
# 大规模 预训练 AI模型 发展,现状和未来

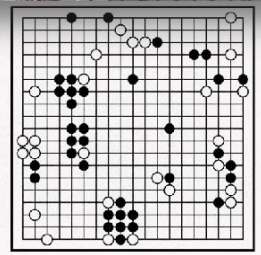
华南师范大学人工智能学院

范晨悠, 马宇函, 江金刚

# 1. AI发展里程碑 (2016年)

**围棋AI模型AlphaGo/Zero, 击败人类世界冠军并登上 N&S 杂志封面。**

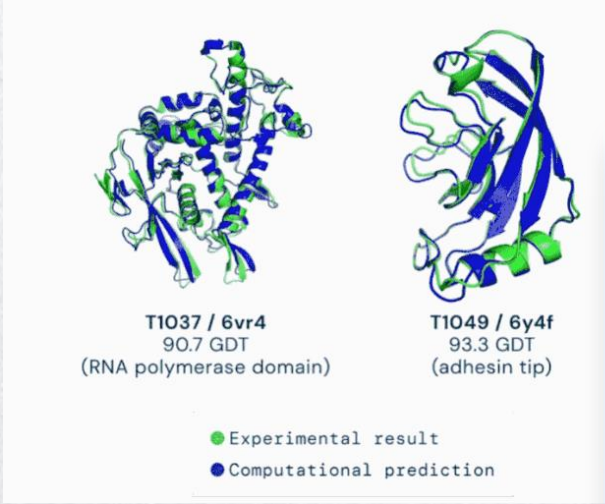


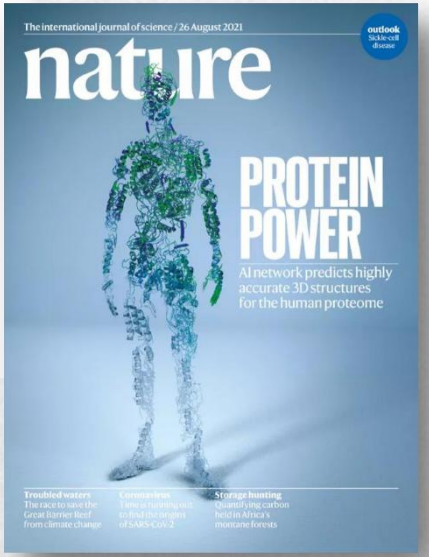


# AI发展里程碑 (2021年)

**蛋白质结构预测算法AlphaFold登上 Nature杂志封面。**

-- “**可准确预测98.5%**的人类蛋白结构”



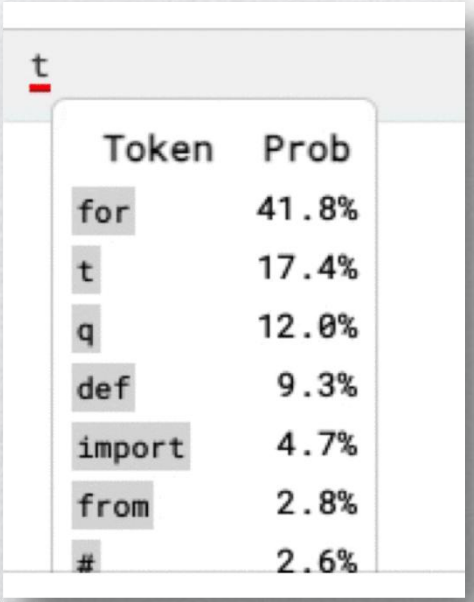




*Highly accurate protein structure prediction with AlphaFold, Nature, 2021*

# AI发展里程碑 (2022年)

**AI编程模型AlphaCode登上Science 封面。**

**该模型在全球顶级计算机编程评测系统Codeforces中击败了半数的人类程序员。**

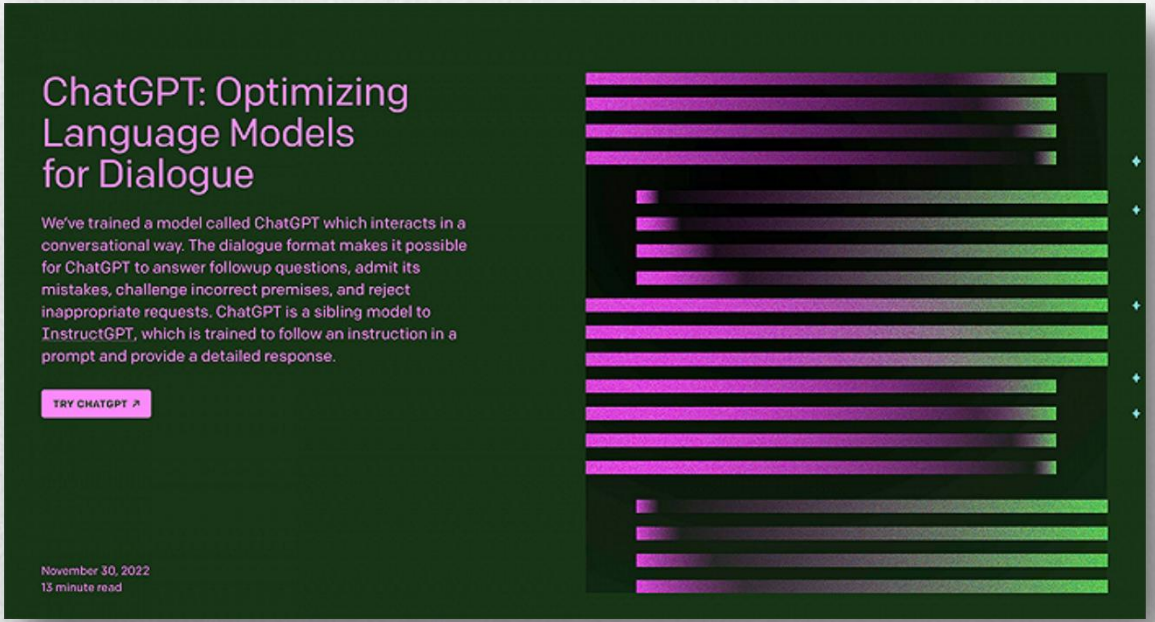




*AI learns to write computer code in 'stunning' advance, Science, 2022*

# AI发展里程碑 (2022年)

**人工智能对话模型ChatGPT发布并突破1亿用户。**



# 深度学习三巨头获图灵奖 (2018年)

“They led significant breakthroughs in AI technologies …”

- 约书亚·本吉奥(Yoshua Bengio)
  - 加拿大蒙特利尔大学
- 杰弗里·辛顿(Geoffrey Hinton)
  - 多伦多大学，Google Brain
- 杨乐昆(Yann LeCun)
  - 纽约大学, Meta首席科学家

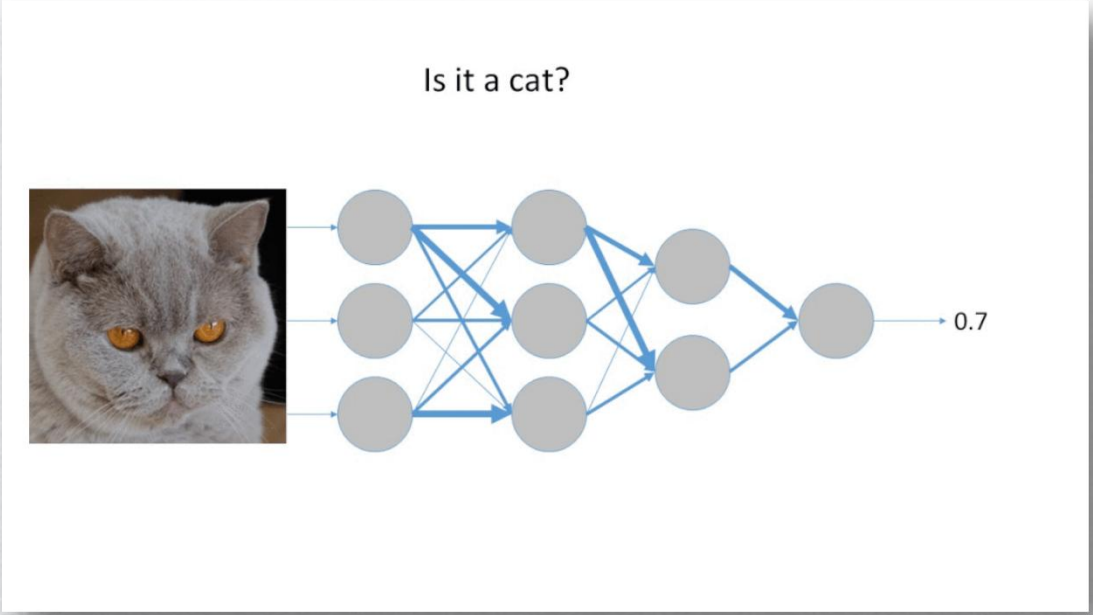
# 2.人工智能模型发展 (技术篇)

- 2012年，以AlexNet为代表的卷积神经网络(百万参数)
  - AI进入**深度学习**时代
- 2017年，以Transformer为代表的自然语言模型(>1亿参数)
  - AI进入**大模型**时代
- 2020~至今，以GPT-3为代表的预训练大模型(>1000亿参数)
  - 进入**超大模型**时代

