



中国科学院大学

University of Chinese Academy of Sciences

自然语言处理

第5讲 预训练

王石 资康莉 刘瑜

2026年春季课程

<https://ictkc.github.io/teaching/>



第五讲 预训练

从理论突破走向实际应用：Word2Vec (13')

□ NNLM和Word2Vec的训练效率对比

Table 6: *Comparison of models trained using the DistBelief distributed framework. Note that training of NNLM with 1000-dimensional vectors would take too long to complete.*

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

训练代价太大，别人训练好了我能直接使用吗？



目 录

1

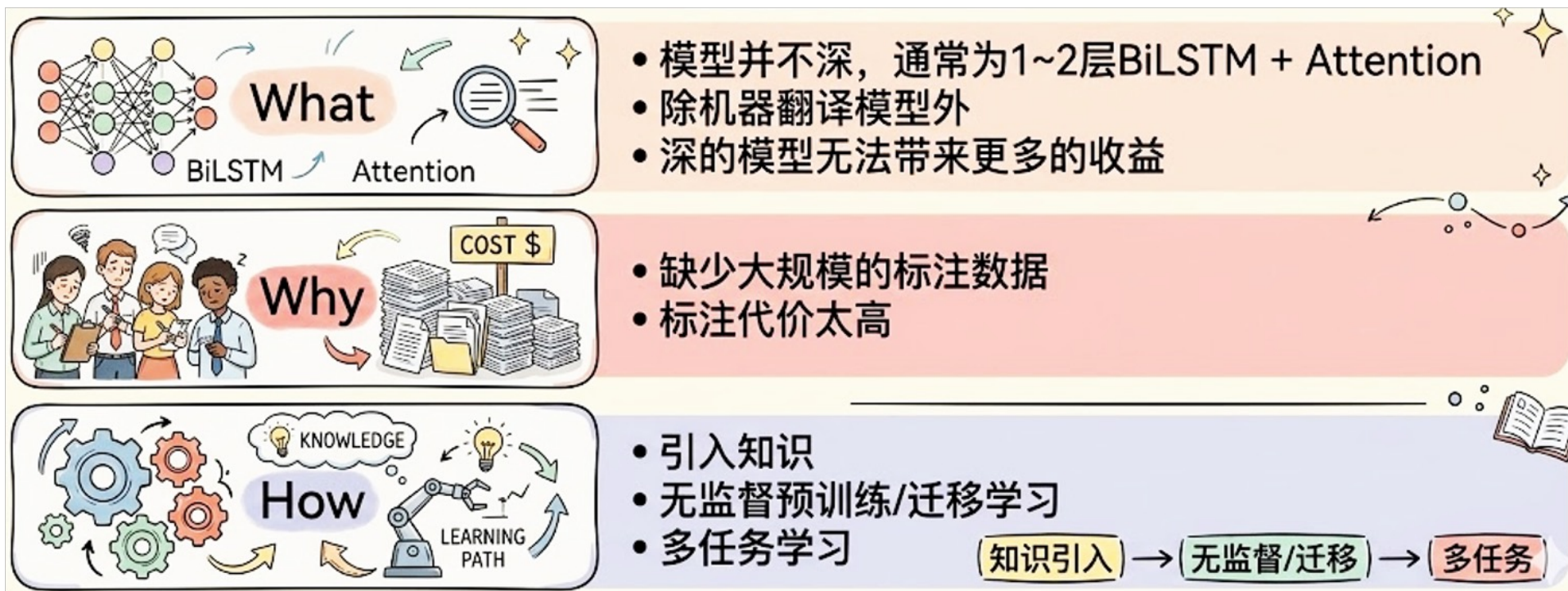
什么是预训练

2

3

为什么要预训练?

□ 深度学习在自然语言处理中的“困境”



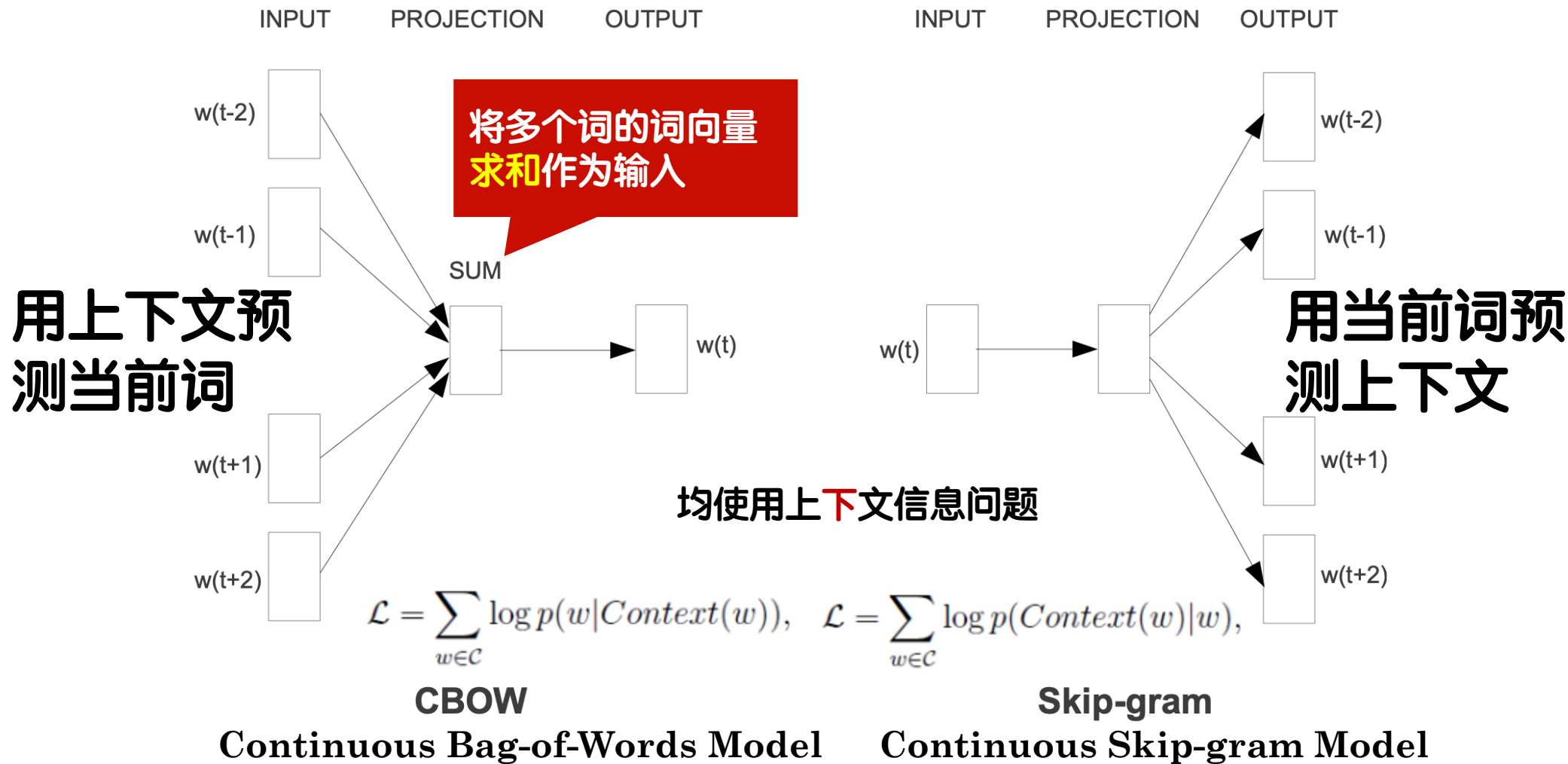
预训练：用无监督学习解决标注数据不足的问题，让深度网络的潜力真正释放出来

什么是预训练模型？

- 在海量无标注数据上进行大规模**自监督**预训练，学习到语言知识和世界知识的模型

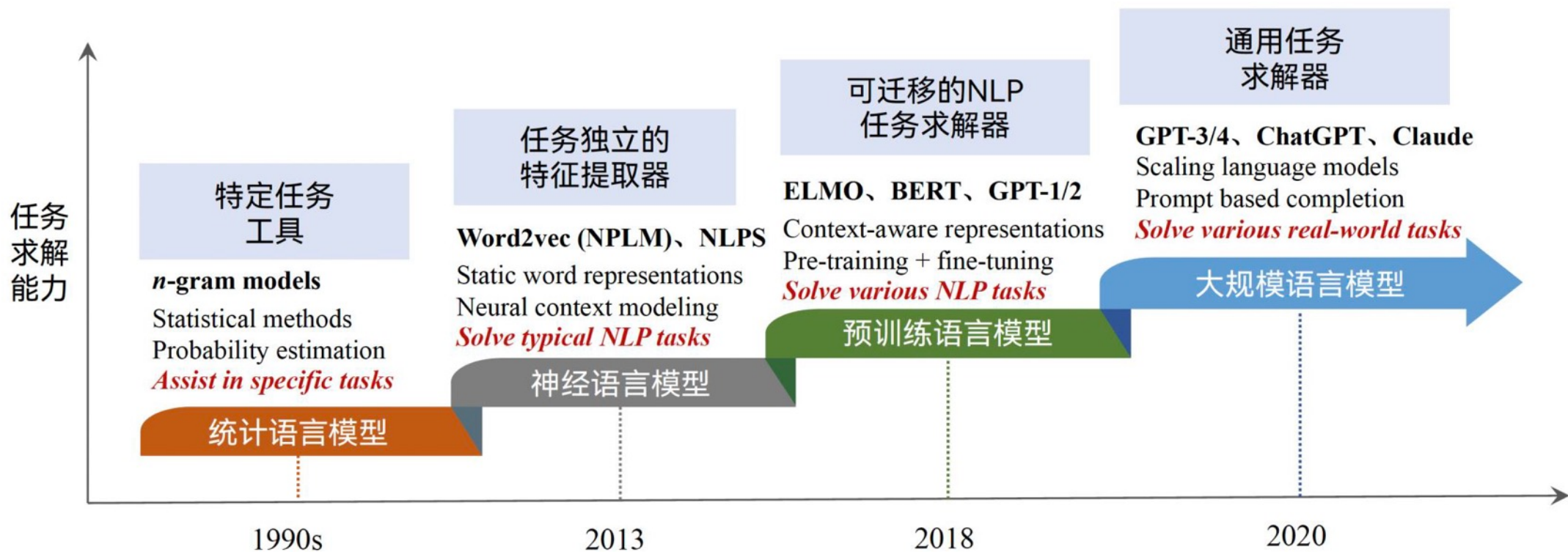


自监督



预训练模型的发展

□ 从小数据到大数据，小模型到大模型，专用到通用的发展历程





目 录

1

什么是预训练

2

预训练模型

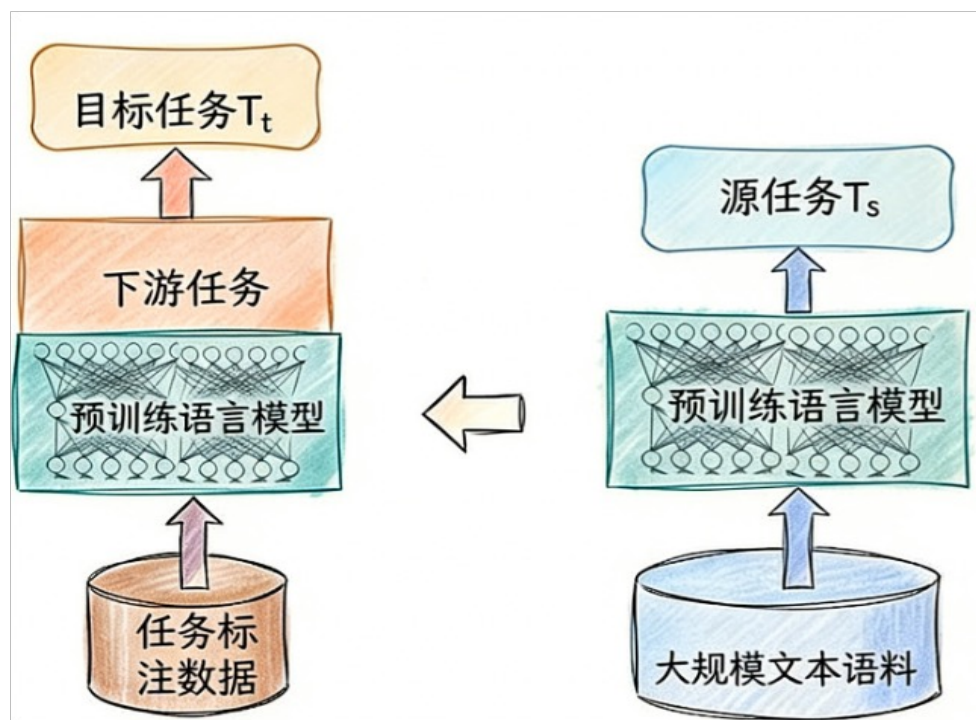
3

基本思想

□ 预训练模型基本思想:



