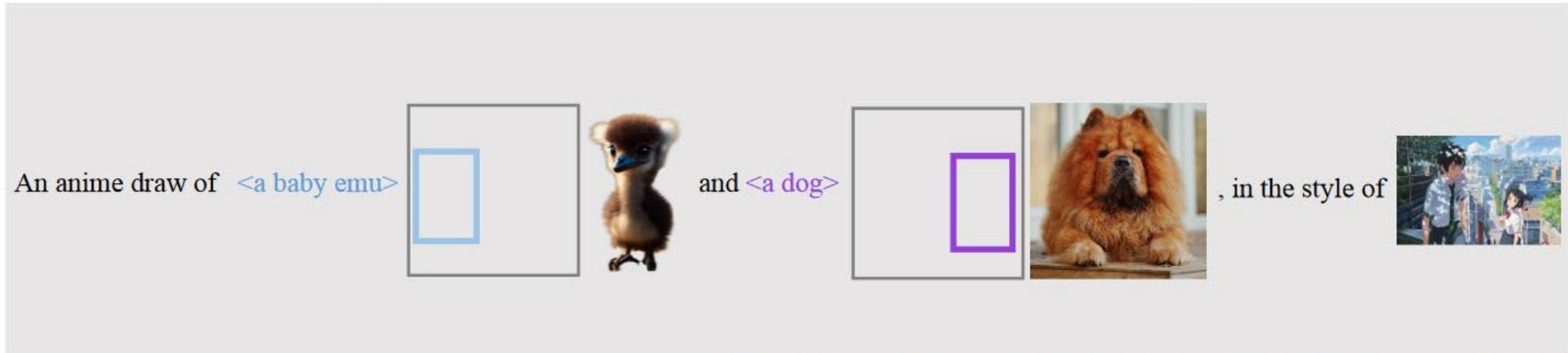
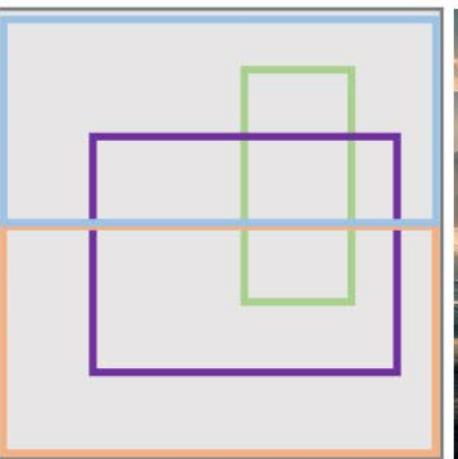
Versatile prompting



impressionist painting of an astronaut in a jungle



An image of <emu> wearing a big sunglasses on the beach



GPT-4V

OpenAI's multimodal LLM

“GPT-4 doesn't take videos as input directly, but we can use vision and the new 128K context window to describe the static frames of a whole video at once.”

<https://cookbook.openai.com/>



Gonzalo Espinoza Graham 💀 ✅

@geepytee

...

GPT-4V + TTS = AI Sports narrator ⚽

Passed every frame of a football video to gpt-4-vision-preview, and with some simple prompting asked to generate a narration

No edits, this is as it came out from the model (aka can be SO MUCH BETTER)



3:45 AM · Nov 7, 2023 · 2.6M Views

<https://x.com/geepytee/status/1721705524176257296?s=20>

Gemini (Dec 2023)

Google Deepmind's multimodal LLMs

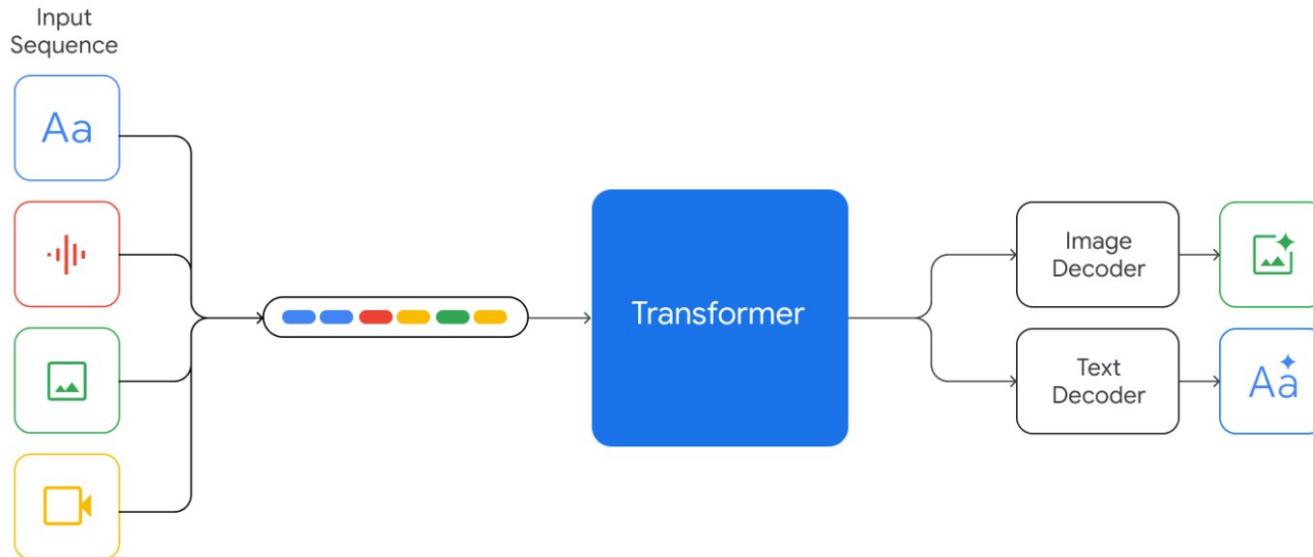


Figure 2 | Gemini supports interleaved sequences of text, image, audio, and video as inputs (illustrated by tokens of different colors in the input sequence). It can output responses with interleaved image and text.

<https://www.youtube.com/watch?v=jV1vkHv4zq8>

https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf

Gemini: A Family of Highly Capable Multimodal Models

Gemini Team, Google¹

This report introduces a new family of multimodal models, Gemini, that exhibit remarkable capabilities across image, audio, video, and text understanding. The Gemini family consists of Ultra, Pro, and Nano sizes, suitable for applications ranging from complex reasoning tasks to on-device memory-constrained use-cases. Evaluation on a broad range of benchmarks shows that our most-capable Gemini Ultra model advances the state of the art in 30 of 32 of these benchmarks — notably being the first model to achieve human-expert performance on the well-studied exam benchmark MMLU, and improving the state of the art in every one of the 20 multimodal benchmarks we examined. We believe that the new capabilities of Gemini models in cross-modal reasoning and language understanding will enable a wide variety of use cases and we discuss our approach toward deploying them responsibly to users.

1. Introduction

We present Gemini, a family of highly capable multimodal models developed at Google. We trained Gemini jointly across image, audio, video, and text data for the purpose of building a model with both strong generalist capabilities across modalities alongside cutting-edge understanding and reasoning performance in each respective domain.

Gemini 1.0, our first version, comes in three sizes: Ultra for highly-complex tasks, Pro for enhanced performance and deployability at scale, and Nano for on-device applications. Each size is specifically tailored to address different computational limitations and application requirements. We evaluate the performance of Gemini models on a comprehensive suite of internal and external benchmarks covering a wide range of language, coding, reasoning, and multimodal tasks.

Gemini advances state-of-the-art in large-scale language modeling (Anil et al., 2023; Brown et al., 2020; Chowdhery et al., 2023; Hoffmann et al., 2022; OpenAI, 2023a; Radford et al., 2019; Rae et al., 2021), image understanding (Alayrac et al., 2022; Chen et al., 2022; Dosovitskiy et al., 2020; OpenAI, 2023b; Reed et al., 2022; Yu et al., 2022a), audio processing (Radford et al., 2023; Zhang et al., 2023), and video understanding (Alayrac et al., 2022; Chen et al., 2023). It also builds on the work on sequence models (Sutskever et al., 2014), a long history of work in deep learning based on neural networks (LeCun et al., 2015), and machine learning distributed systems (Barham et al., 2022; Bradbury et al., 2018; Dean et al., 2012) that enable large-scale training.

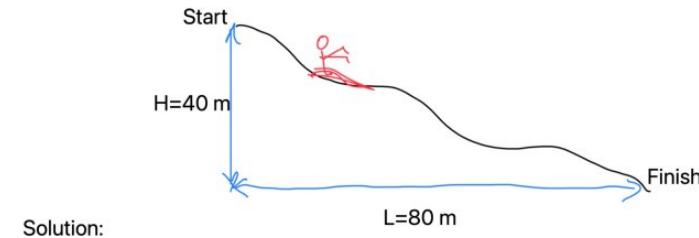
Our most capable model, Gemini Ultra, achieves new state-of-the-art results in 30 of 32 benchmarks we report on, including 10 of 12 popular text and reasoning benchmarks, 9 of 9 image understanding benchmarks, 6 of 6 video understanding benchmarks, and 5 of 5 speech recognition and speech translation benchmarks. Gemini Ultra is the first model to achieve human-expert performance on MMLU (Hendrycks et al., 2021a) — a prominent benchmark testing knowledge and reasoning via a suite of exams — with a score above 90%. Beyond text, Gemini Ultra makes notable advances on challenging multimodal reasoning tasks. For example, on the recent MMMU benchmark (Yue et al., 2023), that comprises questions about images on multi-discipline tasks requiring college-level subject

¹See Contributions and Acknowledgments section for full author list. Please send correspondence to gemini-1-report@google.com

Prompt

Here's a solution to a Physics problem by a student.

23. A skier slides down the frictionless slope as shown. What is the skier's speed at the bottom?



The conservation of energy means that the energy of the skier is same at the beginning and at the end.

$$\text{Start: } E = mgL$$

$$\text{End: } E = \frac{1}{2}mv^2$$

$$\Rightarrow mgL = \frac{1}{2}mv^2$$

$$\Rightarrow v = \sqrt{2gL} = \sqrt{2 \times 9.81 \times 80} = 39.6 \text{ m.s}^{-1}$$

Try to reason about the question step by step.

Did the student get the correct answer? If the solution is wrong, please explain what is wrong and solve the problem.

Make sure to use LaTeX for math and round off the final answer to two decimal places.

Response

The student did not get the correct answer. The student made a mistake in the calculation of the potential energy at the start of the slope. The potential energy at the start of the slope is given by mgh , where m is the mass of the skier, g is the acceleration due to gravity, and h is the height of the slope. The student used the length of the slope, L , instead of the height, H , in the calculation of the potential energy.

The correct solution is:

The conservation of energy means that the total energy of the skier is the same at the beginning and at the end of the slope.

At the start of the slope, the skier has only potential energy, which is given by mgh .

At the end of the slope, the skier has only kinetic energy, which is given by $\frac{1}{2}mv^2$

Therefore, we can write:

$$mgh = \frac{1}{2}mv^2$$

Cancelling out the mass m on both sides, we get:

$$gh = \frac{1}{2}v^2$$

Solving for v , we get:

$$v = \sqrt{2gh}$$

Substituting the given values, we get:

$$v = \sqrt{2 \times 9.81 \times 40} = 28.01 \text{ m/s}$$

Therefore, the skier's speed at the bottom of the slope is 28.01 m/s.

Generating videos

Diffusion models dominate

- One can use a GPT to generate video frame-by-frame and patch-by-patch, but limited context length is a bottleneck for longer videos
- U-Net denoisers scale better, but longer videos are still a challenge

How to control?

A text prompt goes only so far.

DragNUWA allows the user to specify control trajectories.

DRAGNUWA: FINE-GRAINED CONTROL IN VIDEO GENERATION BY INTEGRATING TEXT, IMAGE, AND TRAJECTORY

Shengming Yin^{1*} Chenfei Wu^{2*} Jian Liang³ Jie Shi³ Houqiang Li¹ Gong Ming² Nan Duan^{2†}

¹University of Science and Technology of China ²Microsoft Research Asia ³Peking University

{sheyin@mail.,lihq}@ustc.edu.cn, {chewu,migon,nanduan}@microsoft.com, {j.liang@stu.,jieshi@}pku.edu.cn