# 现代深度神经网络(DNN) - 2012年

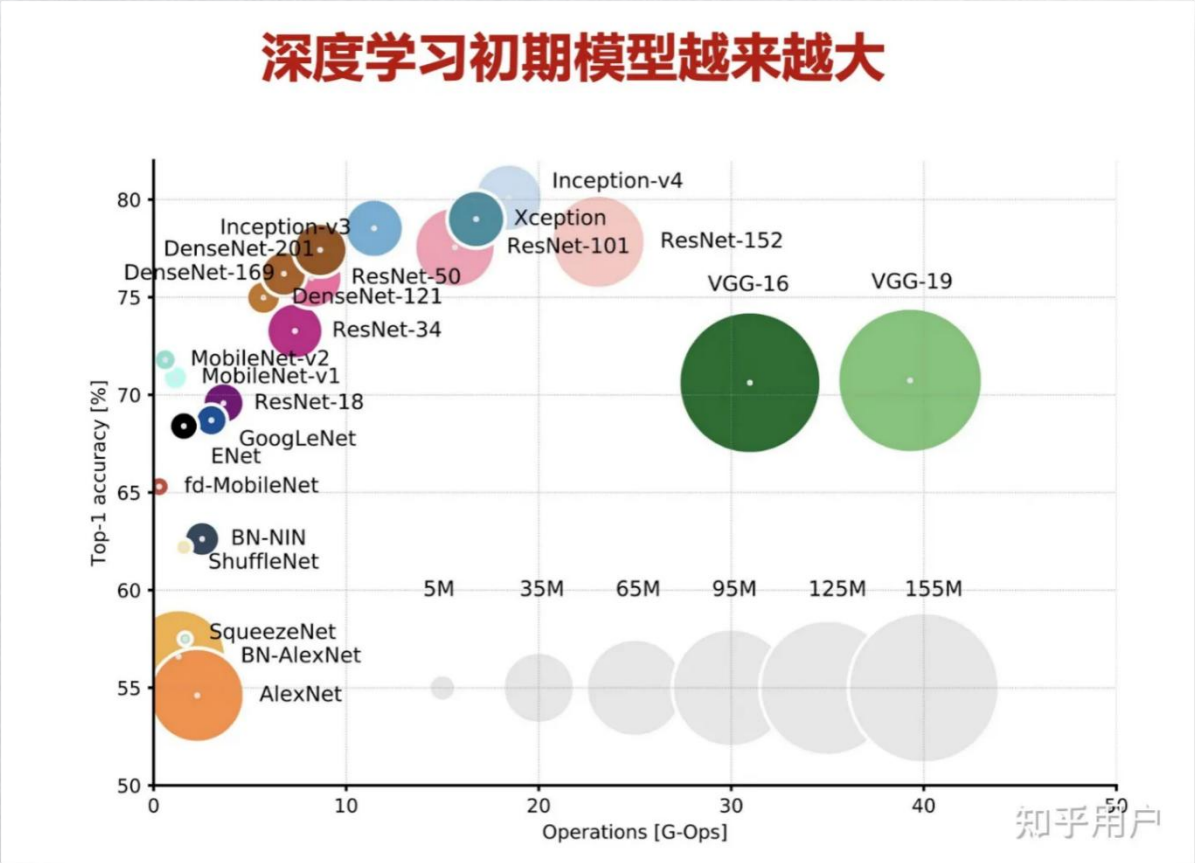
**DNN是一种用于模拟人类神经系统的计算模型。**

- 使用多层神经元组合来处理复杂的输入数据。
- 可以学习到高维输入特征的复杂关系，如图像视频、语音、自然语言等。
- 可扩展, 依赖算力和数据。




ImageNet Classification with Deep Convolutional Neural Networks, 2012.
作者: Alex Krizhevsky, **Ilya Sutskever (OpenAI)**, Geoffrey E. Hinton

# 2012-2017 DNN模型规模稳步增长

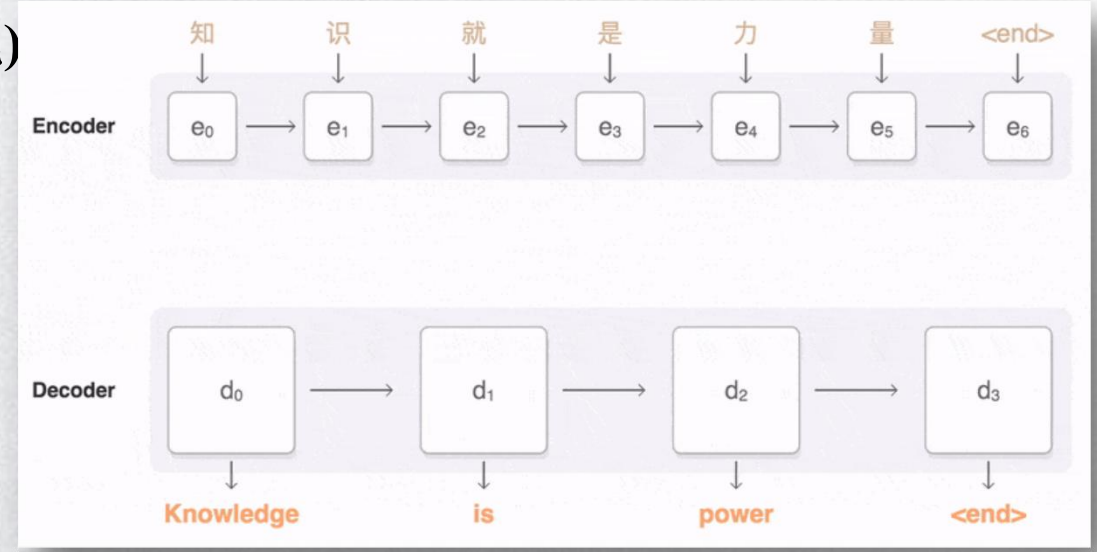



图片来源于知乎

# 语言模型 Transformer - 2017年

**新一代序列数据模型** (参数过亿)

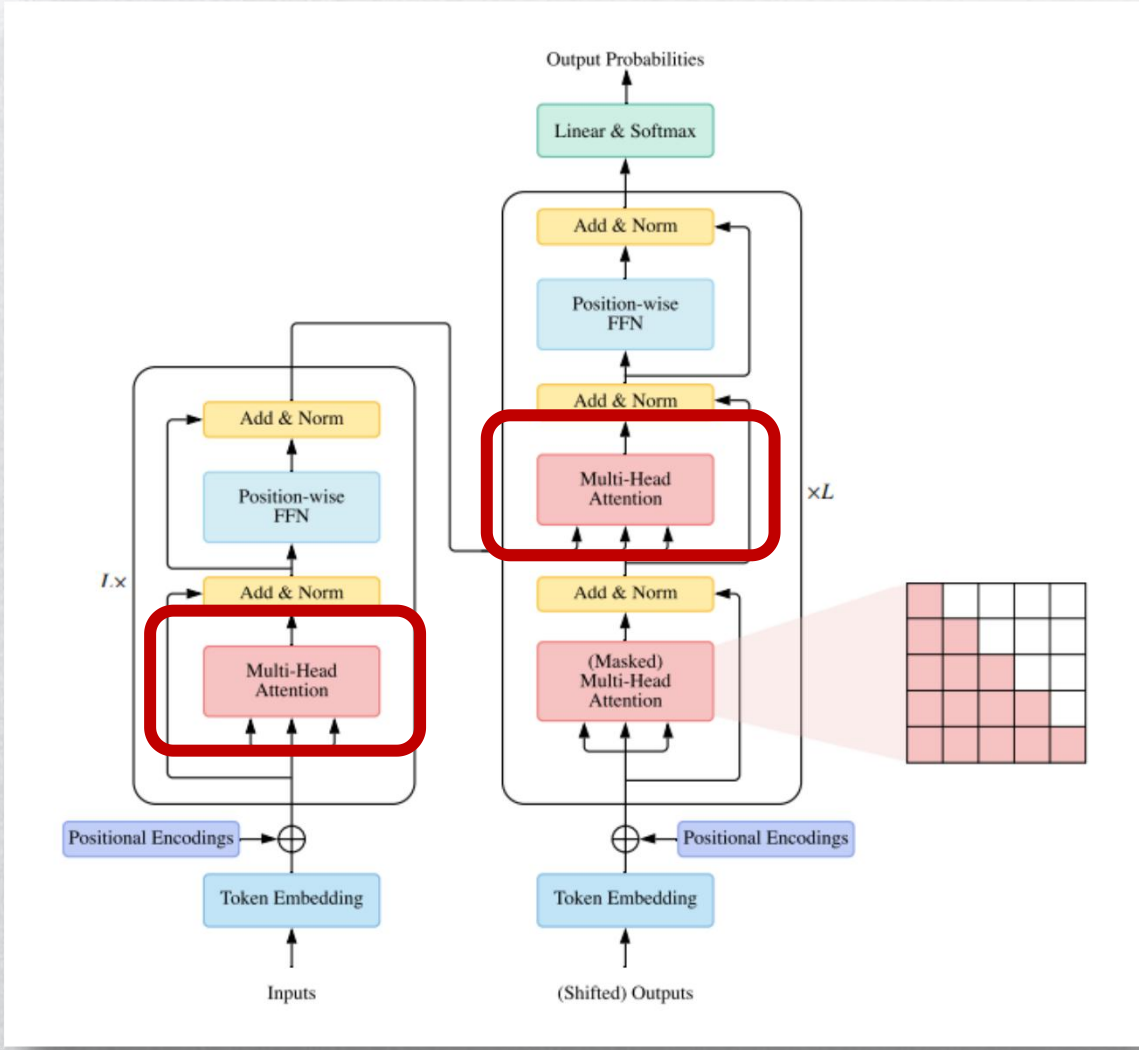
- 基于编码-解码模型
- 计算所有输入单词之间的相关程度(编码)
- 计算所有输入-输出单词之间的相关程度(解码)
- 模型计算量大大增加



# 基本单元: 注意力模块

**使用基于键值对的注意力机制，以实现：**

- 自注意力(self-attention) 对输入进行编码
- 交叉注意力(cross-attention) 对输出进行解码
- 更好理解上下文
- 可以输出更长的文字, 理解历史对话

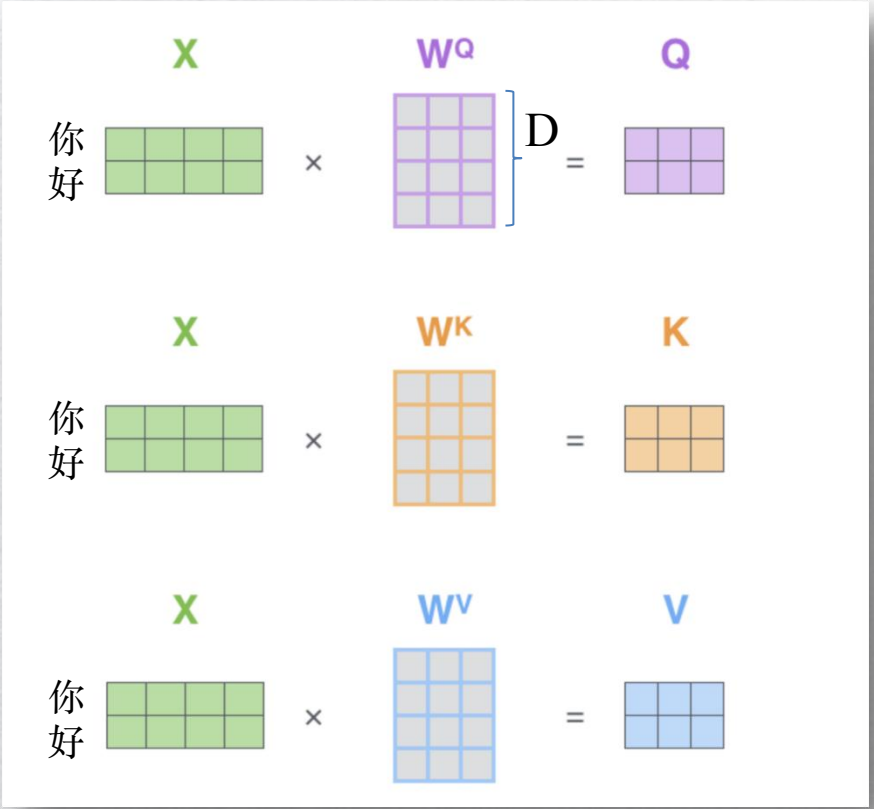


# 基于键值对的注意力机制(主要的计算消耗单元)

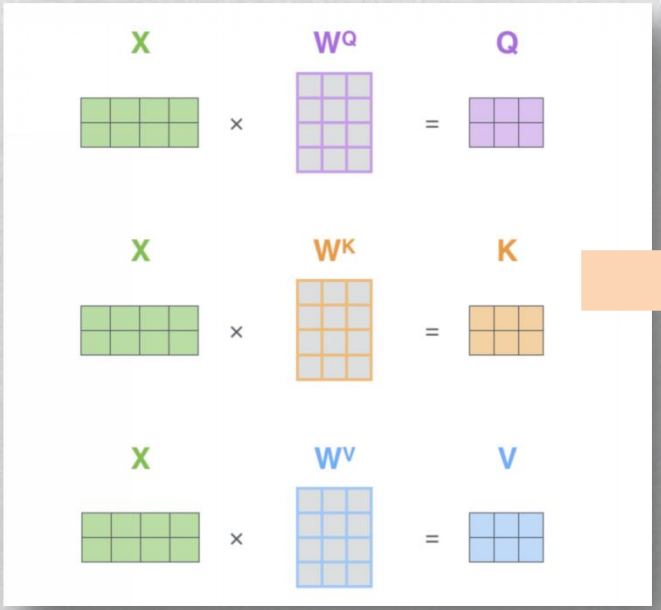
对一句话每一个单词

- 计算Query (查询)
- 计算Key (键)
- 计算Value (值)