问题: 对于任务 t ,
标注数据有限



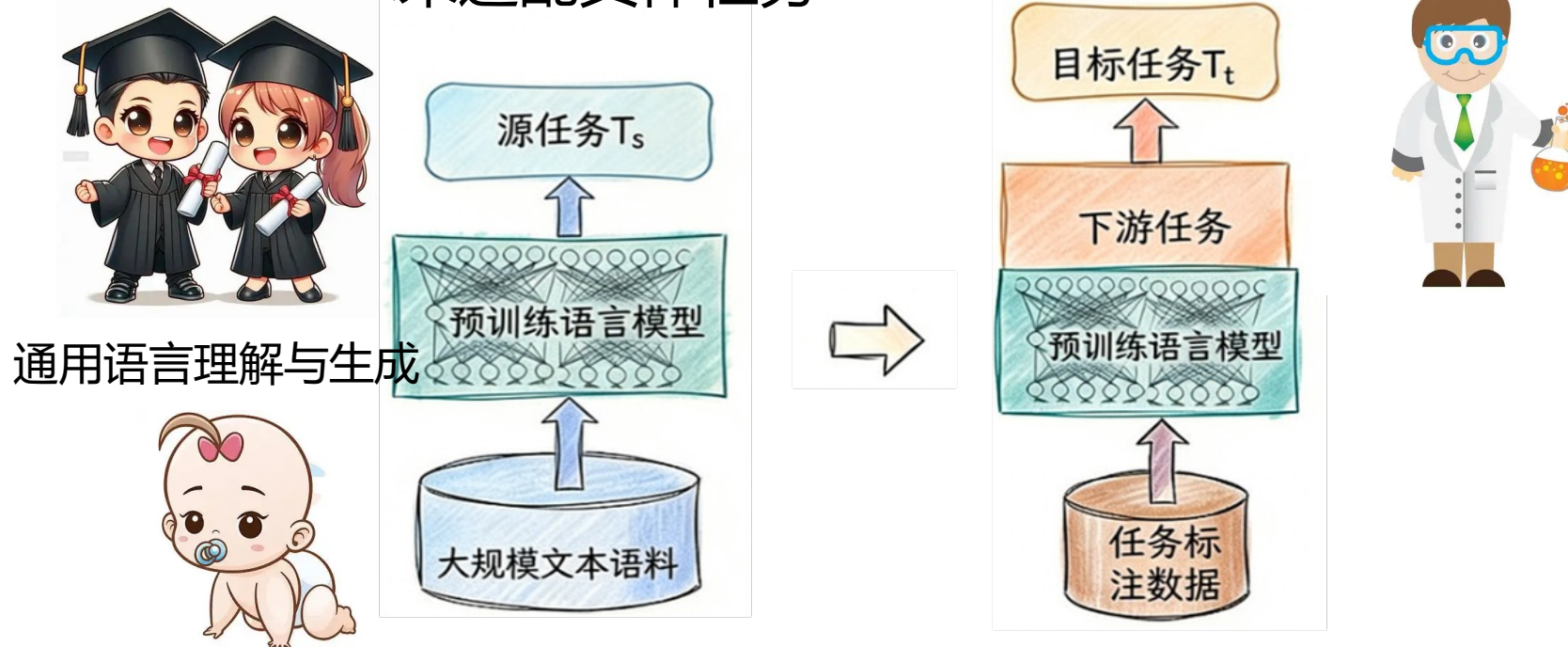
解决方法: 将任务 s 数据或所学知识
迁移到任务 t , 帮助任务 t 学习

预训练模型范式

□ 预训练两个阶段：

第一阶段：Pre-training，利用大型语料库完成预训练模型自监督学习

第二阶段：Fine-tuning，针对特定任务在相应数据集中进行监督学习，通过Fine-tuning技术来适配具体任务



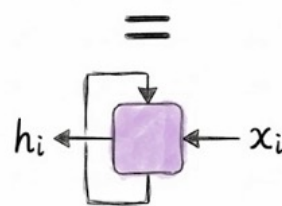
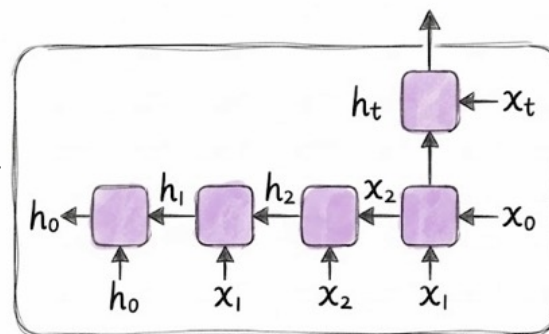
预训练模型架构

□ RNN (LSTM)

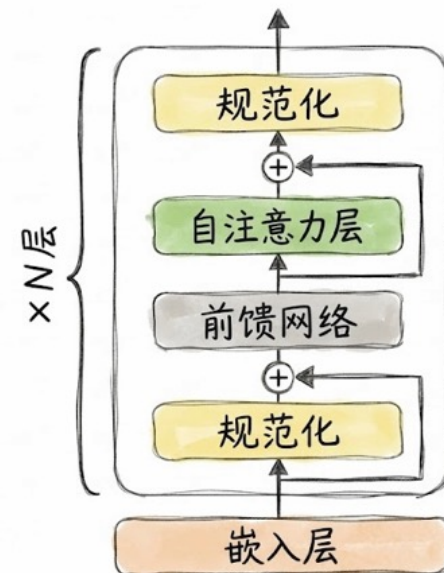
- 存在梯度问题，长距离建模能力有限
- 难以并行计算，训练速度较慢

□ Transformer

- 采用自注意力机制进行全局处理
- 擅长建模长距离依赖，支持并行计算
- 表达能力强，由多头注意力和深度结构提升建模能力



(a) RNN



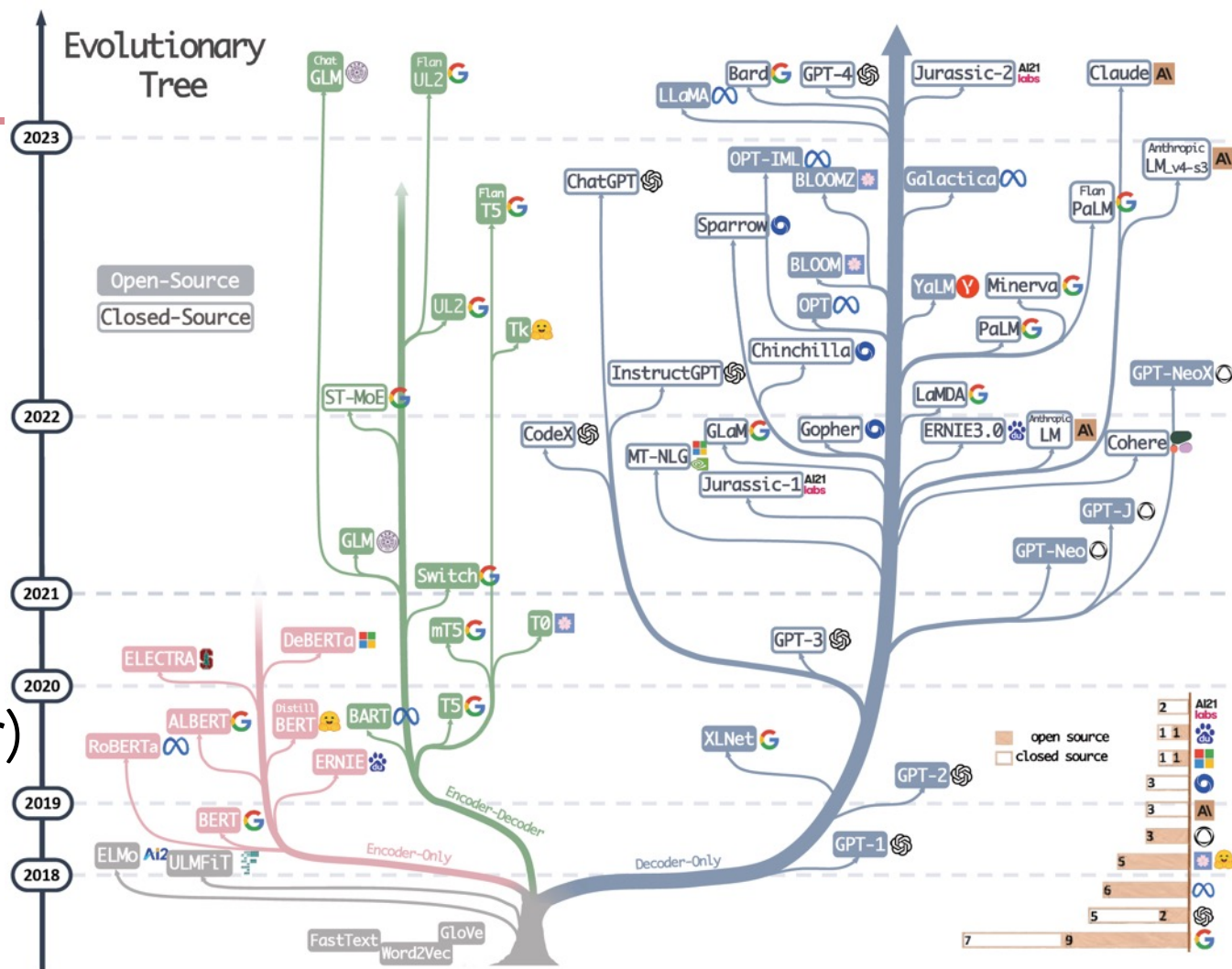
(b) Transformer

从“顺序记忆”向“全局建模” + “高效并行”

预训练模型类型

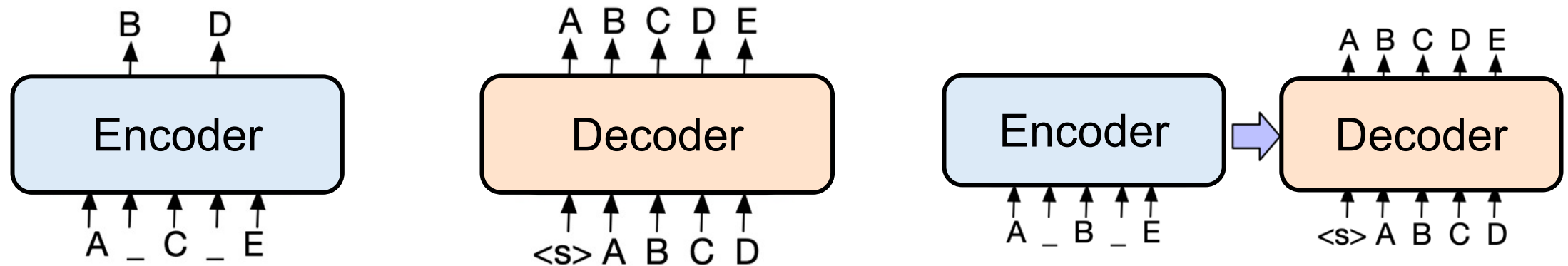
□ 根据利用Transformer的编解码器的不同

- █ 编码器模型 (Encoder-Only)
- █ 解码器模型 (Decoder-Only)
- █ 编解码模型 (Encoder-Decoder)



预训练模型类型

- 从自监督预训练的视角，不同架构的训练目标存在差异
 - 编码器模型：自编码语言建模 (Auto-encoding Language Modeling)
 - 解码器模型：自回归语言建模 (Auto-regressive Language Modeling)
 - 编解码模型：序列到序列建模 (Sequence to Sequence Modeling)



编码器模型

- 训练目标：自编码语言建模（Auto-encoding Language Modeling）

$$\mathcal{L}_{\text{MLM}} = - \sum_{x_m \in M(\mathbf{x})} \log \mathbf{P}(x_m | \mathbf{x} \setminus M(x))$$

- 代表性模型：BERT、RoBERTa、ALBERT等系列

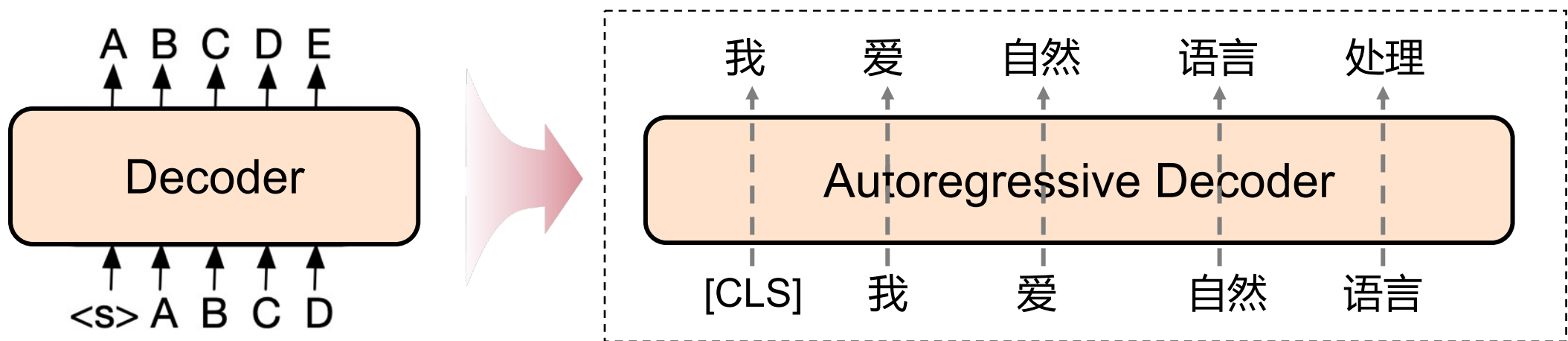


解码器模型

- 训练目标：自回归语言建模 (Auto-regressive Language Modeling)

$$\mathcal{L}_{\text{LM}} = -\log \mathbf{P}(\mathbf{x}) = -\sum_{i=1}^T \log \mathbf{P}(x_i | x_{<i})$$

- 代表性模型：GPT、Llama、PaLM等系列

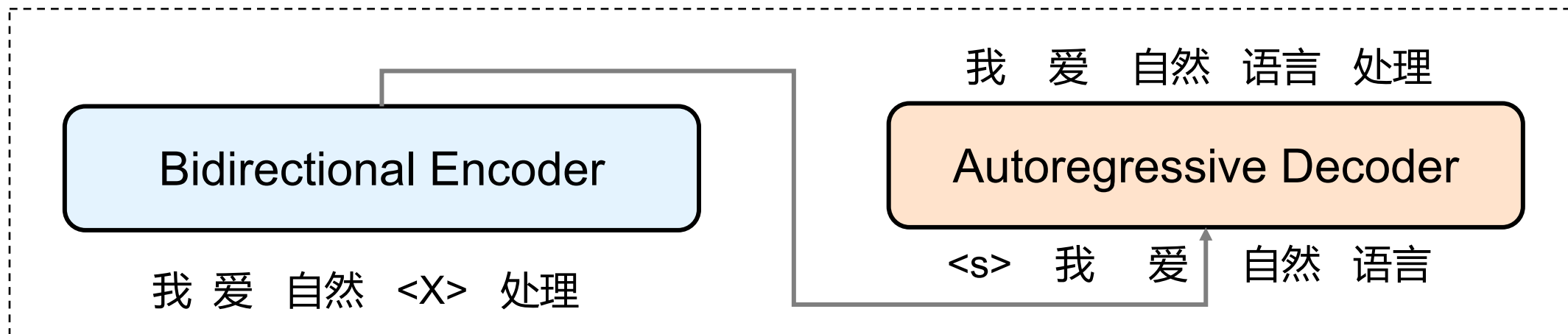


编解码模型

- 训练目标：序列到序列建模 (Sequence to Sequence Modeling)

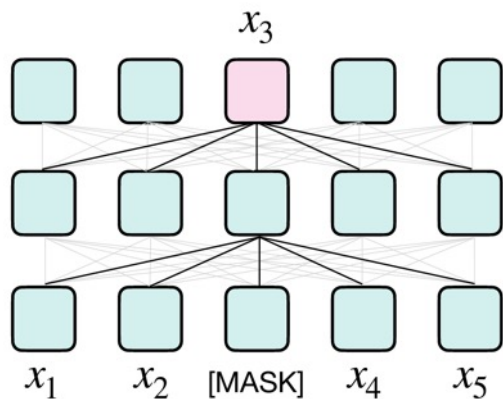
$$\mathcal{L}_{\text{Seq2Seq MLM}} = - \sum_{\mathbf{x}_{i:j} \in M(\mathbf{x})} \sum_{t=i}^j \log \mathbf{P}(x_t | \mathbf{x} \setminus M(\mathbf{x}), \mathbf{x}_{i:t-1})$$