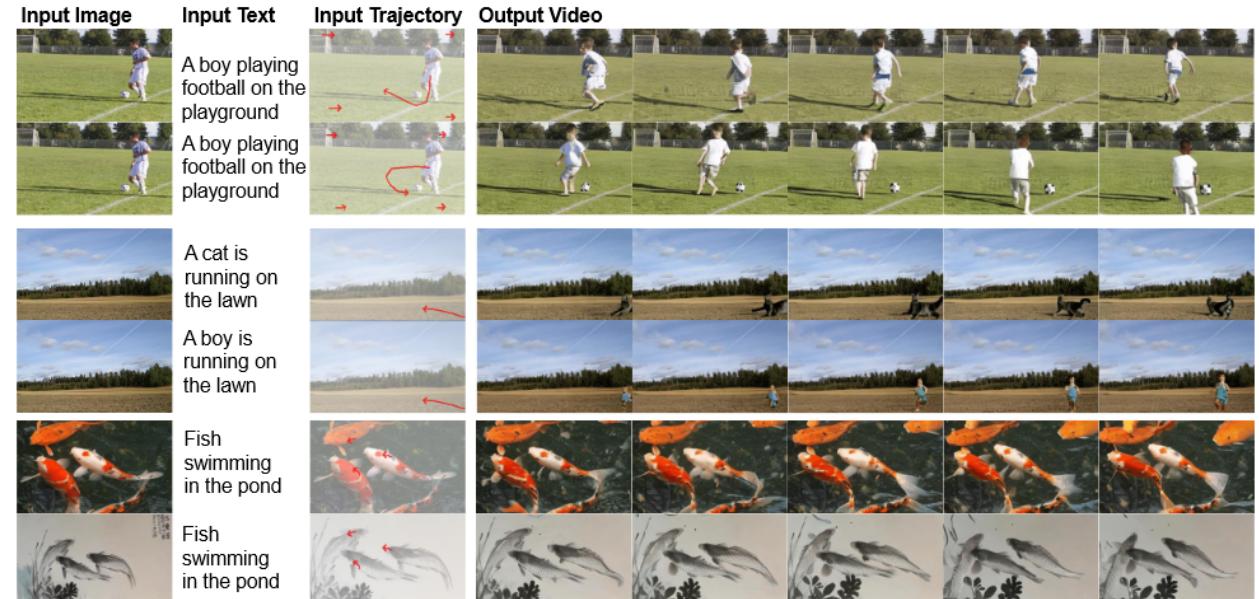
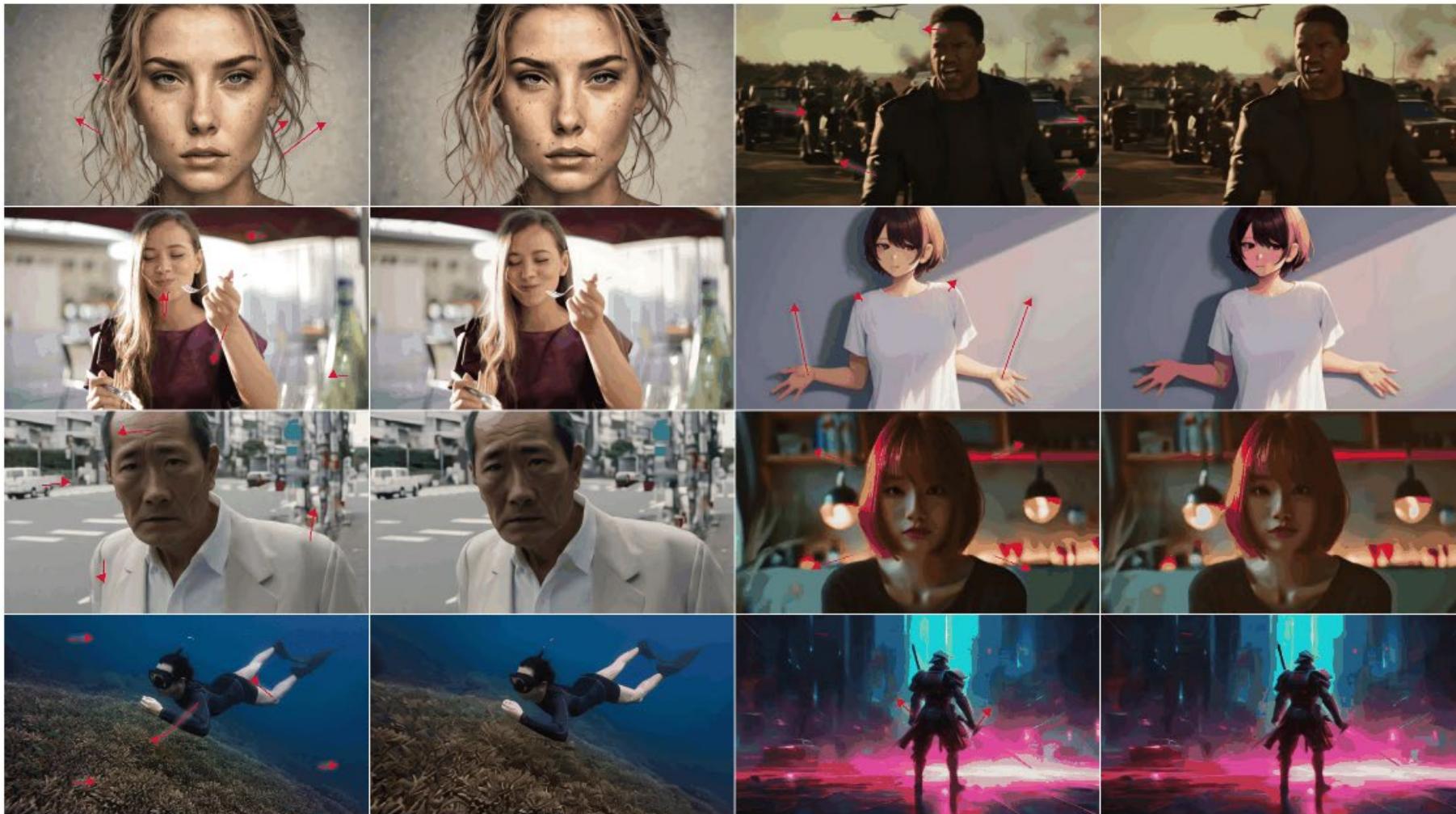


Figure 1: DragNUWA integrates text, image, and trajectory controls to achieve controllable video generation from semantic, spatial, and temporal perspectives. The three groups of examples demonstrate the impact of altering one control while keeping the other two fixed. The first group (Row 1-2) displays the control of complex trajectories, including complex motions (red curved arrows) and camera movements (red rightward arrows). The second group (Row 3-4) illustrates the influence of language control, pairing different text with the same image and trajectory to achieve the effect of introducing new objects in the images. The third group (Row 5-6) demonstrates the impact of image control, showcasing the generation of both real-world and artistic videos.

<https://arxiv.org/abs/2308.08089>

DragNUWA 1.6 will come soon (Updated on Jan 30, 2024)

- Consistency in human facial representation



Stable Video Diffusion

Released in November 2023

Image-to-video, i.e., conditioned both using text and a reference image

Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets

Andreas Blattmann* Tim Dockhorn* Sumith Kulal* Daniel Mendelevitch
Maciej Kilian Dominik Lorenz Yam Levi Zion English Vikram Voleti
Adam Letts Varun Jampani Robin Rombach
Stability AI



"A robot dj is playing the turntables, in heavy raining futuristic tokyo, rooftop, sci-fi, fantasy"



"An exploding cheese house"



"A fat rabbit wearing a purple robe walking through a fantasy landscape"



Figure 1. Stable Video Diffusion samples. Top: Text-to-Video generation. Middle: (Text-to-)Image-to-Video generation. Bottom: Multi-view synthesis via Image-to-Video finetuning.

Abstract

We present Stable Video Diffusion — a latent video diffusion model for high-resolution, state-of-the-art text-to-video and image-to-video generation. Recently, latent diffusion models trained for 2D image synthesis have been turned into generative video models by inserting temporal layers and finetuning them on small, high-quality video datasets. However, training methods in the literature vary widely, and the field has yet to agree on a unified strategy for cu-

rating video data. In this paper, we identify and evaluate three different stages for successful training of video LDMs: text-to-image pretraining, video pretraining, and high-quality video finetuning. Furthermore, we demonstrate the necessity of a well-curated pretraining dataset for generating high-quality videos and present a systematic curation process to train a strong base model, including captioning and filtering strategies. We then explore the impact of finetuning our base model on high-quality data and train a text-to-video model that is competitive with closed-source video generation. We also show that our base

* Equal contributions.

↪ You reposted



Steve Mills ✅ @SteveMills · Nov 24, 2023

SDV (Stable Diffusion Image To Video) Google Colab available here for anyone who wants to play along at home.

colab.research.google.com/github/mkshing...

Generates 3 seconds of video in about 30 seconds using an A100 GPU on Colab+

No control of the actual video in any way at all (yet), but it...

Show more



https://colab.research.google.com/github/mkshing/notebooks/blob/main/stable_video_diffusion_img2vid.ipynb



AI Magic Tools

Gen-2

Create videos in any style you can imagine with Text to Video generation. If you can imagine it, you can generate it.

Try Runway for Free >

<https://runwayml.com/ai-magic-tools/gen-2/>





niceaunties ✅ @niceaunties · Dec 23, 2023

...

'Bear to Bull' (B2B)

A holiday parade of free animals in Auntieverse. Happy Holidays everyone



A prompt share and conversational collab with [@pancakepie360](#). Two videos, each made from own interpretation. Details in [🧵](#)



<https://x.com/niceaunties/status/1738592581448344038?s=20>

https://youtube.com/shorts/UX2uqSlf_2w?si=T333ktMr8vG_SeLU



niceaunties ✅ @niceaunties · Dec 23, 2023

Theme: pancake + auntie

Prompt : pancake + auntie

Images: dalle3, [@Magnific_AI](#) x auntie

Animation: [@runwayml](#) x auntie

Edit: [@capcutapp](#) x auntie

Music : [@suno_ai_](#) x auntie

Location: Auntieverse



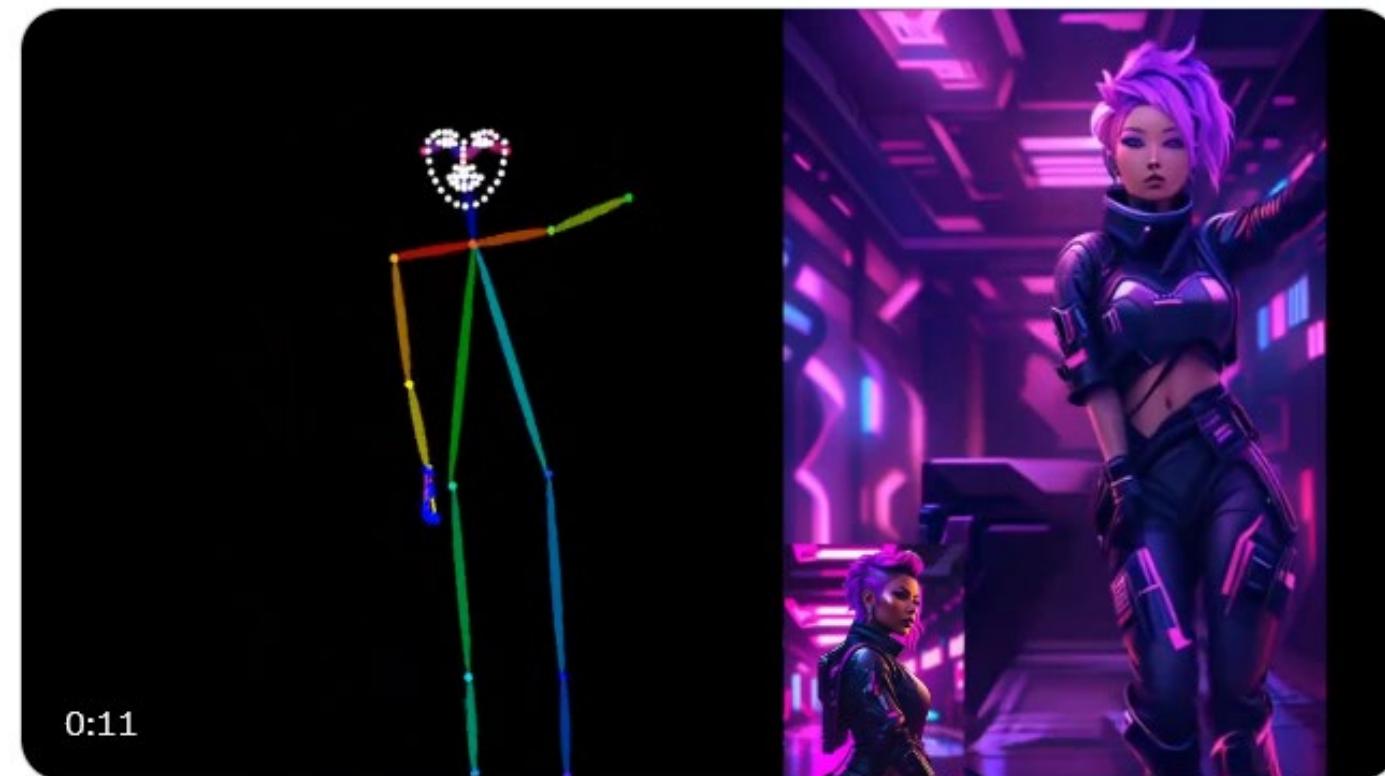
AK ✅ @_akhaliq · Dec 28, 2023 · 📝

...

Alibaba releases DreaMoving demo on Hugging Face

A Human Video Generation Framework
based on Diffusion Models

demo: [huggingface.co/spaces/jiayong...](https://huggingface.co/spaces/jiayong/DreaMoving)



23

427

1.7K

180K



https://x.com/_akhaliq/status/1740380744726594003?s=20



Steve Mills  @SteveMills · Dec 6, 2023

...

Imagine a "Channel" in the future that creates an endless stream of 100% new, totally compelling visuals and music on demand.

Today's experiment involved an exploration of [@runwayml](#) Gen-2 with AI driven prompt discovery. 100% text to image.

Sound On 

Music from Google...

Show more

<https://www.youtube.com/watch?v=ZYXUi51nTc>



104

170

708

89K



Steve Mills  @SteveMills · 4h

...

The Channel is [Ainfinite.tv](#) Live NOW!

You reposted



AK ✅ @_akhaliq · Jan 8

...

Microsoft just released DragNUWA on Hugging Face

Fine-grained Control in Video Generation by Integrating Text, Image, and Trajectory

demo: huggingface.co/spaces/yinsmin...

DragNUWA is a video generation model that utilizes text, images, and trajectory as three essential control factors...

