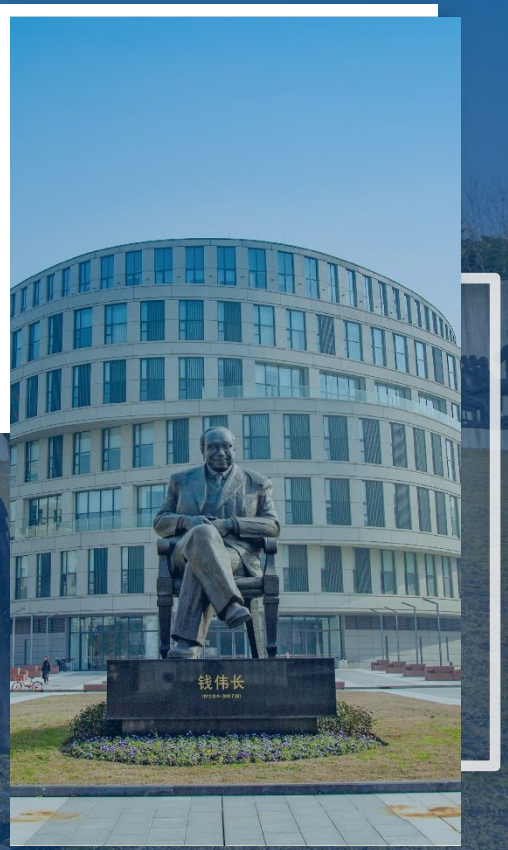
上海大学未来技术学院 | 上海大学人工智能研究院
SCHOOL OF FUTURE TECHNOLOGY, SHANGHAI UNIVERSITY | INSTITUTE OF ARTIFICIAL INTELLIGENCE, SHANGHAI UNIVERSITY

人工智能导论

——第7课：人工智能在大模型的应用

(生成式AI)



叶林奇

未来技术学院（人工智能研究院）

2023冬季学期

提纲

一、生成式AI

二、预训练模型

三、具身智能

菜鸟

咏卷心菜小鸡

卷心菜里藏玄机，
神鸡孕育其中栖。
眼似星辰闪橙光，
嘴如朝阳染霞霓。
天赐灵物人间现，
地育仙禽世间稀。
莫道此物无用处，
寻常百姓盘中餐。

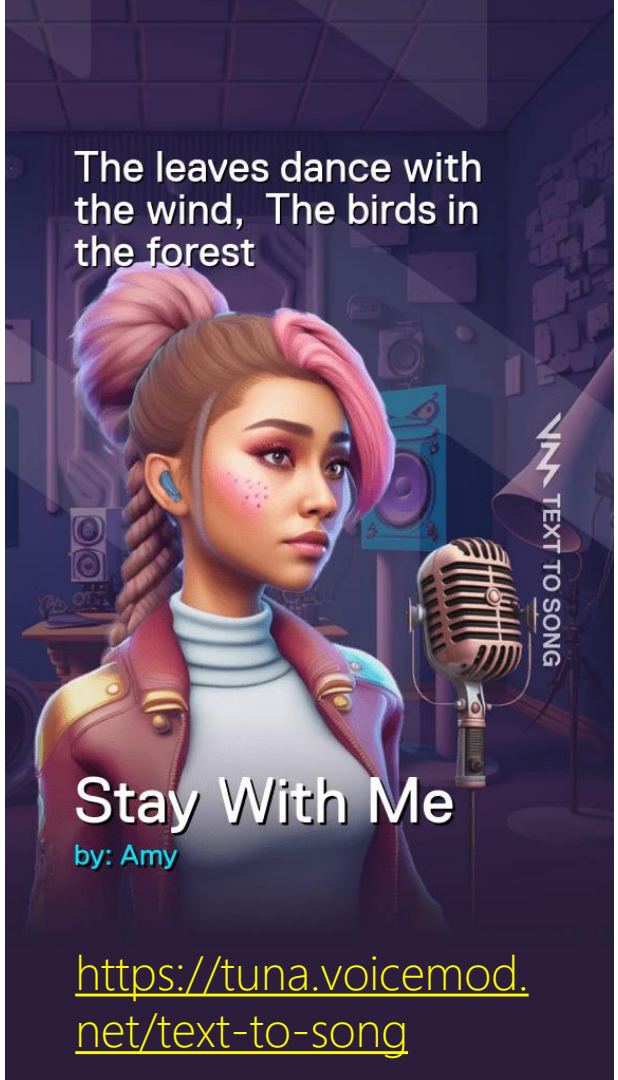
叶随风舞舞翩翩，
林间鸟语声声甜。
奇峰异石入眼帘，
人间仙境在心田。

普通话　上海话
广东话　河南话
东北话　陕西话
山东话　四川话
香港话　台湾话

https://www.text-to-speech.cn/

https://png3d.com/

https://pika.art/



https://tuna.voicemod.net/text-to-song

俯首甘为孺子牛

横眉冷对千夫指

鲁迅先生


A1
大家好我是沐沐


所以让我们可以这样


ARAG
ARAG –
Partner des Sports
ARAG
www.sports
SOLOMONJAGWE
ARAG

https://wonderdynamics.com/

TEXT-TO-VIDEO

PIKA

https://pika.art/

https://research.runwayml.com/gen2

群聊：人工智能导论 2023

该二维码7天内(1月14日前)有效，重新进入将更新

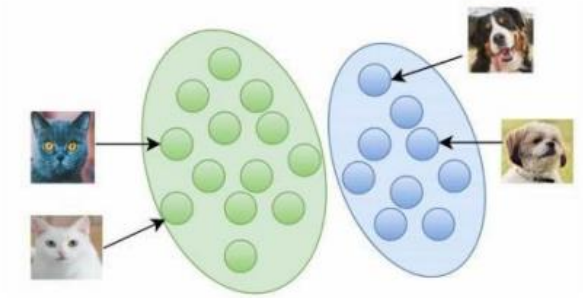
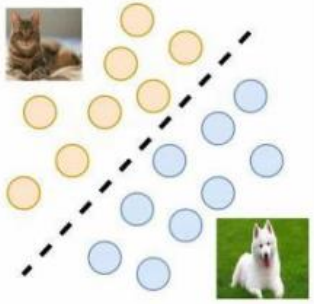
文心一言

通义千问

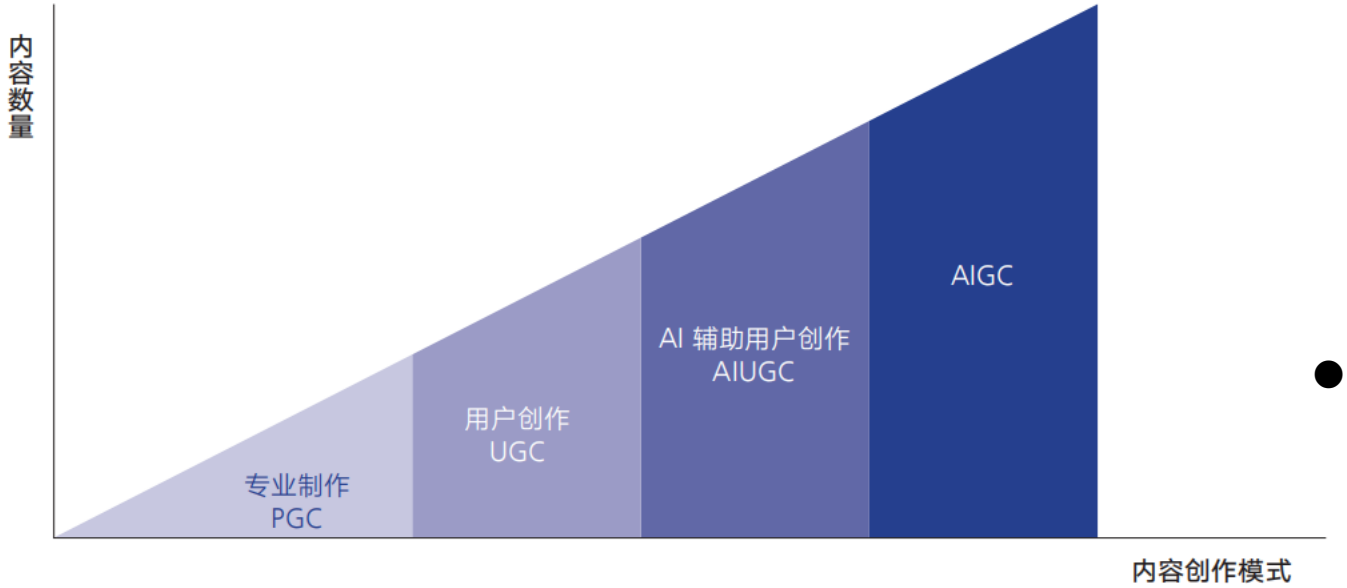
讯飞星火

生成式AI

AI模型可大致分为决策式AI（Discriminant AI）和生成式AI （Generative AI）两类。

类型	决策式AI	生成式AI
技术路径	已知数据分别求解输出类别标签，区分不同类型数据,例如将图像区分为猫和狗 	分析归纳已有数据后创作新的内容，例如生成逼真的猫或狗的图像 
成熟程度	技术成熟，应用广泛，辅助提高非创造性工作效率	2014年开始快速发展，近期发展速度呈指数级爆发，部分领域应用落地
应用方向	推荐系统、风控系统、决策智能体等	内容创作、科研、人机交互以及多个工业领域
应用产品	人脸识别、精准广告推送、金融用户评级、智能辅助驾驶等	文案写作、文字转图片、视频智能配音、智能海报生成、视频智能特效、代码生成、语音人机交互、智能医疗诊断等

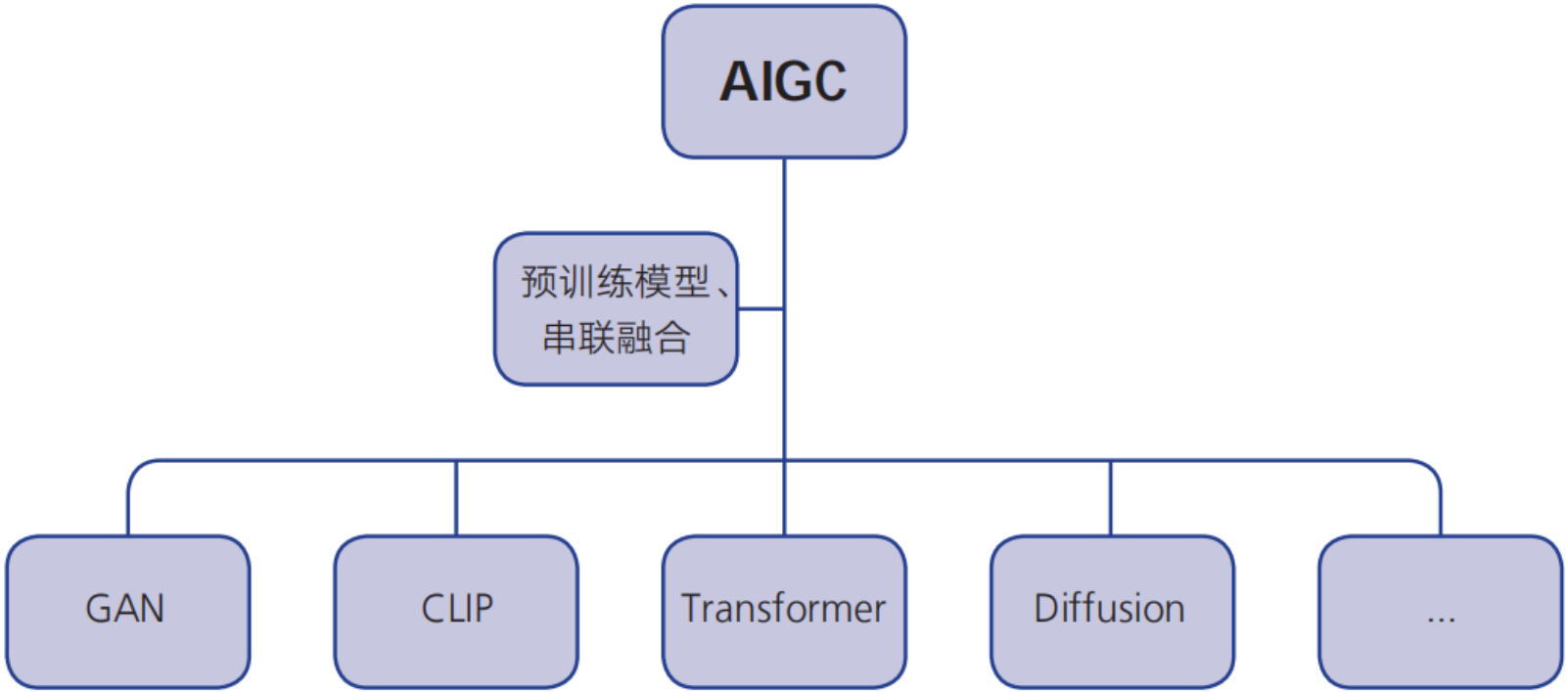
AIGC（AI-Generated Content，人工智能生成内容）



图：内容创作模式的四个发展阶段

- 1957 年莱杰伦·希勒(Leiaren Hiller)和伦纳德·艾萨克森(Leonard Isaacson)完成了人类历史上第一支由计算机创作的音乐作品就可以看作是AIGC 的开端。
- 2022 年才真正算是 AIGC 的爆发之年，人们看到了AIGC 无限的创造潜力和未来应用可能性。

生成式AI

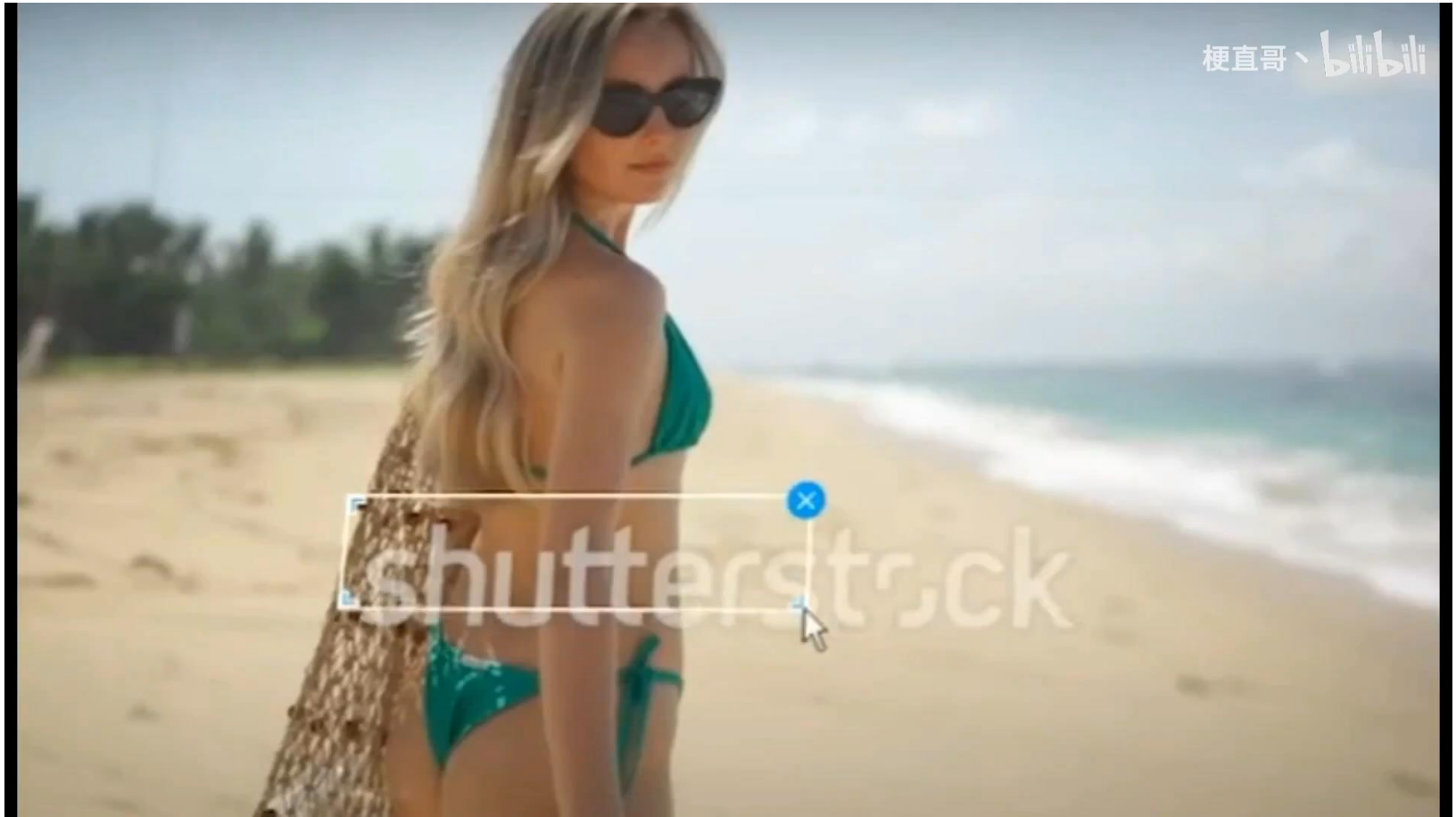


图：AIGC 技术累积融合 [02]

生成式AI

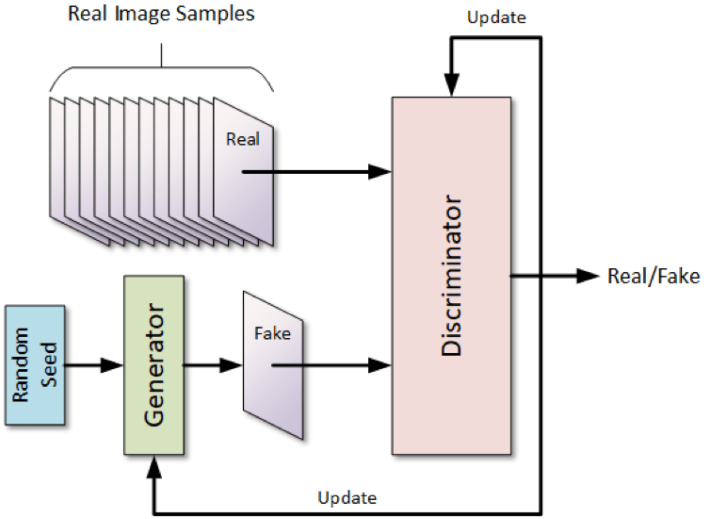
模型	提出时间	模型描述
变分自动编码（Variational Autoencoders，VAE）	2014年	基于变分下界约束得到的Encoder-Decoder模型对
生成对抗网络（GAN）	2014年	基于对抗的Generator-Discriminator模型对
基于流的生成模型（Flow-based models）	2015年	学习一个非线性双射转换（bijective transformation），其将训练数据映射到另一个空间，在该空间上分布是可以因子化的，整个模型架构依靠直接最大化log-likelihood来完成