- 代表性模型：T5、BART等系列



优缺点对比

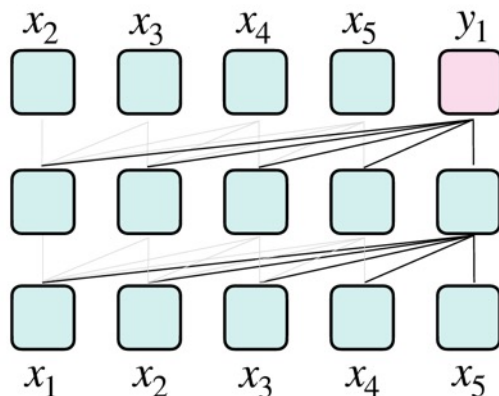
编码器模型



优点： 利用双向上下文，语言理解能力强

缺点： 预训练与下游任务存在不一致，MASK位置稀疏

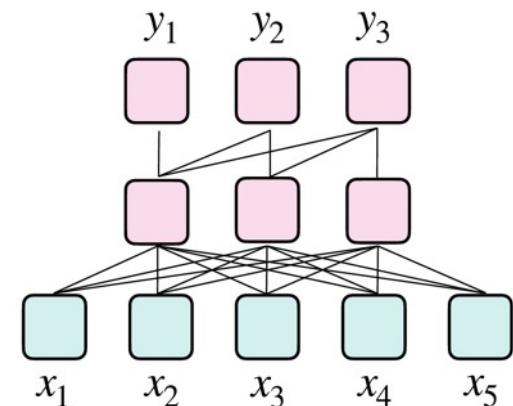
解码器模型



优点： 训练与生成过程一致，可扩展性强

缺点： 单向建模，上下文利用不充分

编解码模型

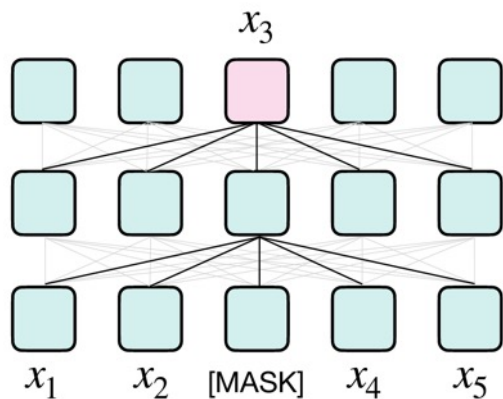


优点： 同时兼顾理解与生成能力，分工明确

缺点： 模型结构复杂，训练和推理成本较高

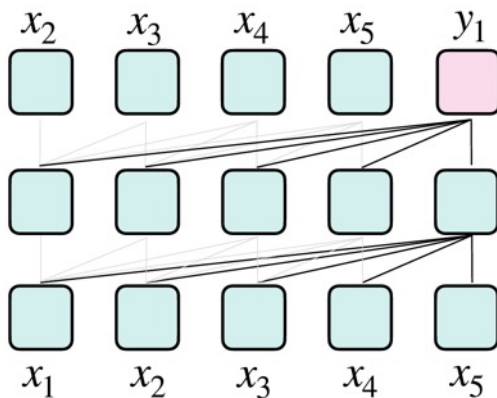
适应场景

编码器模型



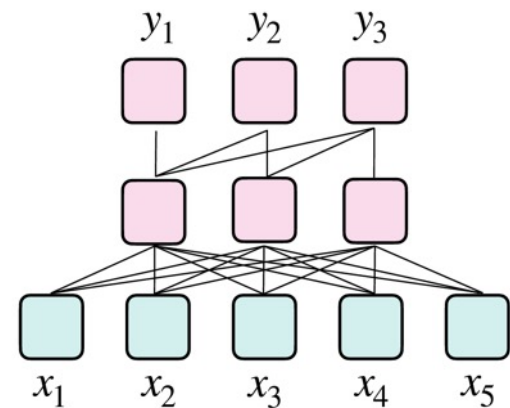
适合**判别式**任务
(分类/匹配/抽取等)

解码器模型



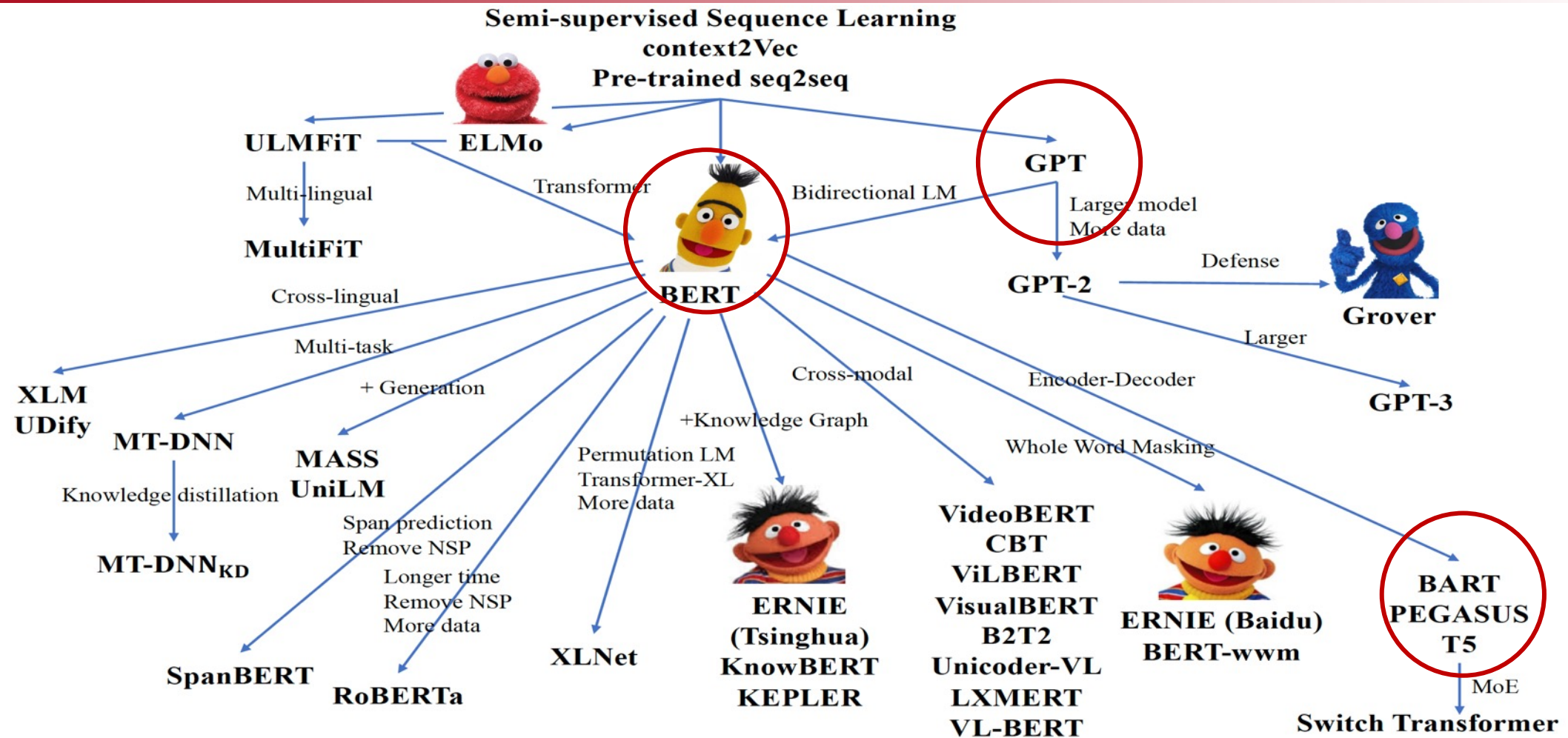
适合**开放生成**任务
(写作/对话等)

编解码模型



兼具**判别与生成**能力
(翻译/摘要等)

预训练模型家族详解





目 录

- ① 什么是预训练
- ② 预训练模型
 - ②.1 编码器模型
- ③

编码器模型BERT

Bert: Pre-training of deep bidirectional transformers for language understanding

J Devlin, [MW Chang](#), [K Lee](#)... - Proceedings of the 2019 ..., 2019 - aclanthology.org

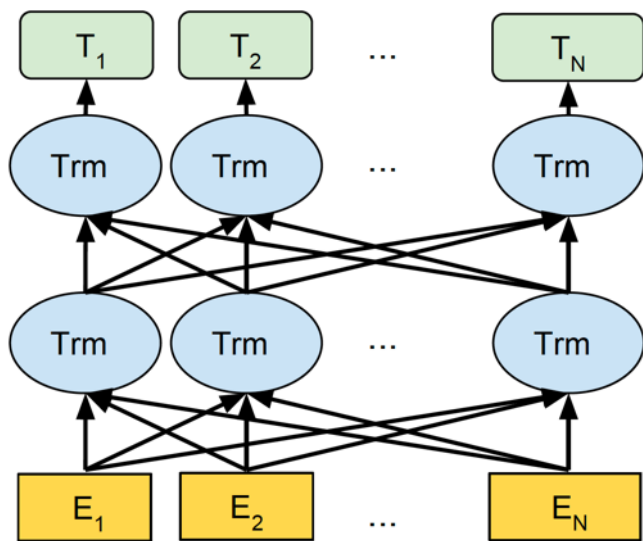
... tion model called **BERT**, which stands for **Bidirectional** Encoder ... 2018), **BERT** is designed to pretrain deep **bidirectional** ... As a result, the pre-trained **BERT** model can be finetuned with ...

☆ 保存 引用 被引用次数: 165454 相关文章 所有 21 个版本 》》

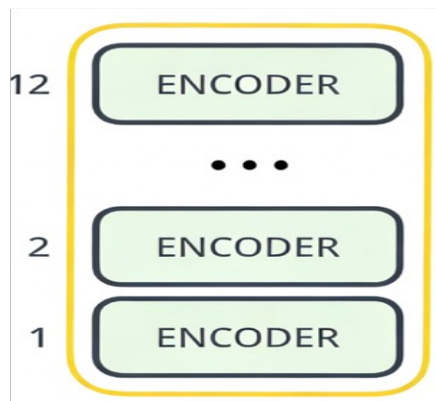


编码器模型BERT

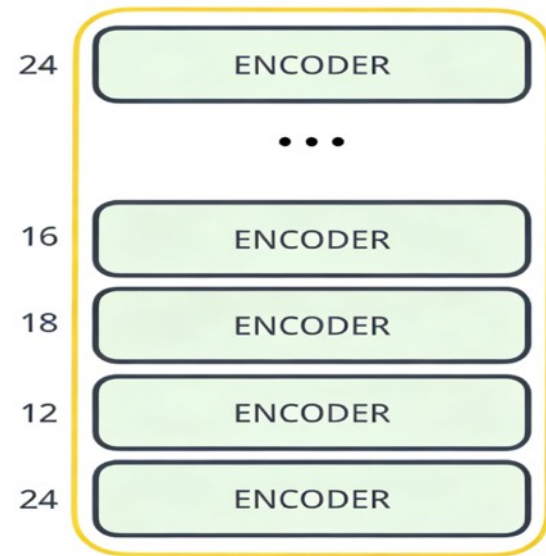
- BERT使用堆叠的**双向Transformer Encoder**，在所有层中共同依赖于左右上下文
- 基础版是12个Encoder (12层)；高级版24个Encoder (24层)



BERT



BERT_Base



BERT_Large



BERT预训练

- BERT预训练阶段主要利用大型语料库完成非监督学习，主要包括两个训练策略

策略一

掩码语言模型 (预测词)
Masked Language Modeling

解决上下文信息泄漏问题，
捕捉词语上下文关系

策略二

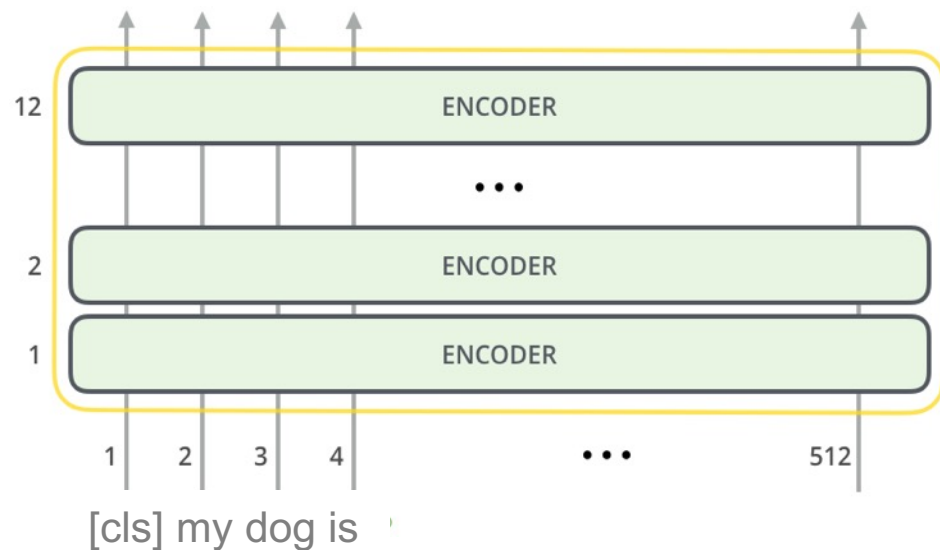
预测下一个句子
Next Sentence Prediction

解决跨句语义理解问题，
建模句子间逻辑关系



BERT输入

- BERT输入是一个句子或一个句对（句子：任意长度的连续文本）
- 每个输入序列以 [CLS] 开头，句子对之间加一个 [SEP]
- 输入表示由三部分组成：词表示、句子表示和位置表示