参数量为 $O(D^2)$



# 基于键值对的注意力机制(主要的计算消耗单元)



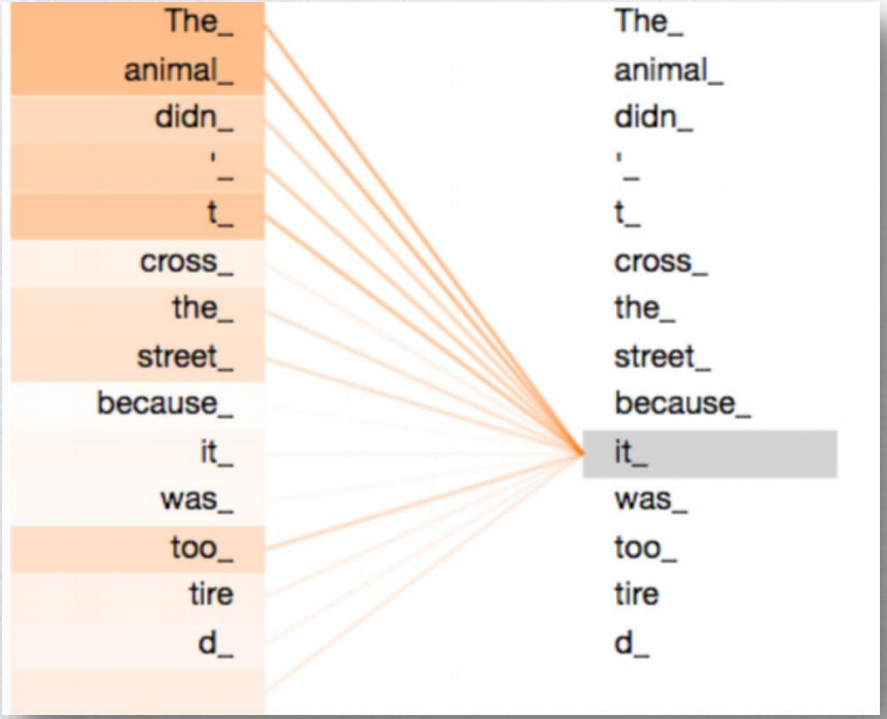

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

具有上下文信息的*你好*

计算量为 $O(N*D^2)$, N为单词个数

# 注意力机制的目的 – 更好的编码和解码!

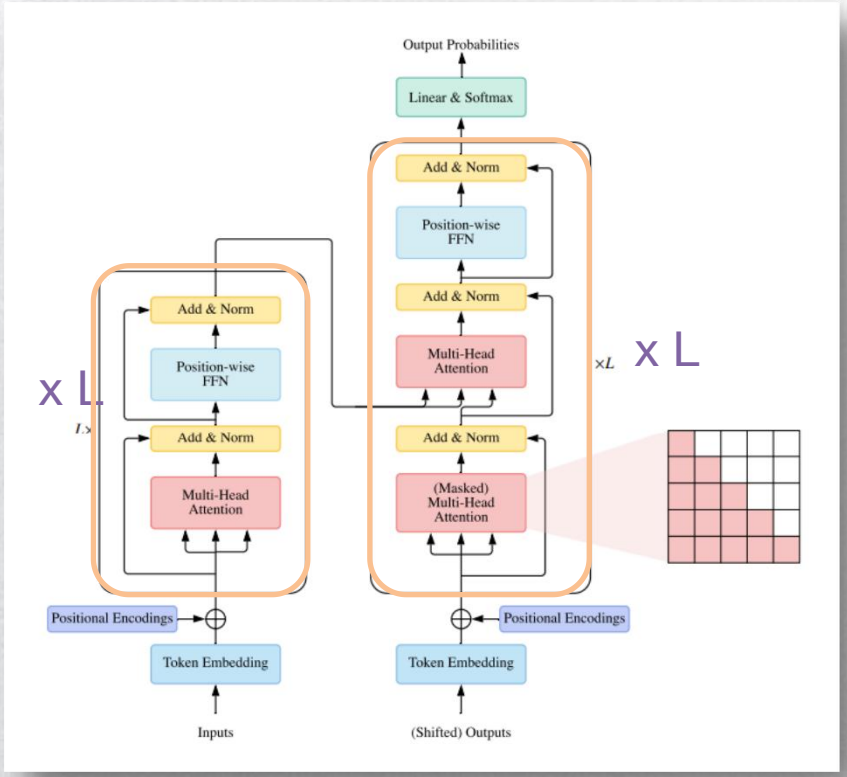
- 通过注意力机制编码
- 每一个单词均能找到合适的上下文信息
- 例如, it 关联到
  - animal,
  - cross-the-street,
  - too tired




The animal didn’t cross the street because it was too tired.

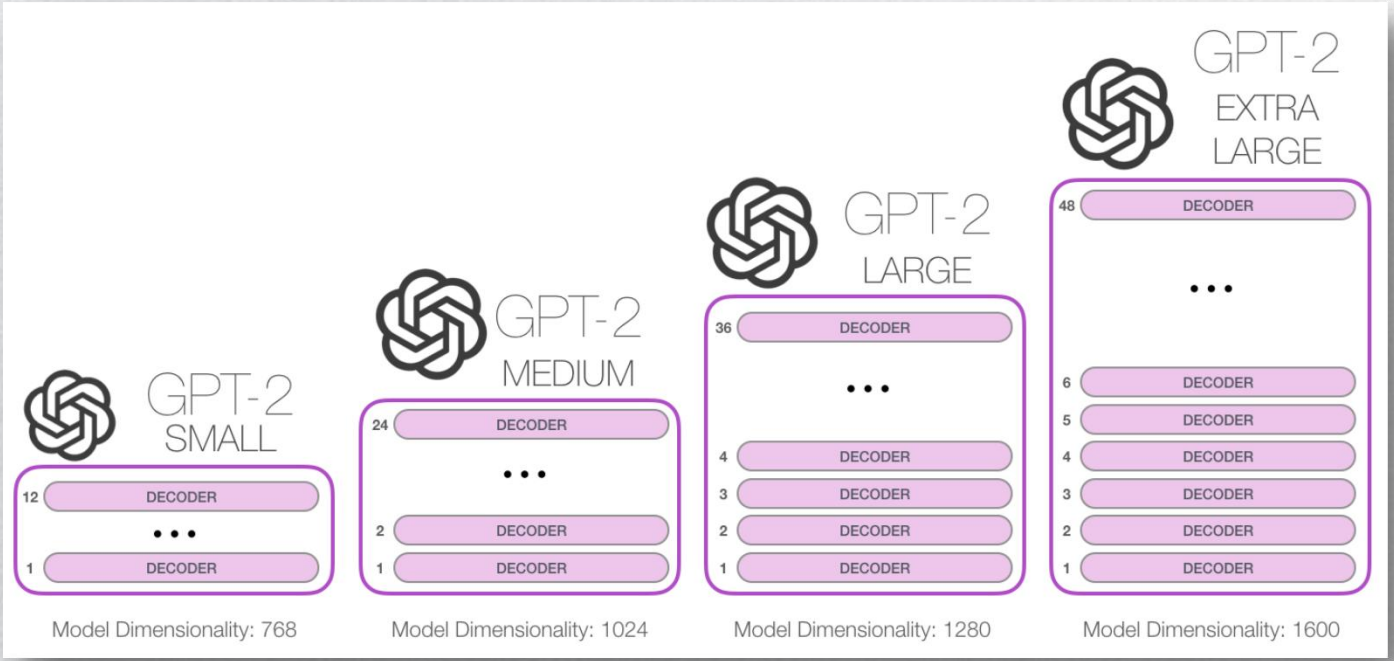
# 模型参数膨胀的秘密

- 大量堆叠注意力模块层数 L
  e.g., L= 96 (ChatGPT)
- 隐藏层维度 D 增长为 2048
- 注意力特征参数 $O(D^2 \cdot L)$
- 其他参数, 如词嵌入, 输出层
  - 达到上亿级别




Vaswani et al. Attention Is All You Need, 2017

# 层数(纵向)叠加




# 维度(横向)增加

# 组件改进

- Positional Embedding 由 sine/cosine 函数改为 Rotary Positional Encoding (RoPE, Su et al. (2021))

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

# Positional Embedding 改进

- 由正弦函数(Sinusoidal)编码改为 旋转位置编码 (RoPE)

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

*想要成为相对位置编码的绝对位置编码*

- RoPE通过变换 q k 使得嵌入相对位置信息

$$\underbrace{\begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix}}_{\boldsymbol{W}_m} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix}$$

$$(\boldsymbol{W}_m \boldsymbol{q})^\top (\boldsymbol{W}_n \boldsymbol{k}) = \boldsymbol{q}^\top \boldsymbol{W}_m^\top \boldsymbol{W}_n \boldsymbol{k} = \boldsymbol{q}^\top \boldsymbol{W}_{n-m} \boldsymbol{k}$$

W是正交矩阵，不会改变向量的模长，稳定

# Linear + ReLU 改成 GLU +GeLU

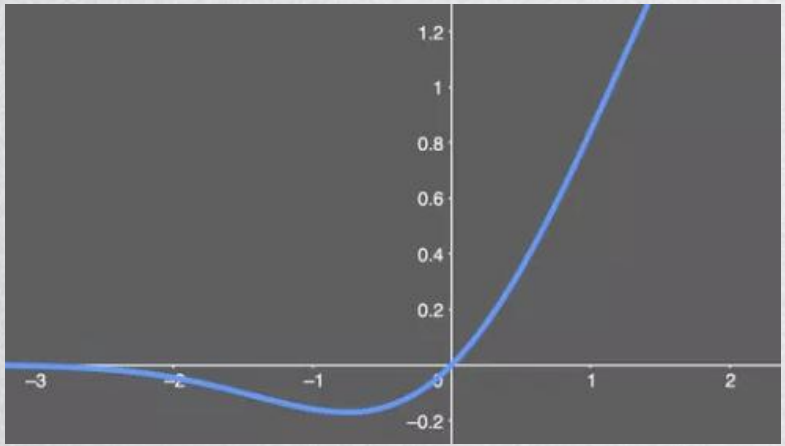
- 门控线性单元 Gated Linear Unit (GLU)
- 引入门控机制，取代仅激活函数，加快收敛