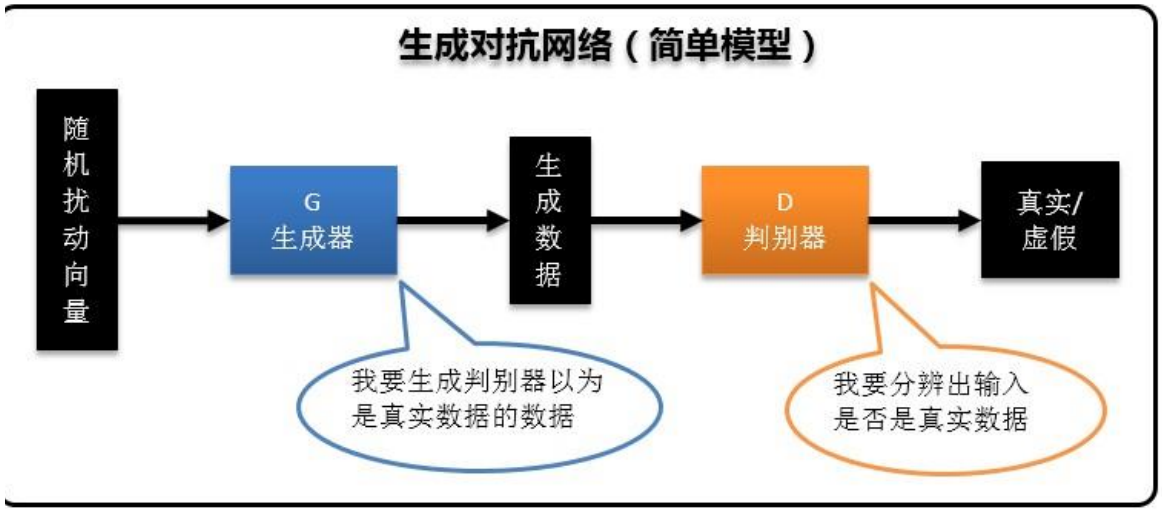
VAE （变分自编码器）原理



GAN（生成对抗网络）原理

2014 年，伊恩·古德费洛(Ian Goodfellow)提出的生成对抗网络(Generative Adversarial Network，GAN) 成为早期最为著名的生成模型。GAN 使用合作的零和博弈框架来学习，被广泛用于生成图像、视频、语音和三维物体模型等。GAN 也产生了许多流行的架构或变种，如DCGAN，StyleGAN，BigGAN，StackGANPix2pix，Age-cGAN，CycleGAN、对抗自编码器(Adversarial Autoencoders，AAE)、对抗推断学习（Adversarially Learned Inference，ALI)等





生成

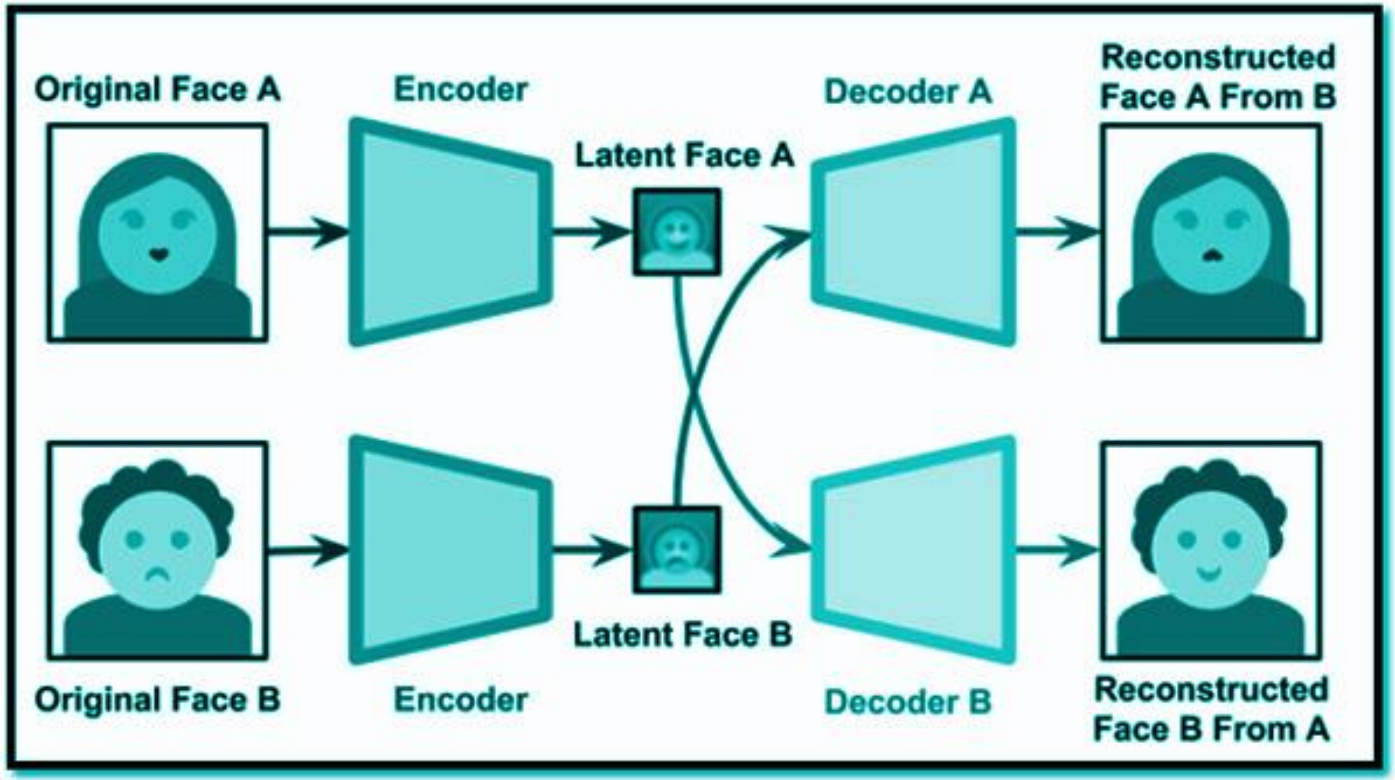
美国科罗拉多州上月举办艺术博览会，一幅名为《太空歌剧院》的画作最终获得数字艺术类别冠军。该作品先由AI制图工具Midjourney生成，再经Photoshop润色而来。

https://www.midjourney.com/


沧海踏笙笑 bilibili
https://www.midjourney.com/

生成式AI

GAN网络应用：AI换脸 是指用另一个人脸来替换一张图片或视频中的一个人脸，合成新的媒体物，它是Deepfake技术最广为人知的一种应用形式。



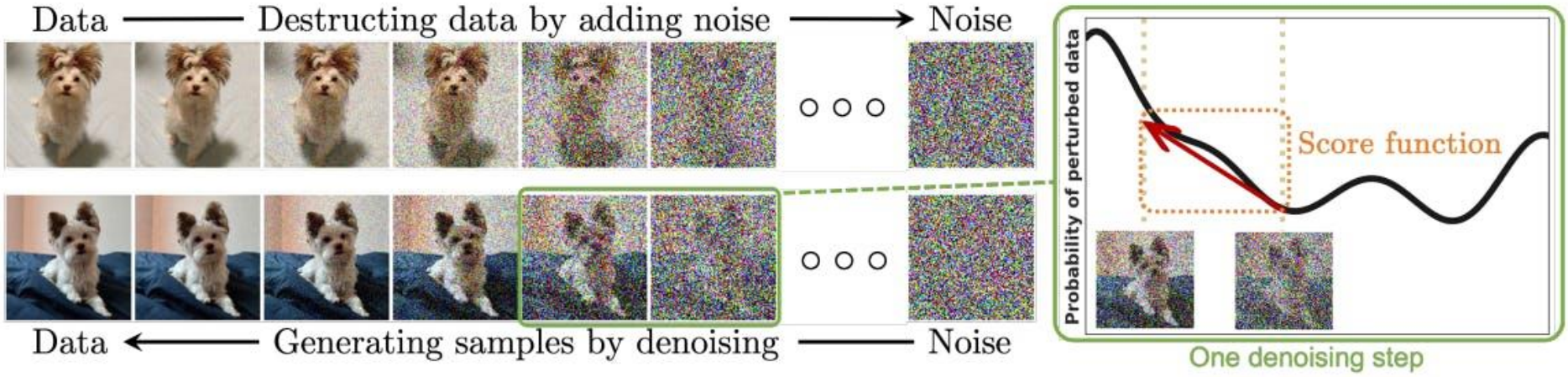

TITANIC

Transformer、基于流的生成模型 (Flow-based models)、扩散模型(Diffusion Model)等深度学习的生成算法相继涌现。
从最优化模型性能的角度出发，扩散模型相对 GAN 来说具有更加灵活的模型架构和精确的对数似然计算，已经取代 GAN 成为最先进的图像生成器。2021 年 6 月，OpenAI 发表论文已经明确了这个结论和发展趋势。

扩散模型(Diffusion Model)	2015年	扩散模型有两个过程，分别为扩散过程和逆扩散过程。在前向扩散阶段对图像逐步施加噪声，直至图像被破坏变成完全的高斯噪声，然后在逆向阶段学习从高斯噪声还原为原始图像的过程。 经过训练，该模型可以应用这些去噪方法，从随机输入中合成新的“干净”数据。
Transformer模型	2017年	一种基于自注意力机制的神经网络模型，最初用来完成不同语言之间的文本翻译任务，主体包含Encoder和Decoder部分，分别负责对源语言文本进行编码和将编码信息转换为目标语言文本

Difussion Model （扩散模型）原理

扩散模型(Diffusion Model)是受非平衡热力学的启发，定义一个扩散步骤的马尔可夫链，逐渐向数据添加随机噪声，然后学习逆扩散过程，从噪声中构建所需的数据样本。扩散模型最初设计用于去除图像中的噪声。随着降噪系统的训练时间越来越长并且越来越好，它们最终可以从纯噪声作为唯一输入生成逼真的图片。

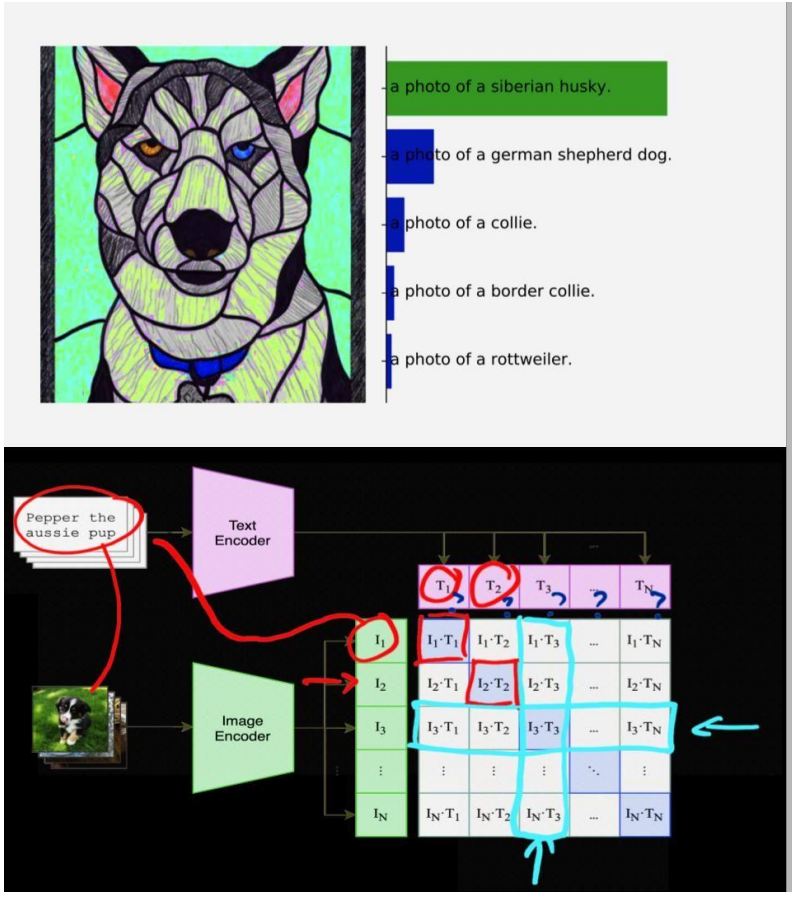


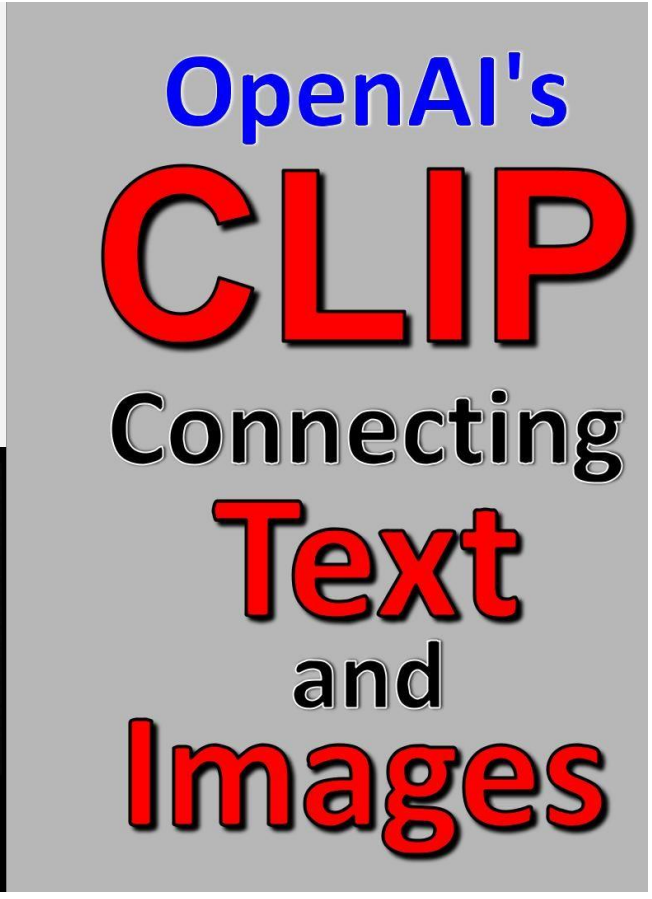

https://stablediffusionweb.com/

生成式AI

神经辐射场（Neural Radiance Field，NeRF）	2020年	它提出了一种从一组输入图像中优化连续5D神经辐射场的表示（任何连续位置的体积密度和视角相关颜色）的方法，要解决的问题就是给定一些拍摄的图，如何生成新的视角下的图
CLIP（Contrastive Language-Image Pre-Training）模型	2021年	1、进行自然语言理解和计算机视觉分析； 2、使用已经标记好的“文字-图像”训练数据。一方面对文字进行模型训练。一方面对另一个模型的训练，不断调整两个模型的内部参数，使得模型分别输出的文字特征和图像特征值并确认匹配。

CLIP（Constastive Language-Image Pretraining）模型：连接图像和文本





提纲

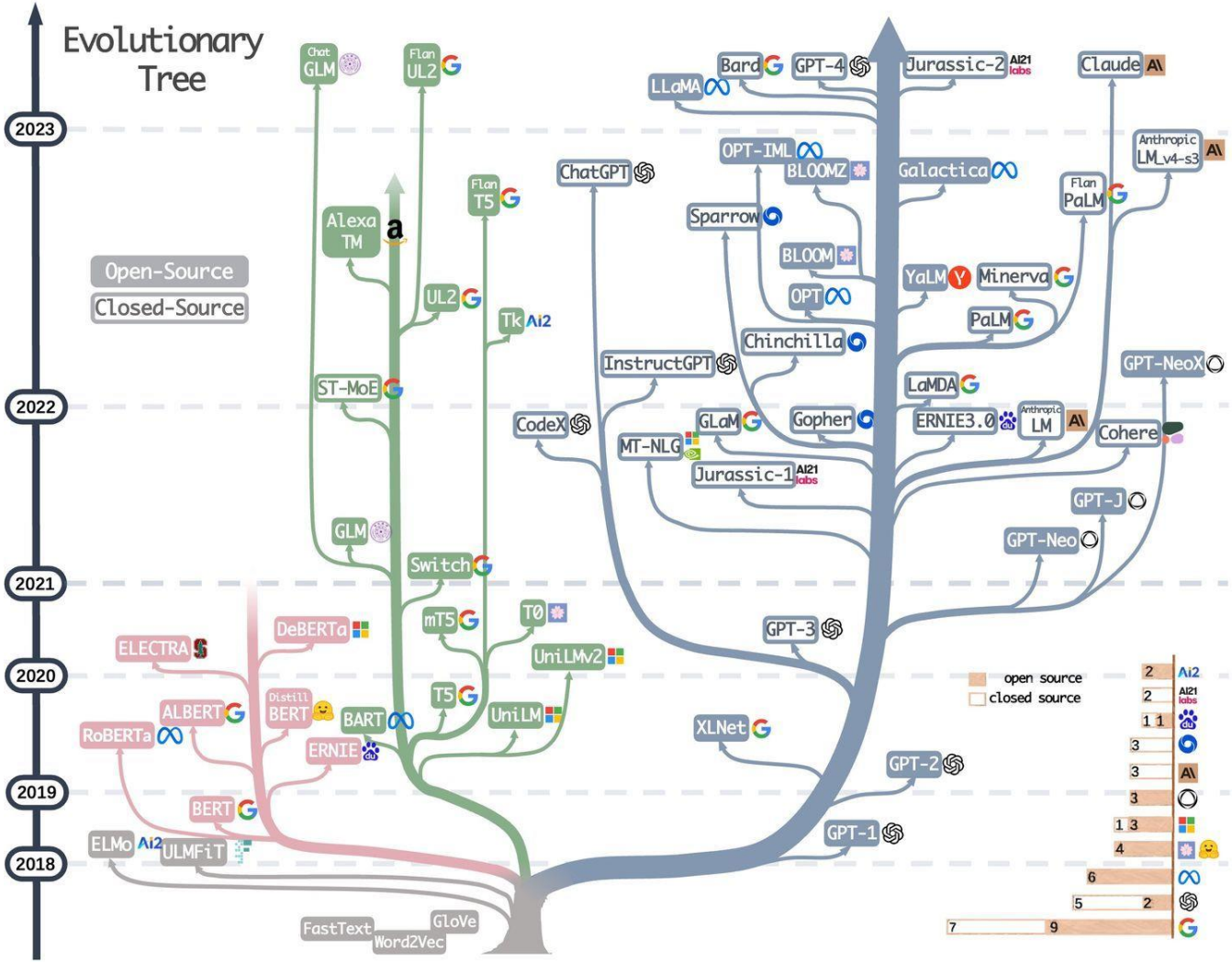
一、生成式AI

二、预训练模型

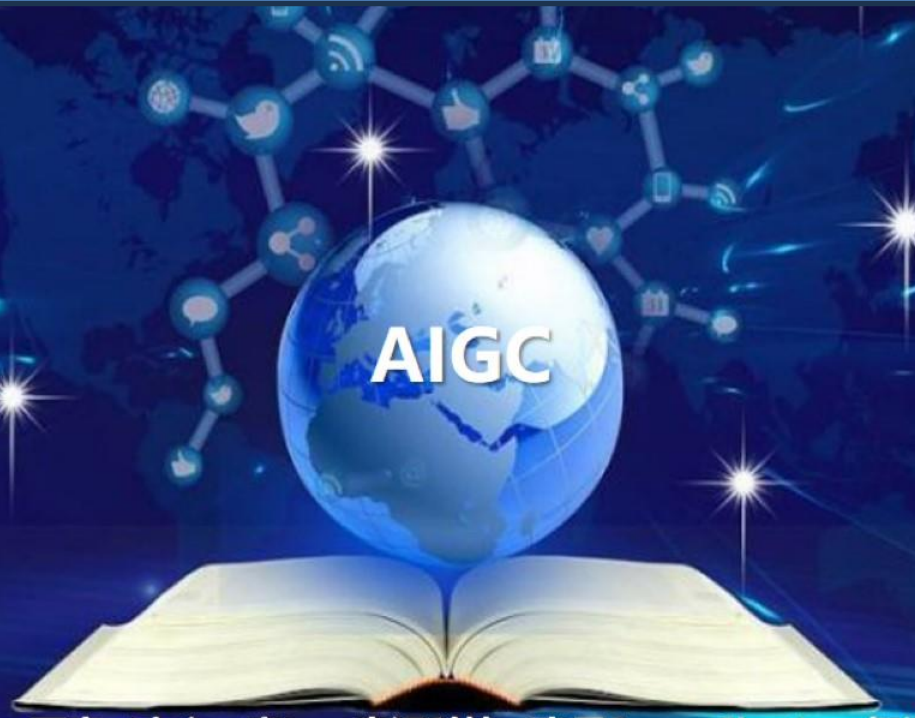
三、具身智能

预训练模型

虽然过去各类生成模型层出不穷，但是使用门槛高、训练成本高、内容生成简单和质量偏低，远远不能满足真实内容消费场景中的灵活多变、高精度、高质量等需求。预训练模型的出现引发了 AIGC 技术能力的质变，以上的诸多落地问题得到了解决。