Show more



https://x.com/_akhaliq/status/1744405937824125211?s=20

Open video models empowering new innovations

ClipDraw-style Bezier curve
renderings guided by a video
Diffusion model

<https://livesketch.github.io/>

Breathing Life Into Sketches Using Text-to-Video Priors

Rinon Gal^{*,1,2}

Yael Vinker^{*,1}

Yuval Alaluf¹

Amit Bermano¹

Daniel Cohen-Or¹

Ariel Shamir³

Gal Chechik²

¹Tel-Aviv University

²NVIDIA

³Reichman University

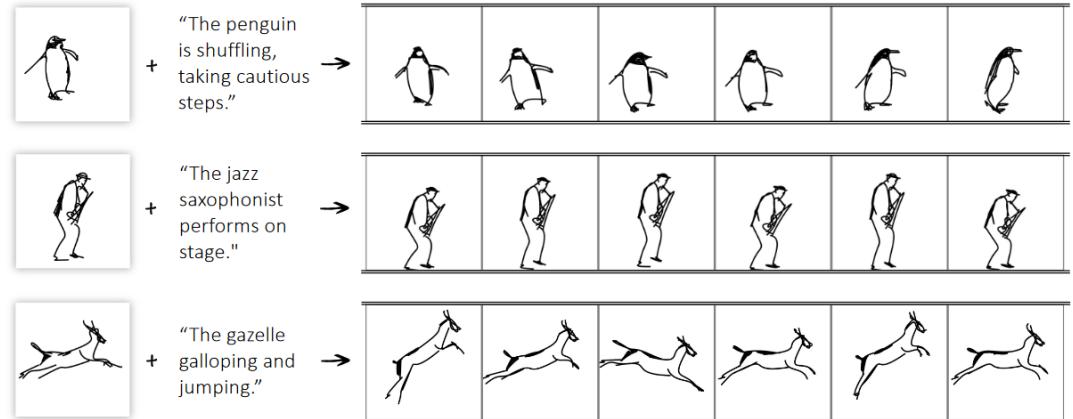


Figure 1. Given a still sketch in vector format and a text prompt describing a desired action, our method automatically animates the drawing with respect to the prompt. Please see the full animations in our project page: <https://livesketch.github.io/>

Abstract

A sketch is one of the most intuitive and versatile tools humans use to convey their ideas visually. An animated sketch opens another dimension to the expression of ideas and is widely used by designers for a variety of purposes. Animating sketches is a laborious process, requiring extensive experience and professional design skills. In this work, we present a method that automatically adds motion to a single-subject sketch (hence, “breathing life into it”), merely by providing a text prompt indicating the desired motion. The output is a short animation provided in vector representation, which can be easily edited. Our method does not require extensive training, but instead leverages the motion prior of a large pretrained text-to-video diffusion model using a score-distillation loss to guide the placement of strokes. To promote natural and smooth motion and to better preserve the sketch’s appearance, we model the learned motion through two components. The first governs small local deformations and the second controls global affine transformations. Surprisingly, we find that even models that struggle to generate sketch videos on their own can still serve as a useful backbone for animating abstract representations.

1. Introduction

Sketches serve as a fundamental and intuitive tool for visual expression and communication [3, 20, 26]. Sketches capture the essence of visual entities with a few strokes, allowing humans to communicate abstract visual ideas. In this paper, we propose a method to “breathe life” into a static sketch by generating semantically meaningful short videos from it. Such animations can be useful for storytelling, illustrations, websites, presentations, and just for fun.

Animating sketches using conventional tools (such as Adobe Animate and Toon Boom) is challenging even for experienced designers [76], requiring specific artistic expertise. Hence, long-standing research efforts in computer graphics sought to develop automatic tools to simplify this process. However, these tools face multiple hurdles, such as a need to identify the semantic component of the sketch, or learning to create motion that appears natural. As such, existing methods commonly rely on user-annotated skeletal key points [17, 74] or user-provided reference motions that align with the sketch semantics [9, 76, 88].

In this work, we propose to bring a given static sketch to life, based on a textual prompt, without the need for any human annotations or explicit reference motions. We do so by leveraging a pretrained text-to-video diffusion model [43]. Several recent works propose using the prior of such mod-

*Indicates Equal Contribution. Order determined by coin flip.

Generating audio

Audio generation

- Early attempts: GANs, RNNs. No conditioning on text.
- 2020: OpenAI Jukebox: LLM + AE + VQ tokenization. Results mindblowing at the time, but quite noisy due to the VQ imprecision
- 2022: Transformers work well (LLM + AE + RVQ tokenization)
- 2023: Diffusion models excel. Instead of CLIP, one uses CLAP (Contrastive Language Audio Pretraining) embeddings for the classifier-free guidance



[Submitted on 16 Nov 2018]

Generating Albums with SampleRNN to Imitate Metal, Rock, and Punk Bands

CJ Carr, Zack Zukowski

This early example of neural synthesis is a proof-of-concept for how machine learning can drive new types of music software. Creating music can be as simple as specifying a set of music influences on which a model trains. We demonstrate a method for generating albums that imitate bands in experimental music genres previously unrealized by traditional synthesis techniques (e.g. additive, subtractive, FM, granular, concatenative). Raw audio is generated autoregressively in the time-domain using an unconditional SampleRNN. We create six albums this way. Artwork and song titles are also generated using materials from the original artists' back catalog as training data. We try a fully-automated method and a human-curated method. We discuss its potential for machine-assisted production.

Comments: 3 pages

Subjects: **Sound (cs.SD)**; Audio and Speech Processing (eess.AS)

Journal reference: Proceedings of the 6th International Workshop on Musical Metacreation (MUME 2018)

Cite as: [arXiv:1811.06633 \[cs.SD\]](#)

(or [arXiv:1811.06633v1 \[cs.SD\]](#) for this version)



Æternal Reborous - neural metal - DADABOTS (2018)



DADABOTS
31,1 t. tilaajaa

Tilaa

Like 50



Jaa

Lataa

Klippi

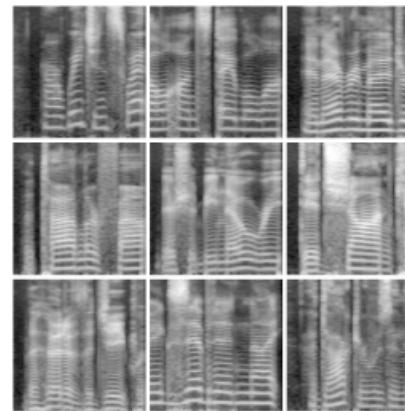
Tallenna

...

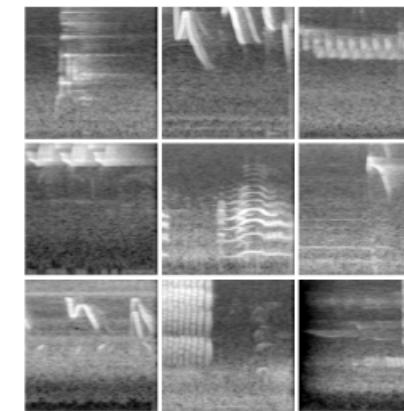
<https://www.youtube.com/watch?v=4zyPl0IEjTg>

Audio and GANs: WaveGAN

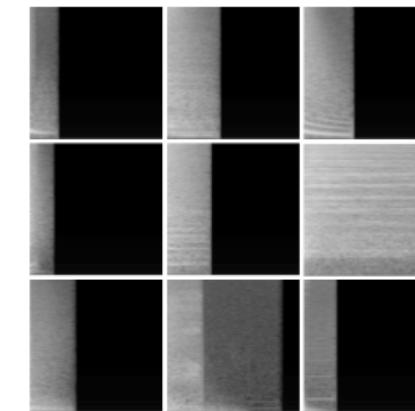
Real



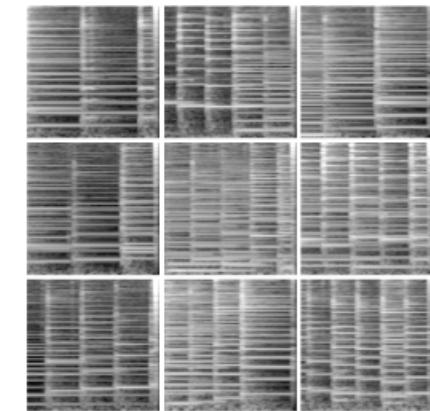
Speech



Birds

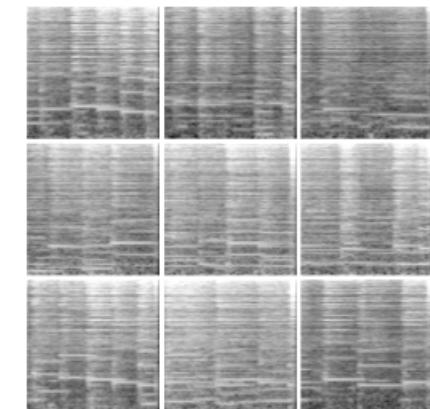
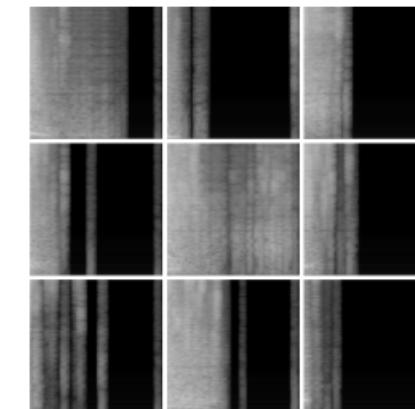
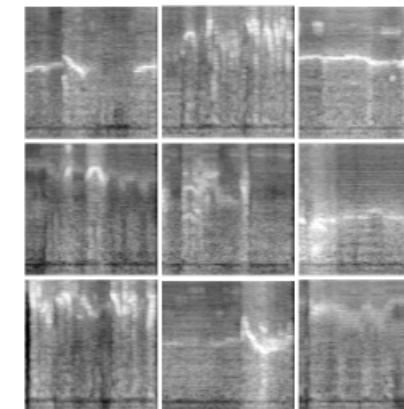
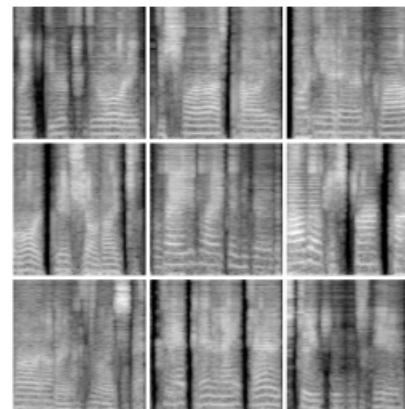


Drums



Piano

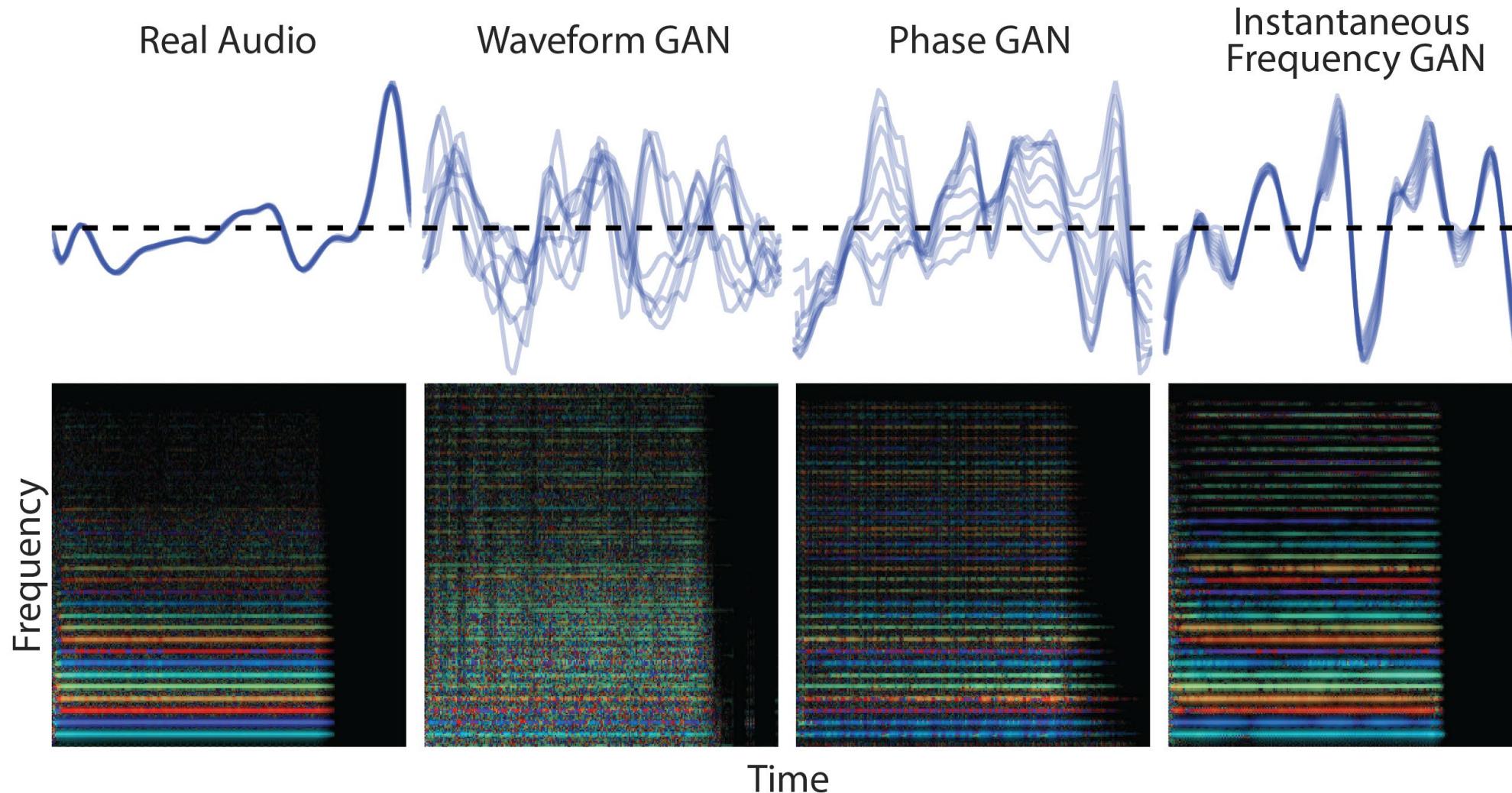
WaveGAN



Code: <https://github.com/chrisdonahue/wavegan>, Interactive demo: <https://chrisdonahue.com/wavegan/>



GANSynth (2019)



<https://magenta.tensorflow.org/gansynth>



AudioCraft: generating high-quality audio and music from text

AudioCraft is a single-stop code base for all your generative audio needs: music, sound effects, and compression after training on raw audio signals. We have released controllable and high-quality models for music and audio generation from text inputs. It represents significant progress in the development of interactive AI systems enabling people to easily and naturally co-create with AI models.