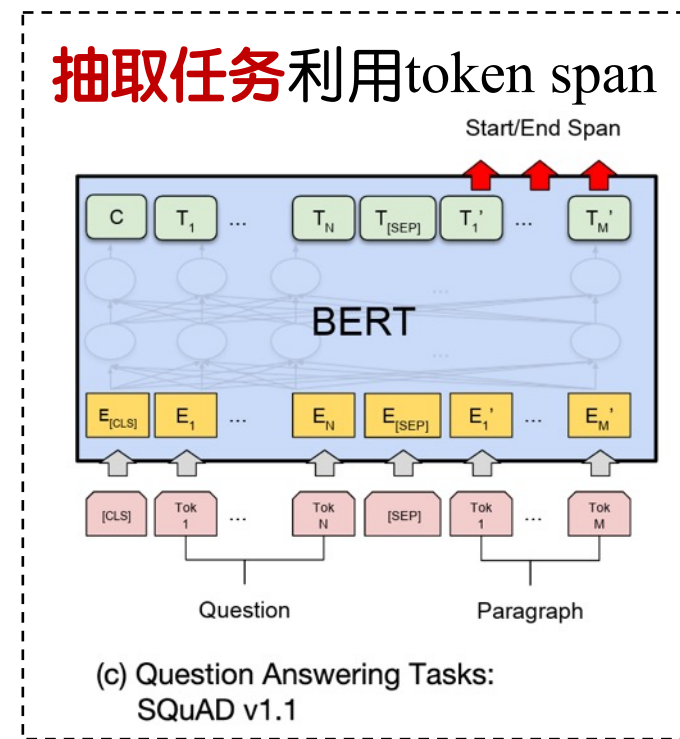
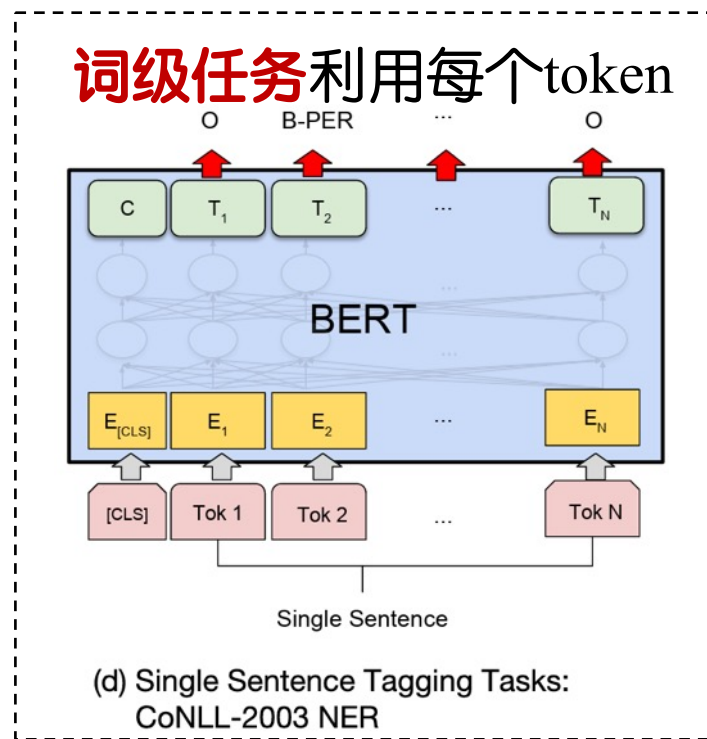
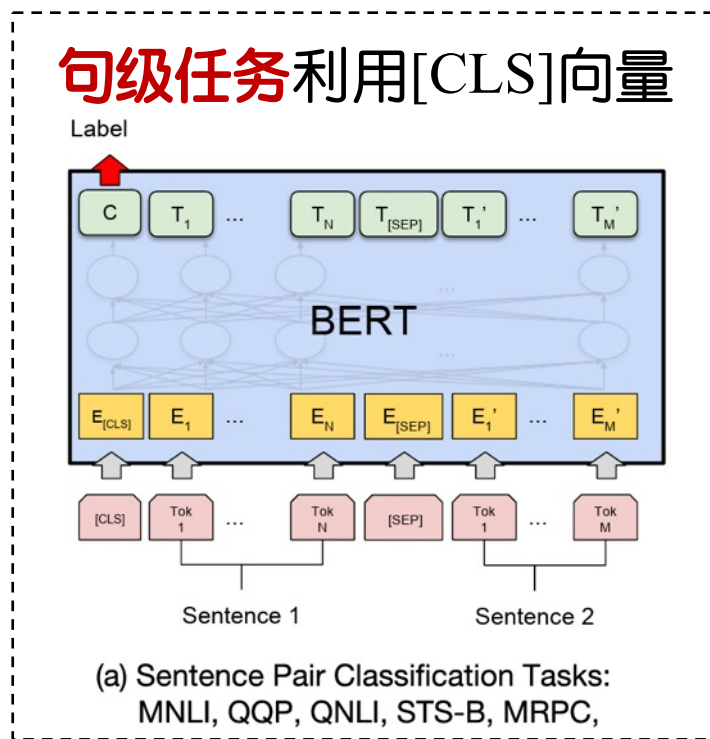


Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{##ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}



BERT输出

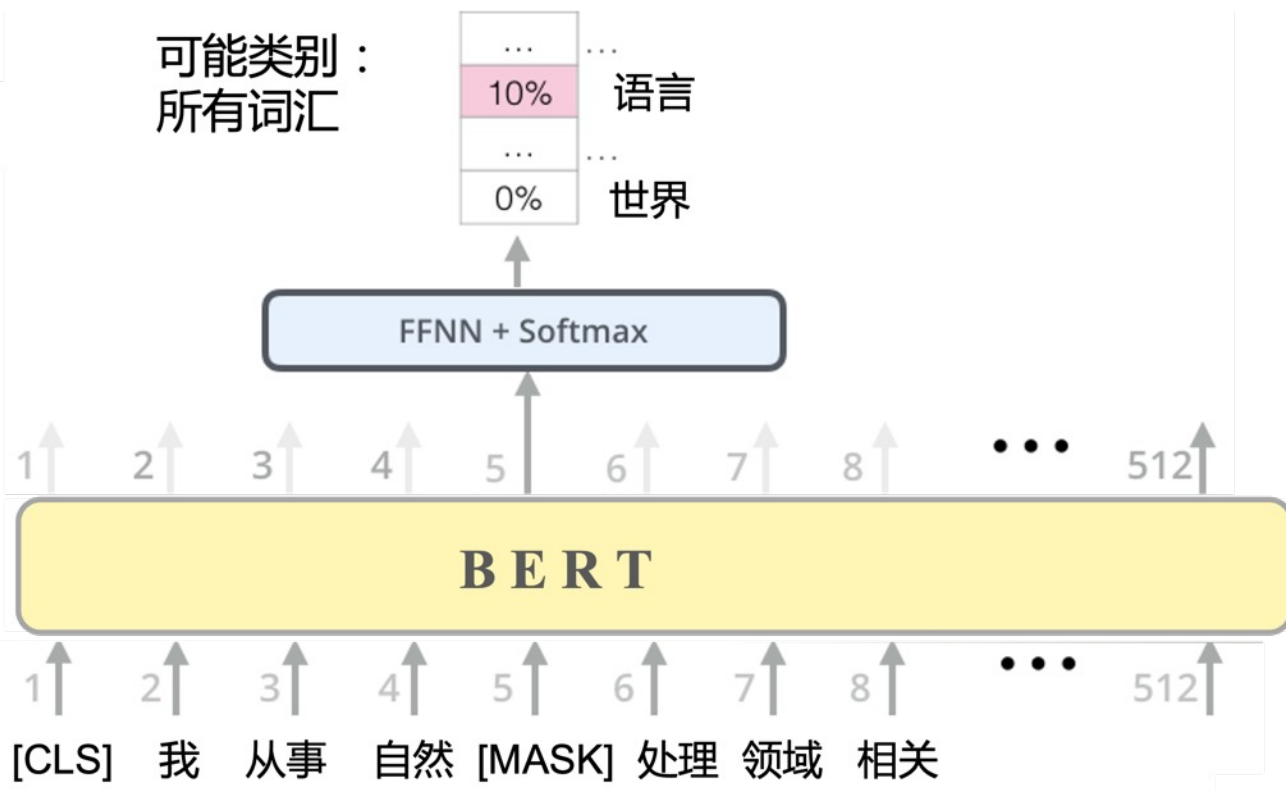
- 根据下游任务不同，BERT采用三种输出方式，核心在于选择不同位置或粒度的表示进行预测





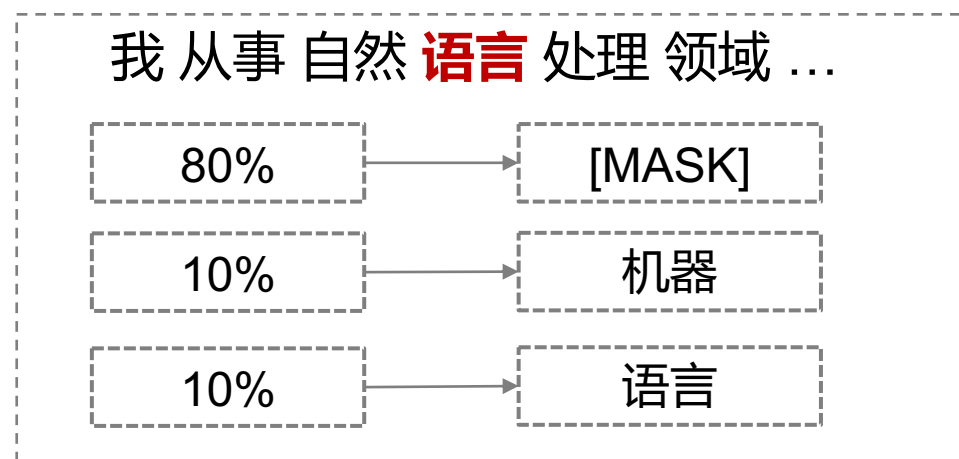
掩码语言模型MLM

□ 随机遮盖mask部分输入词，然后只预测那些被mask的词



□ 训练数据构造方式

随机遮住15%的单词，并进行如下处理。其中：80%用[MASK]来代替，10%用随机词替换，10%保持不变。





预测下一个句子NSP

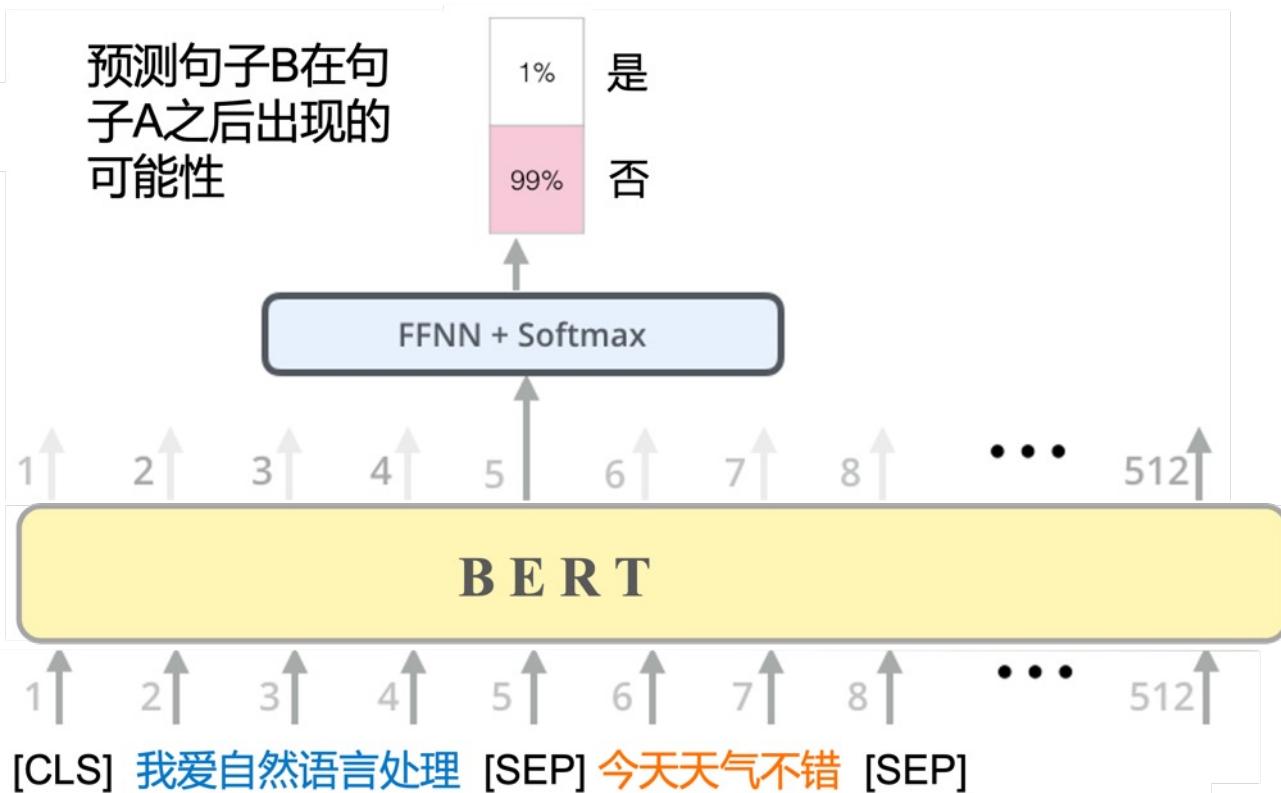
□ 二分类任务，判断两个句子是否连续，建模句间关系

□ 训练数据构造方式

正样本：给定句子A和B，B是A的实际语境下一句；

负样本：在语料库中随机选择的句子作为B

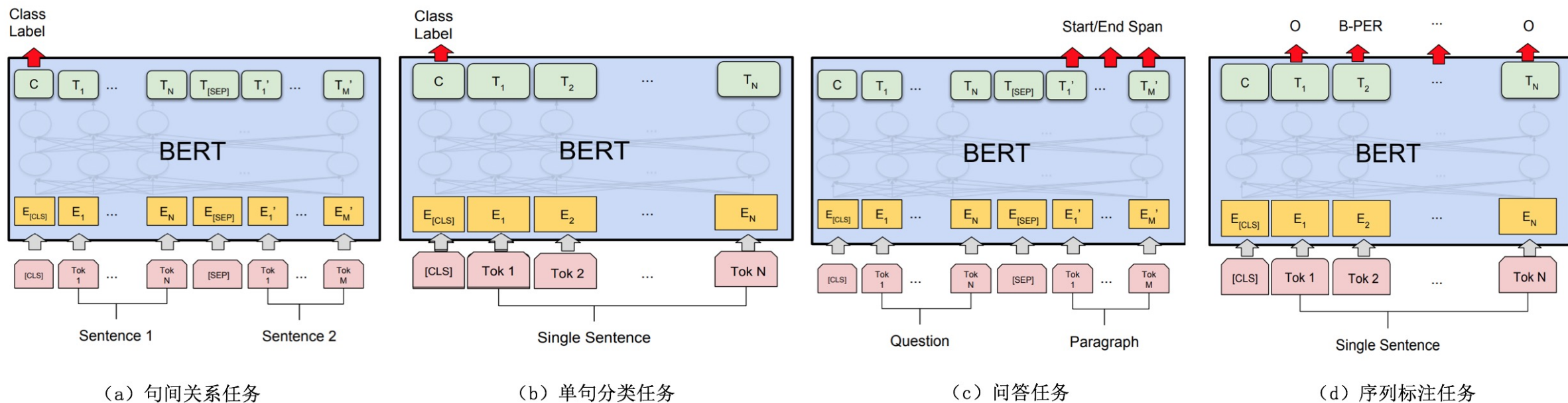
正负样本比例1:1，预测[CLS]位置





BERT微调

- 预训练和微调阶段采用相同的模型结构，预训练的参数可作为初始化
- 微调阶段仅需通过引入一个额外的输出层，可以达到较优效果





BERT实验效果

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.⁸ BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