国内的百模大战



科大讯飞（星火大模型）、百度（文心一言）、商汤科技（商量SenseChat）、智谱AI（GLM智谱清言）、华为（盘古大模型）、腾讯（混元大模型）、百川智能（百川大模型）、抖音（云雀大模型）、中科院（紫东太初）、MiniMax（ABAB大模型）、上海人工智能实验室（书生大模型）、360智脑、……

预训练模型

GPT的全称，是Generative Pre-Trained Transformer（生成式预训练Transformer模型）是一种基于互联网的、可用数据来训练的、文本生成的深度学习模型。

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

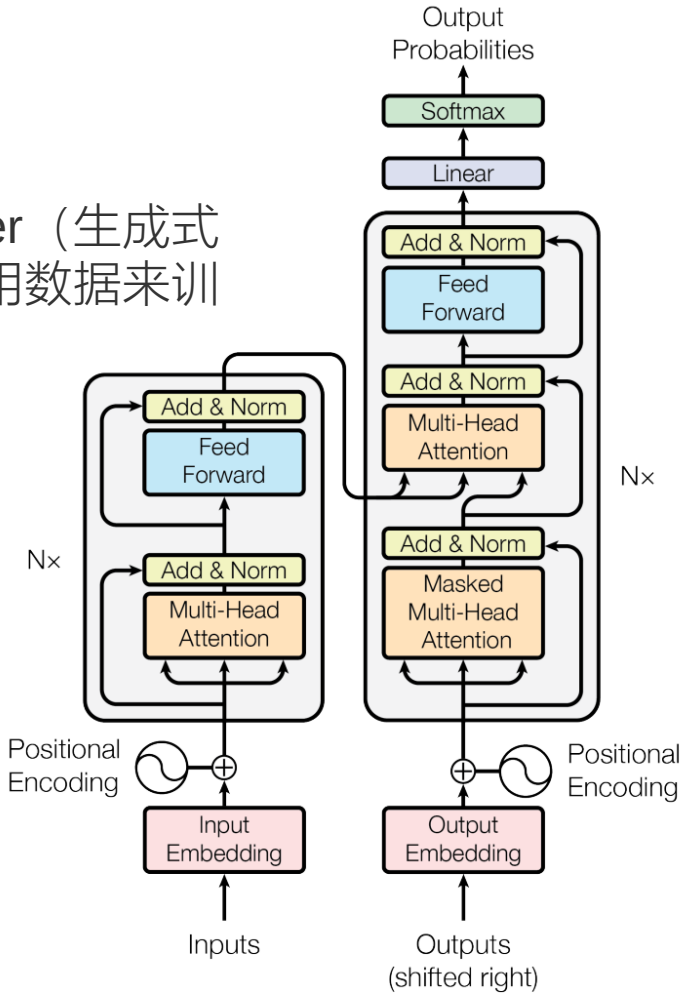
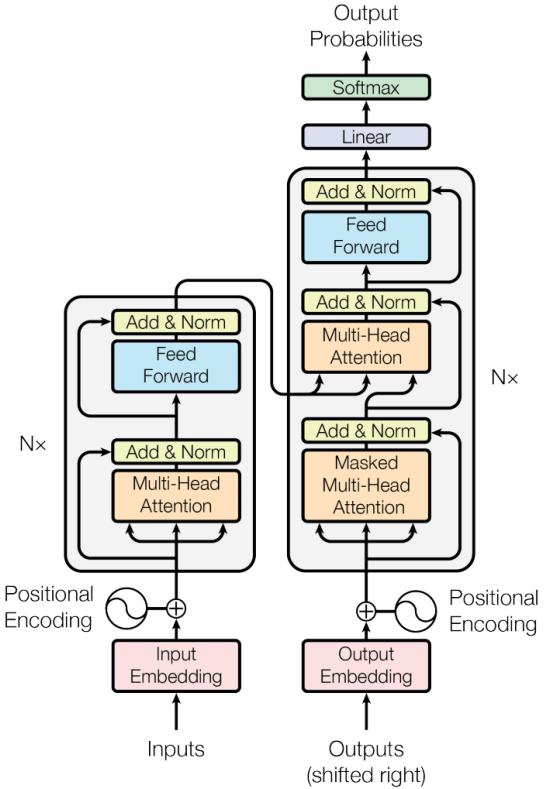
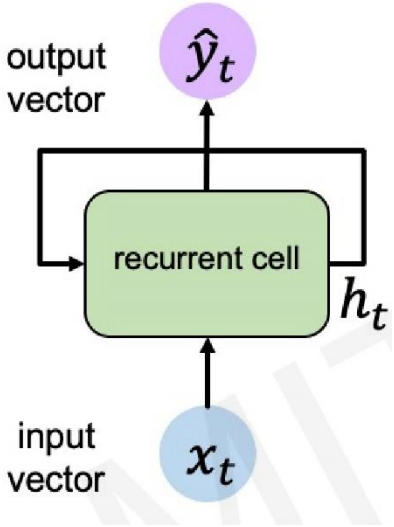


Figure 1: The Transformer - model architecture.

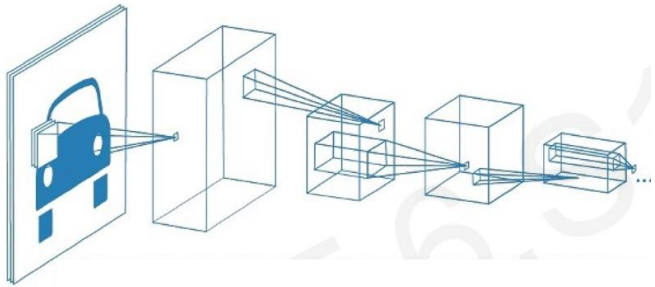
预训练模型



Transformer



RNN



CNN

预训练模型

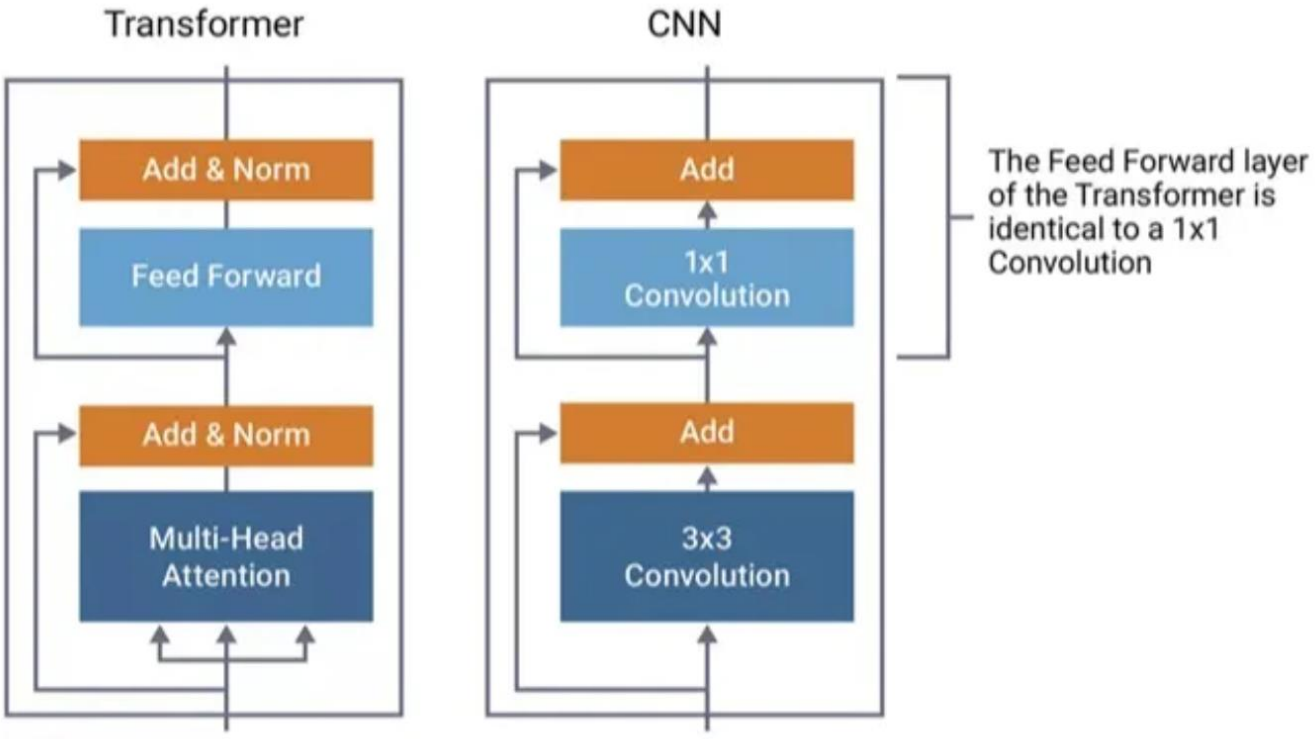


Fig. 2: Comparing Transformer and CNN structures.

预训练模型

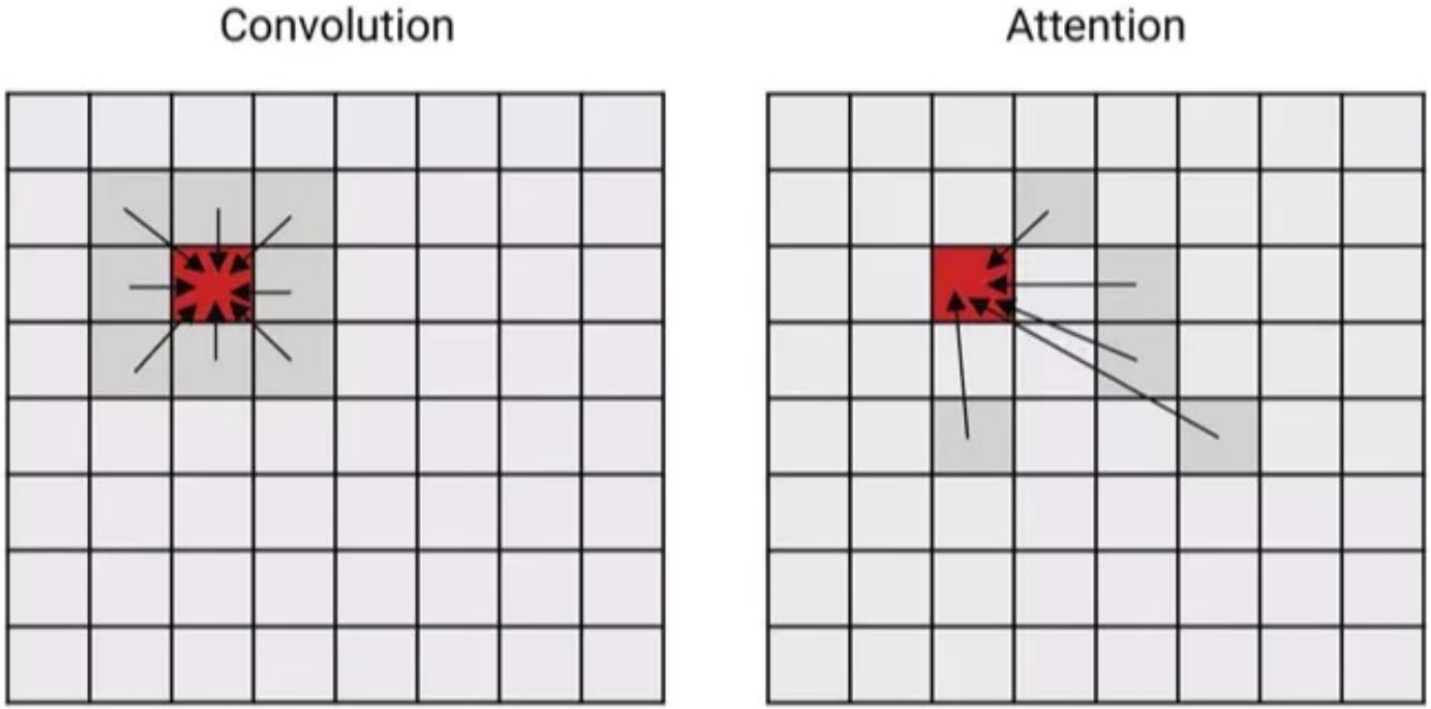
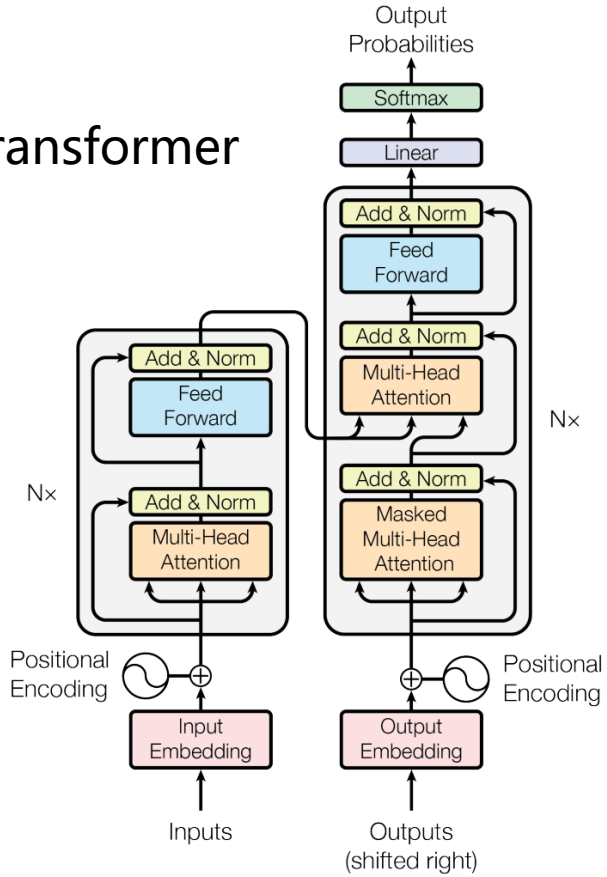


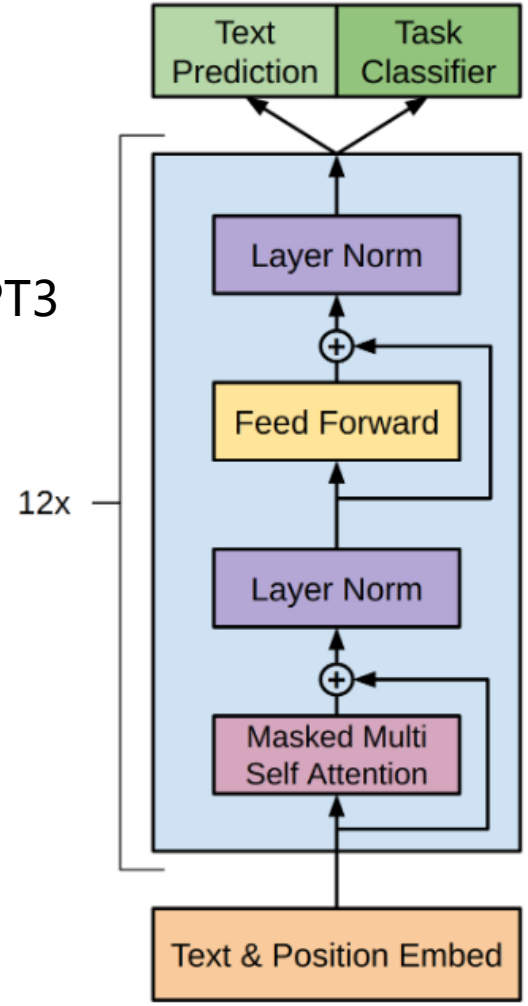
Fig. 3: Illustrating the difference between how a CNN's convolution and a transformer's attention networks mix in features of other tokens/pixels.

预训练模型

Transformer

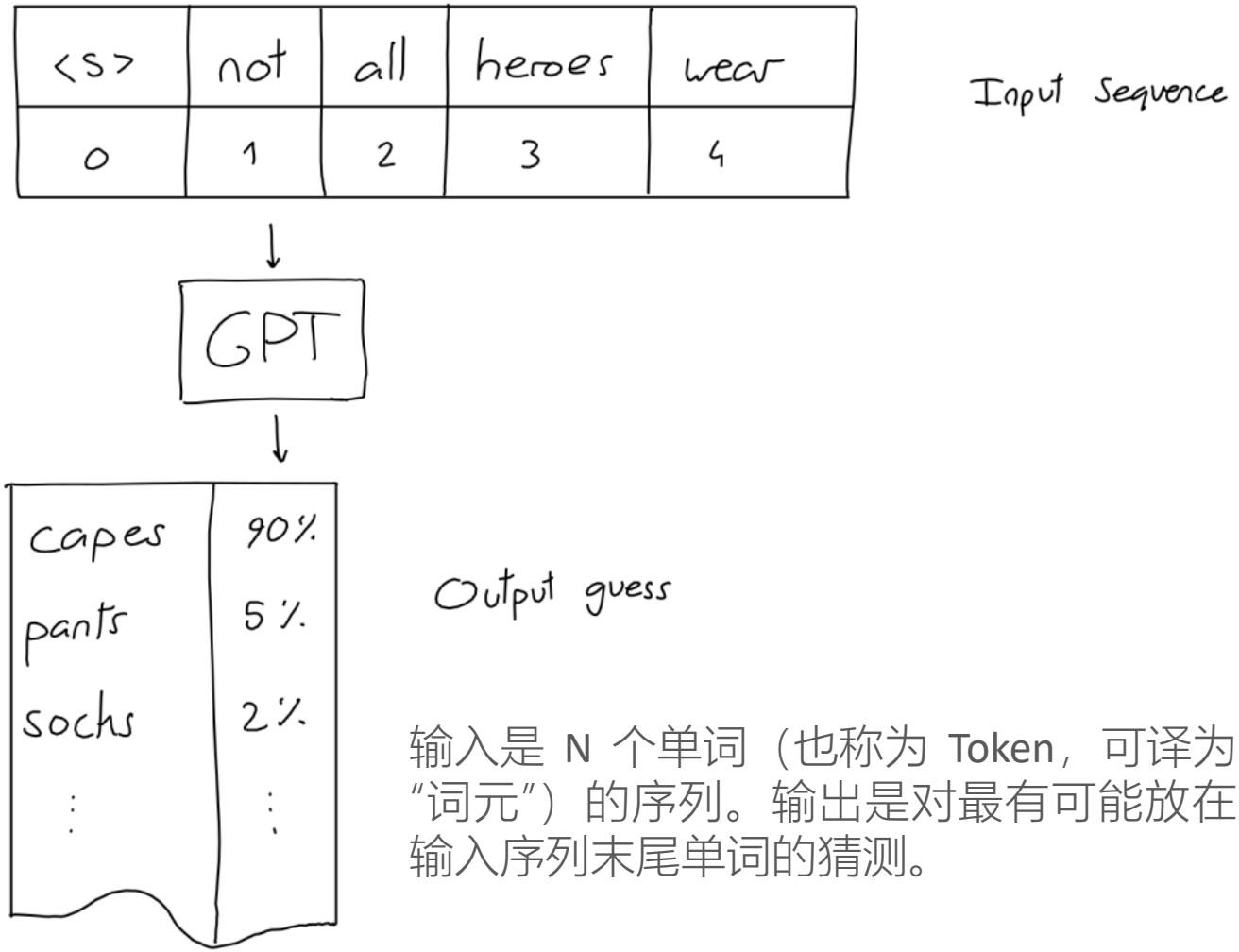


GPT3



预训练模型

GPT 的输入和输出是什么?