<https://ai.meta.com/resources/models-and-libraries/audiocraft/>

The models

Open source, single-stage models, built with simplicity in mind

↗ [MusicGen paper](#)

↗ [AudioGen paper](#)

↗ [EnCodec paper](#)

↗ [Multi-Band Diffusion paper](#)

AudioCraft powers our audio compression and generation research and consists of three models: MusicGen, AudioGen, and EnCodec. MusicGen, which was trained with Meta-owned and specifically licensed music, generates music from text-based user inputs, while AudioGen, trained on public sound effects, generates audio from text-based user inputs.

EnCodec, typically used foundationally in building MusicGen and AudioGen, is a state-of-the-art, real-time, high-fidelity audio codec that leverages neural networks to compress any kind of audio and reconstruct the original signal with high-fidelity. We further propose a diffusion-based approach to EnCodec to reconstruct the audio from the compressed representation with fewer artifacts.

September 2022: AudioGen

Generates both speech and audio effects.

LLM + AE and RVQ for tokenization

Uses classifier-free guidance for adjusting the quality-diversity tradeoff (had not been done before in LLM-based autoregressive sampling)

<https://felixkreuk.github.io/audiogen/>

AUDIOGEN: TEXTUALLY GUIDED AUDIO GENERATION

Felix Kreuk¹, Gabriel Synnaeve¹, Adam Polyak¹, Uriel Singer¹, Alexandre Défossez¹, Jade Copet¹, Devi Parikh¹, Yaniv Taigman¹, Yossi Adi^{1,2}

¹FAIR Team, Meta AI

²The Hebrew University of Jerusalem

felixkreuk@meta.com

<https://arxiv.org/abs/2209.15352>

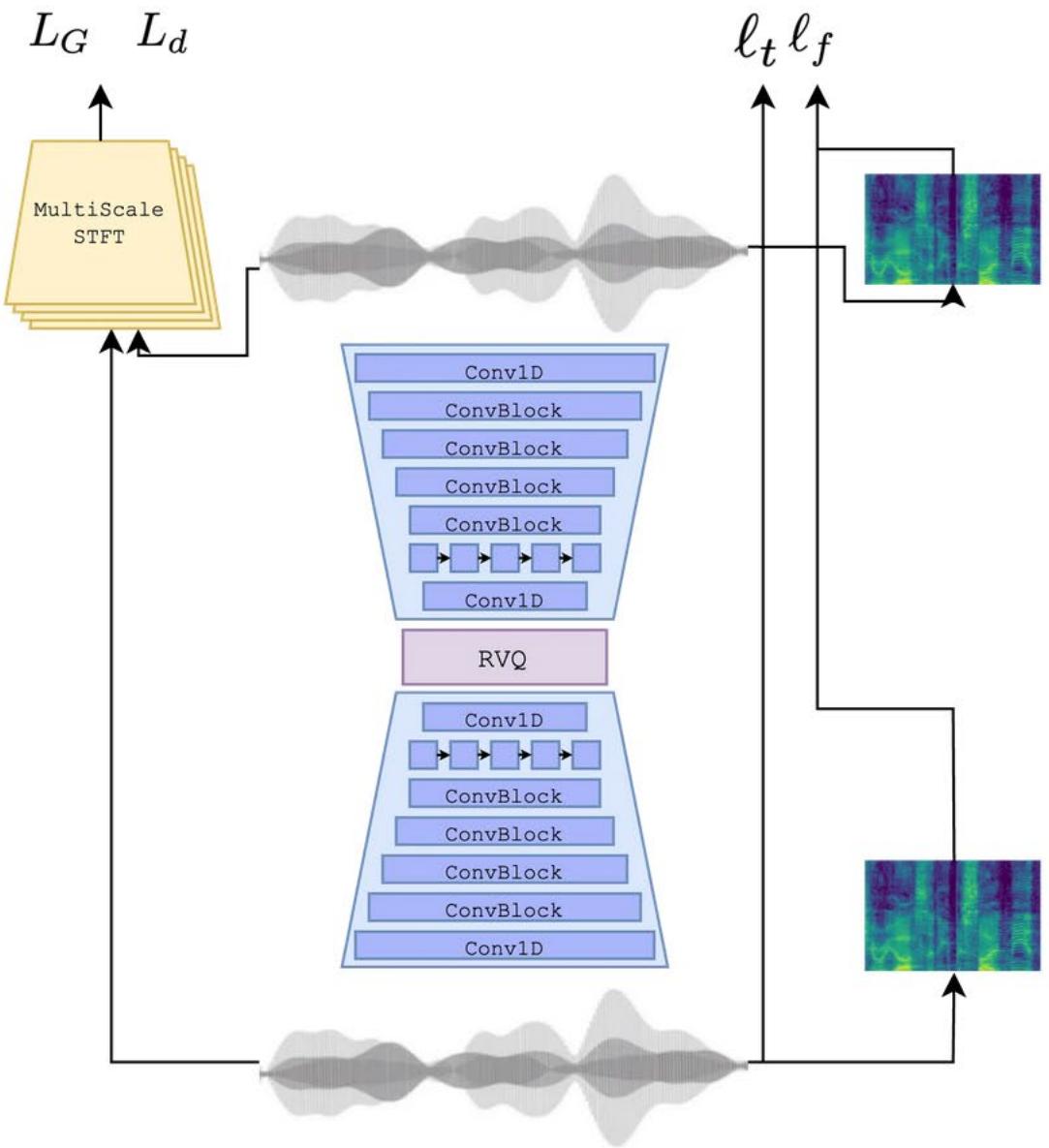
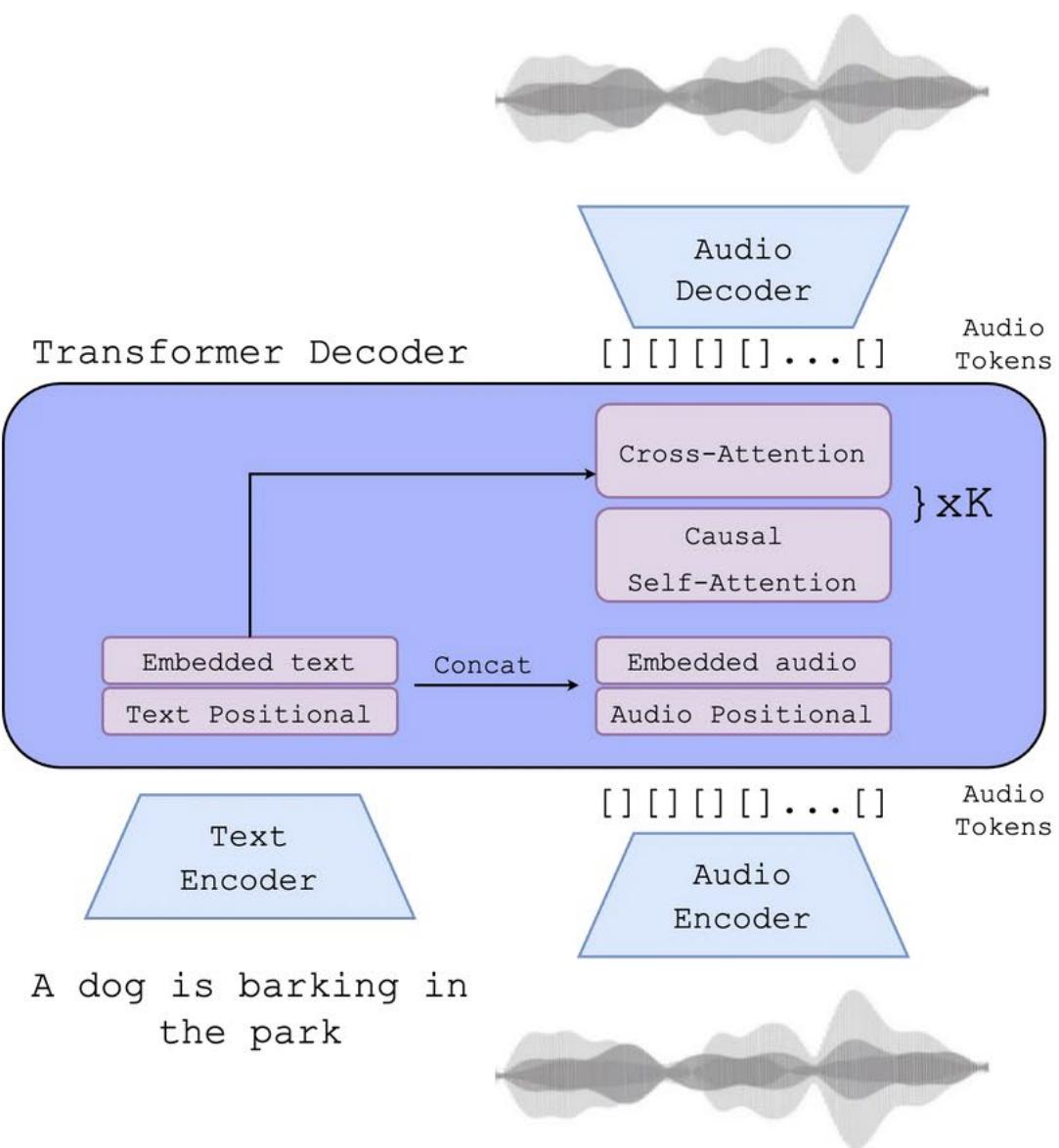
ABSTRACT

We tackle the problem of generating audio samples conditioned on descriptive text captions. In this work, we propose AUDIOPEN, an auto-regressive generative model that generates audio samples conditioned on text inputs. AUDIOPEN operates on a learned discrete audio representation. The task of text-to-audio generation poses multiple challenges. Due to the way audio travels through a medium, differentiating “objects” can be a difficult task (e.g., separating multiple people simultaneously speaking). This is further complicated by real-world recording conditions (e.g., background noise, reverberation, etc.). Scarce text annotations impose another constraint, limiting the ability to scale models. Finally, modeling high-fidelity audio requires encoding audio at high sampling rate, leading to extremely long sequences. To alleviate the aforementioned challenges we propose an augmentation technique that mixes different audio samples, driving the model to internally learn to separate multiple sources. We curated 10 datasets containing different types of audio and text annotations to handle the scarcity of text-audio data points. For faster inference, we explore the use of multi-stream modeling, allowing the use of shorter sequences while maintaining a similar bitrate and perceptual quality. We apply classifier-free guidance to improve adherence to text. Comparing to the evaluated baselines, AUDIOPEN outperforms over both objective and subjective metrics. Finally, we explore the ability of the proposed method to generate audio continuation conditionally and unconditionally. Samples: <https://felixkreuk.github.io/audiogen>.

1 INTRODUCTION

Neural generative models have challenged the way we create digital content. From generating high-quality images (Karras et al., 2019; Park et al., 2019) and speech (Ren et al., 2021; Oord et al., 2016), through generating long textual spans (Brown et al., 2020; Zhang et al., 2022), to the recently proposed text prompted image generation (Ramesh et al., 2022; Rombach et al., 2022), these models have shown impressive results. This begs the question *what would be the audio equivalent to textually guided generative models?* From generating soundscapes to music or speech, a solution to this problem that is high fidelity, controllable, and diverse in its outputs, would be a useful addition to the modern toolbox of creators of movies, video games, and any virtual environments.

While image generation and audio generation have a lot in common, there are a few key differences. Audio is intrinsically a one dimensional signal and thus has less degrees of freedom to differentiate overlapping “objects” (Capon, 1969; Frost, 1972). Real-world audio inherently has reverberations, which makes the task of differentiating objects from the surrounding environment even harder. Moreover, psychoacoustic and psychovisual properties differ, for instance hearing “resolution” (equal-loudness) is U-shaped in frequencies with a dip at 4kHz and bump at 8kHz (Suzuki et al., 2003). Last but not least, the availability of audio data with textual descriptions is orders of magnitude below that of text-image paired data. This makes generating unseen audio compositions a hard task (e.g. generating an audio equivalent of an image of “an astronaut riding a horse in space”).



Oct 2022: EnCodec

Further development of the AudioGen
AE + RVQ tokenizer.

Many later papers use this as a good
off-the-shelf audio tokenizer

High Fidelity Neural Audio Compression

Alexandre Défossez*
Meta AI, FAIR Team, Paris, France

defossez@meta.com

Jade Copet*
Meta AI, FAIR Team, Paris, France

jadecopet@meta.com

Gabriel Synnaeve†
Meta AI, FAIR Team, Paris, France

gab@meta.com

Yossi Adi†
Meta AI, FAIR Team, Tel-Aviv, Israel

adiyoss@meta.com

Abstract

We introduce a state-of-the-art real-time, high-fidelity, audio codec leveraging neural networks. It consists in a streaming encoder-decoder architecture with quantized latent space trained in an end-to-end fashion. We simplify and speed-up the training by using a single multiscale spectrogram adversary that efficiently reduces artifacts and produce high-quality samples. We introduce a novel loss balancer mechanism to stabilize training: the *weight* of a loss now defines the fraction of the overall gradient it should represent, thus decoupling the choice of this hyper-parameter from the typical scale of the loss. Finally, we study how lightweight Transformer models can be used to further compress the obtained representation by up to 40%, while staying faster than real time. We provide a detailed description of the key design choices of the proposed model including: training objective, architectural changes and a study of various perceptual loss functions. We present an extensive subjective evaluation (MUSHRA tests) together with an ablation study for a range of bandwidths and audio domains, including speech, noisy-reverberant speech, and music. Our approach is superior to the baselines methods across all evaluated settings, considering both 24 kHz monophonic and 48 kHz stereophonic audio. Code and models are available at github.com/facebookresearch/encodec.

1 Introduction

Recent studies suggest that streaming audio and video have accounted for the majority of the internet traffic in 2021 (82% according to [Cisco, 2021](#)). With the internet traffic expected to grow, audio compression is an increasingly important problem. In lossy signal compression we aim at minimizing the bitrate of a sample while also minimizing the amount of distortion according to a given metric, ideally correlated with human perception. Audio codecs typically employ a carefully engineered pipeline combining an encoder and a decoder to remove redundancies in the audio content and yield a compact bitstream. Traditionally, this is achieved by decomposing the input with a signal processing transform and trading off the quality of the components that are less likely to influence perception. Leveraging neural networks as trained transforms via an encoder-decoder mechanism has been explored by [Morishima et al. \(1990\)](#); [Rippel et al. \(2019\)](#); [Zeghidour et al. \(2021\)](#). Our research work is in the continuity of this line of work, with a focus on audio signals.