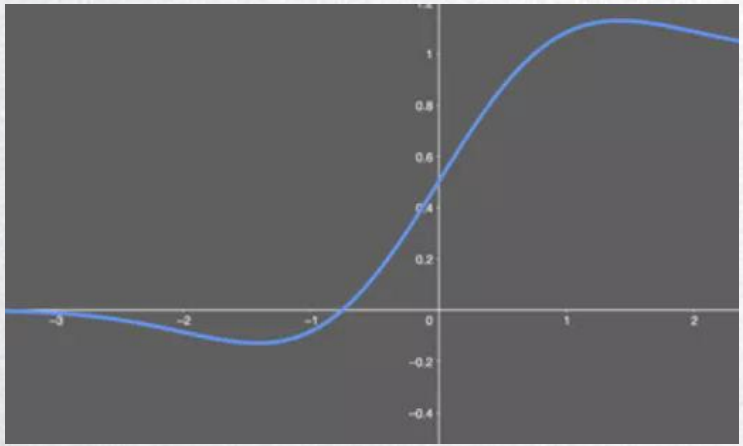
Applies the gated linear unit function $GLU(a, b) = a \otimes \sigma(b)$

## GELU

高斯误差线性单元激活函数在最近的 Transformer 模型（谷歌的 BERT 和 OpenAI 的 GPT-2）中得到了应用。GELU 的论文来自 2016 年，但直到最近才引起关注。这种激活函数的形式为：

$$\mathrm{GELU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{2/\pi}(x + 0.044715x^3)\right)\right)$$



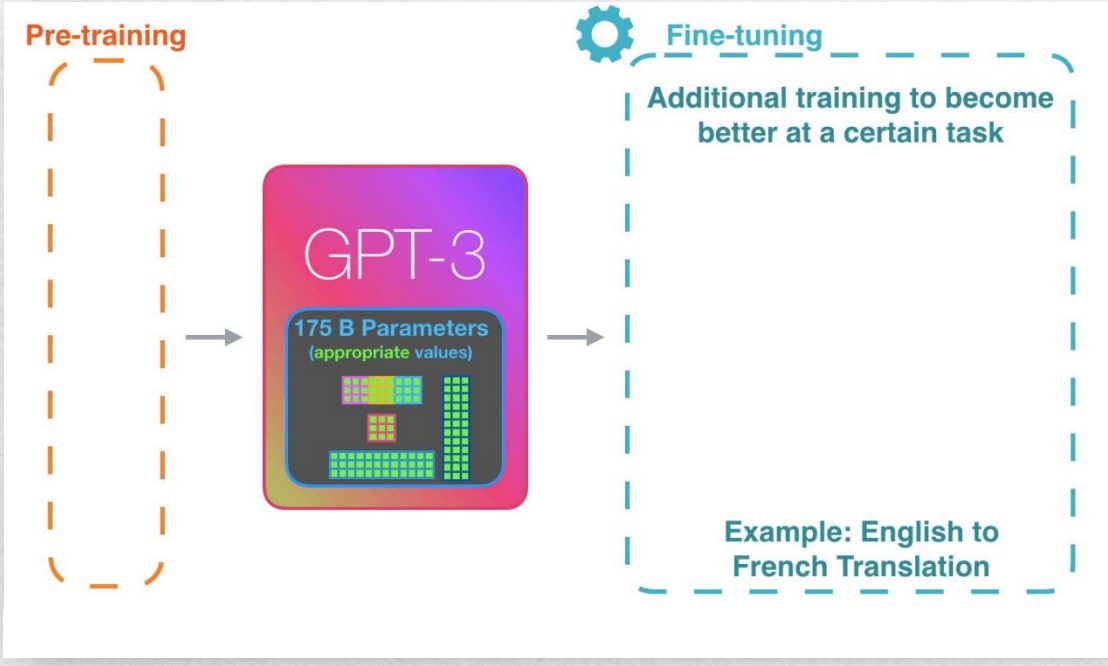



# 前馈神经网络FFN (改进后)

$$\mathrm{FFN}_{\mathrm{GeGLU}}\left(\boldsymbol{x}; \boldsymbol{W}_1, \boldsymbol{V}, \boldsymbol{W}_2\right) = \left(\mathrm{GeLU}(\boldsymbol{x}\boldsymbol{W}_1) \otimes \boldsymbol{x}\boldsymbol{V}\right)\boldsymbol{W}_2$$

# 大语言模型 (Large Language Model)

- 参数个数1千亿-2万亿
- 采用预训练加任务微调
- 超级数据中心进行训练

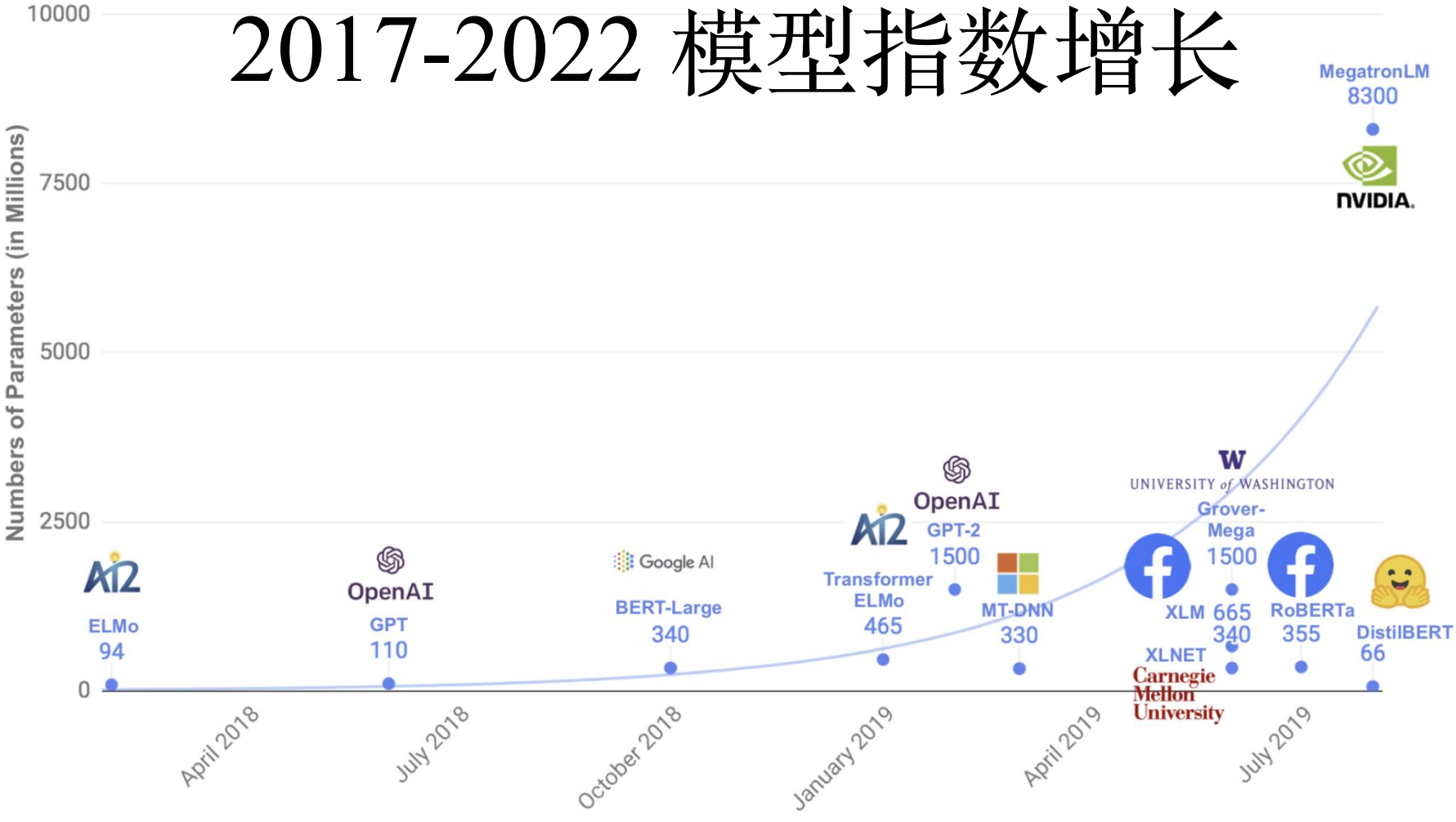


```
ChatGLMModel(
  (word_embeddings): Embedding(130528, 4096)
  (layers): ModuleList(
    (0-27): 28 x GLMBlock(
      (input_layernorm): LayerNorm((4096,), eps=1e-05, elementwise_affine=True)
      (attention): SelfAttention(
        (rotary_emb): RotaryEmbedding()
        (query_key_value): Linear(in_features=4096, out_features=12288, bias=True)
        (dense): Linear(in_features=4096, out_features=4096, bias=True)
      )
      (post_attention_layernorm): LayerNorm((4096,), eps=1e-05, elementwise_affine=True)
      (mlp): GLU(
        (dense_h_to_4h): Linear(in_features=4096, out_features=16384, bias=True)
        (dense_4h_to_h): Linear(in_features=16384, out_features=4096, bias=True)
      )
    )
  )
  (final_layernorm): LayerNorm((4096,), eps=1e-05, elementwise_affine=True)
)
```

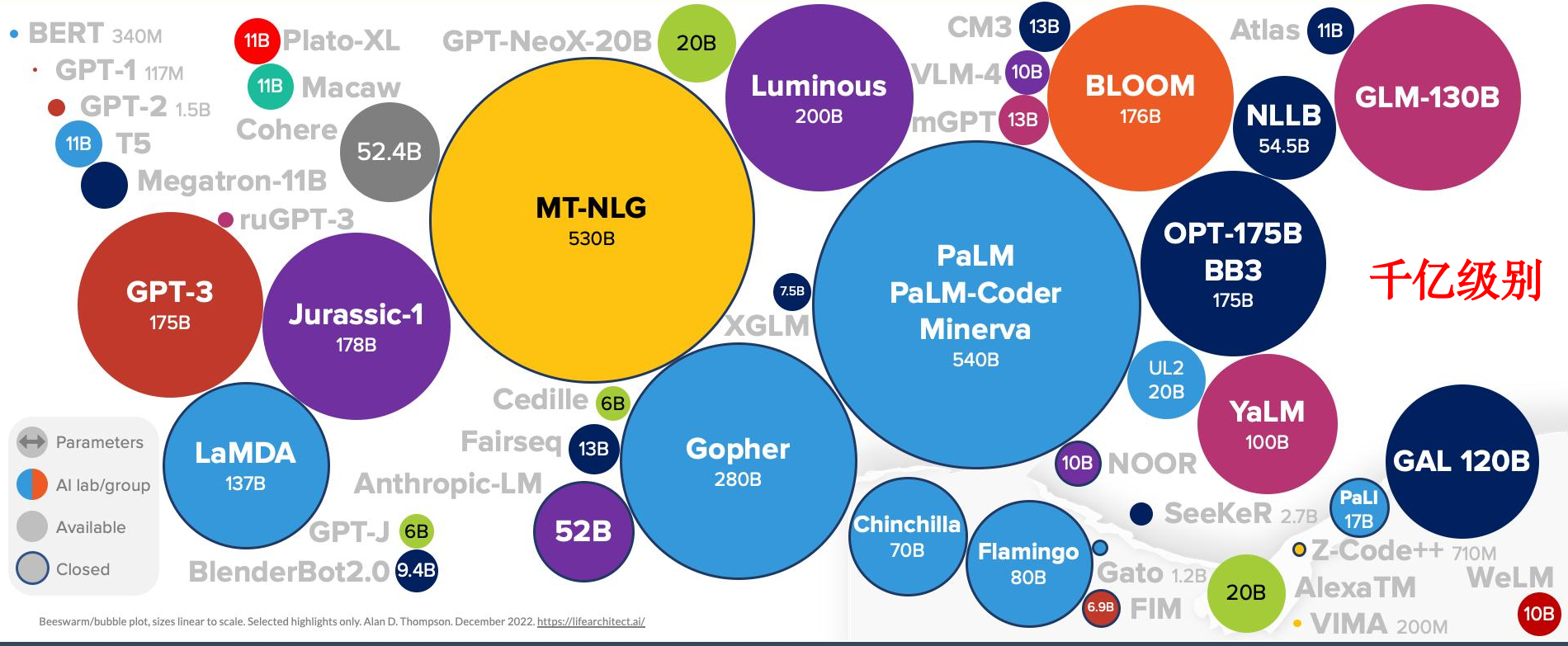
28 layer- 60亿参数模型

70 layer- 1300 亿参数模型

# 2017-2022 模型指数增长



# LANGUAGE MODEL SIZES TO DEC/2022




Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. Alan D. Thompson. December 2022. https://lifearchitect.ai/