输入是 N 个单词（也称为 Token，可译为“词元”）的序列。输出是对最有可能放在输入序列末尾单词的猜测。

预训练模型

所有令人印象深刻的 GPT 对话、故事和示例都是通过这种简单的输入输出方案完成的：给它一个输入序列——得到接下来的一个词。

- **Not all heroes wear -> capes**
- **Not all heroes wear capes -> but**
- **Not all heroes wear capes but -> all**
- **Not all heroes wear capes but all -> villans**
- **Not all heroes wear capes but all villans -> do**

得到下一个单词后，将其添加到输入序列中，再得到下一个单词。

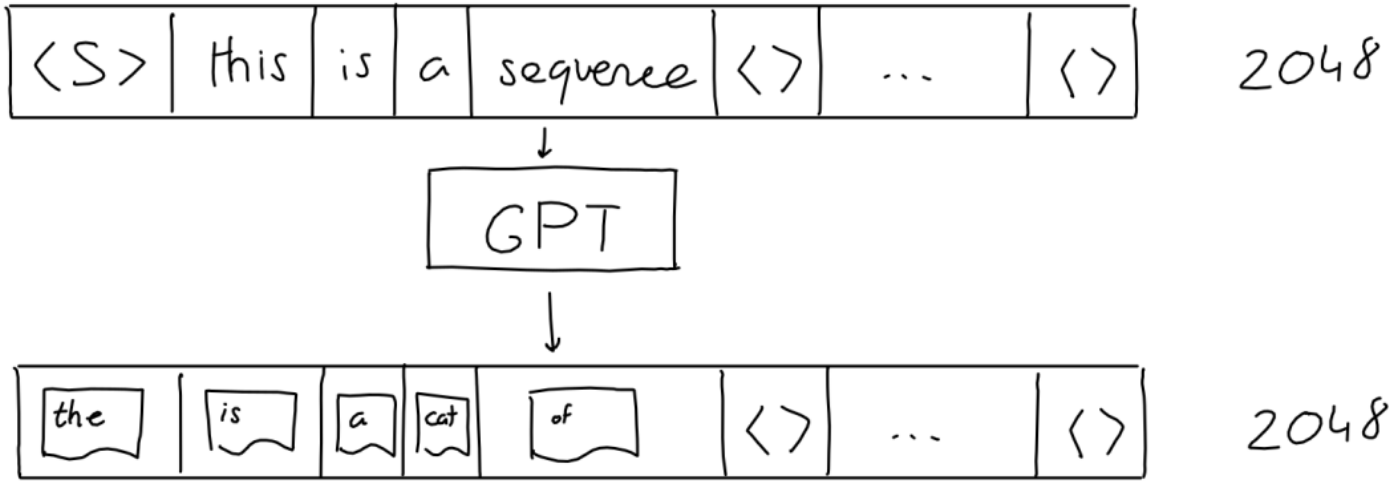
像这样一直重复，最终生成你需要的长文本。

预训练模型

确切来说还有两点需要纠正：

1.输入序列实际上固定为 2048 个 Token（对于 GPT-3 来说）。仍然可以传递短序列作为输入：只需用“空”值填充其他额外的位置。

2.GPT 输出不仅是单个预测，而是一个多预测值（每个可能单词的概率）构成的序列（长度为 2048），每组预测值对应输入序列中的每个单词的“下一个”位置。但是在生成文本时，我们通常只查看序列中最后一个单词的预测。



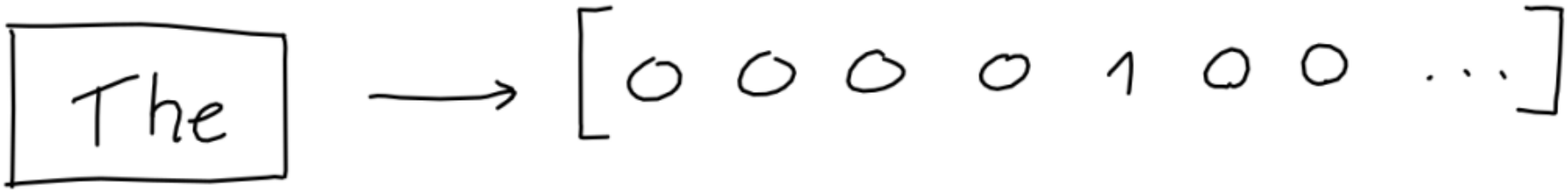
预训练模型

编码（Encoding）

GPT 实际上并不能理解单词含义。作为一种机器学习算法，它是对数字向量进行运算的。那么如何将单词转换为向量呢？

第一步是将所有单词整理为一个词汇表，这使我们能够为每个单词赋予一个值。（GPT 的词汇表包含 50257 个单词）

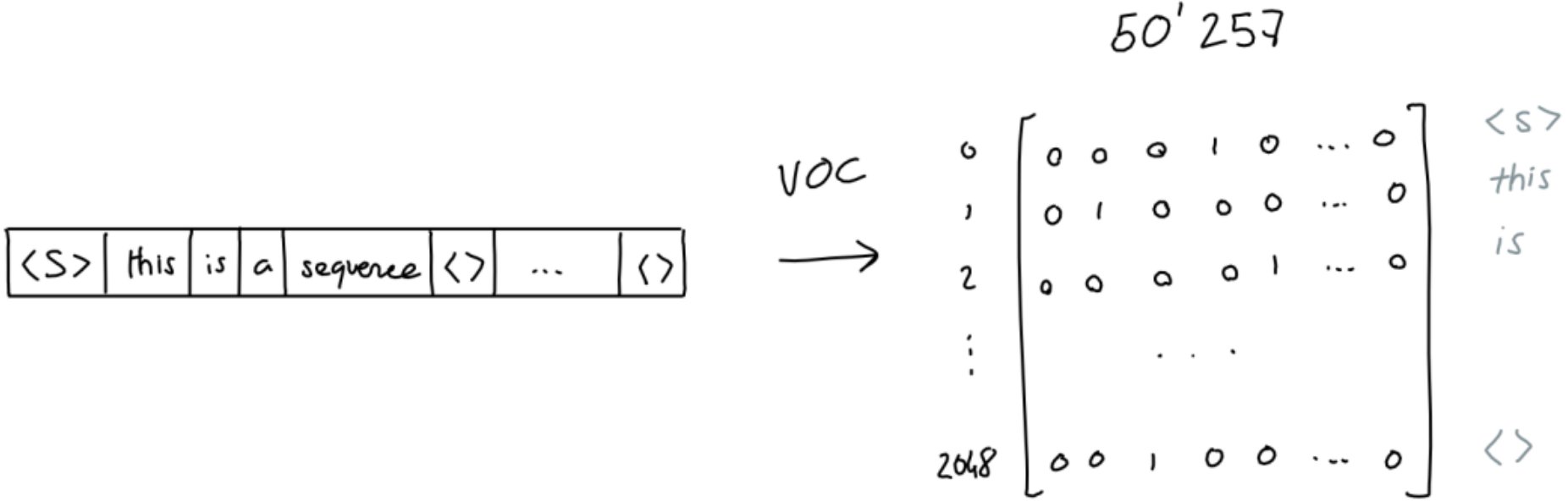
最后，我们可以将每个单词转换为 50257 长度的独热编码（one-hot）向量，其中仅索引 i 处的维（单词的值）为 1，其他维均为 0。



对序列中的每个单词都执行此操作

预训练模型

结果是一个 2048 x 50257 的 1/0 矩阵。

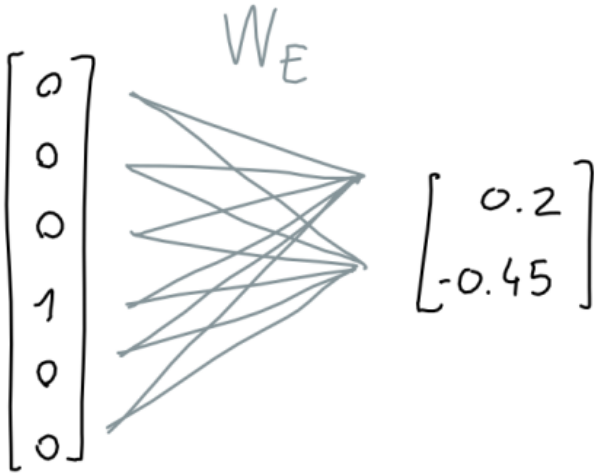


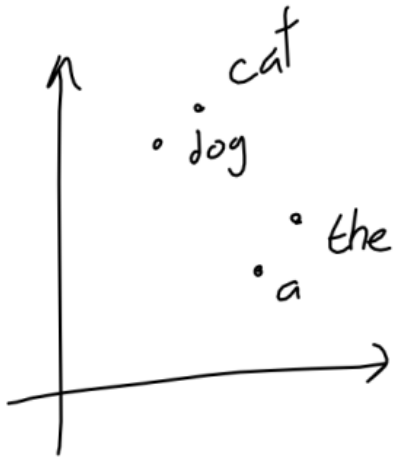
预训练模型

嵌入（Embedding）

对于一个向量来说，50257 相当大了，并且其中大部分都用 0 填充，这样会浪费很多空间。为了解决这个问题，可以学习一个嵌入（Embedding）函数：一个神经网络，以 50257 长度的 1/0 向量为输入，输出一个长度为 n 的数字向量。尝试将单词含义的信息存储（或投影）到较小的空间中。

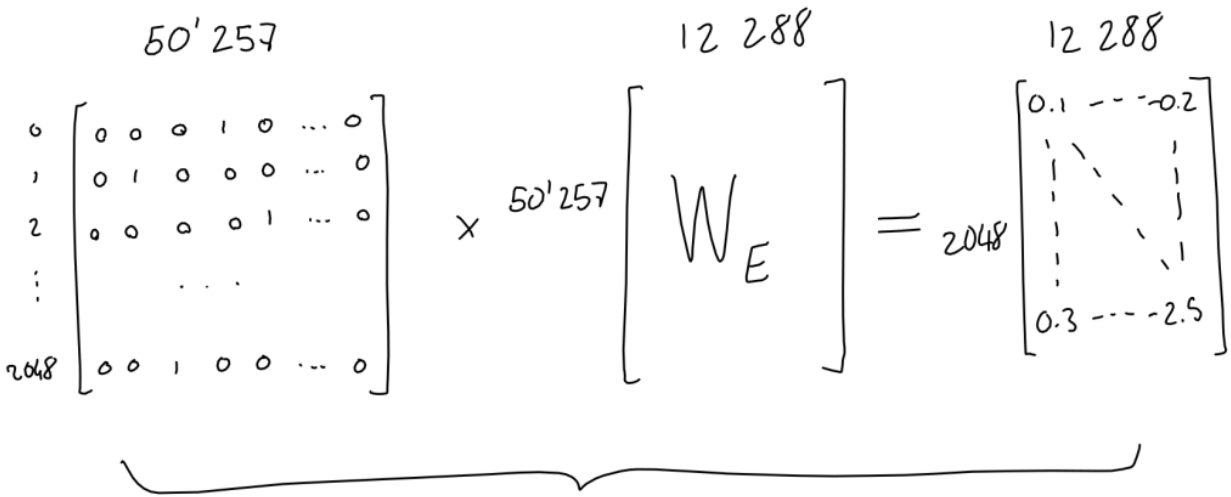
例如，如果嵌入维数为 2，就类似于将每个单词存储在二维空间中的特定坐标处。

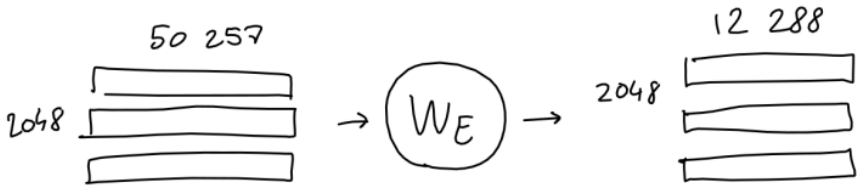




预训练模型

另一种直观的思考方式是，将每个维度都当做虚构的属性，比如“柔软度”或者“亮度”等，为每个属性赋予一个值，我们就可以准确知道哪个词是什么意思。
当然，嵌入维度通常会大于 2：GPT 使用的是 12288 维。
在实践中，每个单词的 one-hot 向量都与学习的嵌入网络权重相乘，最终成为 12288 维嵌入向量。

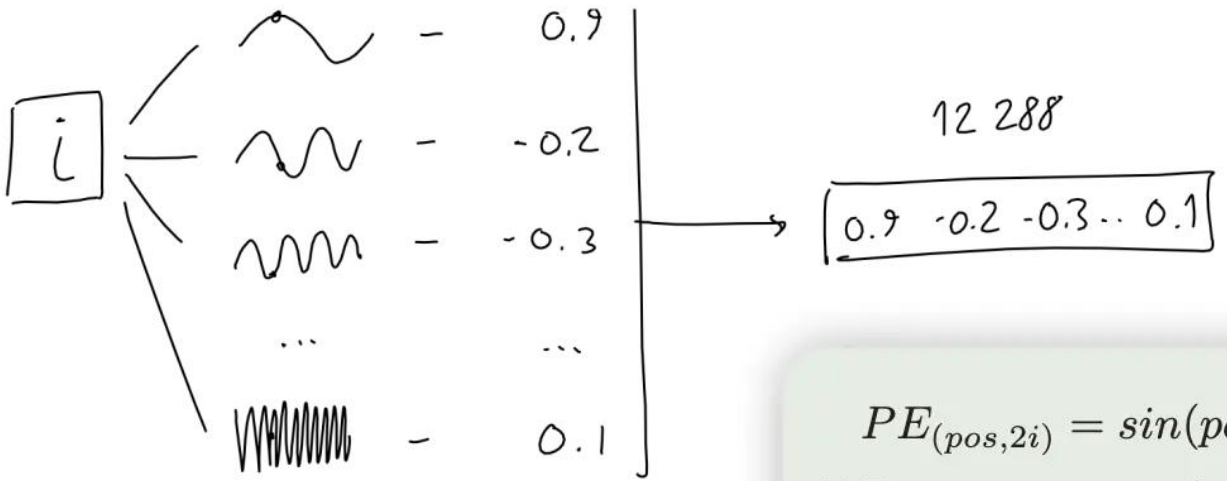




预训练模型

位置编码

为了对当前 Token 在序列中的位置进行编码，作者将 Token 的位置（标量 i，取值范围 [0-2047]）作为参数传递给 12288 个频率不同的正弦函数。



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

预训练模型

这种做法为什么会有效？作者的解释是，生成很多相对位置编码，这对模型很有用。用其他可能的理论来分析这一选择：考虑到信号经常表示为周期性样本之和的方式（参见傅立叶变换），或者语言自然地呈现不同长度循环的可能性（例如诗歌）。

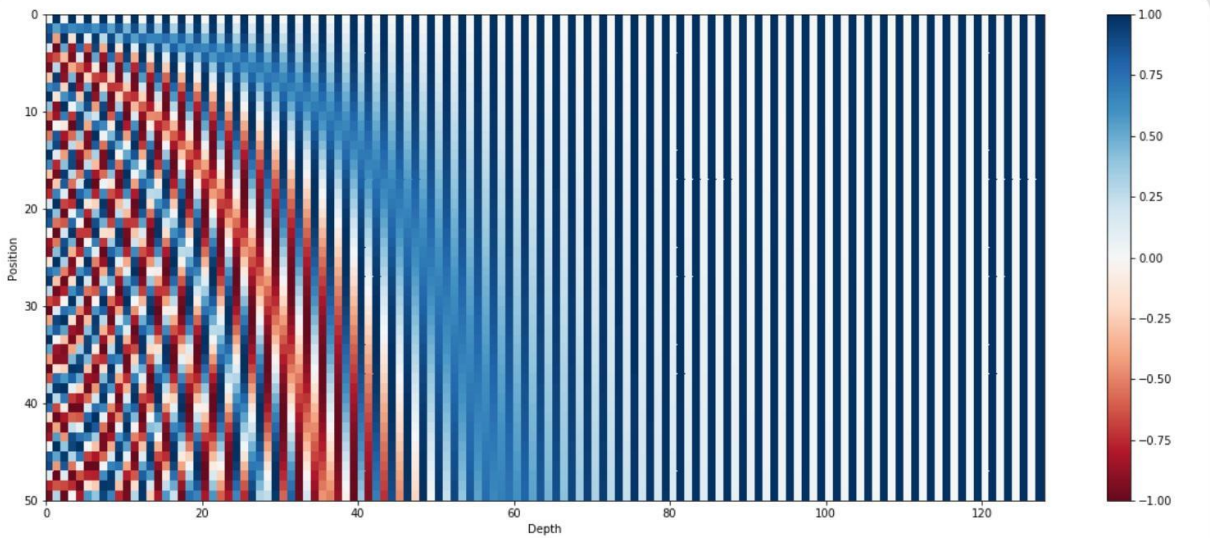
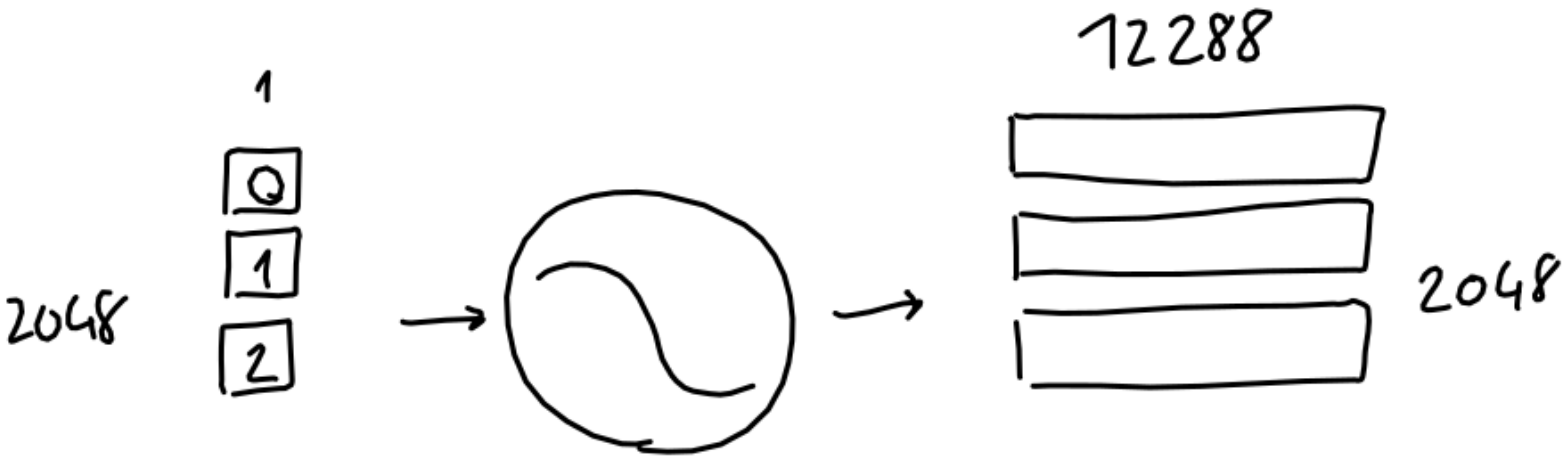