The problems arising in lossy neural compression models are twofold: first, the model has to represent a wide range of signals, such as not to overfit the training set or produce artifact laden audio outside its comfort zone. We solve this by having a large and diverse training set (described in Section [4.1](#)), as well as discriminator networks (see Section [3.4](#)) that serve as perceptual losses, which we study extensively in Section [4.5.1](#), Table [2](#). The other problem is that of compressing efficiently, both in compute time and in size.

*,†Equal contribution.

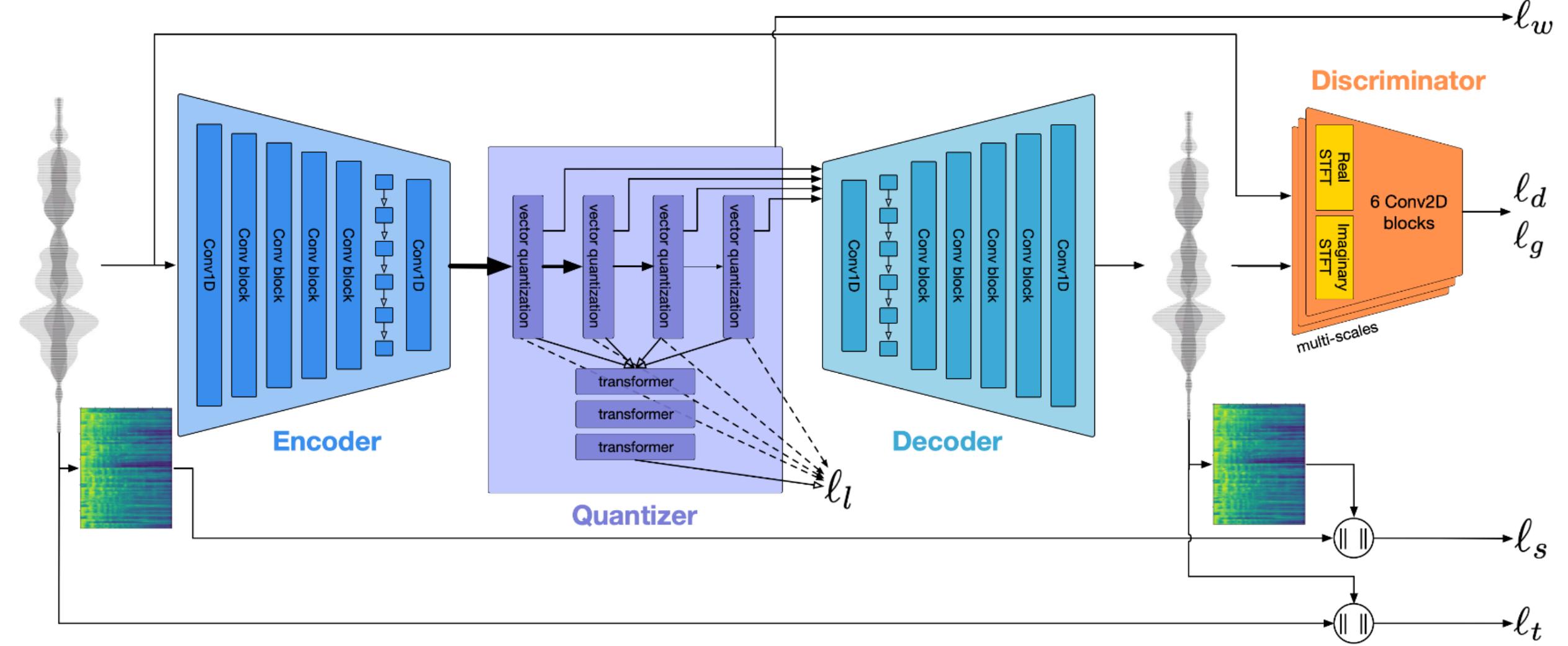


Figure 1: ENCODEC : an encoder decoder codec architecture which is trained with reconstruction (ℓ_f and ℓ_t) as well as adversarial losses (ℓ_g for the generator and ℓ_d for the discriminator). The residual vector quantization commitment loss (ℓ_w) applies only to the encoder. Optionally, we train a small Transformer language model for entropy coding over the quantized units with ℓ_l , which reduces bandwidth even further.

January 2023: MusicLM

- Google joins the race with MusicLM
- Basically still the same “LLM trained with text and audio tokens” approach

MusicLM: Generating Music From Text

Andrea Agostinelli ^{*1} Timo I. Denk ^{*1}
Zalán Borsos ¹ Jesse Engel ¹ Mauro Verzetti ¹ Antoine Caillon ² Qingqing Huang ¹ Aren Jansen ¹
Adam Roberts ¹ Marco Tagliasacchi ¹ Matt Sharifi ¹ Neil Zeghidour ¹ Christian Frank ¹

Abstract

We introduce MusicLM, a model for generating high-fidelity music from text descriptions such as *“a calming violin melody backed by a distorted guitar riff”*. MusicLM casts the process of conditional music generation as a hierarchical sequence-to-sequence modeling task, and it generates music at 24 kHz that remains consistent over several minutes. Our experiments show that MusicLM outperforms previous systems both in audio quality and adherence to the text descriptions. Moreover, we demonstrate that MusicLM can be conditioned on both text and a melody in that it can transform whistled and hummed melodies according to the style described in a text caption. To support future research, we publicly release MusicCaps, a dataset composed of 5.5k music-text pairs, with rich text descriptions provided by human experts. google-research.github.io/seanet/musiclm/examples

1. Introduction

Conditional neural audio generation covers a wide range of applications, ranging from text-to-speech (Zen et al., 2013; van den Oord et al., 2016) to lyrics-conditioned music generation (Dhariwal et al., 2020) and audio synthesis from MIDI sequences (Hawthorne et al., 2022b). Such tasks are facilitated by a certain level of temporal alignment between the conditioning signal and the corresponding audio output. In contrast, and inspired by progress in text-to-image generation (Ramesh et al., 2021; 2022; Saharia et al., 2022; Yu et al., 2022), recent work has explored generating audio from sequence-wide, high-level captions (Yang et al., 2022; Kreuk et al., 2022) such as *“whistling with wind blowing”*. While generating audio from such coarse captions represents a breakthrough, these models remain limited to simple acoustic scenes, consisting of few acoustic events over a

period of seconds. Hence, turning a single text caption into a rich audio sequence with long-term structure and many stems, such as a music clip, remains an open challenge.

AudioLM (Borsos et al., 2022) has recently been proposed as a framework for audio generation. Casting audio synthesis as a language modeling task in a discrete representation space, and leveraging a hierarchy of coarse-to-fine audio discrete units (or *tokens*), AudioLM achieves both high-fidelity and long-term coherence over dozens of seconds. Moreover, by making no assumptions about the content of the audio signal, AudioLM learns to generate realistic audio from audio-only corpora, be it speech or piano music, without any annotation. The ability to model diverse signals suggests that such a system could generate richer outputs if trained on the appropriate data.

Besides the inherent difficulty of synthesizing high-quality and coherent audio, another impeding factor is the scarcity of paired audio-text data. This is in stark contrast with the image domain, where the availability of massive datasets contributed significantly to the remarkable image generation quality that has recently been achieved (Ramesh et al., 2021; 2022; Saharia et al., 2022; Yu et al., 2022). Moreover, creating text descriptions of general audio is considerably harder than describing images. First, it is not straightforward to unambiguously capture with just a few words the salient characteristics of either acoustic scenes (e.g., the sounds heard in a train station or in a forest) or music (e.g., the melody, the rhythm, the timbre of vocals and the many instruments used in accompaniment). Second, audio is structured along a temporal dimension which makes sequence-wide captions a much weaker level of annotation than an image caption.

In this work, we introduce MusicLM, a model for generating high-fidelity music from text descriptions. MusicLM leverages AudioLM’s multi-stage autoregressive modeling as the generative component, while extending it to incorporate text conditioning. To address the main challenge of paired data scarcity, we rely on MuLan (Huang et al., 2022), a joint music-text model that is trained to project music and its corresponding text description to representations close to each other in an embedding space. This shared embedding space eliminates the need for captions at training time alto-

^{*}Equal contribution ¹Google Research ²IRCAM - Sorbonne Université (work done while interning at Google). Correspondence to: Christian Frank <chfrank@google.com>.



<https://google-research.github.io/seanet/musictm/examples/>

MusicLM: Generating Music From Text

| [paper](#) | [dataset](#) |

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, Christian Frank
Google Research

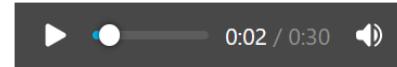
Abstract We introduce MusicLM, a model generating high-fidelity music from text descriptions such as "*a calming violin melody backed by a distorted guitar riff*". MusicLM casts the process of conditional music generation as a hierarchical sequence-to-sequence modeling task, and it generates music at 24 kHz that remains consistent over several minutes. Our experiments show that MusicLM outperforms previous systems both in audio quality and adherence to the text description. Moreover, we demonstrate that MusicLM can be conditioned on both text and a melody in that it can transform whistled and hummed melodies according to the style described in a text caption. To support future research, we publicly release MusicCaps, a dataset composed of 5.5k music-text pairs, with rich text descriptions provided by human experts.

Audio Generation From Rich Captions

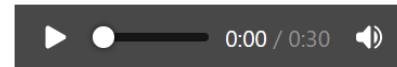
Caption

The main soundtrack of an arcade game. It is fast-paced and upbeat, with a catchy electric guitar riff. The music is repetitive and easy to remember, but with unexpected sounds, like cymbal crashes or drum rolls.

Generated audio

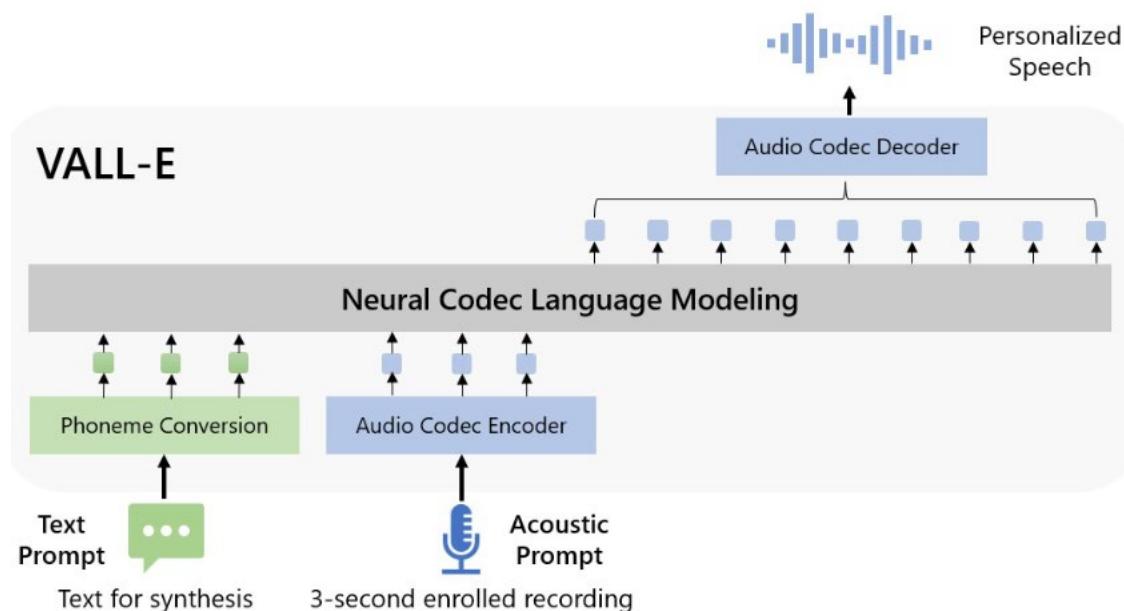


A fusion of reggaeton and electronic dance music, with a spacey, otherworldly sound. Induces the experience of being lost in space, and the music would be designed to evoke a sense of wonder and awe, while being danceable.



January 2023: VALL-E

Microsoft's take on the DALL-E 1 but for audio (speech), i.e., LLM + AE + RVQ



Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers

Chengyi Wang* Sanyuan Chen* Yu Wu* Ziqiang Zhang Long Zhou Shujie Liu
Zhuo Chen Yanqing Liu Huaming Wang Jinyu Li Lei He Sheng Zhao Furu Wei
Microsoft

<https://github.com/microsoft/unilm>

<https://arxiv.org/abs/2301.02111>

Abstract

We introduce a language modeling approach for text to speech synthesis (TTS). Specifically, we train a *neural codec language model* (called VALL-E) using discrete codes derived from an off-the-shelf neural audio codec model, and regard TTS as a conditional language modeling task rather than continuous signal regression as in previous work. During the pre-training stage, we scale up the TTS training data to 60K hours of English speech which is hundreds of times larger than existing systems. VALL-E emerges *in-context learning* capabilities and can be used to synthesize high-quality personalized speech with only a 3-second enrolled recording of an unseen speaker as an acoustic prompt. Experiment results show that VALL-E significantly outperforms the state-of-the-art zero-shot TTS system in terms of speech naturalness and speaker similarity. In addition, we find VALL-E could preserve the speaker's emotion and acoustic environment of the acoustic prompt in synthesis. See <https://aka.ms/valle> for demos of our work.