# 3. 大模型算力成本(载荷)

**2021年 (公布数据)**:

1. GPT-3 在三月份有1百万注册用户
2. GPT-3 输出 3百万 单词/分钟 (WPM)

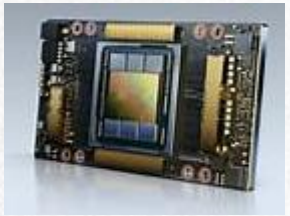
**2022/23 估算**:

1. ChatGPT 有 1亿月活用户
2. 估算ChatGPT 输出量为3亿单词/分钟
3. 5000台服务器可满足全世界的访问要求

# 大模型算力成本(软硬件)

## 模型训练

- 硬件: 384块A100, 单张卡显存80GB, 满足1750亿参数批量训练
- 成本: 960万美元 (48台DGX-A100服务器, 20万美元/台)
- 电费: 单次训练300万美元, 每天5万 × 2个月



## 部署推理

- 硬件：8张A100单次运算，每秒钟能产生大约15-20个单词
- 部署：3亿单词/分钟=> 5000台服务器满足访问=> ~ 10亿美元
- API 收费(最新公开): 100万单词 / $2.7

# Attention 加速

**1. self-attention 公式**

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

在传统的Attention中，Q、K、V作为输入，大小为N×d，如下所示，在计算中需要存储中间值S和P到HBM中，这会极大占用高带宽显存HBM。

$$\mathbf{S} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N\times N}, \quad \mathbf{P} = \mathrm{softmax}(\mathbf{S}) \in \mathbb{R}^{N\times N}, \quad \mathbf{O} = \mathbf{P}\mathbf{V} \in \mathbb{R}^{N\times d},$$

传统的Attention需要将Softmax（$QK^T$）的结果S存入HBM中。

# Flash Attention

- 通过矩阵分块相乘减少显存读取
- 对softmax 做增量计算，无需全局信息
- 只需维护一个全局最大 m 和 求和项

$$Softmax(x) = P(x) = \frac{e^{x_i - m}}{\sum_{i=1}^{N} e^{x_i - m}}$$

# Multi Group Attention

- 减少多头注意力，提高吞吐量

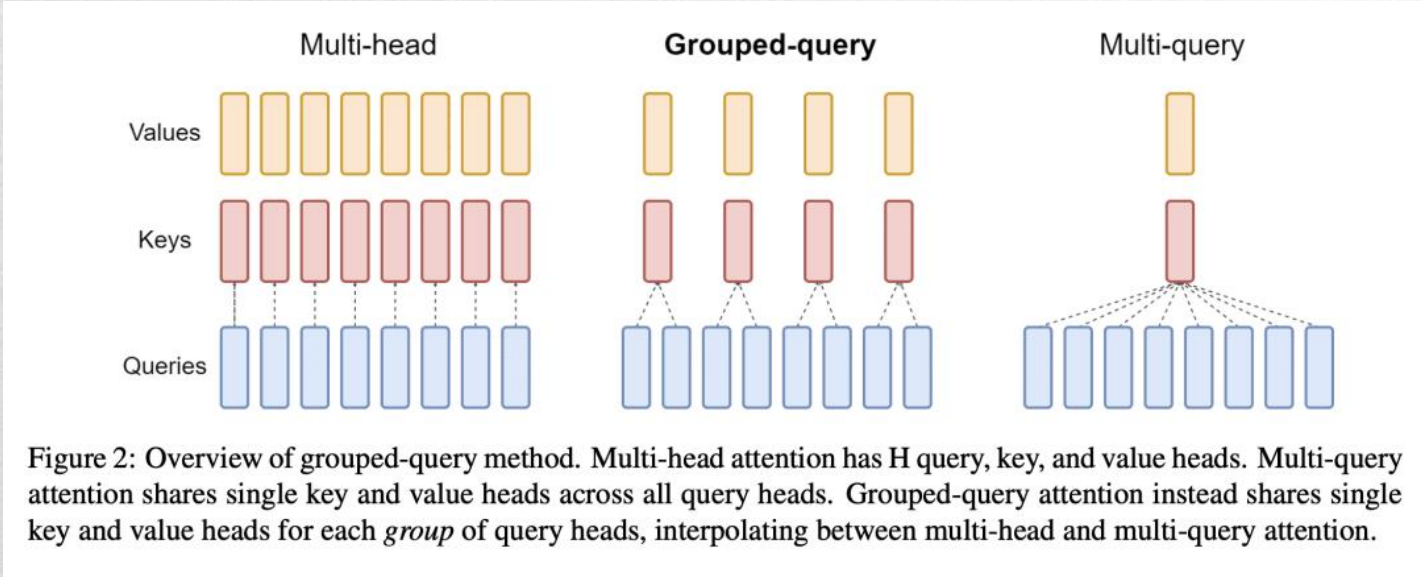
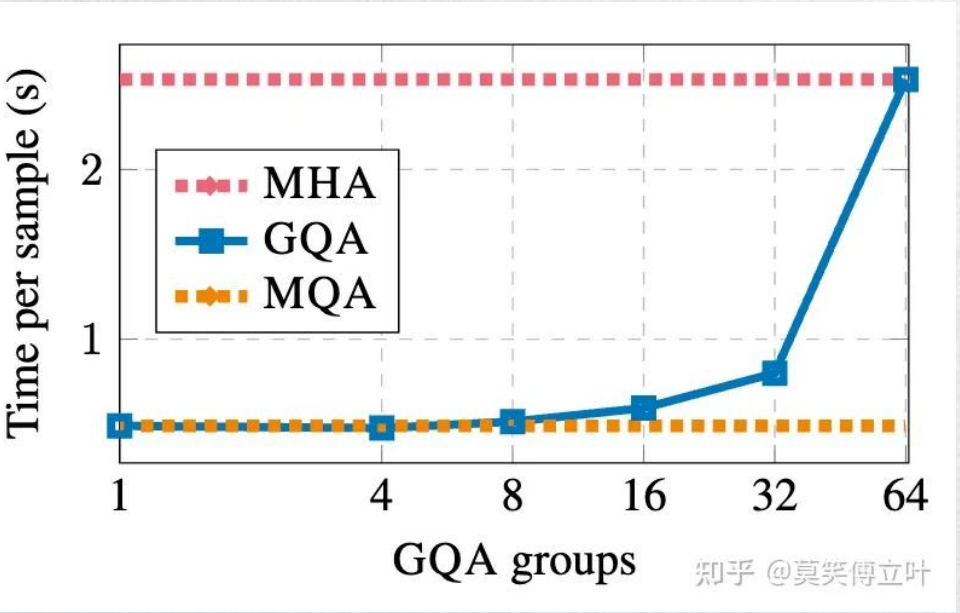
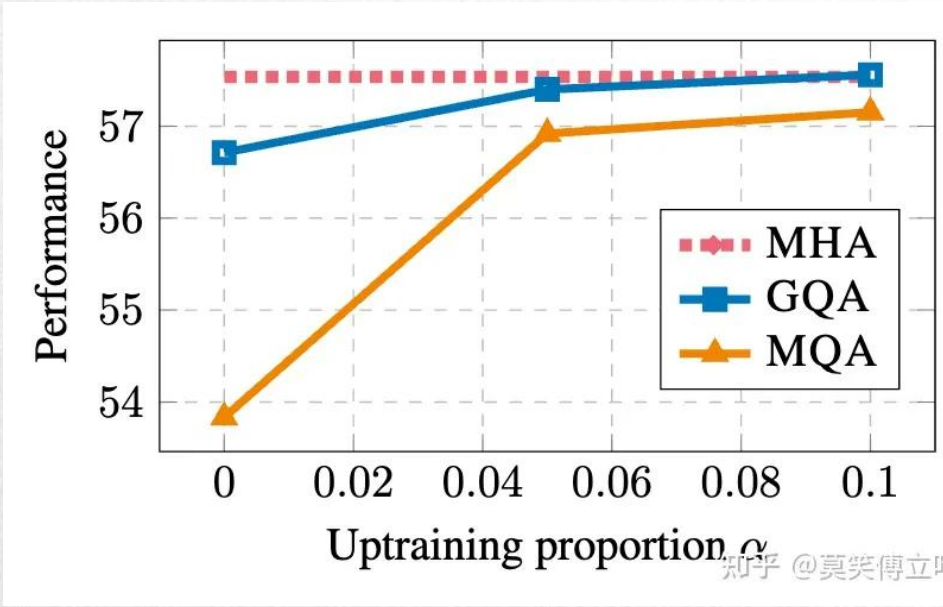


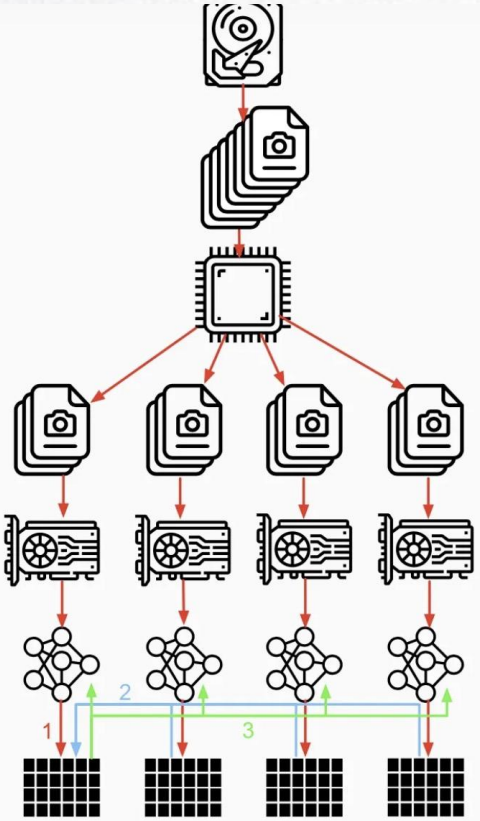

Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.