Figure 2 - The 128-dimensional positonal encoding for a sentence with the maximum lenght of 50. Each row represents the embedding vector $\overrightarrow{p_t}$

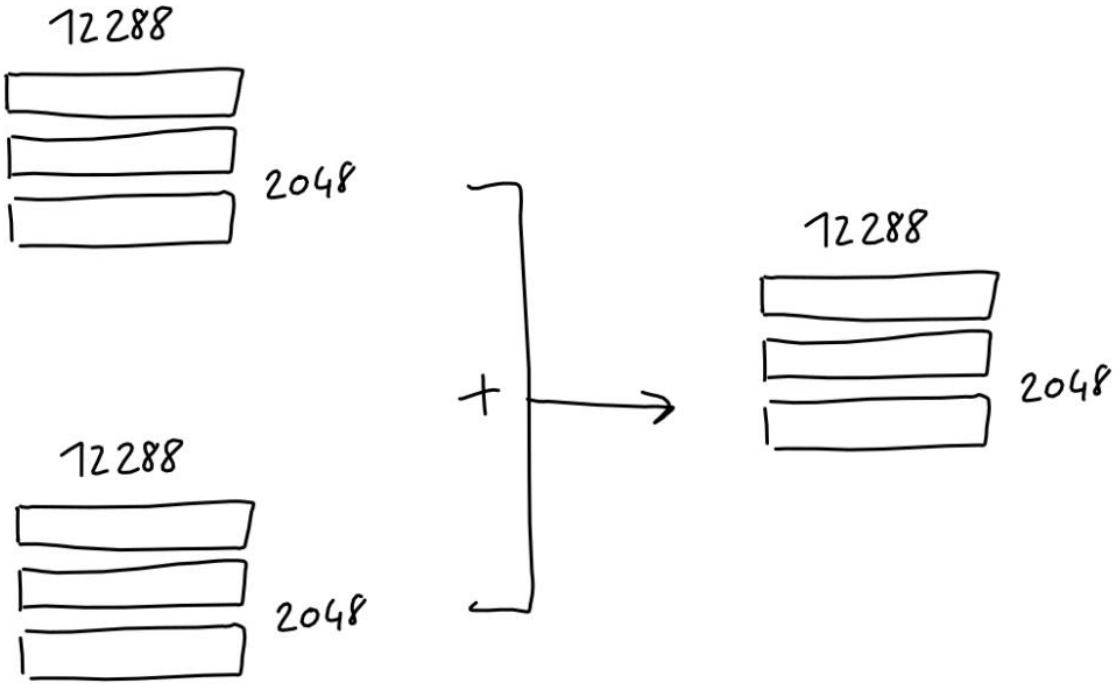
预训练模型

这一过程的结果是，每个 Token 对应一个 12288 维的数字向量。和嵌入操作一样，我们将这些向量组合成具有 2048 行的单一矩阵，其中每一行是序列中每个 Token 的 12288 列位置编码。



预训练模型

最后，与序列嵌入矩阵相同形状的序列位置编码矩阵可以直接添加到嵌入矩阵中。

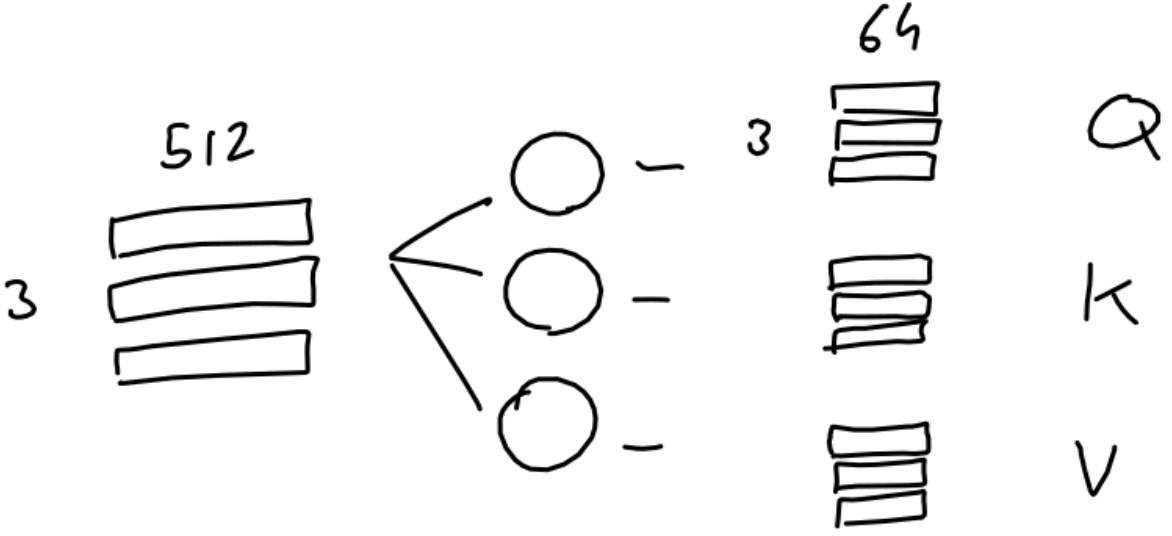


预训练模型

注意力（简化版）

简单来说，注意力（Attention）机制的目的是：对于序列中的每个输出，预测需要关注的输入 Token 是哪些，以及有多少。这里，想象一个由 3 个 Token 组成的序列，每个 Token 都由 512 个值的嵌入表示。

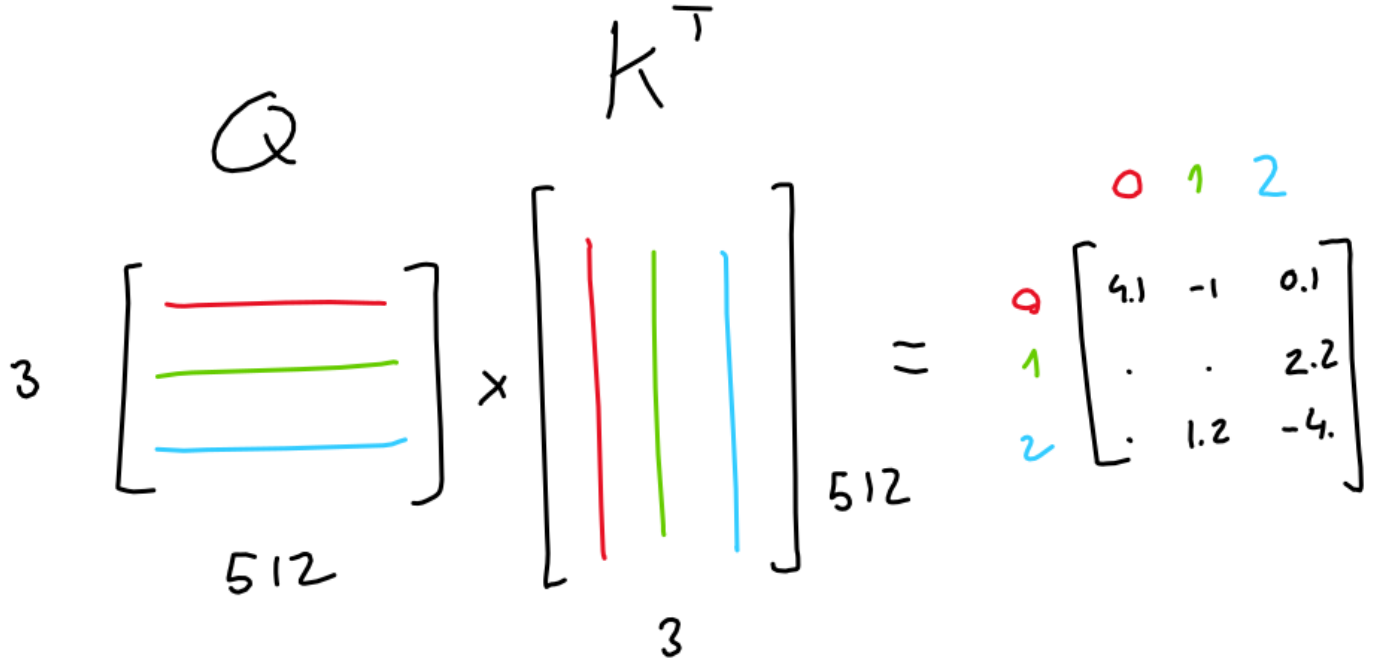
该模型学习 3 个线性投影，所有这些投影都应用于序列嵌入。也可以说是学习了 3 个权重矩阵，这些矩阵将我们的序列嵌入转换为 3 个单独的 3x64 矩阵，每个矩阵分别用于不同的任务。



预训练模型

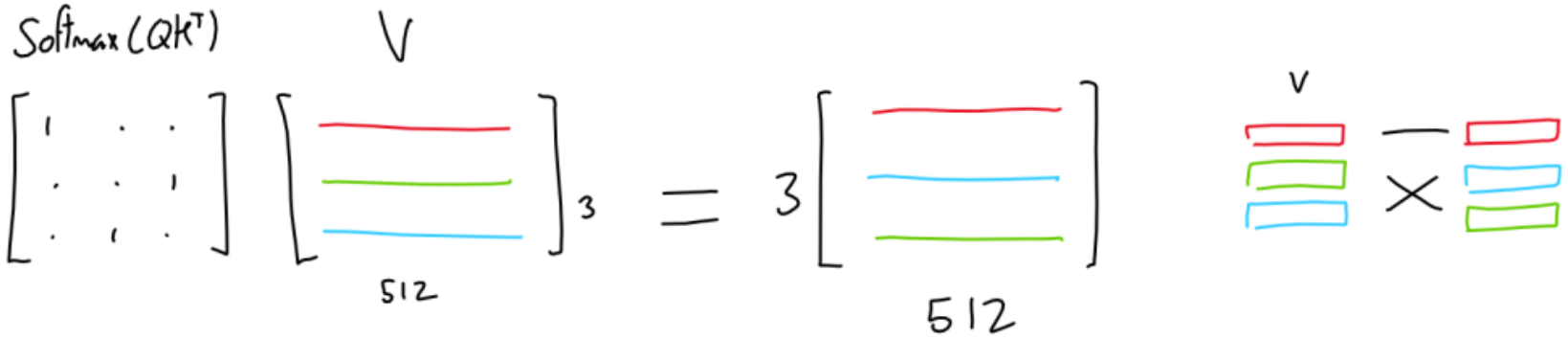
前两个矩阵（“查询 Query”和“键 Key”）相乘（QK^T），得出一个 3x3 矩阵。该矩阵（通过 softmax 归一化）表示每个 Token 相对于其他 Token 的重要性。

注：（QK^T）操作是 GPT 中唯一**跨**单词的操作，也就是矩阵中涉及行交互的唯一操作。



预训练模型

第三个矩阵（“值 Value”）与这个重要性矩阵相乘，从而为每个 Token 生成所有其他 Token 值的混合（按各个 Token 的重要性加权）。

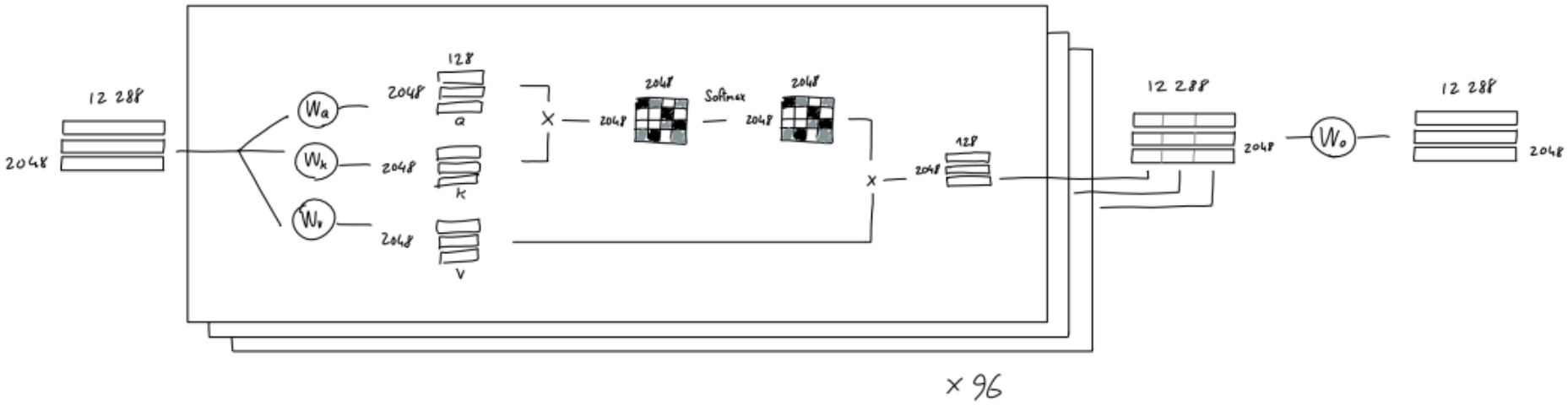


预训练模型

多头注意力（Multi-Head Attention）

GPT 模型中使用了多头注意力。这仅仅意味着，上述过程被重复了很多次（GPT-3 中为 96 次），每次过程都有不同的学习查询、键、值投影权重。

每个注意力头的结果（单个 2048 x 128 矩阵）被拼接在一起，生成一个 2048 x 12288 矩阵，然后将其乘以一个线性投影（不会改变矩阵形状），以达到良好的度量。

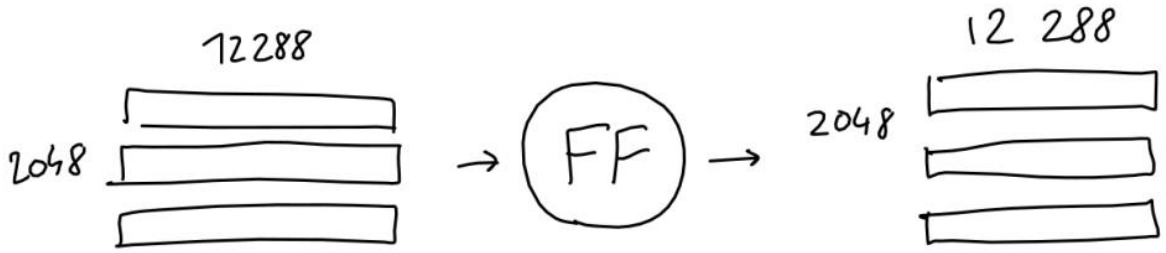


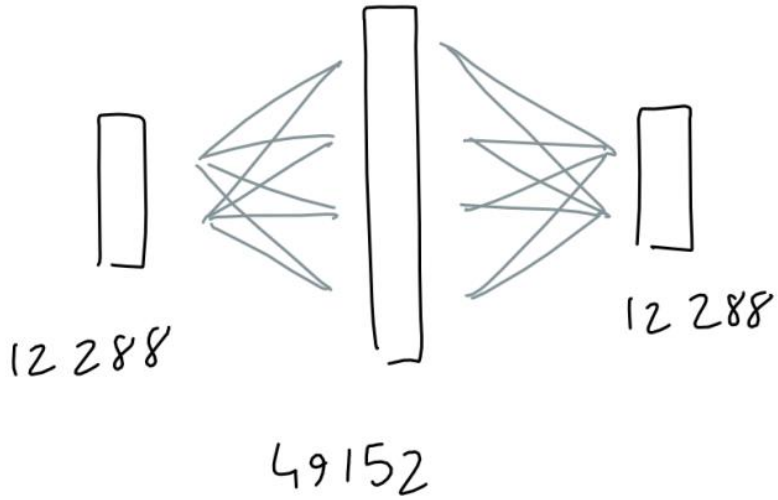
预训练模型

前馈（Feed Forward）

前馈模块是我们熟知的多层感知器（Multi-Later Perceptron，MLP），具有一个隐含层。获取输入，乘以学习的权重，添加学习的偏差，重复该过程，获得结果。