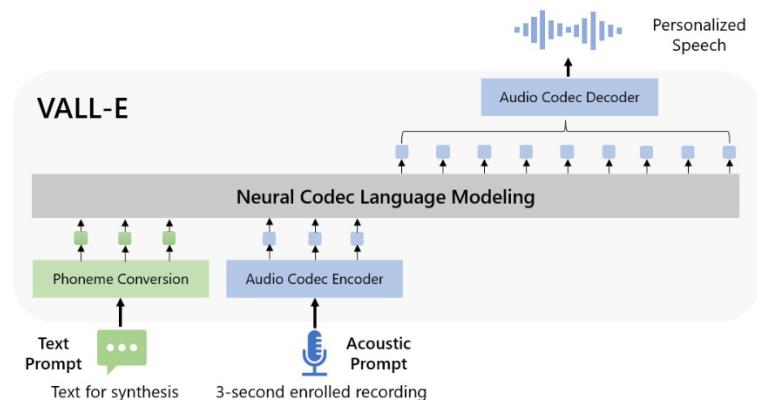


Figure 1: The overview of VALL-E. Unlike the previous pipeline (e.g., phoneme → mel-spectrogram → waveform), the pipeline of VALL-E is phoneme → discrete code → waveform. VALL-E generates the discrete audio codec codes based on phoneme and acoustic code prompts, corresponding to the target content and the speaker's voice. VALL-E directly enables various speech synthesis applications, such as zero-shot TTS, speech editing, and content creation combined with other generative AI models like GPT-3 [Brown et al., 2020].

*These authors contributed equally to this work. Correspondence: {yuwu1,shujie.liu,fuwei}@microsoft.com

June 2023: MusicGen

Like AudioGen, but optimized for music

Demo:

[https://huggingface.co/spaces/facebook
/MusicGen](https://huggingface.co/spaces/facebook/MusicGen)

Jade Copet^{♦◇} Felix Kreuk^{♦◇} Itai Gat Tal Remez David Kant
Gabriel Synnaeve ◇ Yossi Adi[◇] Alexandre Défossez ◇

♦: equal contributions, ◇: core team

Meta AI

{jadecopet, felixkreuk, adiyoss}@meta.com

Abstract

We tackle the task of conditional music generation. We introduce MUSICGEN, a single Language Model (LM) that operates over several streams of compressed discrete music representation, i.e., tokens. Unlike prior work, MUSICGEN is comprised of a single-stage transformer LM together with efficient token interleaving patterns, which eliminates the need for cascading several models, e.g., hierarchically or up-sampling. Following this approach, we demonstrate how MUSICGEN can generate high-quality samples, both mono and stereo, while being conditioned on textual description or melodic features, allowing better controls over the generated output. We conduct extensive empirical evaluation, considering both automatic and human studies, showing the proposed approach is superior to the evaluated baselines on a standard text-to-music benchmark. Through ablation studies, we shed light over the importance of each of the components comprising MUSICGEN. Music samples, code, and models are available at github.com/facebookresearch/audiocraft.

1 Introduction

Text-to-music is the task of generating musical pieces given text descriptions, e.g., “90s rock song with a guitar riff”. Generating music is a challenging task as it requires modeling long range sequences. Unlike speech, music requires the use of the full frequency spectrum [Müller, 2015]. That means sampling the signal at a higher rate, i.e., the standard sampling rates of music recordings are 44.1 kHz or 48 kHz vs. 16 kHz for speech. Moreover, music contains harmonies and melodies from different instruments, which create complex structures. Human listeners are highly sensitive to disharmony [Fedorenko et al., 2012, Norman-Haignere et al., 2019], hence generating music does not leave a lot of room for making melodic errors. Lastly, the ability to control the generation process in a diverse set of methods, e.g., key, instruments, melody, genre, etc. is essential for music creators.

Recent advances in self-supervised audio representation learning [Balestrieri et al., 2023], sequential modeling [Touvron et al., 2023], and audio synthesis [Tan et al., 2021] provide the conditions to develop such models. To make audio modeling more tractable, recent studies proposed representing audio signals as multiple streams of discrete tokens representing the same signal [Défossez et al., 2022]. This allows both high-quality audio generation and effective audio modeling. However, this comes at the cost of jointly modeling several parallel dependent streams.

Kharitonov et al. [2022], Kreuk et al. [2022] proposed modeling multi-streams of speech tokens in parallel following a delay approach, i.e., introduce offsets between the different streams. Agostinelli et al. [2023] proposed representing musical segments using multiple sequences of discrete tokens at different granularity and model them using a hierarchy of autoregressive models. In parallel, Donahue et al. [2023] follows a similar approach but for the task of singing to accompaniment generation. Recently, Wang et al. [2023] proposed tackling this problem in two stages: (i) modeling the first

*Yossi Adi is Affiliated with both The Hebrew University of Jerusalem & MetaAI.



Hugo Flores García^{1,2}

Prem Seetharaman¹

Rithesh Kumar¹

Bryan Pardo²

¹ Descript Inc.
² Northwestern University
hugofg@u.northwestern.edu

July 2023: Vampnet

Training with masked/unknown tokens enables capabilities such as inpainting and looping with variations.

<https://arxiv.org/abs/2307.04686>

ABSTRACT

We introduce VampNet, a masked acoustic token modeling approach to music synthesis, compression, inpainting, and variation. We use a variable masking schedule during training which allows us to sample coherent music from the model by applying a variety of masking approaches (called prompts) during inference. VampNet is non-autoregressive, leveraging a bidirectional transformer architecture that attends to all tokens in a forward pass. With just 36 sampling passes, VampNet can generate coherent high-fidelity musical waveforms. We show that by prompting VampNet in various ways, we can apply it to tasks like music compression, inpainting, outpainting, continuation, and looping with variation (vamping). Appropriately prompted, VampNet is capable of maintaining style, genre, instrumentation, and other high-level aspects of the music. This flexible prompting capability makes VampNet a powerful music co-creation tool. Code³ and audio samples⁴ are available online.

1. INTRODUCTION

In recent years, advances in discrete acoustic token modeling have resulted in significant leaps in autoregressive generation of speech [1, 2] and music [3]. Meanwhile, approaches that use non-autoregressive parallel iterative decoding have been developed for efficient image synthesis [4, 5]. Parallel iterative decoding promises to allow faster inference than autoregressive methods and is more suited to tasks like infill, which require conditioning on both past and future sequence elements.

In this work, we combine parallel iterative decoding with acoustic token modeling, and apply them to music audio synthesis. To the best of our knowledge, ours is the first ¹ extension of parallel iterative decoding to neural audio music generation. Our model, called VampNet, can be

¹ While our work was under peer review, Google released SoundStorm [6], which leverages a similar parallel iterative decoding approach to ours.

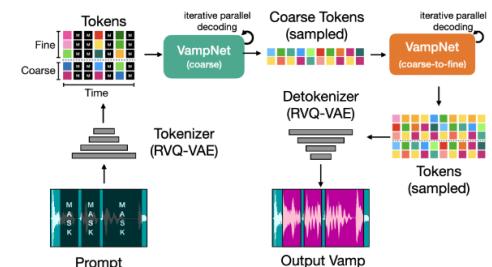


Figure 1. VampNet overview. We first convert audio into a sequence of discrete tokens using an audio tokenizer. Tokens are masked, and then passed to a masked generative model, which predicts values for masked tokens via an efficient iterative parallel decoding sampling procedure at two levels. We then decode the result back to audio.

flexibly applied to a variety of applications via token-based prompting. We show that we can guide VampNet's generation with selectively masked music token sequences, asking it to fill in the blanks. The outputs of this procedure can range from a high-quality audio compression technique to variations on the original input music that match the original input music in terms of style, genre, beat and instrumentation, while varying specifics of timbre and rhythm.

Unlike auto-regressive music models [2, 3], which can only perform music continuations – using some prefix audio as a prompt, and having the model generate music that could plausibly come after it – our approach allows the prompts to be placed anywhere. We explore a variety of prompt designs, including periodic, compression, and musically informed ones (e.g. masking on the beat). We find that our model responds well to prompts to make loops and variations, thus the name VampNet ². We make our code open source ³ and highly encourage readers to listen to our audio samples ⁴.

² To vamp is to repeat a short passage of music with variation.
³ <https://github.com/hugofloresgarcia/vampnet>
⁴ audio samples: <https://tinyurl.com/bdfj7rdx>

How to control?

Researchers and designers are still figuring out how to control generative audio.

Musicfy focuses on voice-based control:

<https://musicfy.lol/>



dadabots @dadabots · Nov 21, 2023

I've learned many instruments, but voice is the most immediately expressive. In this sick video, Ummet Ozcan uses Musicfy's couple dozen (RVC?) instrument models.

...

Lately I've been beatboxing into our prompt models that generate full bands/subgenres. Can't wait to share more!!



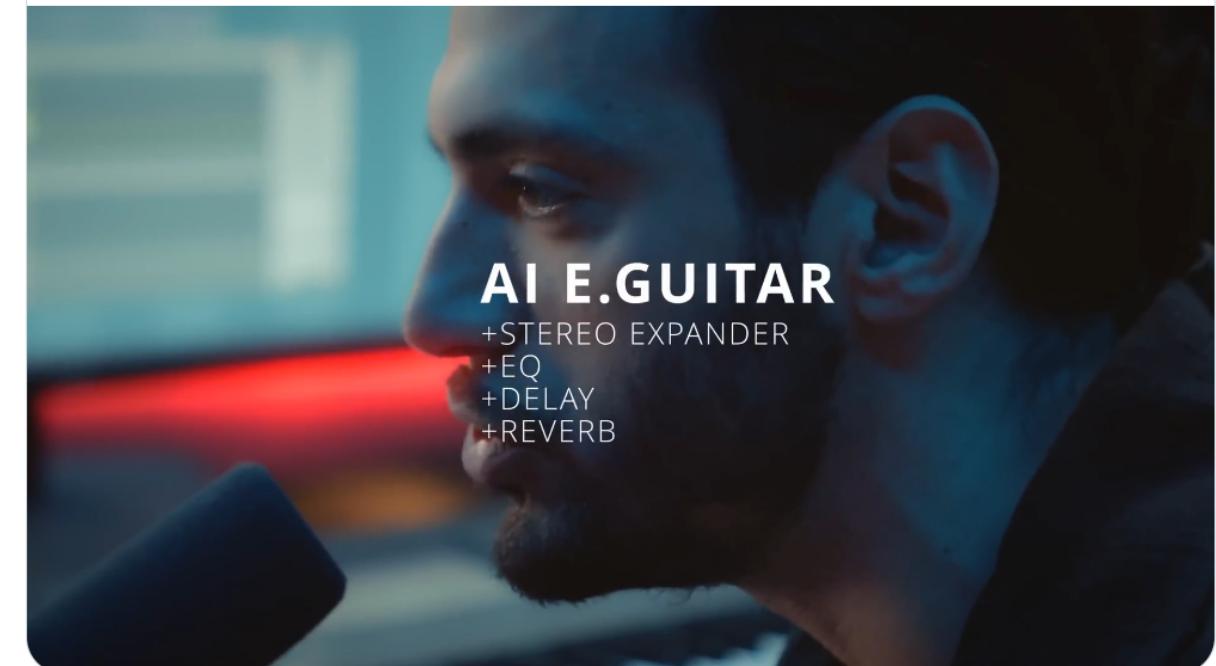
Arib 🇺🇸 🇹🇷 ✅ @aribk24 · Nov 20, 2023

voice to music

we just launched a feature that allows you to sing and turn your notes into any instrument you want

...

[Show more](#)





Commercial services

The screenshot shows the ElevenLabs website. At the top, there's a navigation bar with links for Products, Research, Pricing, Resources, and Company. The main feature is a large dark blue title "Generative Voice AI". Below it is a text input field with placeholder text: "Convert text to speech online for free with our AI voice generator. Create natural AI voices instantly in any language - perfect for video creators, developers, and businesses." A yellow button at the bottom left says "Get Started Free →". To the right, there's a section titled "Click on a language to convert text to speech:" followed by a grid of language names: English, Chinese, Spanish, Hindi, Portuguese, French, German, Japanese, Arabic, Russian, Korean, Indonesian, Italian, Dutch, Turkish, Polish, Swedish, Filipino, Malay, Romanian, Ukrainian, Greek, Czech, Danish, Finnish, Bulgarian, Croatian, Slovak, and Tamil. Below this is a sample text "That's insane!", he shouted angrily" spoken by a voice named "Daniel". A play button and a progress bar are visible below the text. In the bottom right corner, there are icons for refresh and download.

ElevenLabs

Products Research Pricing Resources Company

Generative Voice AI

Convert text to speech online for free with our AI voice generator. Create natural AI voices instantly in any language - perfect for video creators, developers, and businesses.

Get Started Free →

Click on a language to convert text to speech:

English Chinese Spanish Hindi Portuguese French German
Japanese Arabic Russian Korean Indonesian Italian Dutch Turkish Polish Swedish Filipino Malay
Romanian Ukrainian Greek Czech Danish Finnish Bulgarian Croatian Slovak Tamil

"That's insane!", he shouted angrily

— Daniel ▾

36 / 333

https://elevenlabs.io/



Create music with AI.

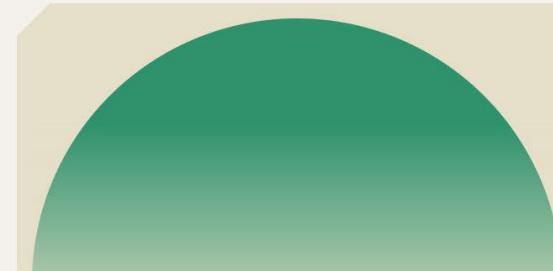


Generate high-quality audio that you can use commercially.
Get started for free.

Try it out

Trance, Ibiza, Beach, Sun, 4 AM, Progressive, Synthesizer, 909, Dramatic Chords, Choir, Euphoric, Nostalgic, Dynamic, Flowing

<https://www.stableaudio.com/>



Trance, Ibiza, Beach, Sun, 4 AM, Progressive,...