Time per sample (s)
2
1
MHA
GQA
MQA
1
4
8
16
32
64
GQA groups


Performance
57
56
55
54
MHA
GQA
MQA
0
0.02
0.04
0.06
0.08
0.1
Uptraining proportion $\alpha$

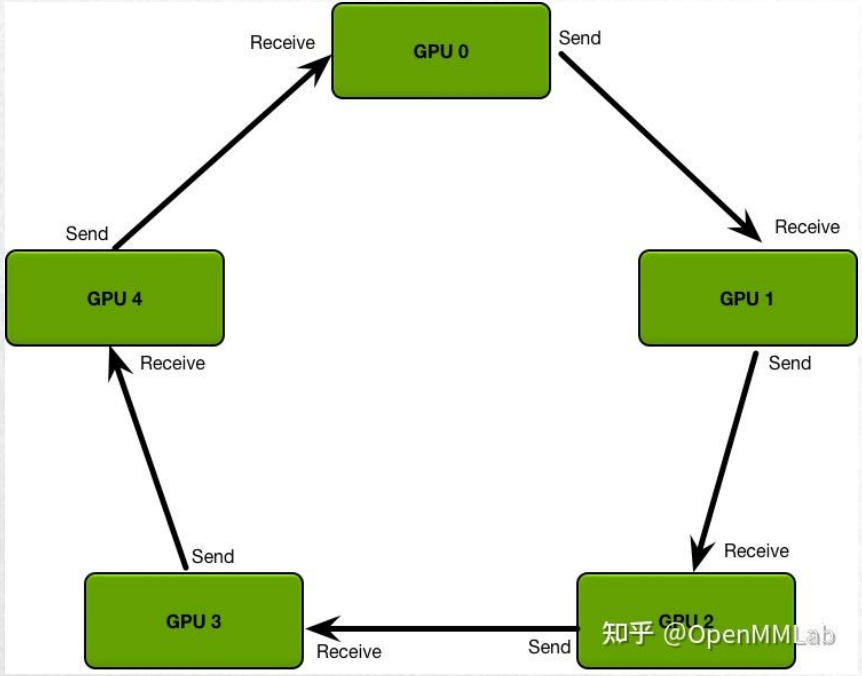
# 优化策略：DP - 数据并行

- DataParallel
- 单进程多线程
- 模型权重在GPU:0上计算，再分发到其他GPU
- 显存和速度瓶颈



# 优化策略：DDP

- Distributed DP
- 多进程控制多 GPU 一起训练模型。
- 每个进程保存一份完整模型，通过通信梯度实现同步。



# 优化策略

- ZeRO（Zero Redundancy Optimizer）大幅减少内存占用。每一张GPU仅保存部分的模型参数、计算部分的梯度，维护部分的优化器参数。

- 混合精度训练：训练过程中同时使用FP16（半精度浮点数）和FP32（单精度浮点数）两种精度的技术，大大减少内存占用。

# 优化策略 - Low Rank Adaptation (LoRA)

预训练的矩阵为 $W_0 \in \mathbb{R}^{d\times k}$，它的更新可表示为：

$$W_0 + \Delta W = W_0 + BA, B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k}$$

其中秩$r << min(d, k)$。

- 70% Params. Pre-training
- 30% Params. Fine-tuning
- 更容易在少样本上微调！

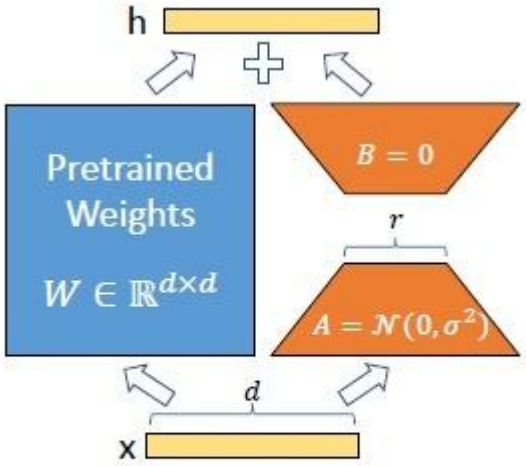


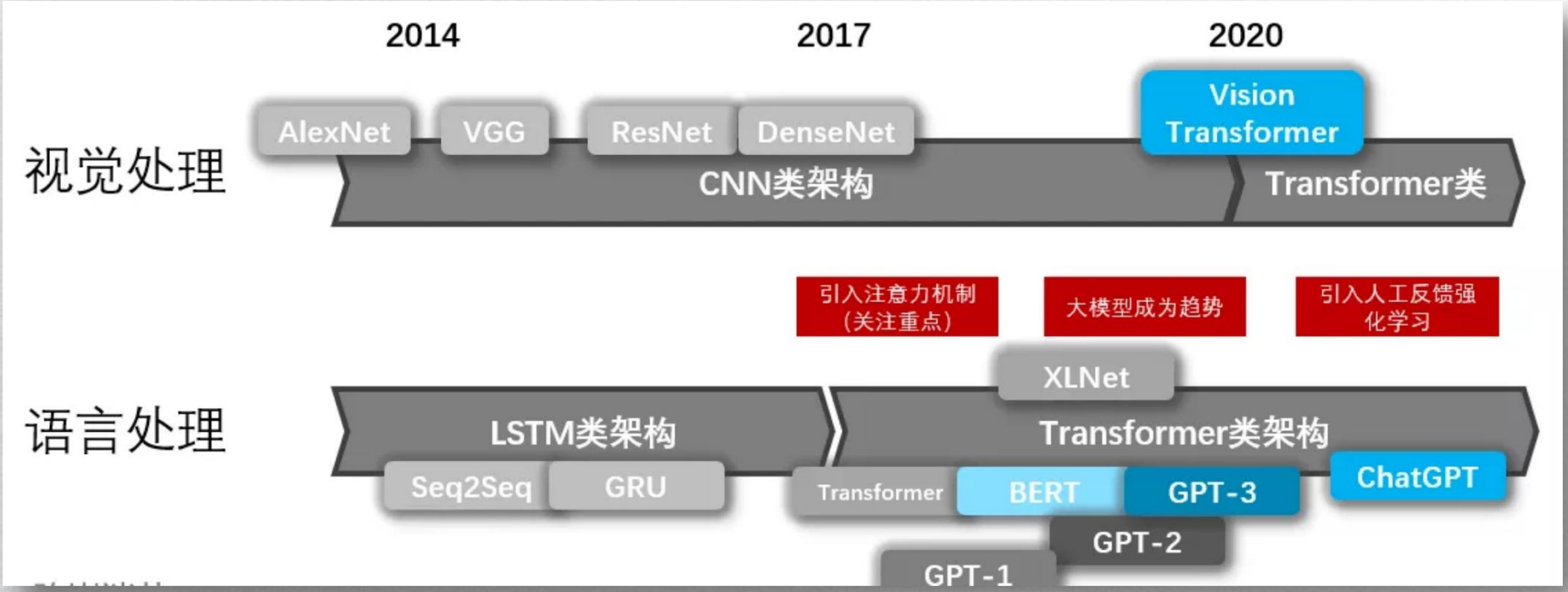

Figure 1: Our reparametrization. We only train $A$ and $B$.

# 计算框架

- Lightning Fabric  - 轻量灵活,单机多卡推荐
- Accelerate (Hugginface)

- ColossalAI (Meta)
- DeepSpeed (微软) - 超大规模，多机多卡推荐

# 4. 视觉及多模态大模型

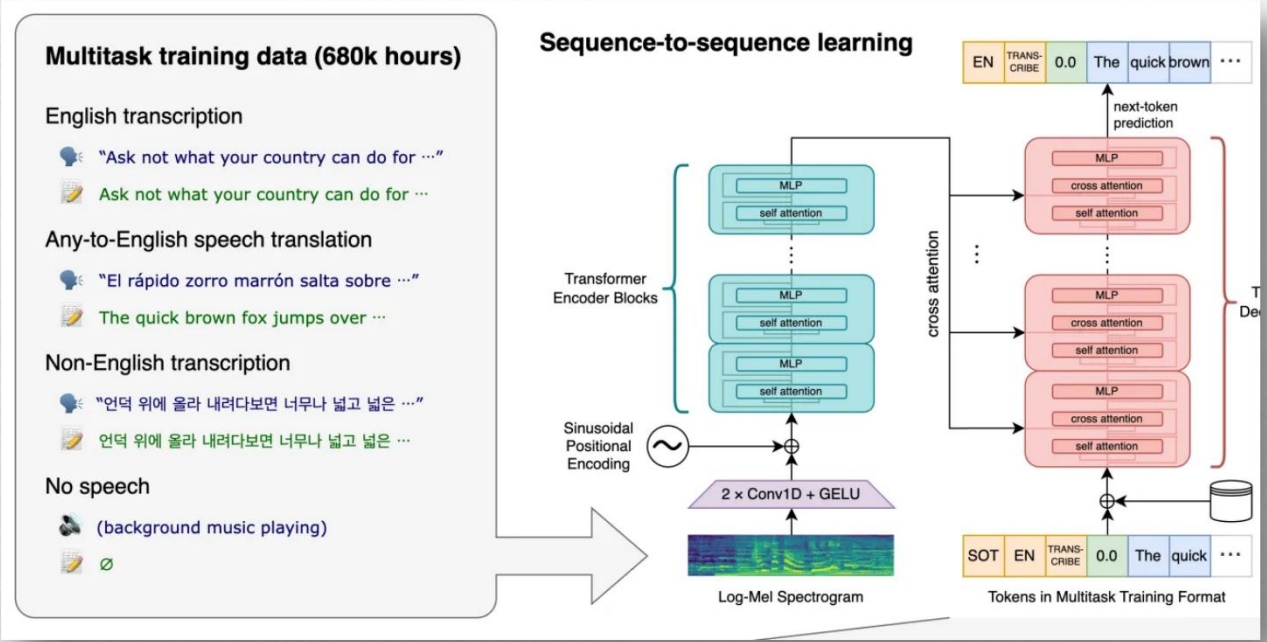

2014
2017
2020
视觉处理
AlexNet
VGG
ResNet
DenseNet
Vision Transformer
CNN类架构
Transformer类
引入注意力机制（关注重点）
大模型成为趋势
引入人工反馈强化学习
语言处理
XLNet
LSTM类架构
Transformer类架构
Seq2Seq
GRU
Transformer
BERT
GPT-3
ChatGPT
GPT-2
GPT-1

# 多模态 预训练大模型 (PLM)

- 图像识别
- 文本-图像生成
- 语音识别和翻译
- 情感分析

# Whisper 语音识别模型

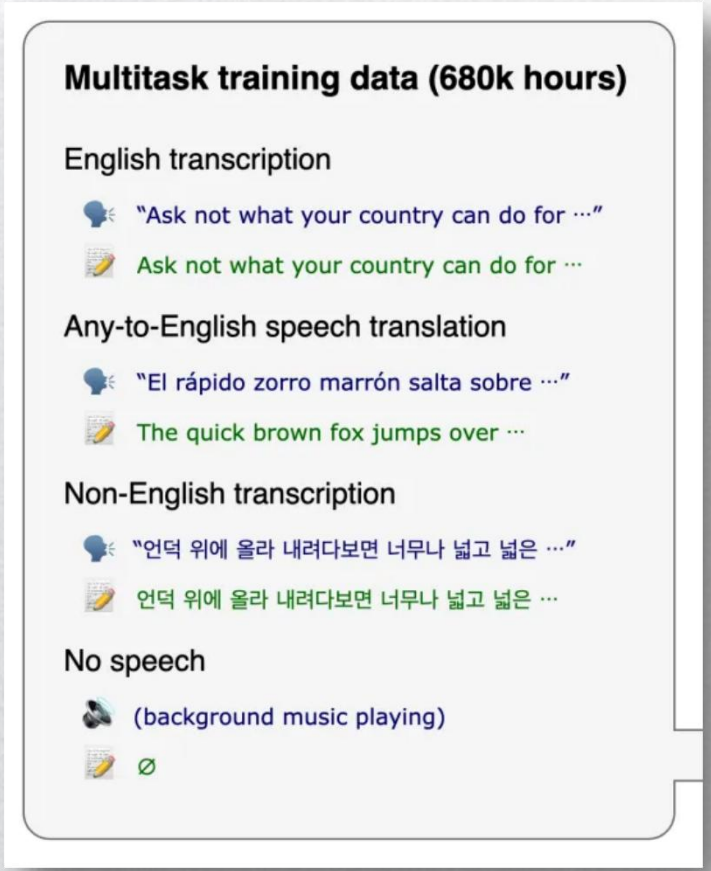
- 15 亿参数
- 96种语言
- 68万小时音频/字幕数据
- 接近人类水平