此处，输入和输出形状都相同（2048 x 12288），但是隐藏层的大小为 4 x 12288。

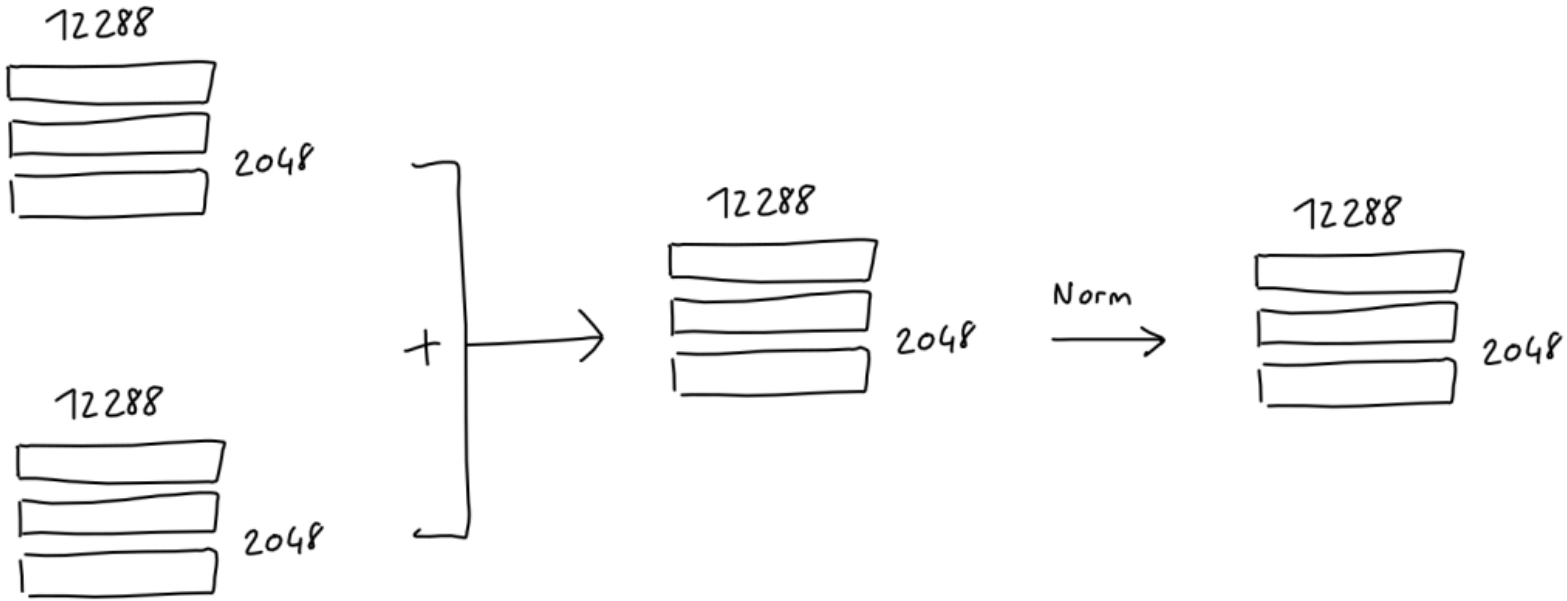




预训练模型

相加和归一化

在“多头注意力”和“前馈”模块之后，将模块的输入添加到输出中，然后对结果进行归一化。这在深度学习模型中很常见（自从 ResNet 之后）。

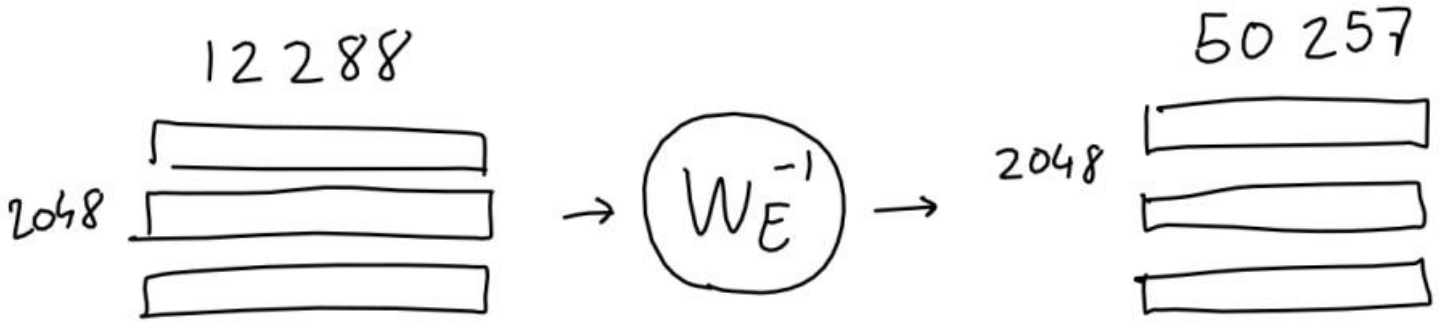


预训练模型

解码（Decoding）

马上就要完成了！通过所有 96 层 GPT-3 的注意力/神经网络机制后，输入已处理为一个 2048 x 12288 矩阵。对于序列中 2048 个输出位置，该矩阵都应该对应包含一个 12288 维向量，其中包含了可能的单词信息。那么，要如何将这些信息提取出来呢？

回想“嵌入”部分，我们学习了一种映射，该映射将给定单词（的独热编码）转换为了一个 12288 维向量嵌入。实际上，我们可以反转此映射，将输出的 12288 维向量嵌入转换回 50257 维单词编码。这一思路就是，既然已经花费了大量精力学习从单词到数字的良好映射，不妨重新利用！



预训练模型

当然，这样操作不会得到与最开始相同的 1/0 值，但这是一件好事：在快速 softmax 之后，我们可以将结果值视为每个单词的概率。
此外，GPT 论文还提到了参数 top-k，该参数将输出中要采样的可能单词数量限制为 k 个最可能的值。例如，当 top-k 参数为 1 时，选择的就是最有可能的单词。

预训练模型

随着 2018 年谷歌发布基于 Transformer 机器学习方法的自然语言处理预训练模型 BERT，人工智能领域进入了大炼模型参数的预训练模型时代。AI 预训练模型，又称为大模型、基础模型（foundation model），即基于大量数据(通常使用大规模自我监督学习)训练的、拥有巨量参数的模型，可以适应广泛的下游任务。

	预训练模型	应用	参数量	领域
谷歌	BERT	语言理解与生成	4810 亿	NLP
	LaMDA	对话系统		NLP
	PaLM	语言理解与生成、推理、代码生成	5400亿	NLP
	Imagen	语言理解与图像生成	110亿	多模态
	Parti	语言理解与图像生成	200亿	多模态

预训练模型

微软	Florence	视觉识别	6.4亿	CV
	Turing-NLG	语言理解、生成	170亿	NLP
Facebook	OPT-175B	语言模型	1750亿	NLP
	M2M-100	100种语言互译	150亿	NLP
Deep Mind	Gato	多面手的智能体	12亿	多模态
	Gopher	语言理解与生成	2800亿	NLP
	AlphaCode	代码生成	414亿	NLP

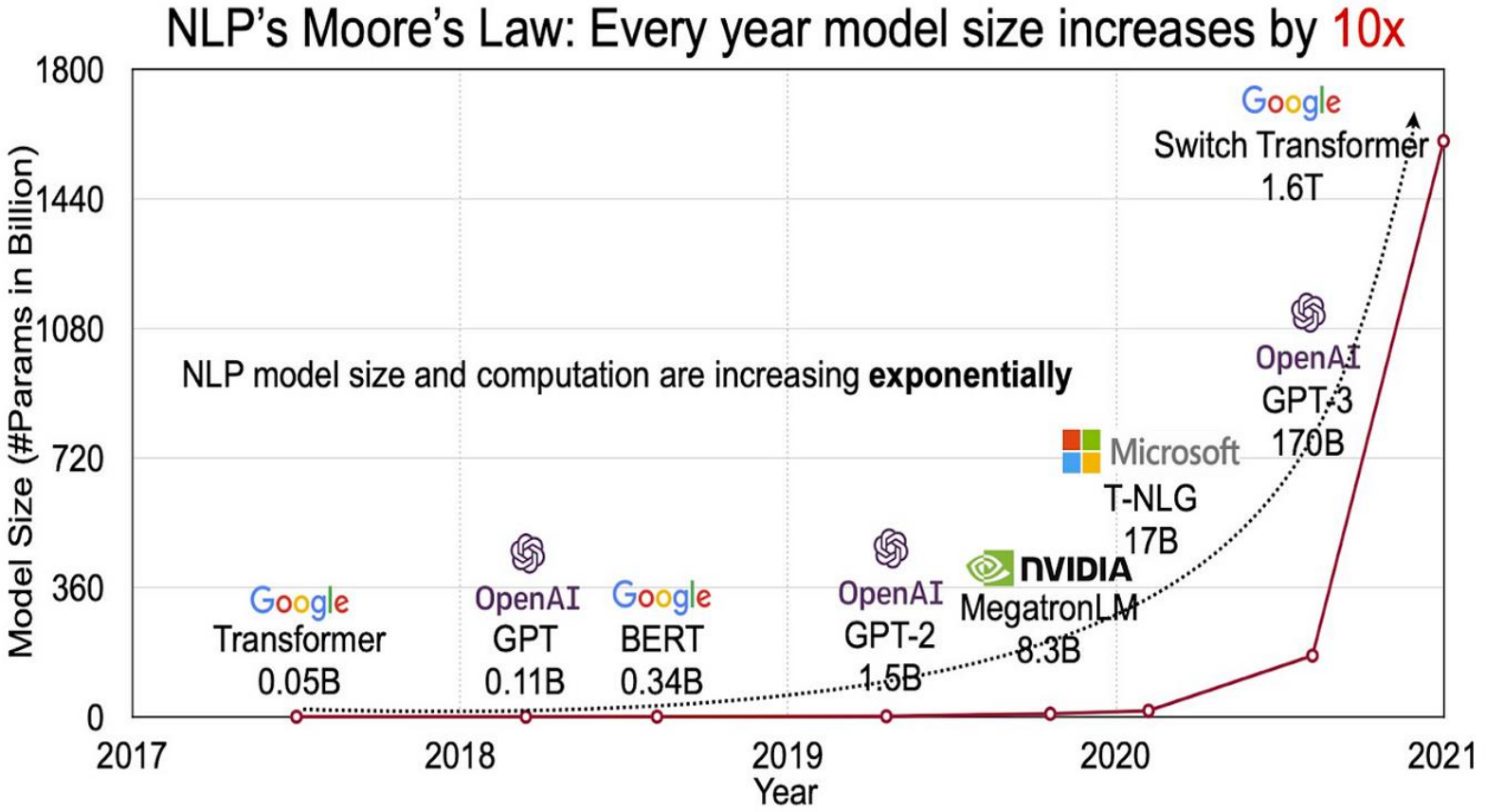
预训练模型

Open AI	GPT3	语言理解与生成、推理等	1750亿	NLP
	CLIP&DALL-E	图像生成、跨模态检索	120亿	多模态
	Codex	代码生成	120亿	NLP
	ChatGPT	语言理解与生成、推理等		NLP
英伟达	Megatron-Turing NLG	语言理解与生成、推理	5300亿	NLP
Stability AI	Stable Diffusion	语言理解与图像生成		多模态

预训练模型

模型	年份	关键创新	训练集	参数数量	上下文窗口
GPT-1	2018	Transformer 解码器, 自监督训练 + 微调	4.5 GB	117M	512
GPT-2	2019	修改的归一化层,	40 GB	1.5B	1,024
GPT-3	2020	稀疏注意力层，支持零样本	57 TB	175B	2,048 令牌
GPT-4	2023	多模态输入（文本 + 图像）	?	1.76T	32,000 令牌

模型参数量指数增长


NLP's Moore's Law: Every year model size increases by 10x
Model Size (#Params in Billion)
1800
1440
1080
720
360
0
2017
2018
2019
2020
2021
Year
NLP model size and computation are increasing exponentially
Google
Transformer
0.05B
OpenAI
GPT
0.11B
Google
BERT
0.34B
OpenAI
GPT-2
1.5B
NVIDIA
MegatronLM
8.3B
Microsoft
T-NLG
17B
Google
Switch Transformer
1.6T
OpenAI
GPT-3
170B

智能涌现

Sam Altman
@sama

a new version of moore's law that could start soon:

the amount of intelligence in the universe doubles every 18 months

12:24 AM · Feb 27, 2023 · **3.8M** Views

1,923 Retweets **702** Quotes **14.8K** Likes

大模型训练成本

Optimal LLM Training Cost				
Model	Size (# Parameters)	Tokens	GPU	Optimal Training Compute Cost
MosaicML GPT-30B	30 Billion	610 Billion	A100	$ 325,855
Google LaMDA	137 Billion	168 Billion	A100	$ 368,846
Yandex YaLM	100 Billion	300 Billion	A100	$ 480,769
Tsinghua University Zhipu.AI GLM	130 Billion	400 Billion	A100	$ 833,333
Open AI GPT-3	175 Billion	300 Billion	A100	$ 841,346
AI21 Jurassic	178 Billion	300 Billion	A100	$ 855,769
Bloom	176 Billion	366 Billion	A100	$ 1,033,756
DeepMind Gopher	280 Billion	300 Billion	A100	$ 1,346,154
DeepMind Chinchilla	70 Billion	1,400 Billion	A100	$ 1,745,014
MosaicML GPT-70B	70 Billion	1,400 Billion	A100	$ 1,745,014
Nvidia Microsoft MT-NLG	530 Billion	270 Billion	A100	$ 2,293,269
Google PaLM	540 Billion	780 Billion	A100	$ 6,750,000


Google DeepMind
INTRODUCING
Gemini

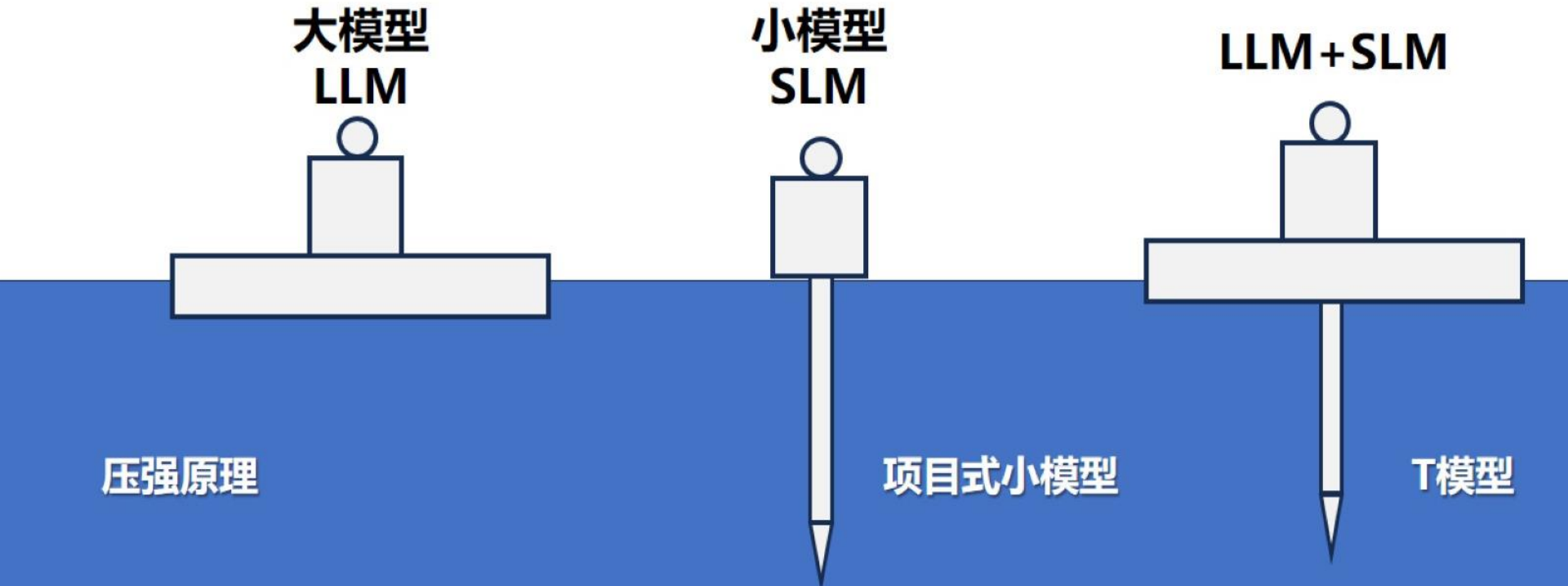
生成式人工智能大型语言模型的不足
（Large Language Models， LLM）

- 准确性问题，“一本正经的胡说八道”。
- 需要大算力大语料大资金支持，训练维护LLM是资源密集型，导致高成本低效率；
- LLM会将预训练数据偏差引入其生成的文本中，产生不准确的信息，无法100%满足教育领域严格管规要求；
- 目前通用型LLM在教育教学领域无法适应学科专业性要求；
- 信息偏见、信息安全问题。

未来的小模型发展方向
(Small Language Models， SLM)

- 小语言模型是未来发展方向：
  - **高度定制：** 根据用户需求，聚焦具体目标，例如；可以为一门学科，一本教材定制小模型。
  - **高效率低成本：** 由于SLM规模较小，用更少的数据训练，部署效率更高，可在功能较弱的硬件上运行，不仅节省成本，而且更实用，更高回报。
  - **高准确性：** SLM通过对特定专业目标数据集进行有针对性训练，可以有效控制训练数据的质量和完整性，能够更可靠地提供高质量准确的结果，这对教育尤其重要！
  - **高安全性：** 与大语言模型相比，小模型的小代码库和更少的参数，这种低复杂性最大限度地减少安全漏洞，从而使小模型在安全性方面更具优势。
  - **高透明度/可控度：** 小模型更透明和可解释的运作方式，让学校用户可以确保小模型符合安全协议和法规要求。小模型在本地或受控环境中处理数据，有助于保护敏感信息和数据隐私，从而防范数据泄露或未经授权访问的风险。

预训练模型



生成式人工智能小模型编辑训练系统

SLM × N

星火小模型编辑训练系统

- 用户自定制场景;
- 用户自有数据特训;
- 用户外挂专属知识库;
- 用户提示词调优/评估;
- 星火大模型提供算法、算力;
- 用户自编辑训练专属小模型。

星火大模型指令集

提示词/指令模板化，简化操作

LLM × 1

星火认知大模型

星火助手中心

细分应用场景，最小成本调教的极简小模型

举例

大算法 · 大算力 · 大数据 ， 大资金

提纲

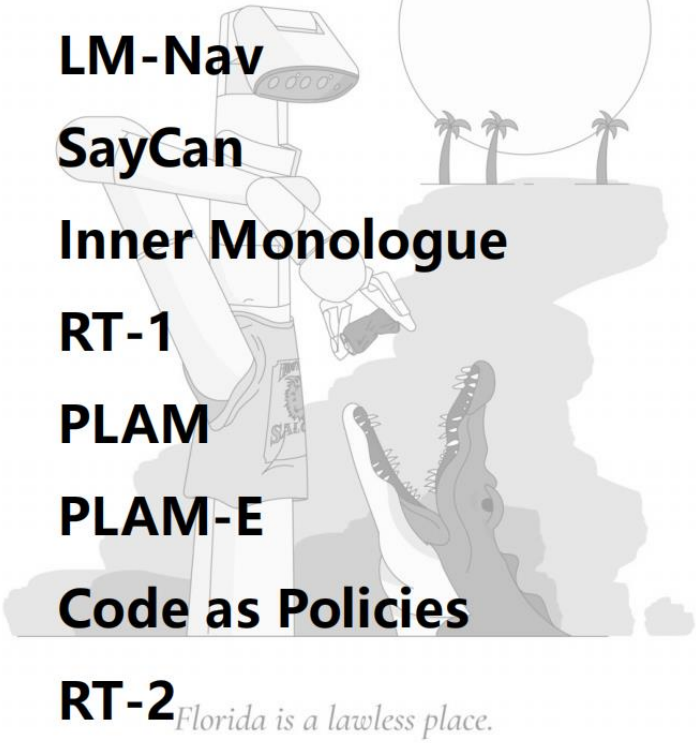
一、生成式AI

二、预训练模型

三、具身智能

具身智能

Robotics at Google



LM-Nav

SayCan

Inner Monologue

RT-1

PLAM

PLAM-E

Code as Policies

RT-2

Other Reseachers

VoxPoser - Fei-Fei Li

Diffusion Policy - Shuran Song

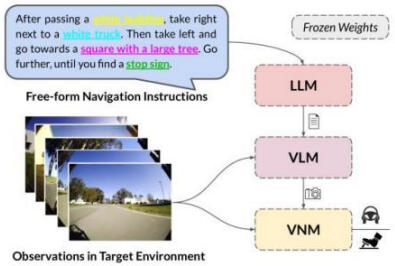
3D-LLM - Chuang Gan

etc.