Stable Audio

AI music creation

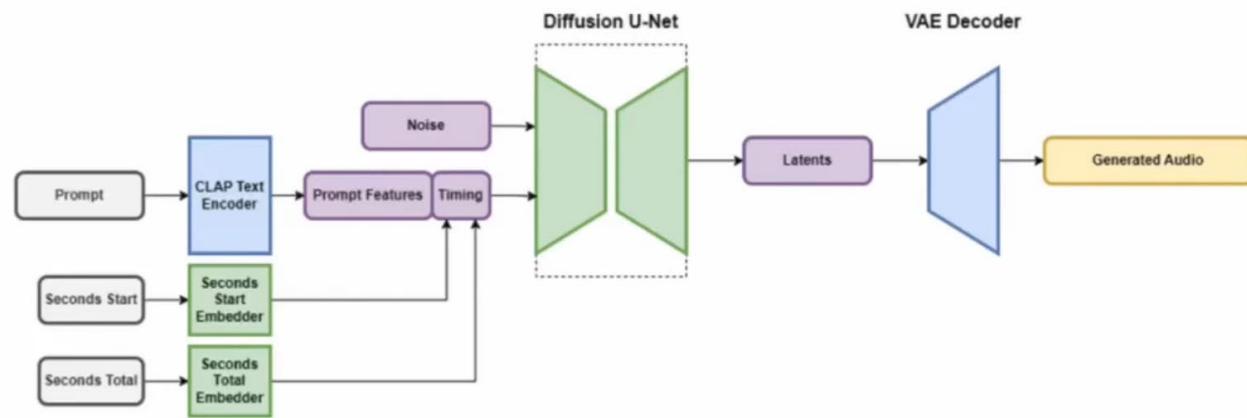


0:00



1:35

https://www.youtube.com/watch?v=6MVysjDl_0E



But Can It Do Death Metal? Announcing New AI Music Model



DADABOTS
31,1 t. tilaaaja

Tilaa

128

?

Jaa

Lataa

Klippi

Tallenna

...

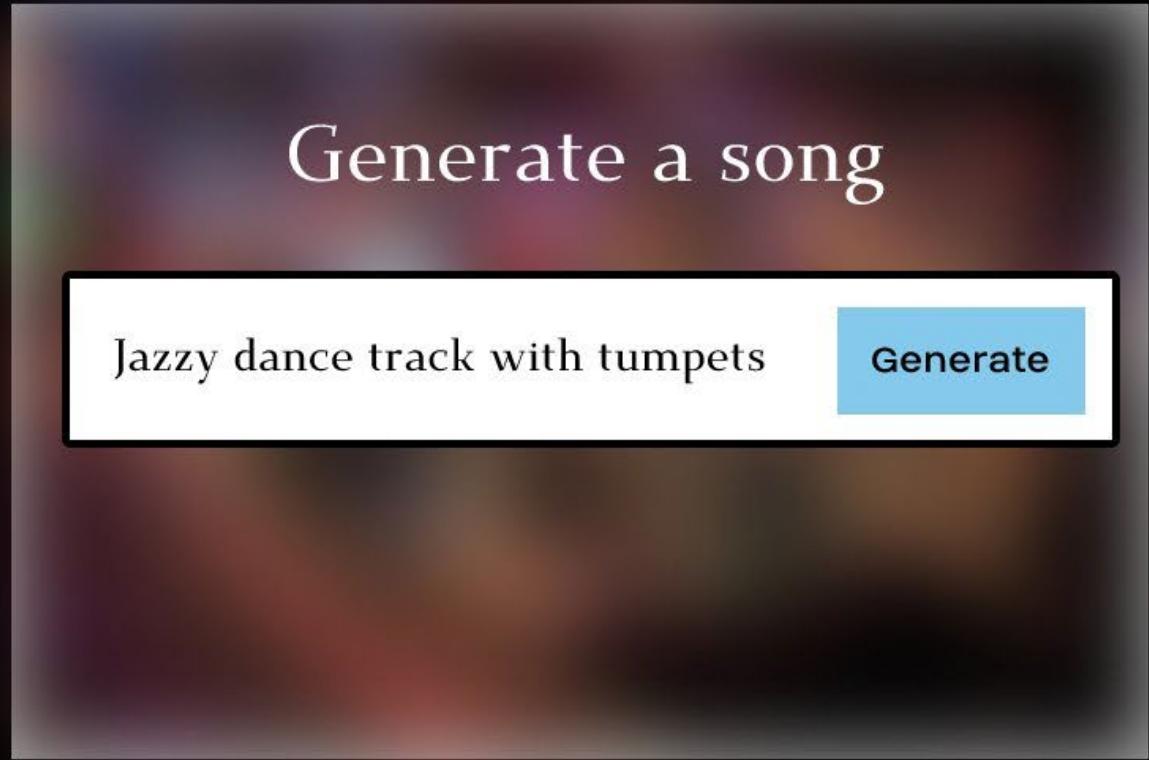
Valid critique

Music Producer: "I have countless specialized interfaces to make music, using my entire body. I love them all."



After extensive training with various music generation neural networks and dedicating countless hours to prompting them, it's become even more evident to me that relying solely on text prompts as interface for music creation significantly limits the creative process.

AI Scene: "All you need is Text Prompts as interface to a Generative AI model to make music. Throw away all else."



Emvoice

Melody and text control

<https://emvoiceapp.com/>



AaltoMediaAI
@aaltomediaai

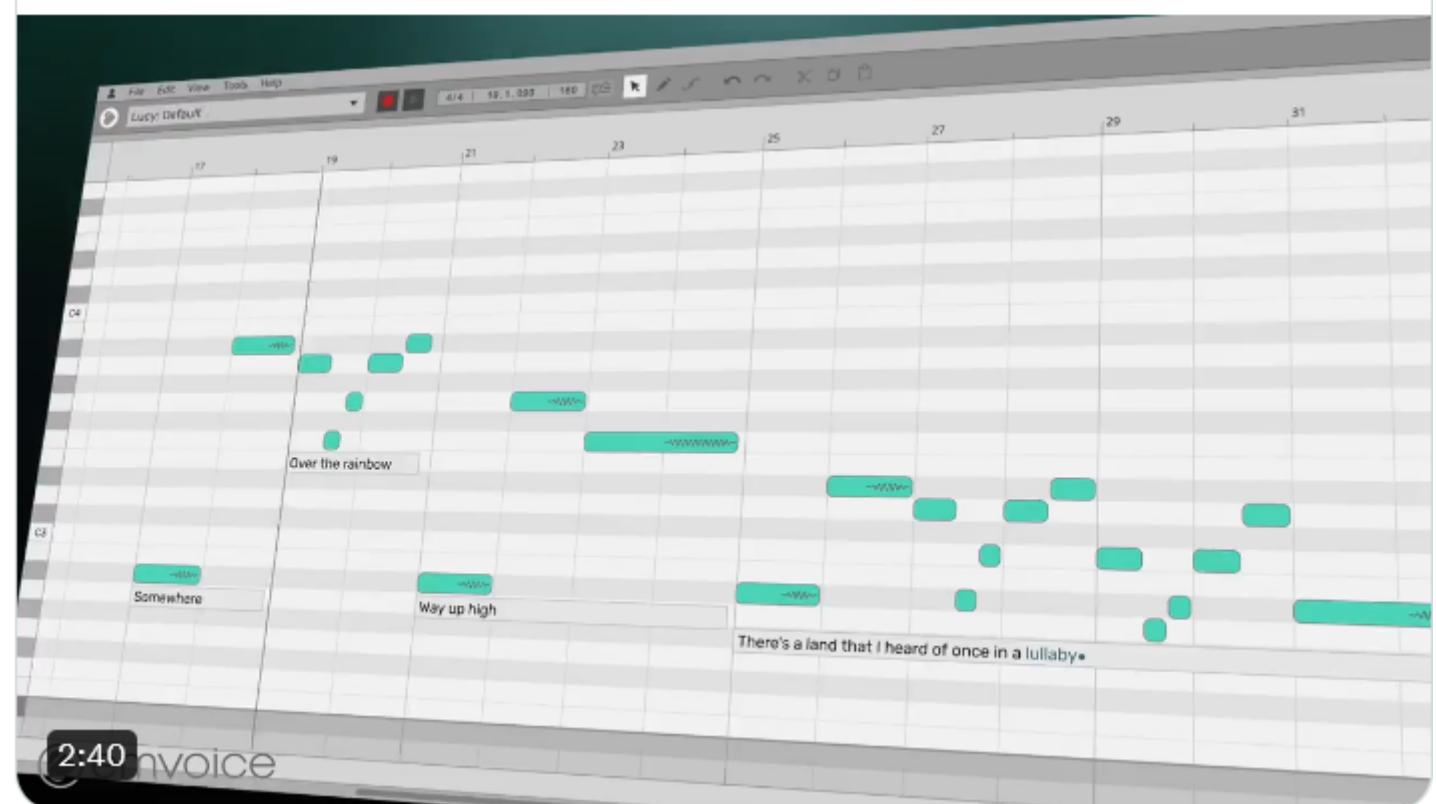
...

Emvoice lets you input both lyrics and melody and choose between 3 AI singing voices. Might be good for initial exploration of ideas if not real production.



Emvoice ✅ @emvoiceapp · Sep 15, 2023

The legendary @MaartenVorwerk used Lucy as the lead vocalist in Jeckyll & Hyde's new single, "Over The Rainbow." Take a listen!





okio ✓

@okio_ai

...

Introducing Remix - a tool made with Nendo that generates remixes of any song in any style. Upload a song or YT video & have fun! (For research purposes only. Usage is at your own risk)

Colab: colab.research.google.com/drive/1P1BEGArC...

Repo: github.com/okio-ai/nendo-...

Examples in this thread



5:50 PM · Nov 24, 2023 · 230.6K Views

<https://github.com/okio-ai/nendo-example-apps>



Core Cloud Community Blog Company Docs

AI ↗️ Audio,
Simplified.

nendo

The AI audio tool suite for developers,
powering next-gen audio applications

Generating 3D animation
(Still in progress, we'll continue on Friday)

Both diffusion and LLMs work

- Body poses (root translation + rotation, joint rotations) can be tokenized with RVQ for LLMs, or represented as real-valued arrays for Diffusion



Zhengyi (Zen) Luo @zhengyiluo · Dec 29, 2023 · 📝

...

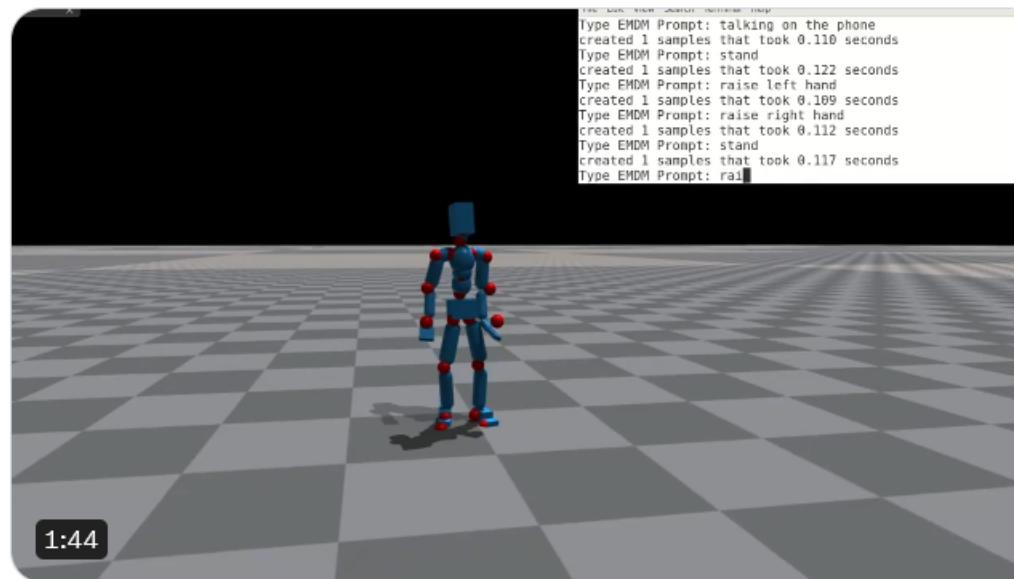
You can now ask your simulated humanoid to perform actions, in REAL-TIME 🙌

Powered by the amazing EMDM ([@frankzydou](#), [@Alex_wangjingbo](#), et al) and PHC.

EMDM: [frank-zy-dou.github.io/projects/EMDM/...](https://frank-zy-dou.github.io/projects/EMDM/)

PHC: github.com/ZhengyiLuo/Per...

Simulation: Isaac Gym



<https://x.com/zhenyiluo/status/1740506602615214231?s=20>

Generating 3D models

LLMs and NeRFs

- LLMs: 3D geometry can be represented as text, e.g., XML
- NeRFs (Neural Radiance Fields): A representation of both geometry and surface appearance that is suitable for Diffusion models. Results can be converted to a triangle mesh (not trivial, quality not yet optimal)

MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers

Yawar Siddiqui¹ Antonio Alliegro² Alexey Artemov¹
Tatiana Tommasi² Daniele Sirigatti³ Vladislav Rosov³ Angela Dai¹ Matthias Nießner¹
Technical University of Munich¹ Politecnico di Torino² AUDI AG³

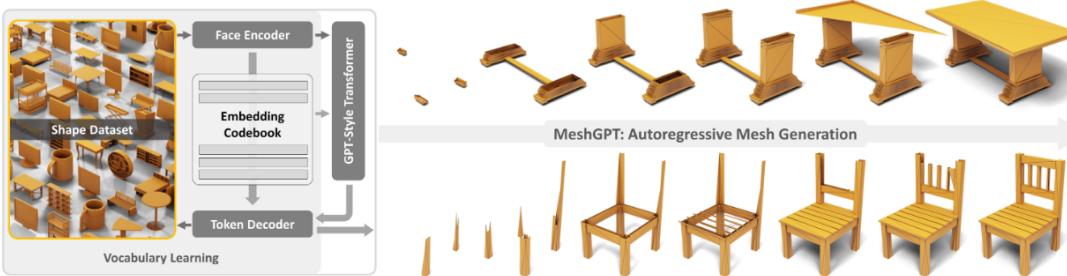


Figure 1. Our method creates triangle meshes by autoregressively sampling from a transformer model that has been trained to produce tokens from a learned geometric vocabulary. These tokens can then be decoded into the faces of a triangle mesh. Our method generates clean, coherent, and compact meshes, characterized by sharp edges and high fidelity.