刷新了句子关系判断及分类任务、抽取式任务 (SQuAD)、序列标注任务、分类任务、文本生成任务等11项自然语言处理任务榜单

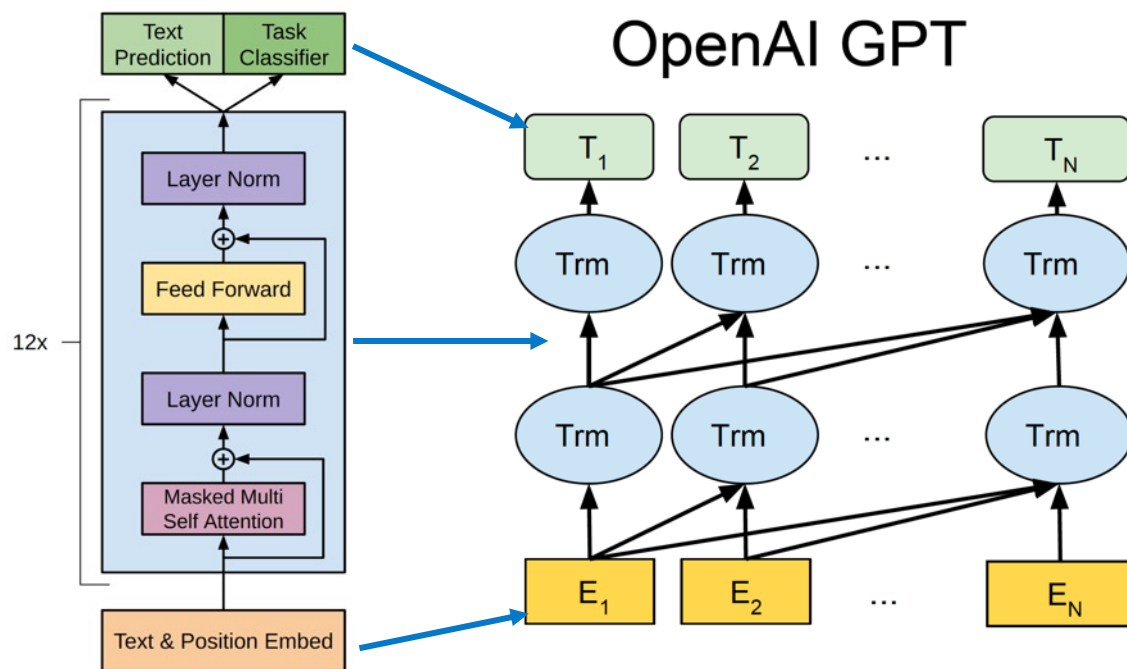


目 录

- 1 什么是预训练
- 2 预训练模型
 - 2.1 编码器模型
 - 2.2 解码器模型
- 3
- 4

解码器模型GPT

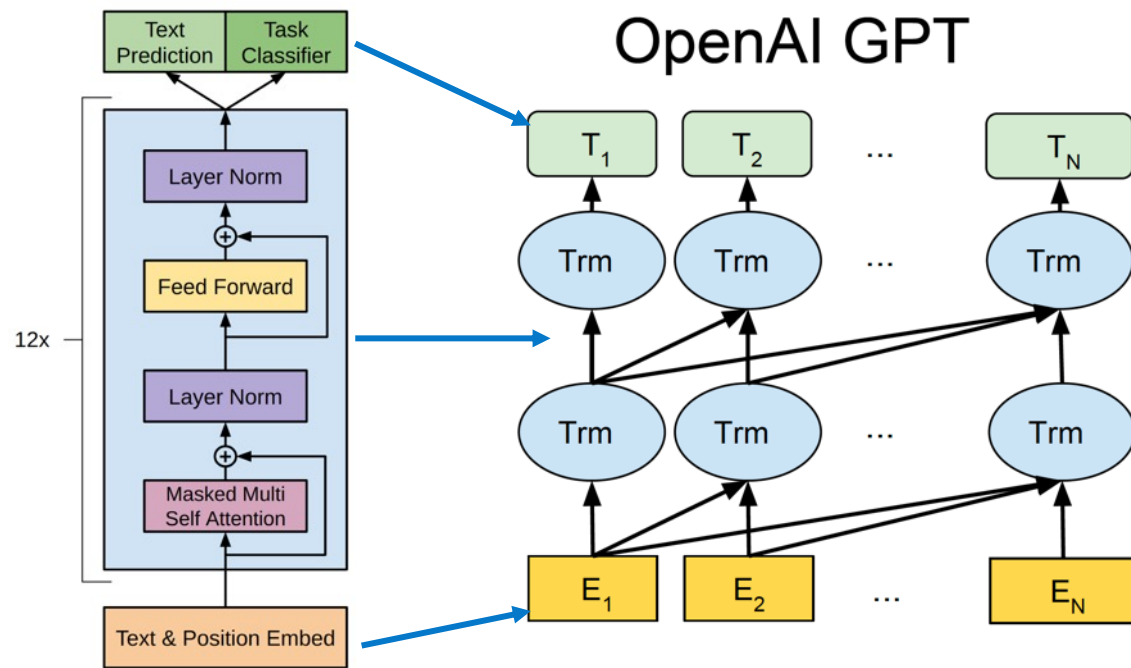
- GPT (Generative Pre-trained Transformer) 采用**Transformer Decoder**部分，只采用上文词来预测当前词，避免“自己看见自己”问题



- 每一层只有一个Masked Multi-Head Attention和一个Feed Forward
- 模型共叠加使用了12层

GPT预训练

- GPT在Pre-training阶段主要利用大型语料库完成非监督学习，其目标是最大化语言模型



□ 输入输出

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer_block}(h_{l-1}) \forall i \in [1, n]$$

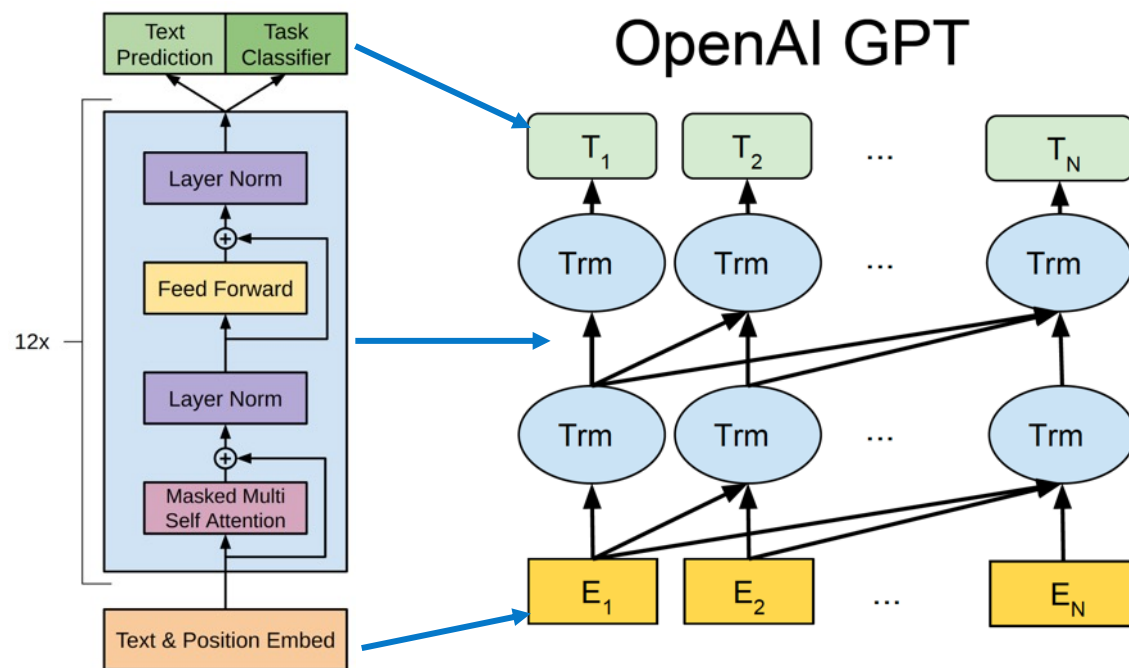
$$P(u) = \text{softmax}(h_n W_e^T)$$

□ 目标函数

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

GPT任务微调

□ 在Fine-tune阶段，将预训练模型提供给下游的任务，预训练模型与下游任务模型联合优化



□ 任务微调有2种方式

● 冻结预训练参数，只调任务参数

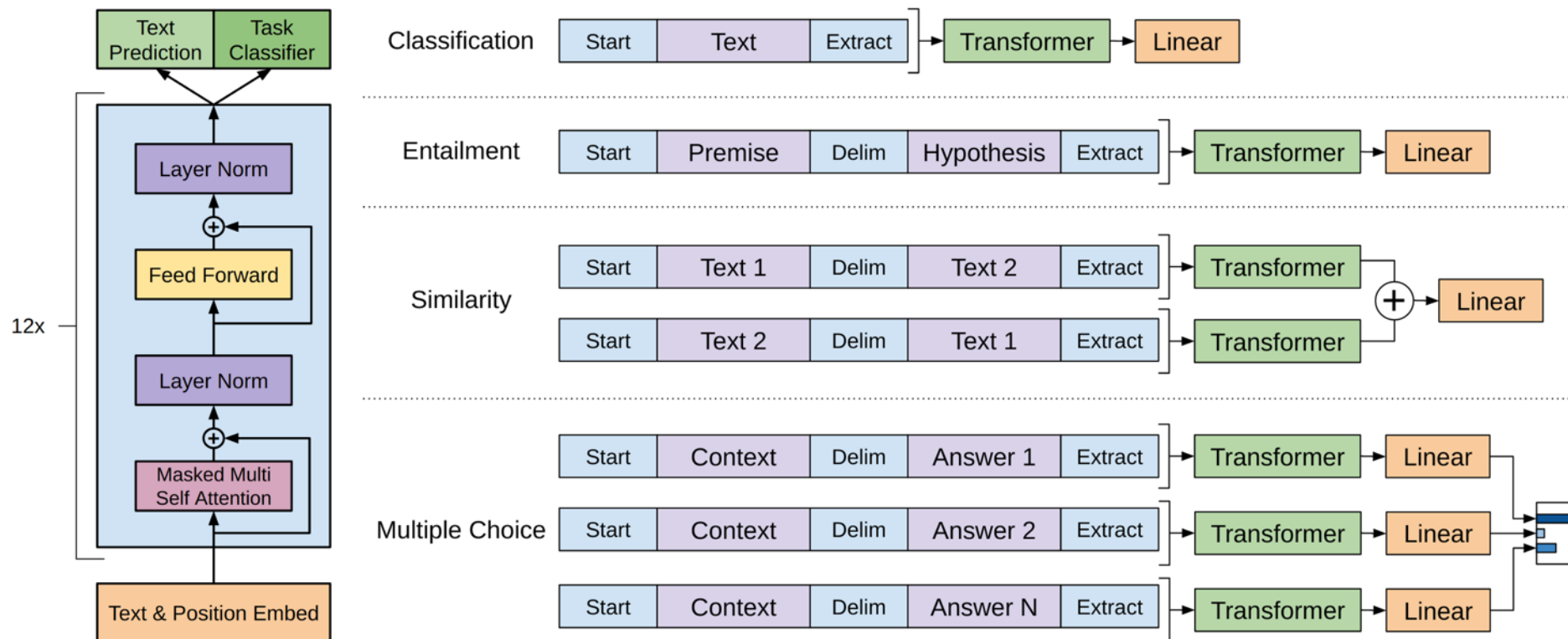
$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m)$$

● 预训练和任务参数参数一起调

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

GPT任务转换

- 对于不同的输入任务，采用相应的文本拼接方式，再使用Transformer层进行处理，最后使用Linear层完成特定的监督学习任务



GPT任务转换

文本分类 (Classification) :

任务描述: 将文本划分到预定义的类别中。

GPT策略: 直接微调模型。在模型的输出层添加一个线性层, 将GPT的输出转换为对应类别的概率分布。

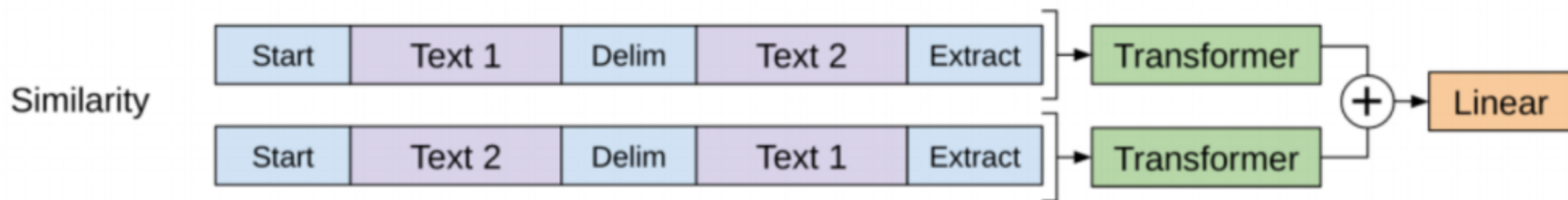


GPT任务转换

相似性 (Similarity) :