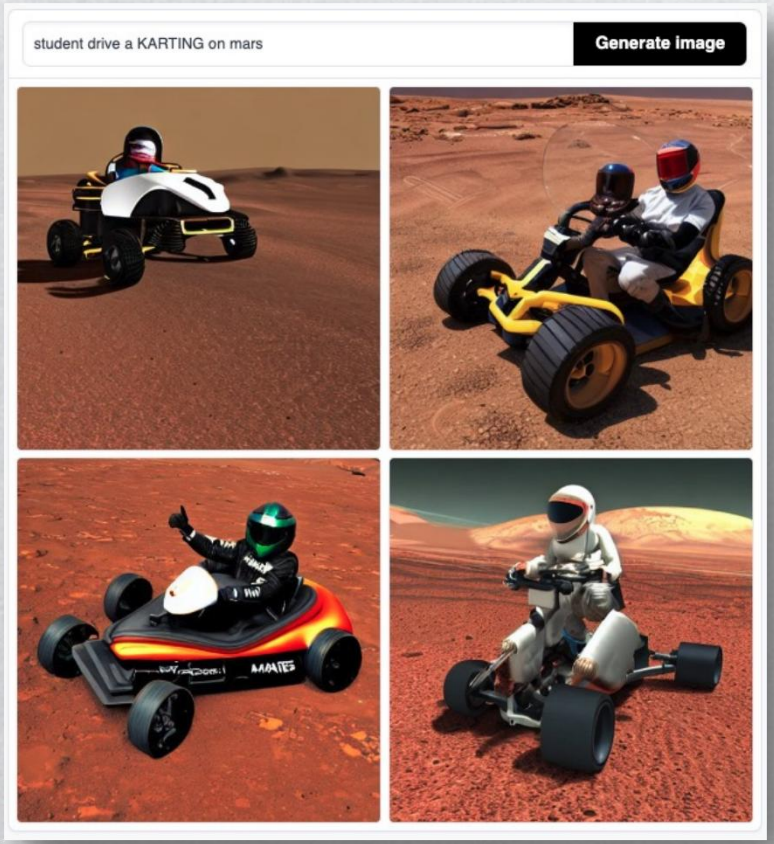

*Robust Speech Recognition via Large-Scale Weak Supervision,* OpenAI, 2022

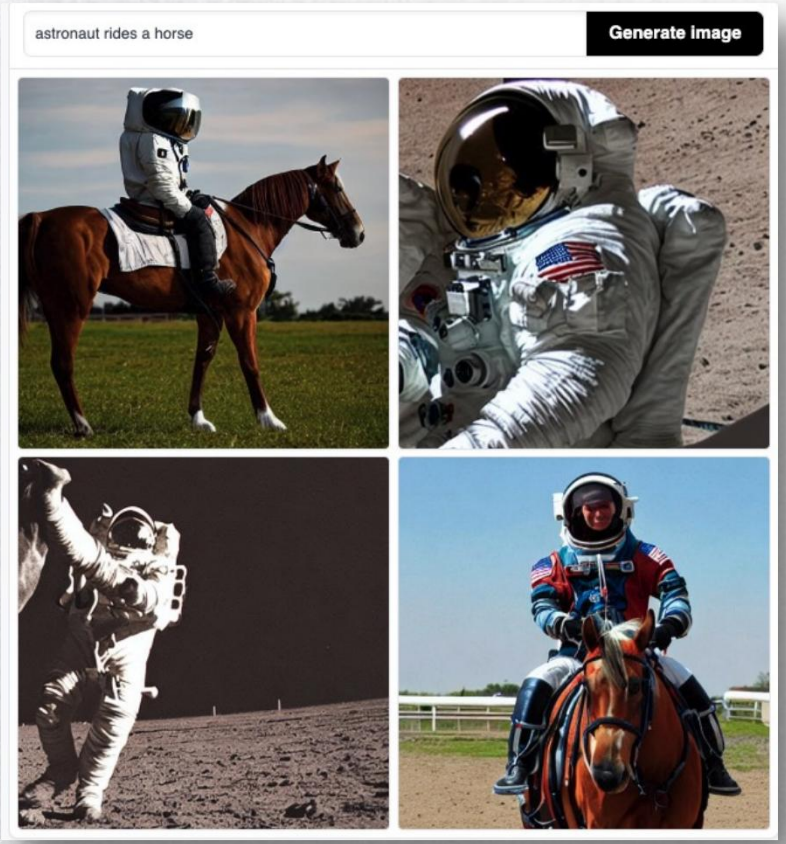
# Whisper 语音模型

- 英文语音转录成英文文本；
- 其他语言语音翻译成英文；
- 其他语言转录成该语言；
- 识别环境是否有人说话。



# 图像生成 - Diffusion扩散模型

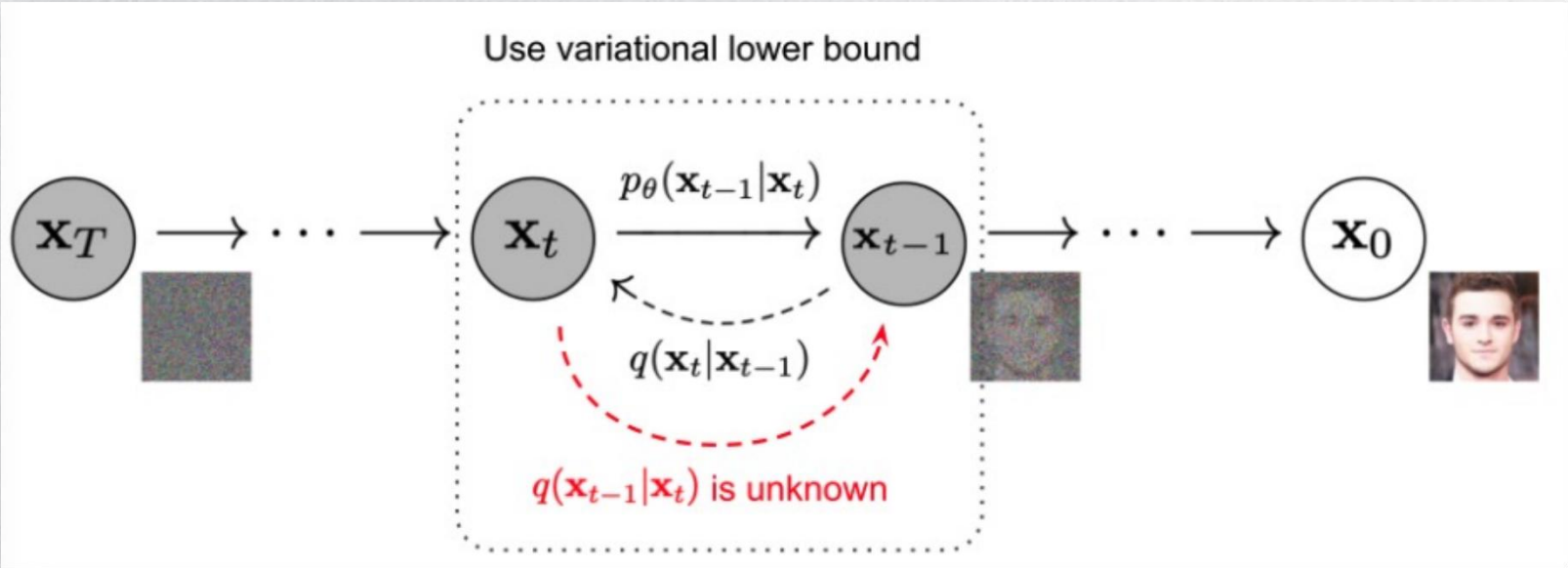



https://stablediffusionweb.com/#demo

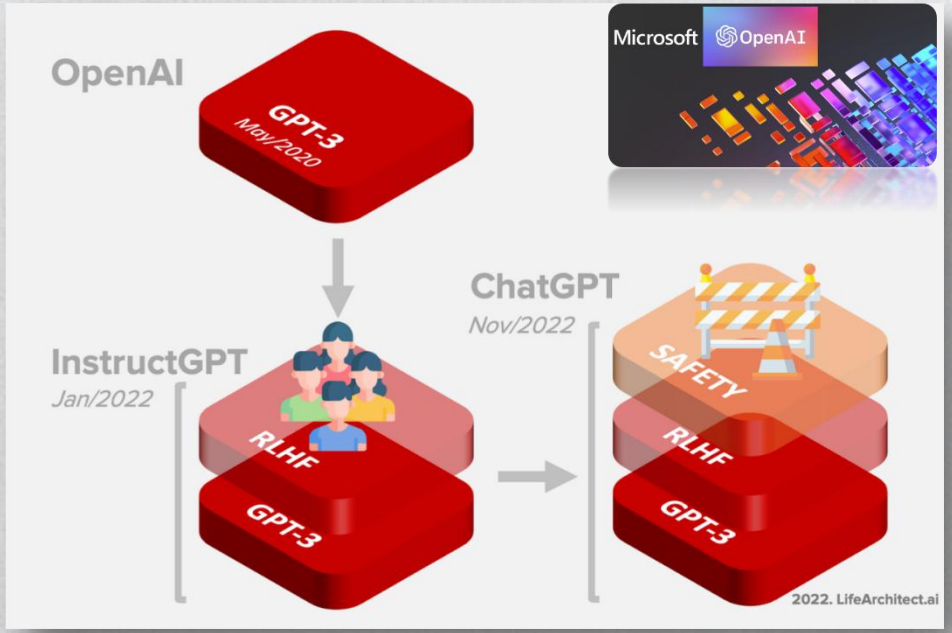
# 扩散模型 (~50亿参数)

## 学习到人类图片的所有纹理和语义特征




*Denoising Diffusion Probabilistic Models*, 2020.