Timeline

LM-Nav

Navigation based LM

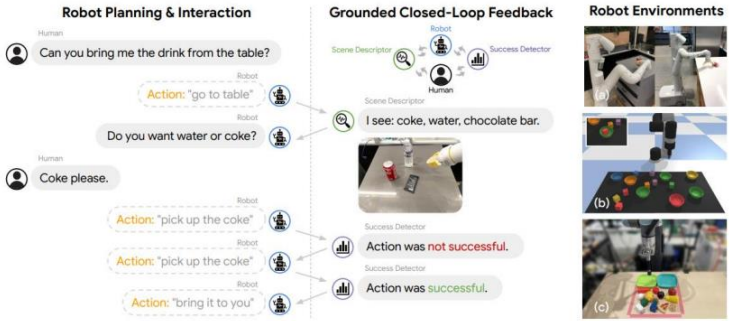


SayCan

LLM Planner + BC-Z Skills

Inner Monologue

Build Feedback System for SayCan



*cite from Robotics at Google

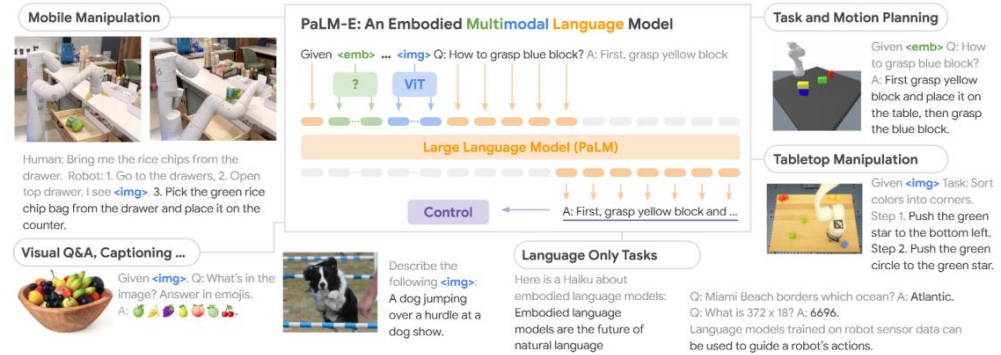
具身智能

RT-1

Transformer for low-level skills

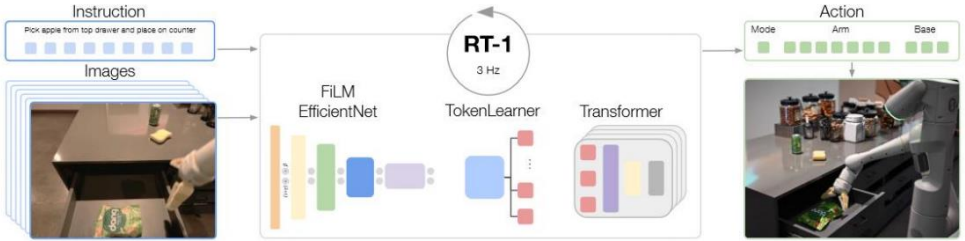
PLAM

Google self-designed LLM



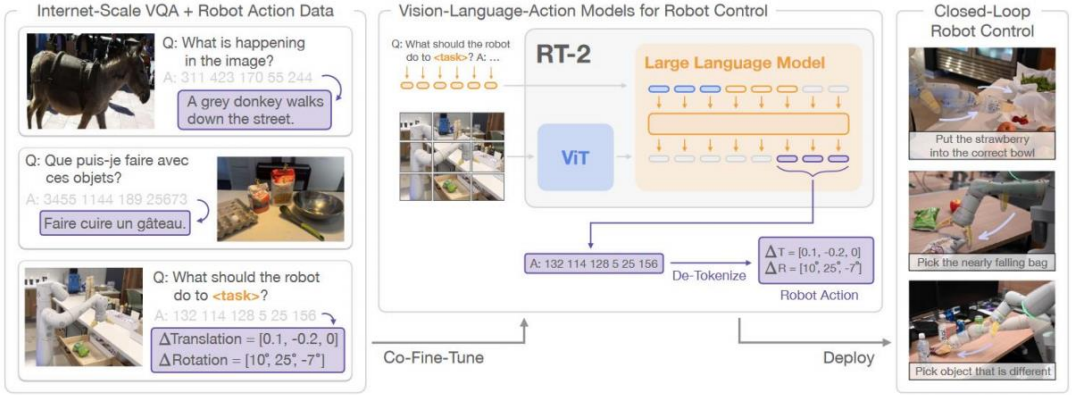
Embodied Multi-modal LM

PLAM-E



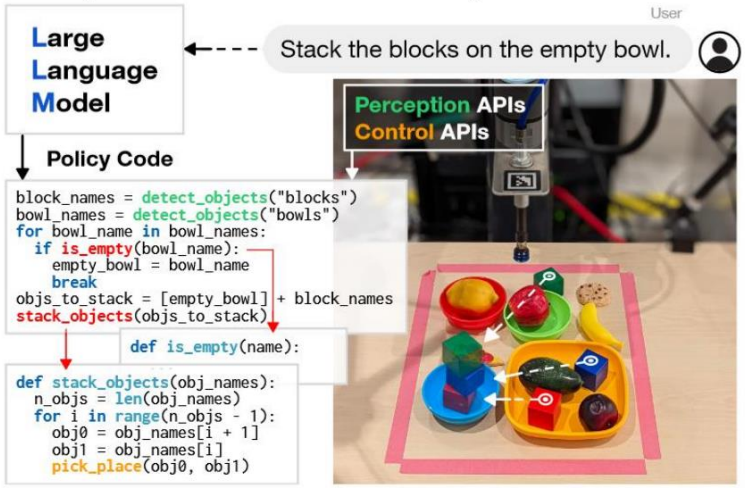
*cite from Robotics at Google

具身智能



Code as Policies

Prompt to generate codes



end-to-end low-level controller

RT-2

*cite from Robotics at Google

具身智能

SayCan

Paper

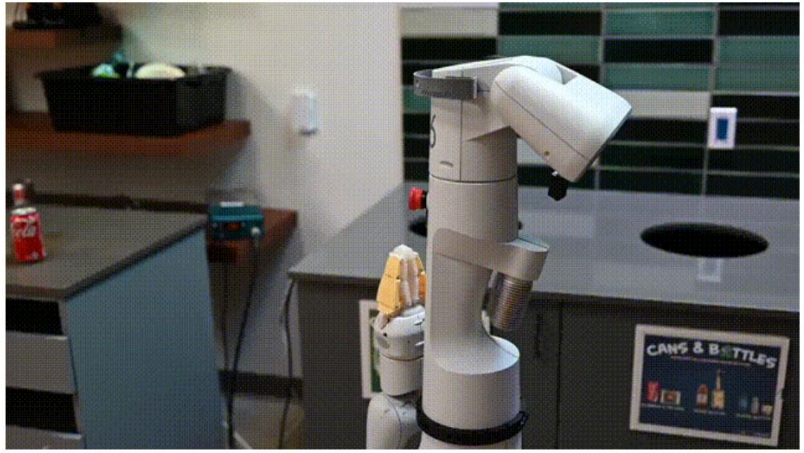
Do As I Can, Not As I Say:
Grounding Language in Robotic Affordances

Michael Ahn* Anthony Brohan* Noah Brown* Yevgen Chebotar* Omar Cortes* Byron David* Chelsea Finn*
Chuyuan Fu* Keerthana Gopalakrishnan* Karol Hausman* Alex Herzog* Daniel Ho* Jasmine Hsu* Julian Ibarz*
Brian Ichter* Alex Irpan* Eric Jang* Rosario Jauregui Ruano* Kyle Jeffrey* Sally Jesmonth* Nikhil Joshi*
Ryan Julian* Dmitry Kalashnikov* Yuheng Kuang* Kuang-Huei Lee* Sergey Levine* Yao Lu* Linda Luu* Carolina Parada*
Peter Pastor* Jornell Quiambao* Kanishka Rao* Jarek Rettinghouse* Diego Reyes* Pierre Sermanet* Nicolas Sievers*
Clayton Tan* Alexander Toshev* Vincent Vanhoucke* Fei Xia* Ted Xiao* Peng Xu* Sichun Xu* Mengyuan Yan* Andy Zeng*

Robotics at Google Everyday Robots

Overview

Demo



*cite from Robotics at Google


EXIT
076
076

具身智能

RT-1

Paper

RT-1: ROBOTICS TRANSFORMER FOR REAL-WORLD CONTROL AT SCALE

Anthony Brohan*, Noah Brown*, Justice Carbajal*, Yevgen Chebotar*, Joseph Dabis*, Chelsea Finn*, Keerthana Gopalakrishnan*, Karol Hausman*, Alex Herzog†, Jasmine Hsu*, Julian Ibarz*, Brian Ichter*, Alex Irpan*, Tomas Jackson*, Sally Jesmonth*, Nikhil J Joshi*, Ryan Julian*, Dmitry Kalashnikov*, Yuheng Kuang*, Isabel Leal*, Kuang-Huei Lee‡, Sergey Levine*, Yao Lu*, Utsav Malla*, Deeksha Manjunath*, Igor Mordatch‡, Ofir Nachum‡, Carolina Parada*, Jodilyn Peralta*, Emily Perez*, Karl Pertsch*, Jornell Quiambao*, Kanishka Rao*, Michael Ryoo*, Grecia Salazar*, Pannag Sanketi*, Kevin Sayed*, Jaspiar Singh*, Sumedh Sontakke‡, Austin Stone*, Clayton Tan*, Huong Tran*, Vincent Vanhoucke*, Steve Vega*, Quan Vuong*, Fei Xia*, Ted Xiao*, Peng Xu*, Sichun Xu*, Tianhe Yu*, Brianna Zitkovich*

*Robotics at Google, †Everyday Robots, ‡Google Research, Brain Team

Details

1. Action Spaces

arm: x, y, z, roll, pitch, yaw, opening of grasper

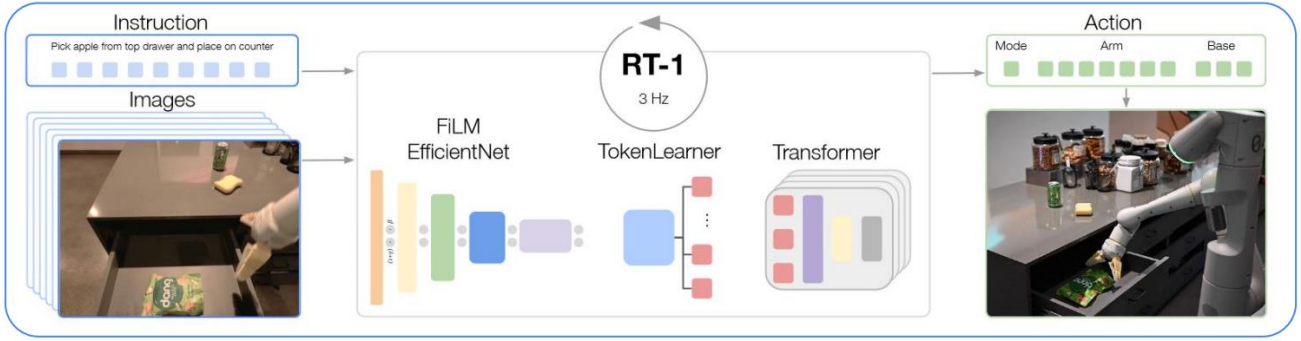
base: x, y, yaw

switch mode: {control the arm, the base, termination}

control frequency = 3Hz

Action tokenization. To tokenize actions, each action dimension in RT-1 is discretized into 256 bins. As mentioned previously, the action dimensions we consider include seven variables for the arm movement (x, y, z, roll, pitch, yaw, opening of the gripper), three variables for base movement (x, y, yaw) and a discrete variable to switch between three modes: controlling arm, base or terminating the episode. For each variable, we map the target to one of the 256 bins, where the bins are uniformly distributed within the bounds of each variable.

Overview



*cite from Robotics at Google

具身智能之大模型

PALM-E

Paper

PaLM-E: An Embodied Multimodal Language Model

Danny Driess[1,2] Fei Xia[1] Mehdi S. M. Sajjadi[3] Corey Lynch[1] Aakanksha Chowdhery[3]
Brian Ichter[1] Ayzaan Wahid[1] Jonathan Tompson[1] Quan Vuong[1] Tianhe Yu[1] Wenlong Huang[1]
Yevgen Chebotar[1] Pierre Sermanet[1] Daniel Duckworth[3] Sergey Levine[1] Vincent Vanhoucke[1]
Karol Hausman[1] Marc Toussaint[2] Klaus Greff[3] Andy Zeng[1] Igor Mordatch[3] Pete Florence[1]

[1] Robotics at Google [2] TECHNISCHE UNIVERSITÄT BERLIN [3] Google Research

Details

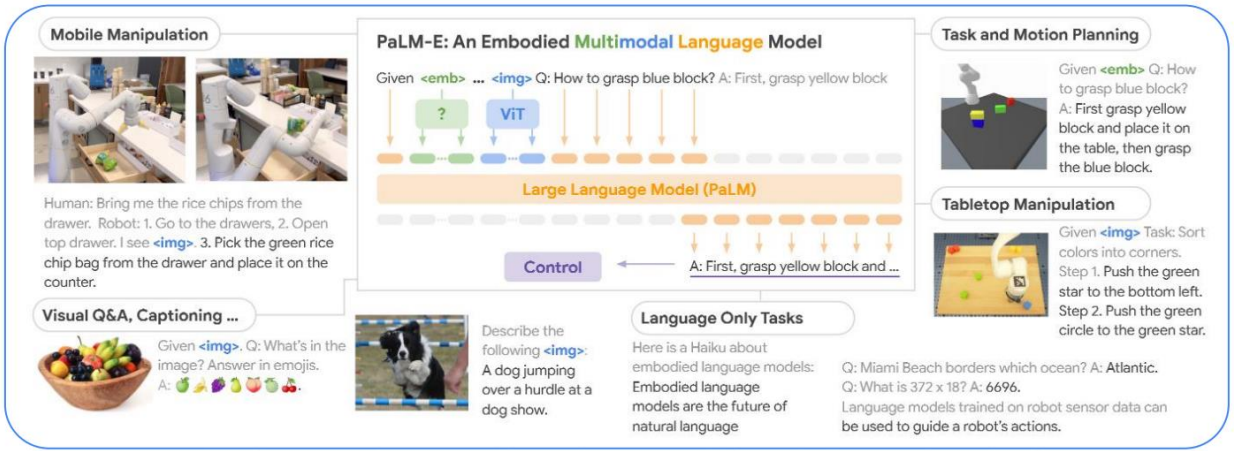
1. Multi-modal Imputs

1.1 State estimation

1.2 Images

1.3 Language

Overview




4x speed
106

具身智能

Code as Polices

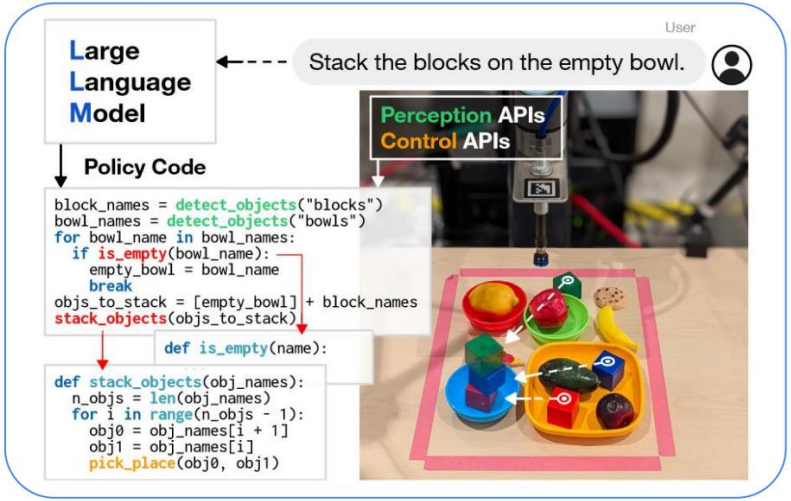
Paper

Code as Policies:
Language Model Programs for Embodied Control

Jacky Liang Wenlong Huang Fei Xia Peng Xu Karol Hausman Brian Ichter Pete Florence Andy Zeng

Robotics at Google

Overview



Details

1. APIs

Perception APIs

of LMP-based policies. For example, in real-world experiments below, we use recently developed open-vocabulary object detection models like ViLD [3] and MDETR [2] off-the-shelf to obtain object positions and bounding boxes.

Control APIs

architect a dynamic codebase. We demonstrate across several robot systems that LLMs can autonomously interpret language commands to generate LMPs that represent reactive low-level policies (e.g., PD or impedance controllers), and waypoint-based policies (e.g., for vision-based pick and place, or trajectory-based control).

where `put_first_on_second` is an existing open vocabulary pick and place primitive (e.g., CLIPort [36]). For new embodiments, these active function calls can be replaced with available control APIs that represent the action space (e.g., `set_velocity`) of the agent. Hierarchical code-gen with verbose variable names

*cite from Robotics at Google

RT-2

Paper

RT-2: Vision-Language-Action Models

Transfer Web Knowledge to Robotic Control

Anthony Brohan Noah Brown Justice Carbajal Yevgen Chebotar Xi Chen Krzysztof Choromanski Tianli Ding
Danny Driess Avinava Dubey Chelsea Finn Pete Florence Chuyuan Fu Montse Gonzalez Arenas Keerthana Gopalakrishnan
Kehang Han Karol Hausman Alex Herzog Jasmine Hsu Brian Ichter Alex Irpan Nikhil Joshi Ryan Julian
Dmitry Kalashnikov Yuheng Kuang Isabel Leal Lisa Lee Tsang-Wei Edward Lee Sergey Levine Yao Lu Henryk Michalewski
Igor Mordatch Karl Pertsch Kanishka Rao Krista Reymann Michael Ryoo Grecia Salazar Pannag Sanketi Pierre Sermanet
Jaspiar Singh Anikait Singh Radu Soricut Huong Tran Vincent Vanhoucke Quan Vuong Ayzaan Wahid Stefan Welker
Paul Wohlhart Jialin Wu Fei Xia Ted Xiao Peng Xu Sichun Xu Tianhe Yu Brianna Zitkovich

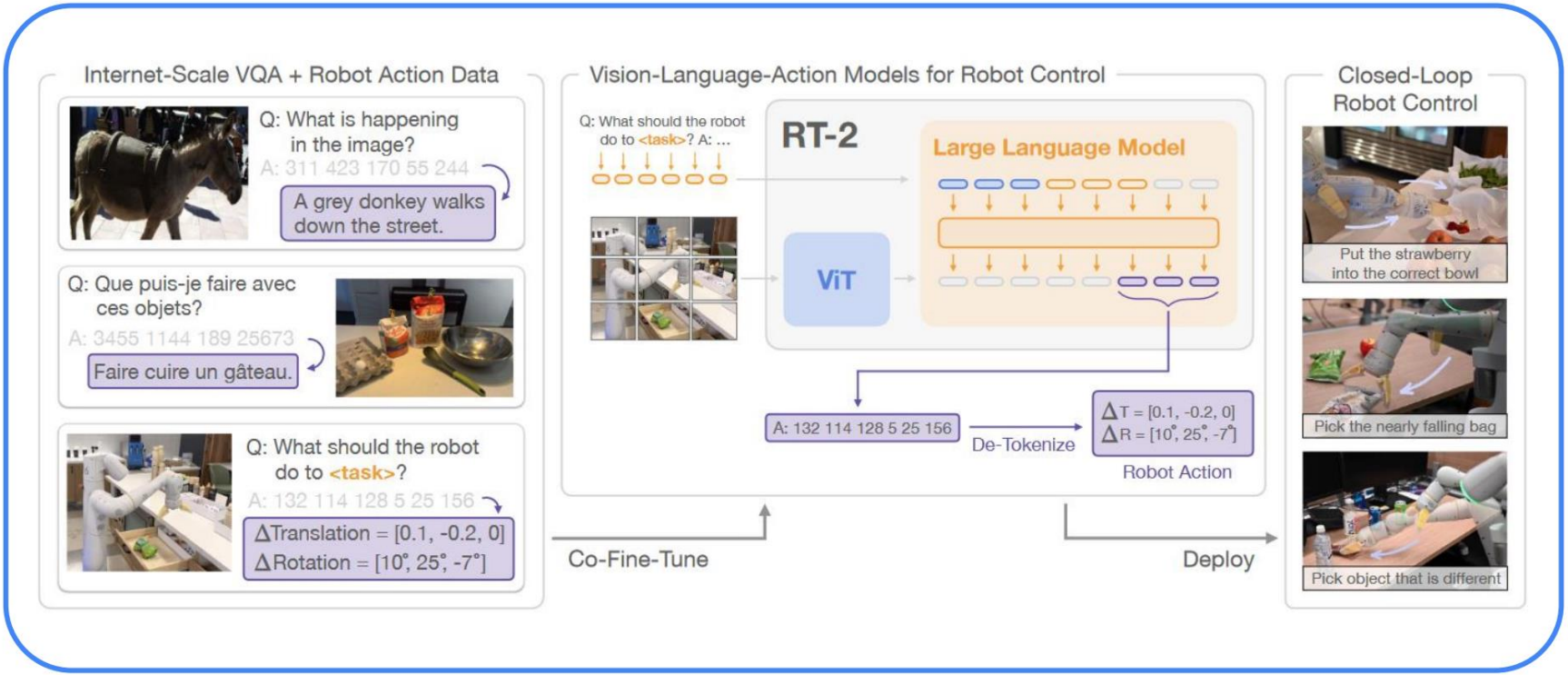
Authors listed in alphabetical order (see paper appendix for contribution statement).