任务描述: 判断两个文本之间的相似性。

GPT策略: 由于句子顺序不固定, 处理时生成两种可能的句子顺序, 并分别得到它们的表示。将这些表示相加后, 输入到线性输出层中进行相似性判断。

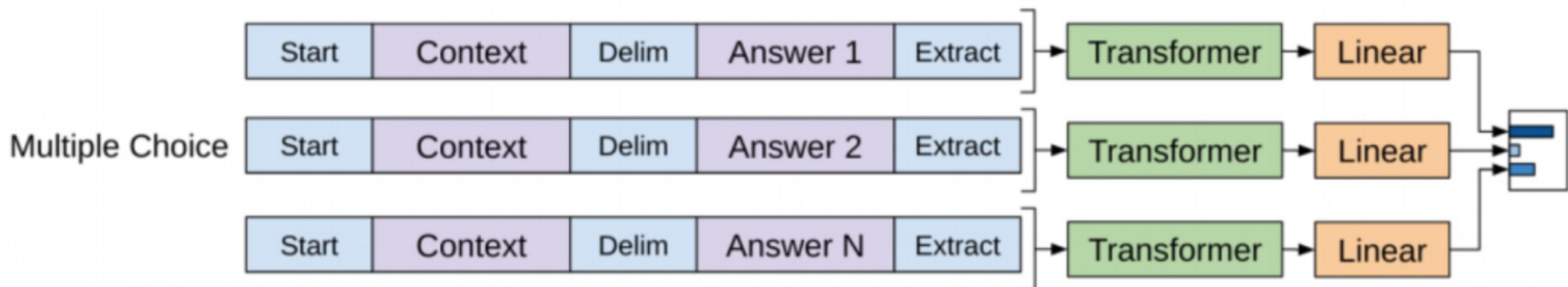


GPT任务转换

问答和常识推理 (Multiple Choice) :

任务描述: 给定上下文、问题和一组候选答案, 选择最合适的答案。

GPT策略: 将上下文、问题和每个候选答案连接起来, 中间插入分隔符。分别处理这些序列, 并通过softmax层生成答案的概率分布。选择概率最高的答案作为输出。



GPT实验效果

推理任务:

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

问答和常识推理任务:

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	86.5	62.9	57.4	59.0

语义相似度和分类任务:

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	93.2	-	-	-	-
TF-KLD [23]	-	-	86.0	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	<u>81.0</u>	-	-
Single-task BiLSTM + ELMo + Attn [64]	<u>35.0</u>	90.2	80.2	55.5	<u>66.1</u>	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	<u>68.9</u>
Finetuned Transformer LM (ours)	45.4	91.3	82.3	82.0	70.3	72.8

总结：在多项任务上达到了最好结果；
 文本生成任务效果提升显著；分类任务效果较差

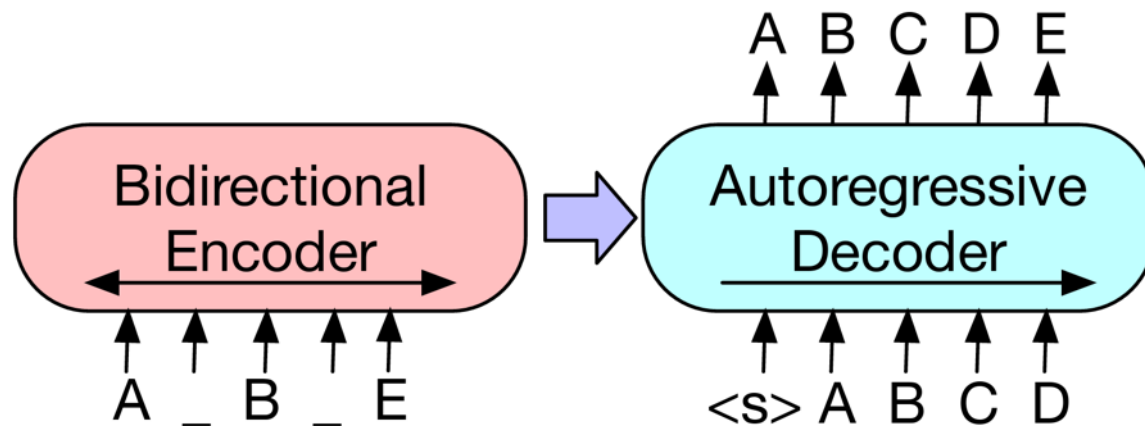


目 录

- 1 什么是预训练
- 2 预训练模型
 - 2.1 编码器模型
 - 2.2 解码器模型
 - 2.3 编解码模型
- 3

编解码模型BART

- BART (Bidirectional and Auto-Regressive Transformers) 是一种基于Encoder-Decoder结构的预训练模型，兼具双向语义理解能力和自回归生成能力

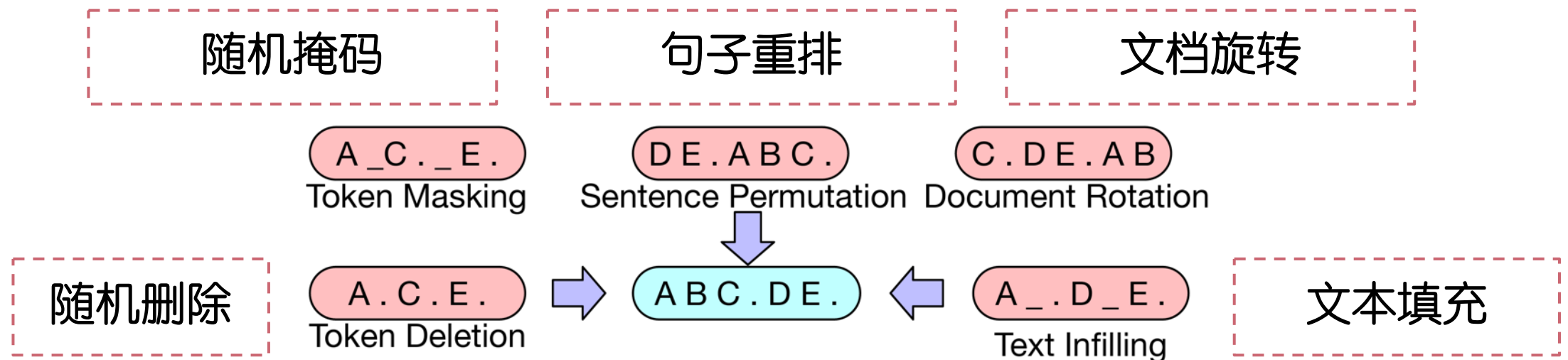


- 采用标准Transformer 模型，包括Encoder和Decoder部分
- 使用Decoder输出进行逐步预测，不需要额外的FFN

BART预训练

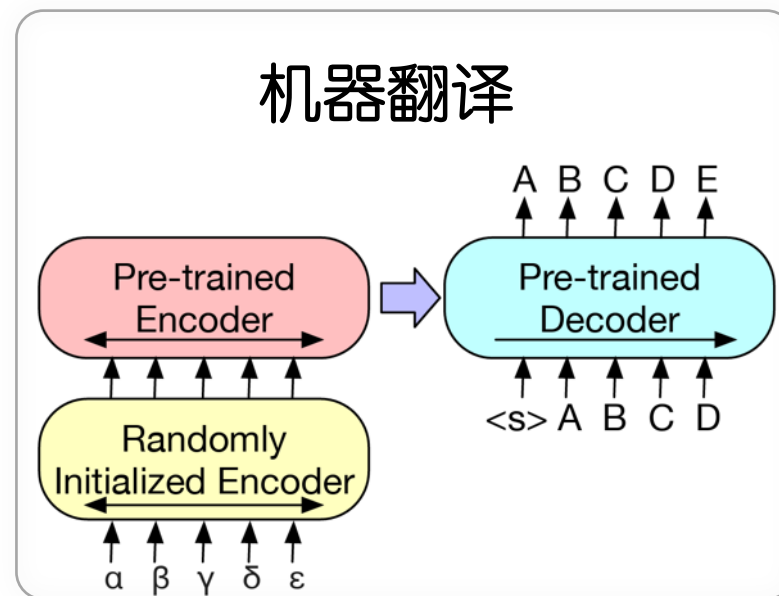
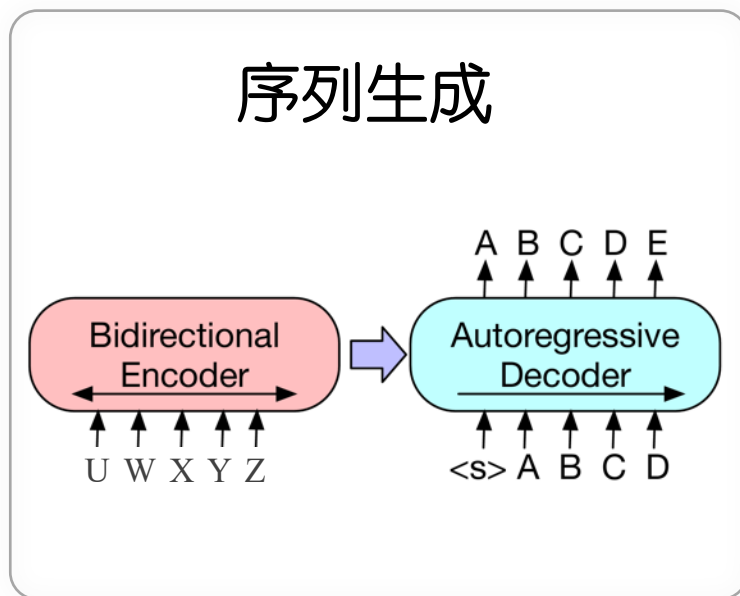
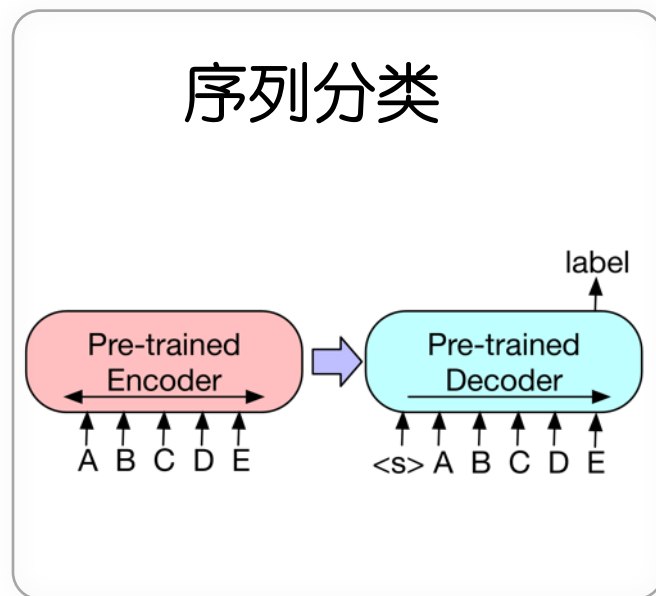
□ BART采用无监督去噪自编码（Denoising Autoencoder）策略，通过“破坏文本 + 重建原文”方式进行预训练

□ 目标函数： $P(x | \tilde{x}) = \prod_t P(x_t | x_{<t}, \tilde{x})$ x ：原始文本； \tilde{x} ：破坏文本



BART微调

- BART将不同 NLP 任务统一建模为“条件文本生成问题，在微调时通常只需保持原模型结构，根据任务替换或添加输出层



BART实验效果

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	84.3	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq Language Model	87.0	82.1	23.40	6.80	11.43	6.19
Permutated Language Model	76.7	80.1	21.40	7.00	11.51	6.56
Multitask Masked Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	90.8	84.0	24.26	6.61	11.05	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	90.8	83.8	24.17	6.62	11.12	5.41

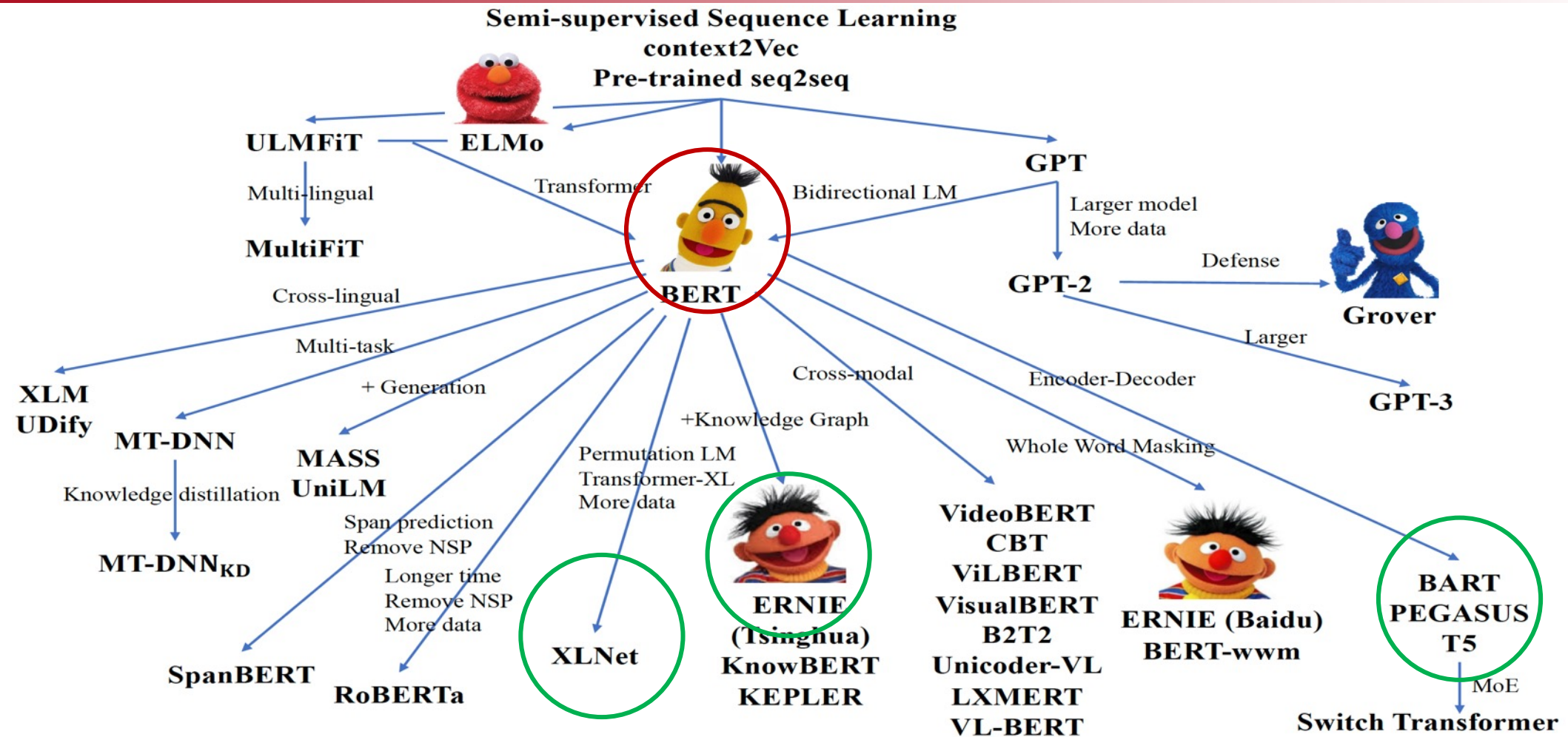
使用Text Infilling的效果非常好, 并且, 预训练的有效性高度取决于任务, 自回归式的模型有利于解决生成类任务