Abstract

We introduce *MeshGPT*, a new approach for generating triangle meshes that reflects the compactness typical of artist-created meshes, in contrast to dense triangle meshes extracted by iso-surfacing methods from neural fields. Inspired by recent advances in powerful large language models, we adopt a sequence-based approach to autoregressively generate triangle meshes as sequences of triangles. We first learn a vocabulary of latent quantized embeddings, using graph convolutions, which inform these embeddings of the local mesh geometry and topology. These embeddings are sequenced and decoded into triangles by a decoder, ensuring that they can effectively reconstruct the mesh. A transformer is then trained on this learned vocabulary to predict the index of the next embedding given previous embeddings. Once trained, our model can be autoregressively sampled to generate new triangle meshes, directly generating compact meshes with sharp edges, more closely imitating the efficient triangulation patterns of human-crafted meshes. *MeshGPT* demonstrates a notable improvement over state-of-the-art mesh generation methods, with a 9% increase in shape coverage and a 30-point enhancement in FID scores across various categories.

1. Introduction

Triangle meshes are the main representation for 3D geometry in computer graphics. They are the predominant representation for 3D assets used in video games, movies, and

virtual reality interfaces. Compared to alternative 3D shape representations such as point clouds or voxels, meshes provide a more coherent surface representation; they are more controllable, easier to manipulate, more compact, and fit directly into modern rendering pipelines, attaining high visual quality with far fewer primitives. In this paper, we tackle the task of automated generation of triangle meshes, streamlining the process of crafting 3D assets.

Recently, 3D vision research has seen great interest in generative 3D models using representations such as voxels [3, 62], point clouds [37, 67, 68], and neural fields [14, 19, 31, 35, 41]. However, these representations must then be converted into meshes through a post-process for use in downstream applications, for instance by iso-surfacing with Marching Cubes [36]. Unfortunately, this results in dense, over-tessellated meshes that often exhibit oversmoothing and bumpy artifacts from the iso-surfacing, as shown in Figure 2. In contrast, artist-modeled 3D meshes are compact in representation, while maintaining sharp details with much fewer triangles.

Thus, we propose *MeshGPT*¹ to generate a mesh representation directly, as a set of triangles. Inspired by powerful recent advances in generative models for language, we adopt a direct sequence generation approach to synthesize triangle meshes as sequences of triangles. Following text generation paradigms, we first learn a vocabulary of triangles. Triangles are encoded into latent quantized embeddings through an encoder. To encourage learned trian-

¹nihilasid.github.io/mesh-gpt

LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching

Yixun Liang^{*1} Xin Yang^{*1,2} Jiantao Lin¹ Haodong Li¹ Xiaogang Xu^{3,4} Yingcong Chen^{**1,2}
¹ HKUST (GZ) ² HKUST ³ Zhejiang Lab ⁴ Zhejiang University

yliang982@connect.hkust-gz.edu.cn xin.yang@connect.ust.hk jlin695@hkust-gz.edu.cn
hli736@connect.hkust-gz.edu.cn xgxu@zhejianglab.com yingcongchen@ust.hk

arXiv:2311.11284v3 [cs.CV] 2 Dec 2023

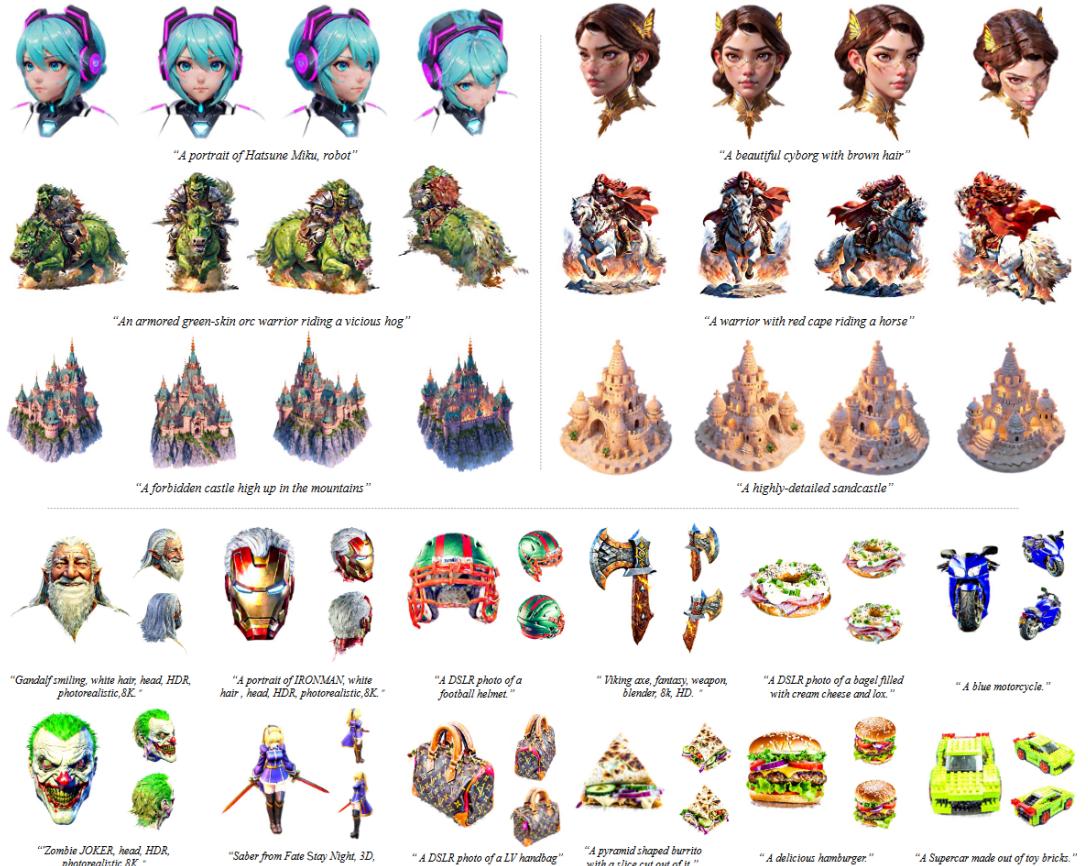


Figure 1. Examples of text-to-3D content creations with our framework. We present a text-to-3D generation framework, named the *LucidDreamer*, to distill high-fidelity textures and shapes from pretrained 2D diffusion models (detailed shows on Sec. 4) with a novel **Interval Score Matching** objective and an *Advanced 3D distillation pipeline*. Together, we achieve superior 3D generation results with photorealistic quality in a short training time. Please zoom in for details.

^{**} Corresponding author.

^{*}The first two authors contributed equally to this work.

^{*}Conceptualization: Yixun Liang: 60%, Xin Yang: 40%,

Methodology: Xin Yang: 60%, Yixun Liang: 40%.

Real-world production pipelines

The reality

- Text prompting almost never gives exactly what you want
- Control interfaces are still emerging
- Most AI tools are highly dedicated
- Typical: inpaint to fix mistakes, chain multiple AI tools, e.g., DALL-E 3 image to Stable Video Diffusion

"Generative AI in Game Development"



▶ ▶ | 0:00 / 53:44 • Intro >

Jussi Kemppainen

Designer, Dinosaurs Are Better



GAMES
2023-2024
NOW! A!
Aalto University

#gamesnowaalto.fi



StableDiffusion and ControlNet integrated in a painting app

[https://x.com/dreamwieber/
status/17275488227449079
26?s=20](https://x.com/dreamwieber/status/1727548822744907926?s=20)

[https://apps.apple.com/us/
app/sagebrush-ai-
painter/id6472820126](https://apps.apple.com/us/app/sagebrush-ai-painter/id6472820126)

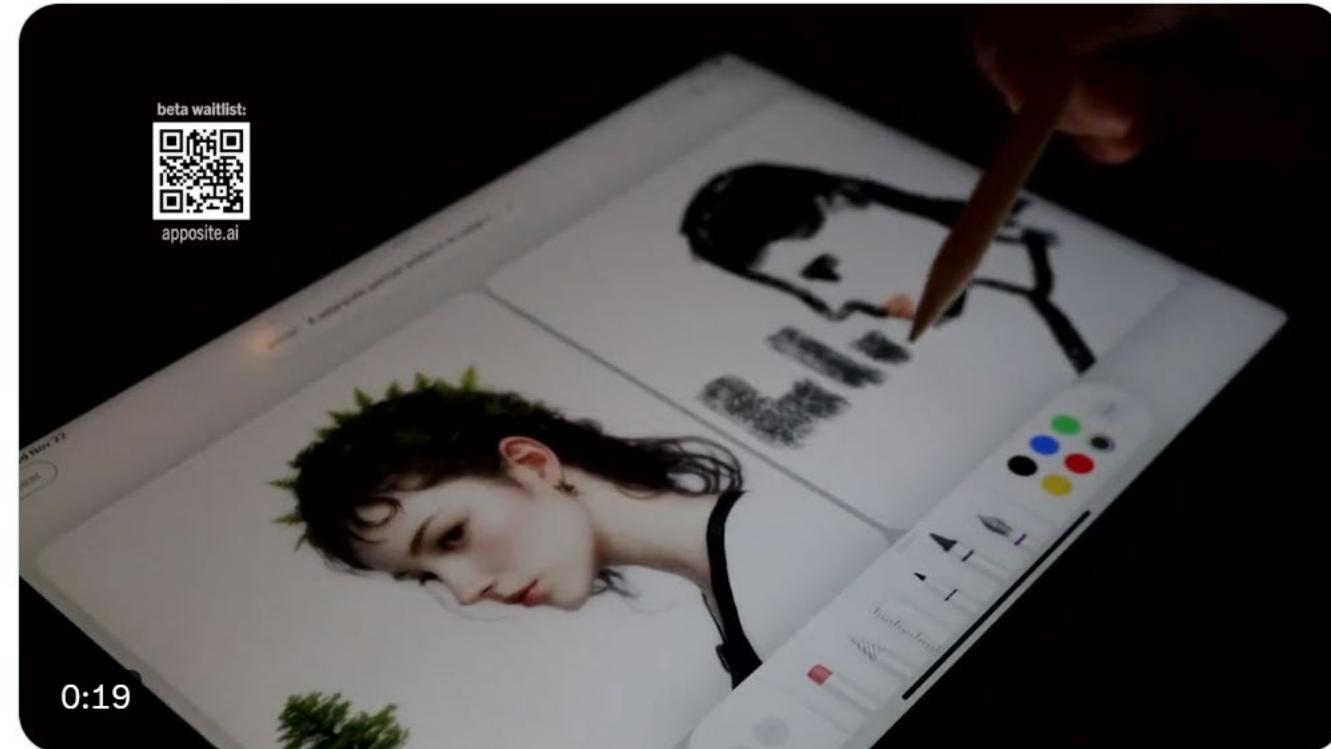


Gregory Wieber ✅ @dreamwieber · Nov 23, 2023

My new iPad app that lets you paint with **#ai** just got approved for TestFlight 🚀

So much interest, thank you! I'll be rolling out to close friends first, and scaling up over time.

Sign up for the waitlist, and don't forget to follow here! ✨



Same in ComfyUI using a
webcam capture node to feed
live 3D modeling as
ControlNet input

↻ You reposted

toyxyz ✅ @toyxyz3 · Nov 22, 2023

ComfyUI LCM real time Normal Controlnet test(4x) #stablediffusion #AI
ラスト #AI #ComfyUI

The image shows the ComfyUI software interface. On the left, there is a node graph with nodes for 'VAE Decode' and 'KSampler'. A pink arrow points from the 'vae' output of the VAE Decode node to the 'latent_image' input of the KSampler node. The KSampler node has various parameters: seed (4532421566), control_after_generate (fixed), steps (4), cfg (1.4), sampler_name (lcm), scheduler (sgm_uniform), and denoise (1.00). To the right of the graph is a 'Preview Image' window showing a young man wearing a white chef's hat in profile. Below the preview is a 'Webcam Capture' window showing a 3D surface model of the same person's head and shoulders, colored in a rainbow gradient.

0:17

18 188 1K 122K

<https://x.com/toyxyz3/status/1727343696692277367?s=20>



@levelsio ✅
@levelsio

✨ I made my first video game with ChatGPT:

- 1) ChatGPT generates a text-based adventure game with DALL-E 3 generating images for it
- 2) Every time you play the game is different because it generates the story and images live
- 3) The images from DALL-E are sent to [@runwayML](#) which turns images into video
- 4) The text is sent to [@elevenlabs](#) which turns the text adventure into a pirate narrator voice
- 5) It's merged into a video
- 6) Interactive buttons are overlayed

The game is called:

 The Secret of Monkey Island: Amsterdam (unofficial)

...

And you can play it here:

monkeyislandamsterdam.com

(video + TTS + buttons doesn't work auto yet, for now manual but text + img works, I'm building an interface for it now)



9:49 PM · Nov 16, 2023 · 2.7M Views

SceneTex: High-Quality Texture Synthesis for Indoor Scenes via Diffusion Priors

Dave Zhenyu Chen¹, Haoxuan Li¹, Hsin-Ying Lee², Sergey Tulyakov², Matthias Nießner¹,

¹Technical University of Munich, ²Snap Research

Paper

arXiv

Video

Code

<https://daveredrum.github.io/SceneTex/>



SCENETEX generates high-quality textures for 3D indoor scenes from the given text prompts. At its core, SceneTex proposes a multiresolution texture field to implicitly encode the mesh appearance. We optimize the target texture via a score-distillation-based objective function in respective RGB renderings. To further secure the style consistency across views, we introduce a cross-attention decoder to predict the RGB values by cross-attending to the pre-sampled reference locations in each instance. Our method enables various and accurate texture synthesis for 3D-FRONT scenes, demonstrating significant improvements in visual quality and prompt fidelity over the prior texture generation methods.