# GPT (Generative Pre-Training) 发展历史




| 模型 | 年份 | 参数量 | 数据量 |
|---|---|---|---|
| GPT-1 | 2018 | 1亿 | 5GB |
| GPT-2 | 2019 | 15亿 | 40GB |
| GPT-3 | 2020 | 1750亿 | 45TB |
| GPT-3.5 | 2022 | 1750亿 | - |

# Chat-GPT：丰富多彩的生成能力

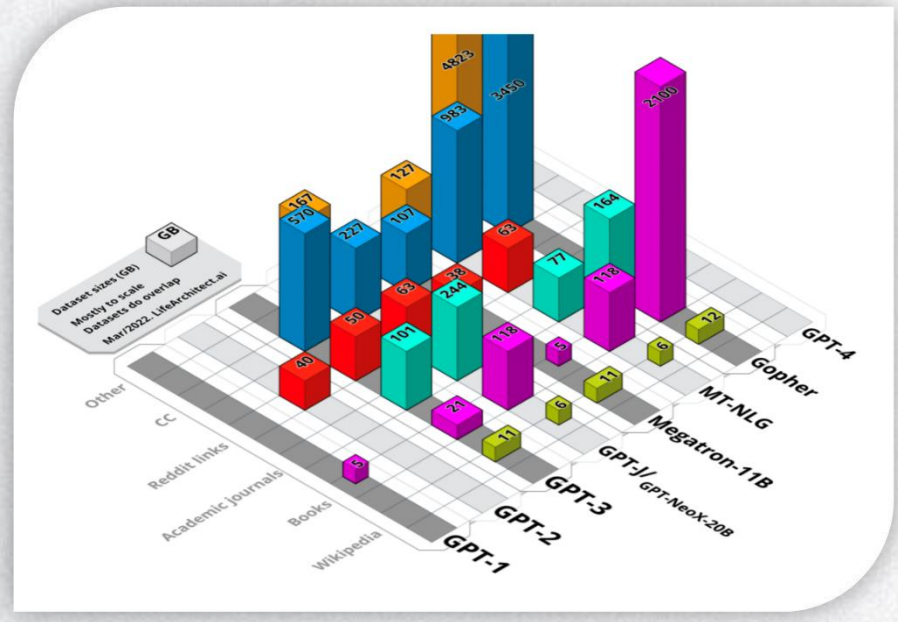
- **多种体裁:**
  - **小说,诗歌,邮件,学术,代码**
- **角色: 模仿各种人物写作风格**
  - **李白, 丘吉尔, 夏洛克**
- **指定语气和情感进行**
- **语言: 超过90种语言支持**

用幽默的口气向老师建议考试简单一些.

老师，你让我们太难了，让我们来一场轻松的考试吧！也许可以考考我们的“把一只猫放进冰箱”的答案！

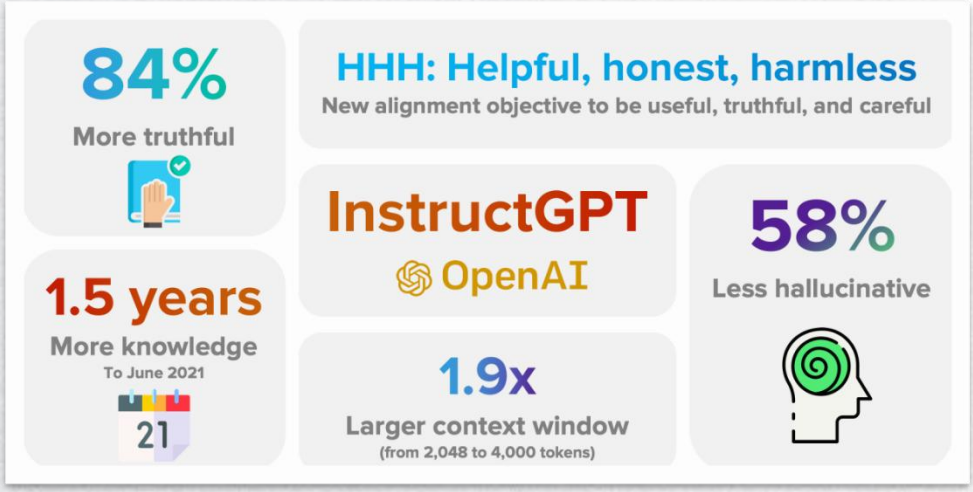
# ChatGPT 数据 - 网络抓取

- 45 TB 人类自然语言数据
- 包含1万亿不同语言的单词
- 来源推测 (未公开)
  - 60%来自于 C4 (互联网)
  - 22% 来自于 WebText2
  - 16% 来自于Books
  - 3%  来自于Wikipedi
- 有害内容审查 (时薪2美元)



# ChatGPT 数据 - 人类反馈

- 工程师设计提示文本，由人类回答进行有监督地微调训练
  - 1万条对话
- 人工对模型输出打分排序，训练模型分辨不当内容
  - 5万条对话
- 基于打分，使用强化学习算法进行微调，输出无害的、人类可理解的答案



# 应用举例

- 作为学生, 可以使用大模型做哪些工作？
- 作为AI/CS专业的学生,如何提高自己的代码和写作水平?
- 作为日常使用工具,如何使用大模型提高工作效率?

# 示例代码生成

Q:用python写一个快速排序

```
# 快速排序
def quick_sort(array):
    if len(array) < 2:
        return array
    else:
        pivot = array[0]
        less = [i for i in array[1:] if i <= pivot]
        greater = [i for i in array[1:] if i > pivot]
        return quick_sort(less) + [pivot] + quick_sort(greater)
```

# 代码 debug

Q:how to correct the code a[1,2,3,4,5] print(a[5])

A:

The code should be changed to print(a[4]) since the index of the last element in the list is 4.

# 修改英文语法

Q:修改英文语法Please revise. He go to work early everyday.

A:

He goes to work early every day.

# 写一封请假信

Q:写一封中文请假信，请假10天，说明我要陪父母去海南玩

A:

尊敬的老师：

你好！

我是你的学生XXX，因为家里有事，我需要请假10天。具体原因是，我的父母要去海南旅游，我想陪着他们一起去，所以我需要请假10天。

# 文献检索

宇函

提供一些多模态和chatgpt结合的论文链接

以下是一些关于多模态和ChatGPT结合的论文链接，供您参考：

1. "M-MIMIC: Multimodal Medical Information Generation with Iterative Refinement and Context Fusion" (2022) by Yifan Zhang et al. https://arxiv.org/abs/2201.01553
2. "Multimodal Language Generation with Recurrent Variational Autoencoders" (2021) by Xinyu Zhang et al. https://arxiv.org/abs/2106.07228
3. "Multi-Modal Conversational AI with Dense Generative Transformer Networks" (2021) by Sumanth Dathathri et al. https://arxiv.org/abs/2103.03229
4. "MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations" (2019) by Soujanya Poria et al. https://arxiv.org/abs/1810.02508

这些论文探讨了多模态和ChatGPT结合的应用，从不同的角度研究了该领域的问题和挑战，并提出了一些创新性的方法和解决方案。希望这些链接对您有所帮助！

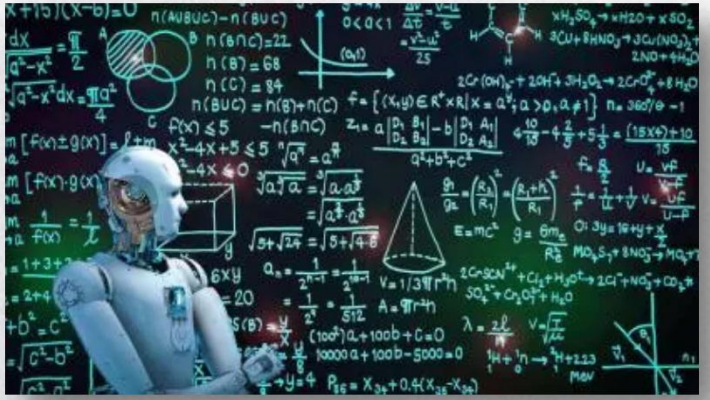
# 令人深思的话题

- 我们能够信任AI给的答案的准确性吗? AI 医生的建议可靠吗?
- AI 真的能够理解一段完整的对话吗? 它会遗忘我们的历史对话吗? 它有逻辑推理能力吗?
- 人类还有必要思考吗, 如果我们的专业知识都能够被AI轻易的记住并回答. 我们会被AI取代吗?
  - **No.小学数学题平均得分55分*!**

* *Training Verifiers to Solve Math Word Problems*, 2021

# 模型滥用的担忧

- **输出虚假、不实的信息和知识**
  - 论文引用和链接不存在
  - 专业领域的事实错误
- **数据泄露问题**
  - 隐私泄露被AI爬取
  - 研究资料通过对话被套取
- **恶意内容生成**
  - 病毒代码, “毁灭人类计划书”

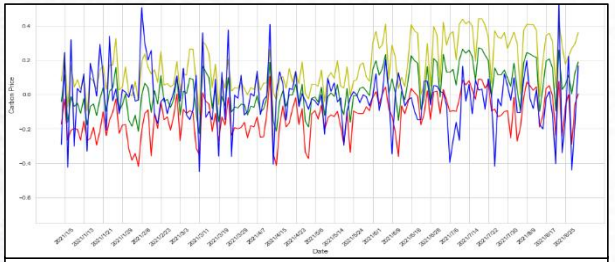


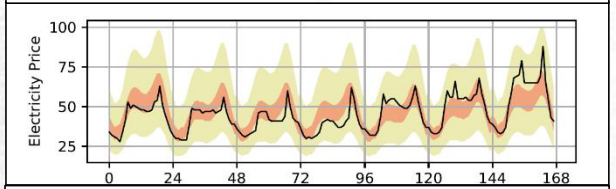
# 数据、算力垄断和霸权

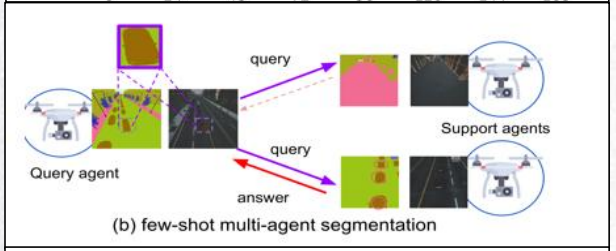
- **大模型的数据和算力门槛过高**
  - 少数企业和机构掌握了模型和计算技术
- **芯片技术封锁**
  - 最新的计算卡如A100已被限制出售
- **数据先发优势导致的垄断**
  - 垄断大量用户输入语料
  - 自动驾驶大厂掌握全世界大量道路、地图信息等

# 致谢和广告

- 感谢研究生同学马宇函和庞天琦
- 感谢人工智能学院陈寅副院长的倡议，林娜婷与王曼老师的技术协助
- 欢迎对人工智能感兴趣的同学关注我们的研究。我们在TPAMI, CVPR, NeurIPS, MM, AAAI 等人工智能顶刊发表论文
- CV&NLP, 时序预测, 分布式学习, 强化学习等研究方向
- https://www.scholat.com/fanchenyou





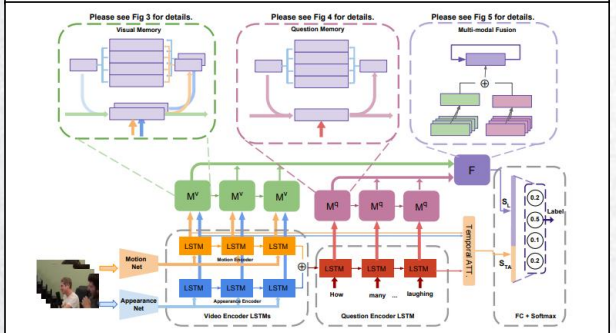




Figure 2. Our proposed VideoQA pipeline with highlighted visual memory, question memory, and multimodal fusion layer.

THANKS

QA