目 录

- 1 什么是预训练
- 2 预训练模型
 - 2.1 编码器模型
 - 2.2 解码器模型
 - 2.3 编解码模型
 - 2.4 扩展
- 3
- 4

预训练模型的扩展

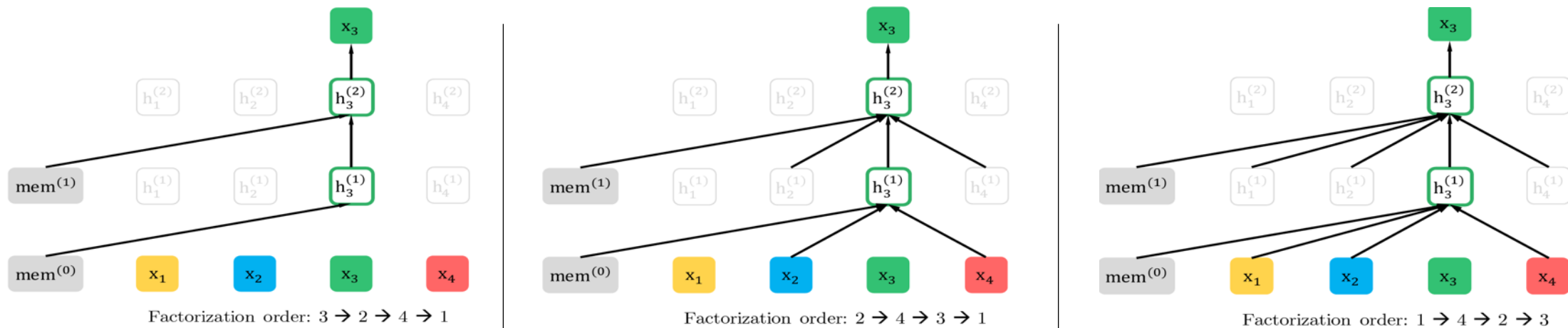


XLNet

□ XLNet提出排列语言建模（Permutation Language Modeling），结合自回归语言模型和自编码语言模型的优势

□ 对于长度T的句子，考虑T!个语言模型

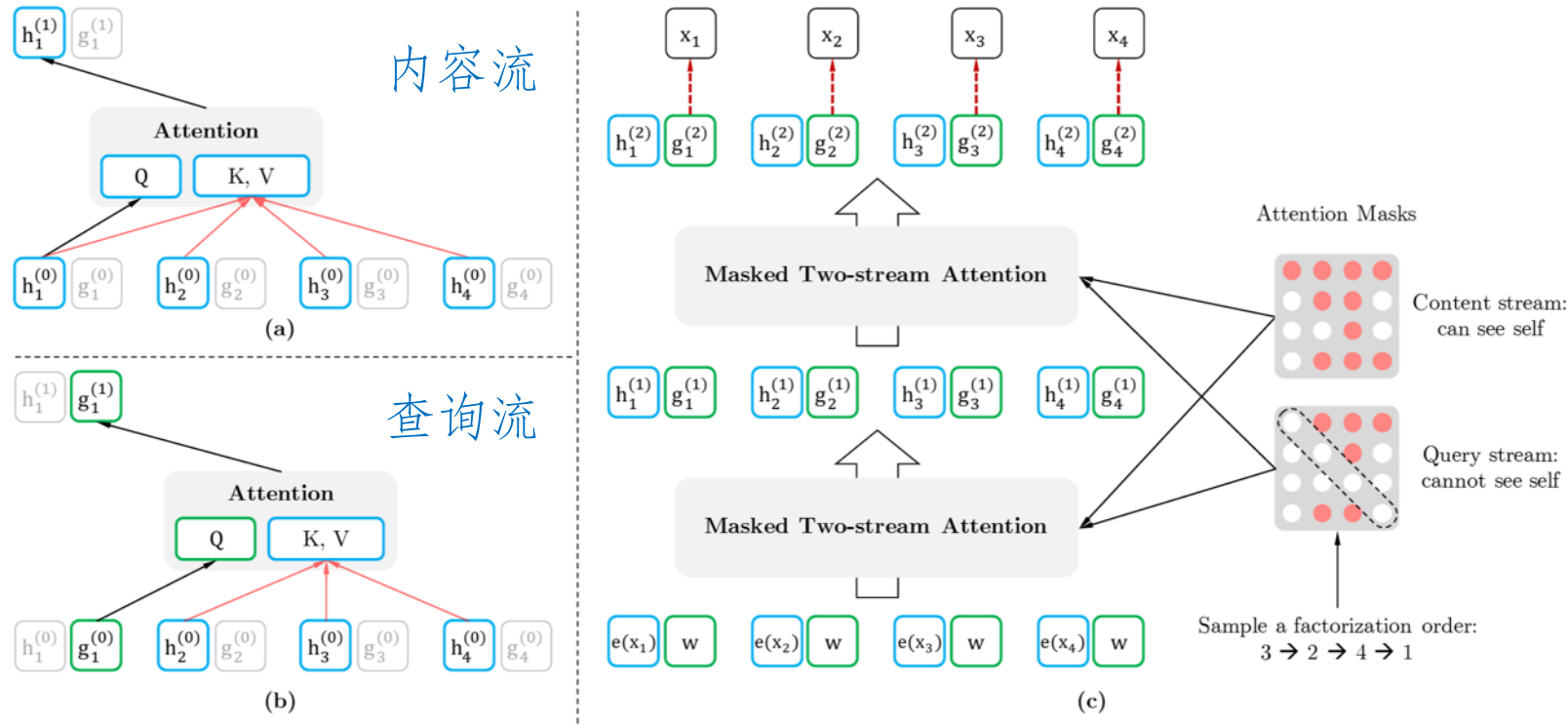
$$\max_{\theta} \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[\sum_{t=1}^T \log p_{\theta}(x_{z_t} | \mathbf{x}_{\mathbf{z}_{<t}}) \right]$$



给定输入序列x，在不同排列顺序时，预测x3的排序语言建模目标示意图

XLNet

模型结构：双流自注意力及MASK机制



内容流：
标准自注意力

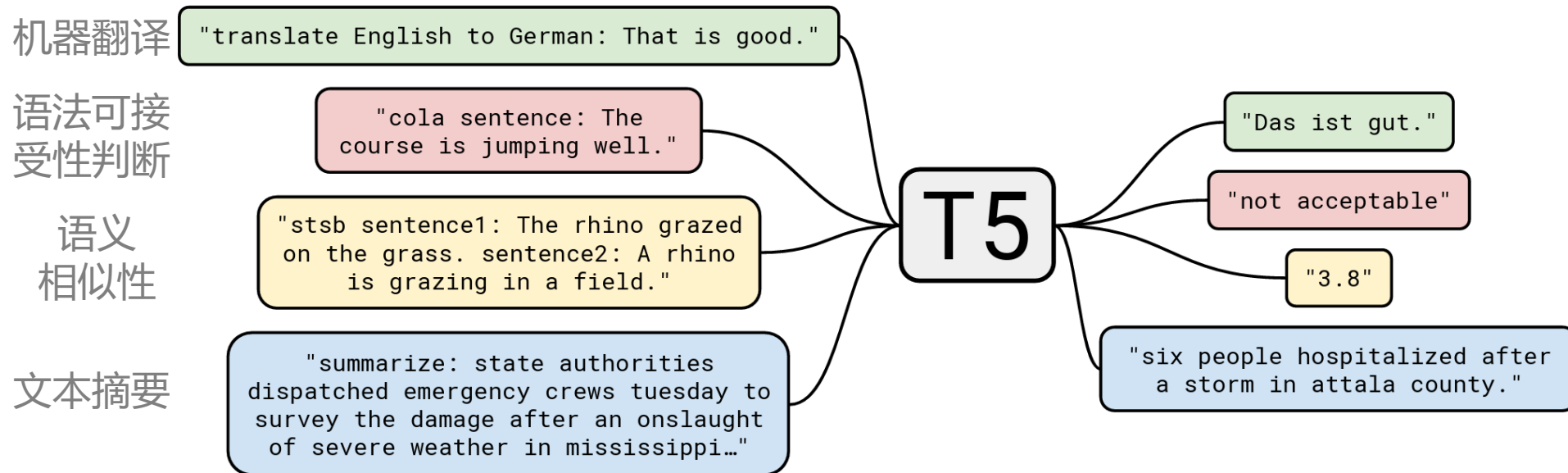
查询流：
只使用位置信息，不使用当前词内容信息

$$g_{z_t}^{(m)} \leftarrow \text{Attention}(Q = g_{z_t}^{(m-1)}, KV = \mathbf{h}_{z < t}^{(m-1)}; \theta), \quad (\text{query stream: use } z_t \text{ but cannot see } x_{z_t})$$

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = \mathbf{h}_{z \leq t}^{(m-1)}; \theta), \quad (\text{content stream: use both } z_t \text{ and } x_{z_t}).$$