Google DeepMind

具身智能

RT-2

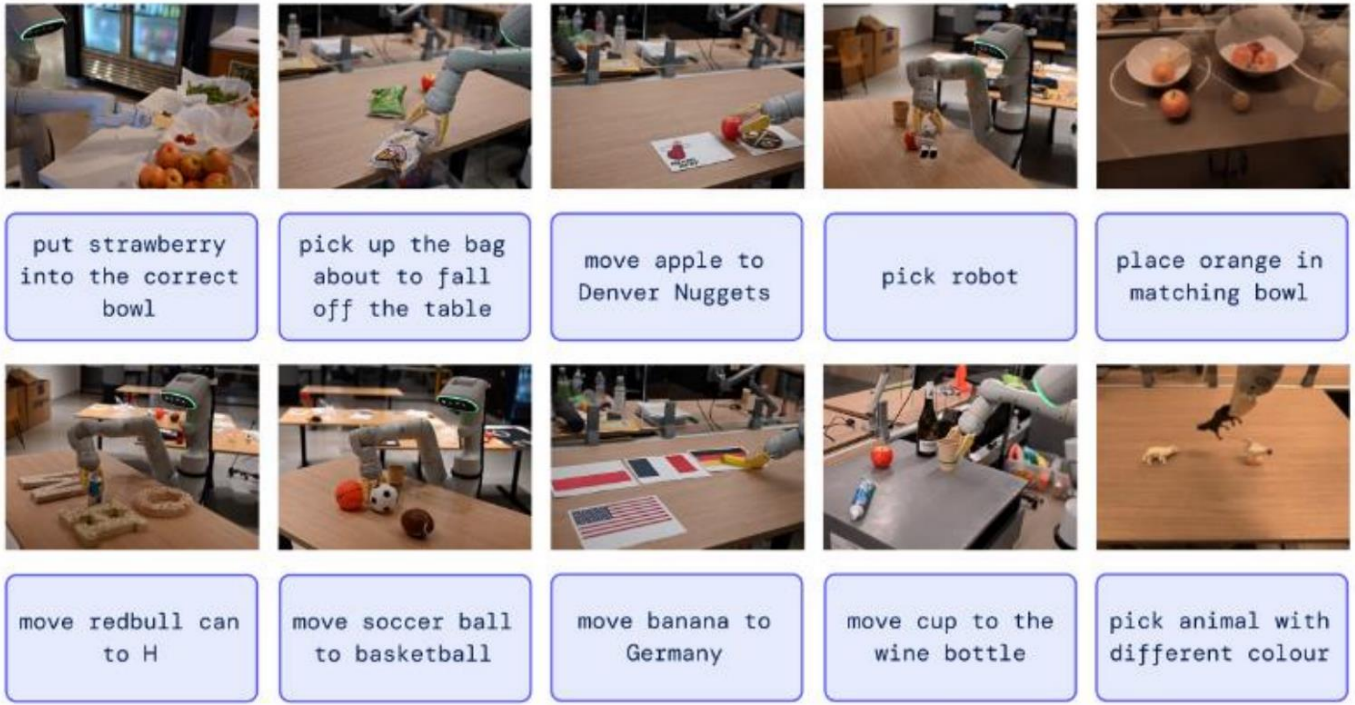
Overview



具身智能

RT-2

Demo



具身智能

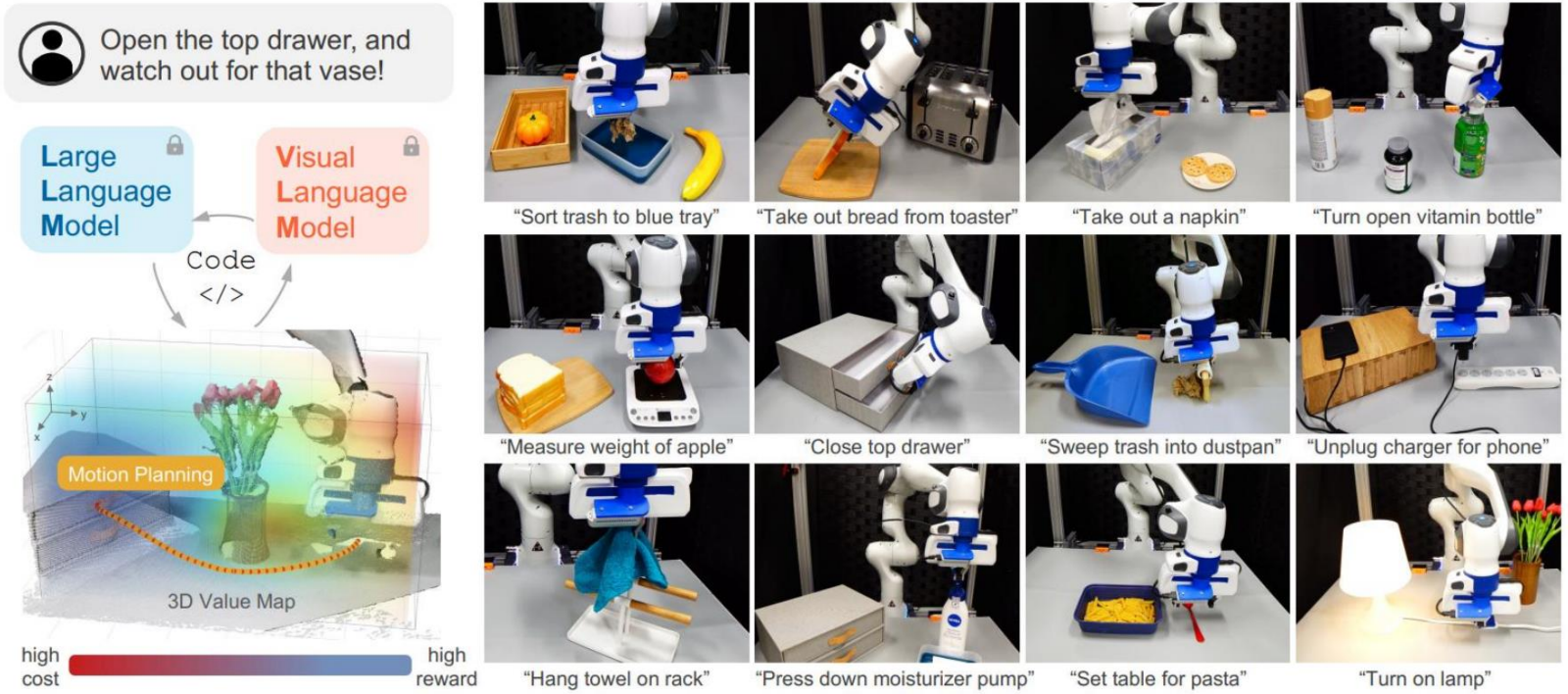


Figure 1: VoxPoser extracts language-conditioned **affordances** and **constraints** from LLMs and grounds them to the perceptual space using VLMs, using a code interface and without additional training to either component. The composed map is referred to as a 3D value map, which enables **zero-shot** synthesis of trajectories for large varieties of everyday manipulation tasks with an **open-set of instructions** and an **open-set of objects**.

This video has audio

VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, Li Fei-Fei

具身智能

斯坦福大学的科研团队近日开发了 Mobile ALOHA，可以执行打开厨房用具柜、洗锅、炸虾、做菜、打扫卫生、整理衣物、套被套等 50 多项家务。这款家用机器人成本仅 3.2 万美元。

Figure 1: ***Mobile ALOHA*** . We introduce a low-cost mobile manipulation system that is bimanual and supports whole-body teleoperation. The system costs $32k including onboard power and compute. *Left:* A user teleoperates to obtain food from the fridge. *Right: Mobile ALOHA* can perform complex long-horizon tasks with imitation learning.

具身智能

其算法Action Chunking with Transformers （ACT）采用了神经网络模型Transformers，因此**具备模仿学习能力。只需要15分钟的演示，机械臂就可以学会一个动作**——直接从真实演示中执行端到端模仿学习，并通过自定义远程操作界面收集。

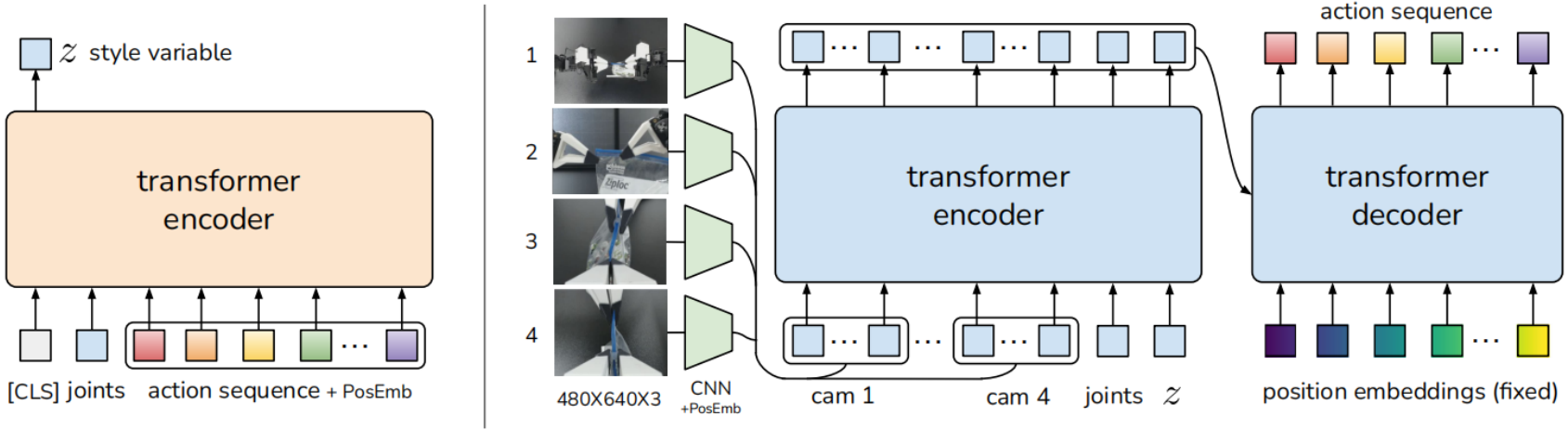


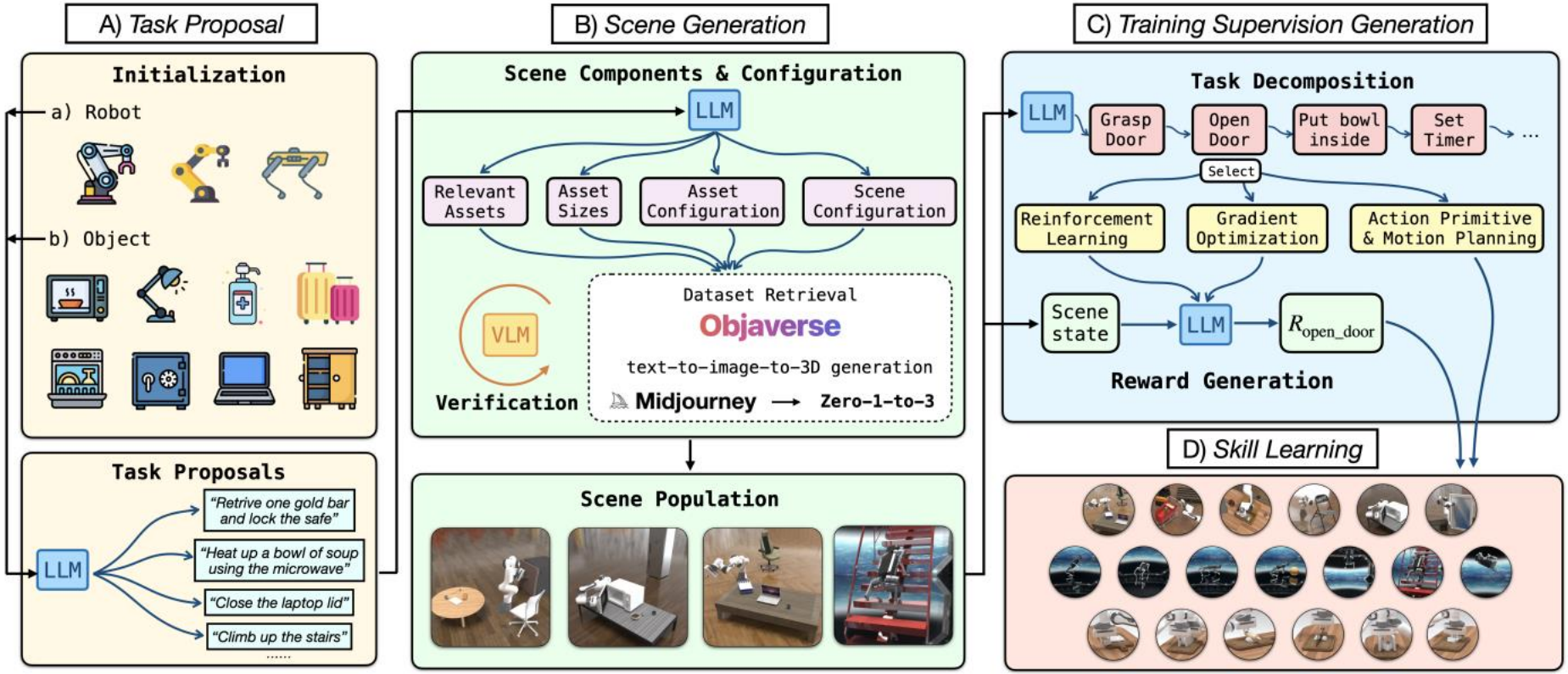
Fig. 4: *Architecture of Action Chunking with Transformers (ACT).* We train ACT as a Conditional VAE (CVAE), which has an encoder and a decoder. *Left:* The encoder of the CVAE compresses action sequence and joint observation into z, the style variable. The encoder is discarded at test time. *Right:* The decoder or policy of ACT synthesizes images from multiple viewpoints, joint positions, and z with a transformer encoder, and predicts a sequence of actions with a transformer decoder. z is simply set to the mean of the prior (i.e. zero) at test time.


Cook Shrimp
(autonomous)
6x speed
Stanford
University

具身智能

CMU、**清华**、MIT、UMass等机构提出RoboGen，是一个完全自动化的机器人学习系统，利用最新的LLMs模型来生成多样化的任务、场景和训练监督。通过自动生成的方式，它将大型模型中的知识转移到机器人身上。

1、任务提案：RoboGen自动生成一系列机器人可能感兴趣或需要学习的任务，从简单物理动作到复杂交互任务。
2、场景生成：确定任务后，RoboGen使用生成模型构建相应的模拟环境，例如模拟一个厨房来学习做饭。
3、训练监督生成：为每个任务生成训练监督信号，如奖励函数，指导机器人完成任务。这是自动生成的关键。
4、技能学习：机器人选择最优的学习方法，通过实践反复试错在模拟环境学习优化策略，直到掌握所提出技能。

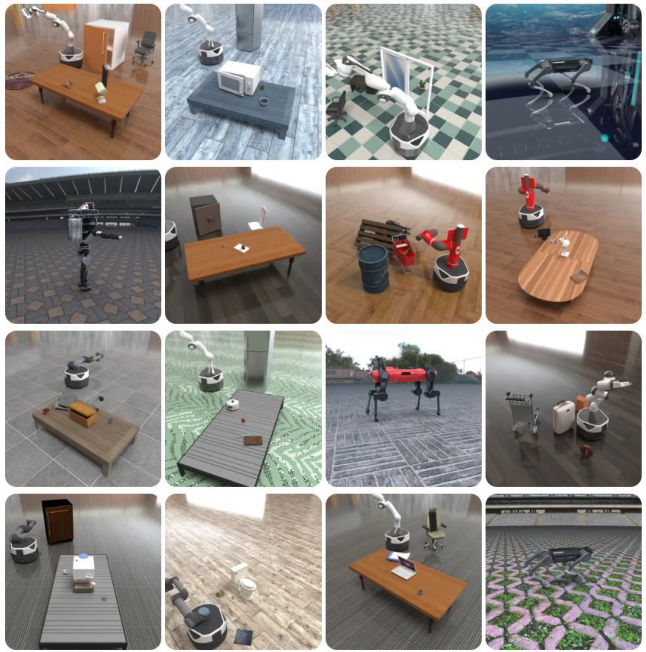


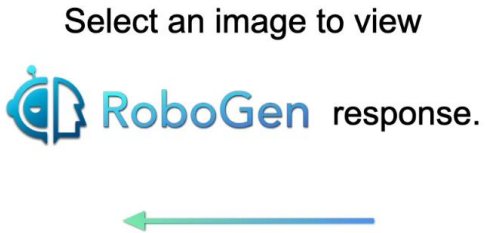
具身智能

RoboGen的特点在于全自动生成的流程，不断查询和执行，为机器人提供无尽的技能学习机会。这种方法具有以下优势：

1、自动化：减少了人工编写任务和环境的需求，提高了效率。
2、多样性：能够生成各种任务和环境，增加了机器人学习的广度和适应性。
3、可扩展性：通过自我引导的学习循环，机器人能够提出新任务、生成新环境，学习新技能。
4、模拟到现实的迁移：虽然学习在模拟环境中进行，但目标是使学到的技能适用于现实世界的机器人。

https://robogen-ai.github.io/





谢谢大家