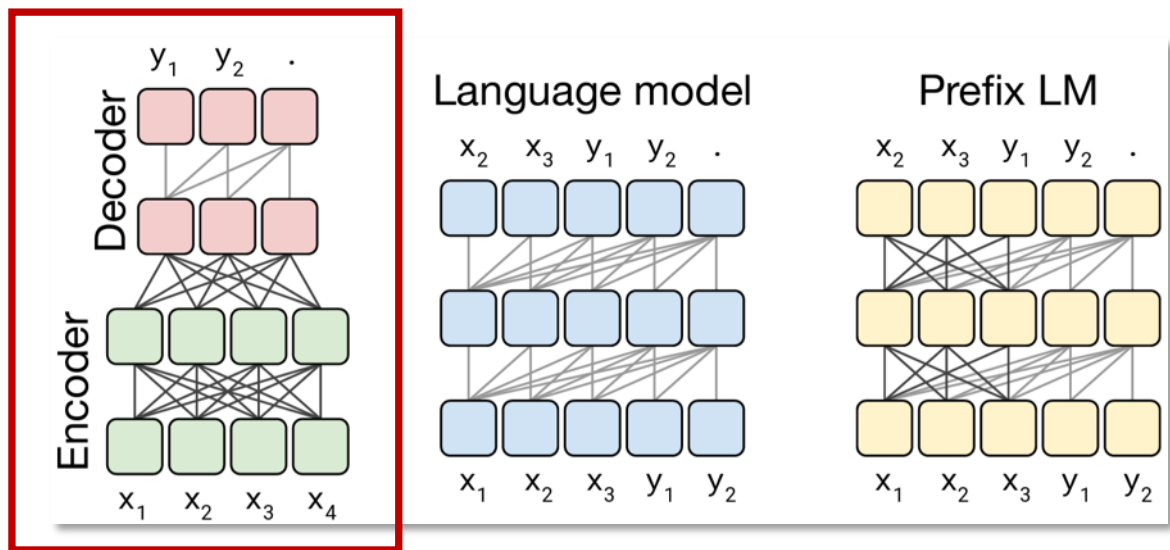
T5

- T5 (Text-to-Text Transfer Transformer) 提供了一个通用框架，把所有 NLP 任务统一转换为“文本到文本”问题



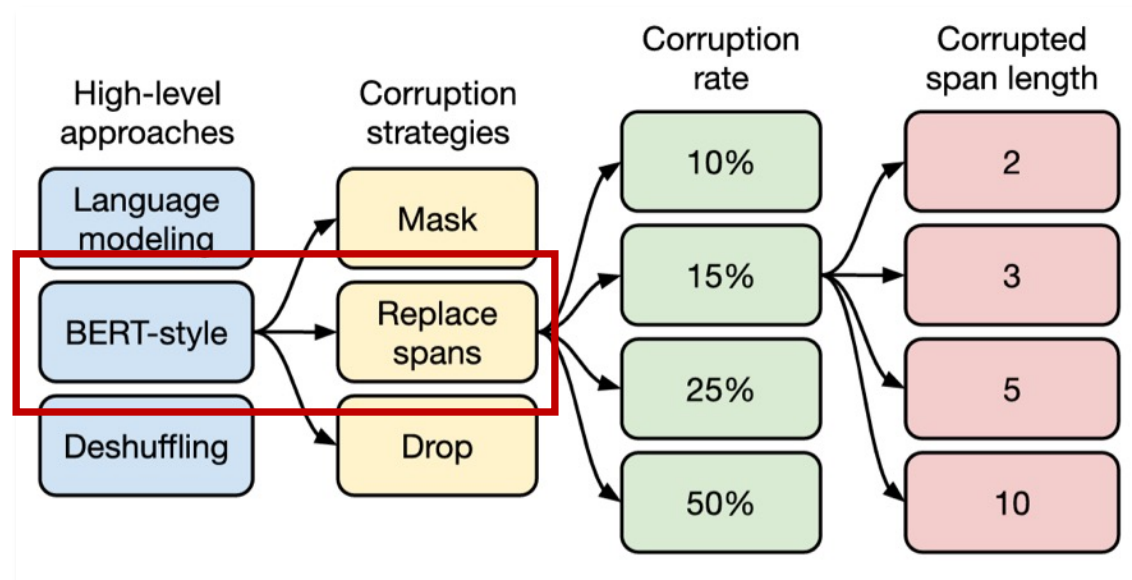
T5

预训练模型架构选择



模型选择：编码器-解码器

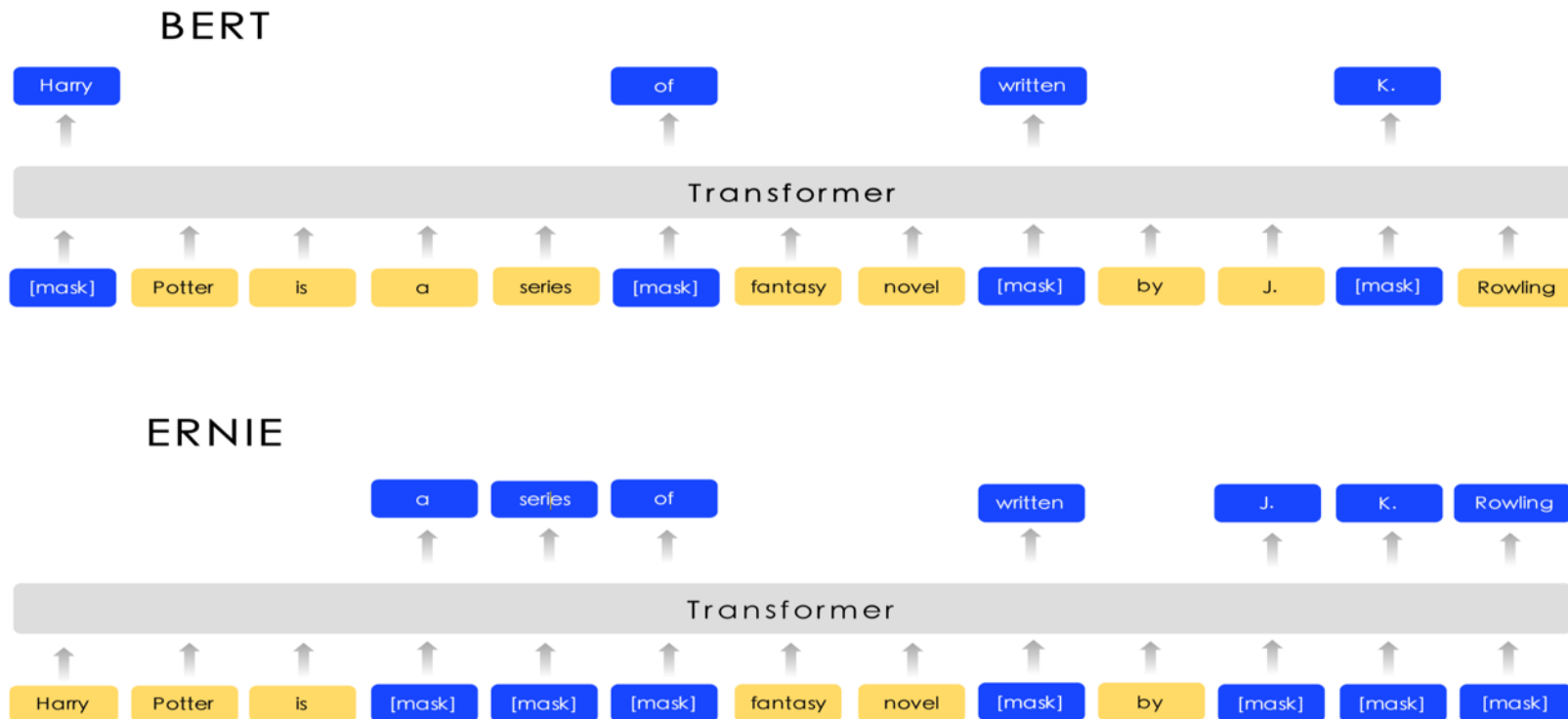
自监督目标



训练方式：BERT-style中片段掩码

ERNIE (Baidu)

- ERNIE是百度提出的知识增强预训练语言模型，旨在通过引入结构化知识和语义信息提升预训练模型的代表能力



ERNIE 1.0

□ 两种新的 masking 策略

phrase-level masking: 短语类如a series of, written等

entity-level masking: 人名, 位置, 组织, 产品等名词

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

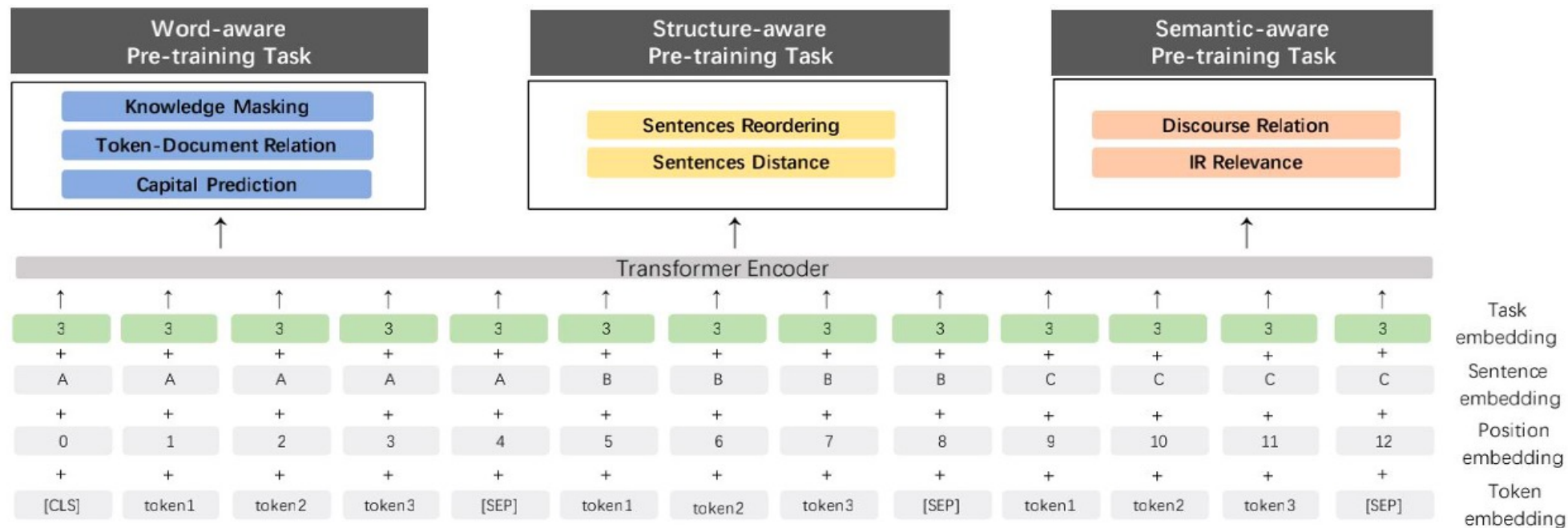
Figure 2: Different masking level of a sentence

ERNIE 2.0

□ 预训练连续学习(Continual Learning)

用大量的数据与先验知识构建不同的预训练任务

用多个预训练任务顺序更新ERNIE 模型

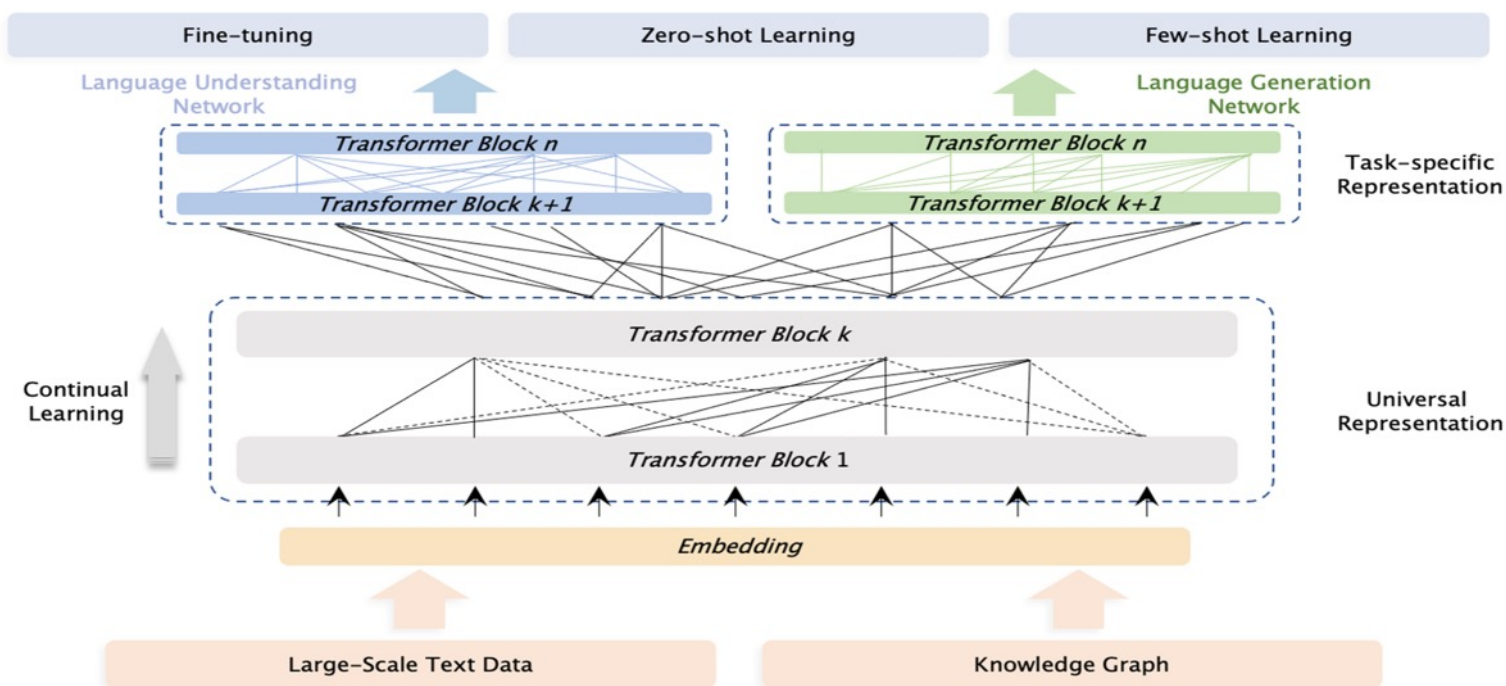


ERNIE 3.0

□ 知识增强预训练模型

知识增强预训练，在文本语料和结构化知识上进行预训练

支持多任务、多模态、多语言，实现统一的理解和生成框架





目 录

1

概述

2

预训练方法

3

大语言模型

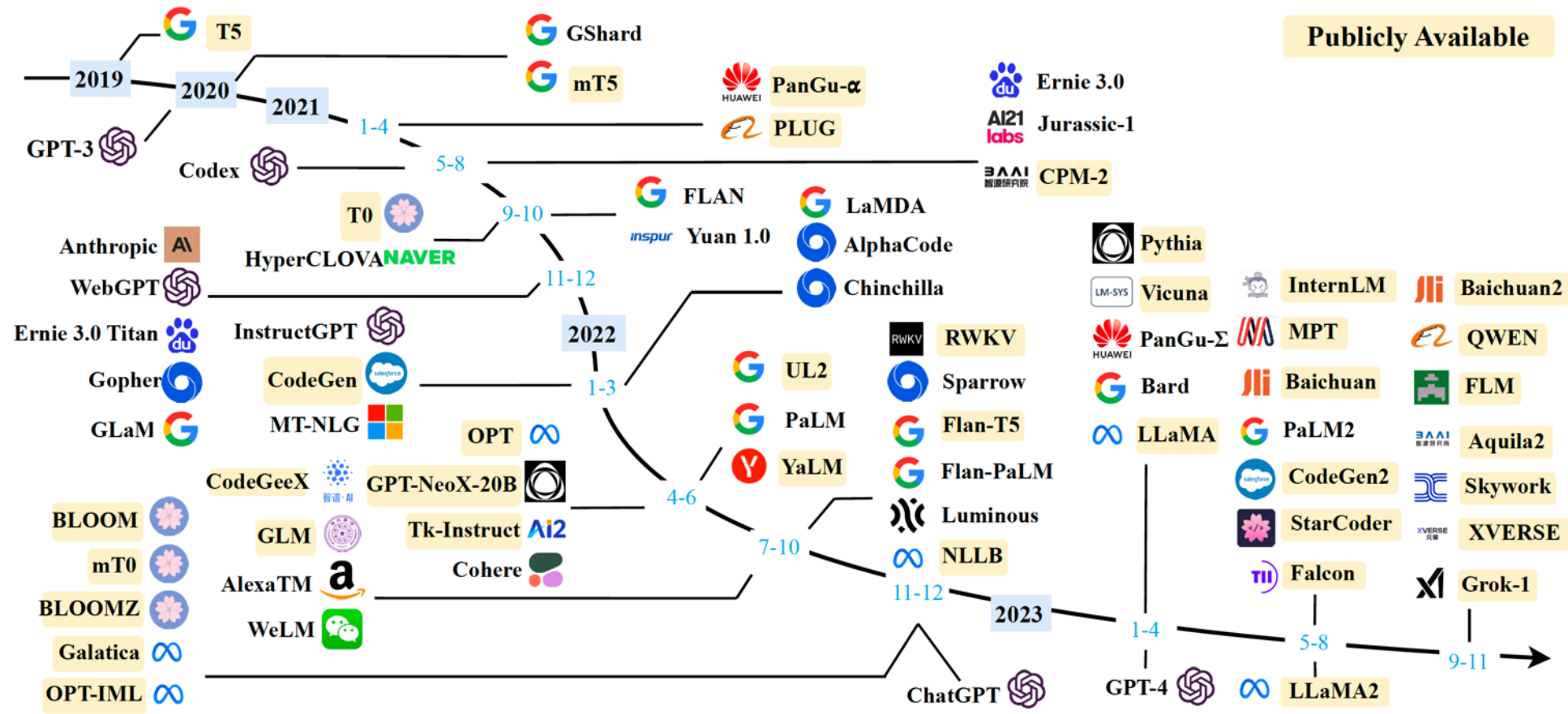
大语言模型

- 大语言模型（LLMs）是指在**海量数据上进行预训练、参数规模庞大，并具备强大语言理解与生成能力**的语言模型



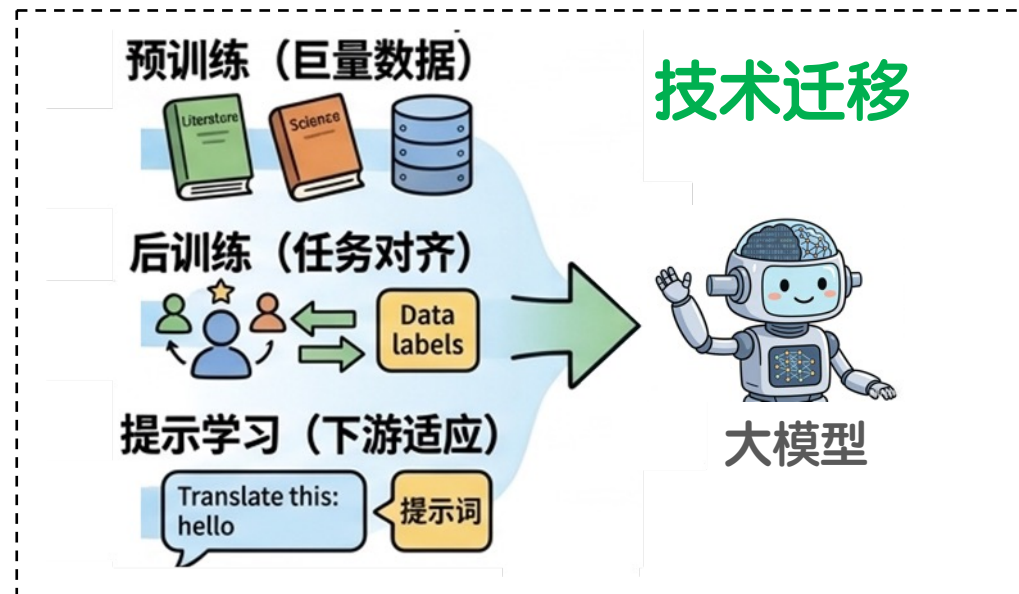
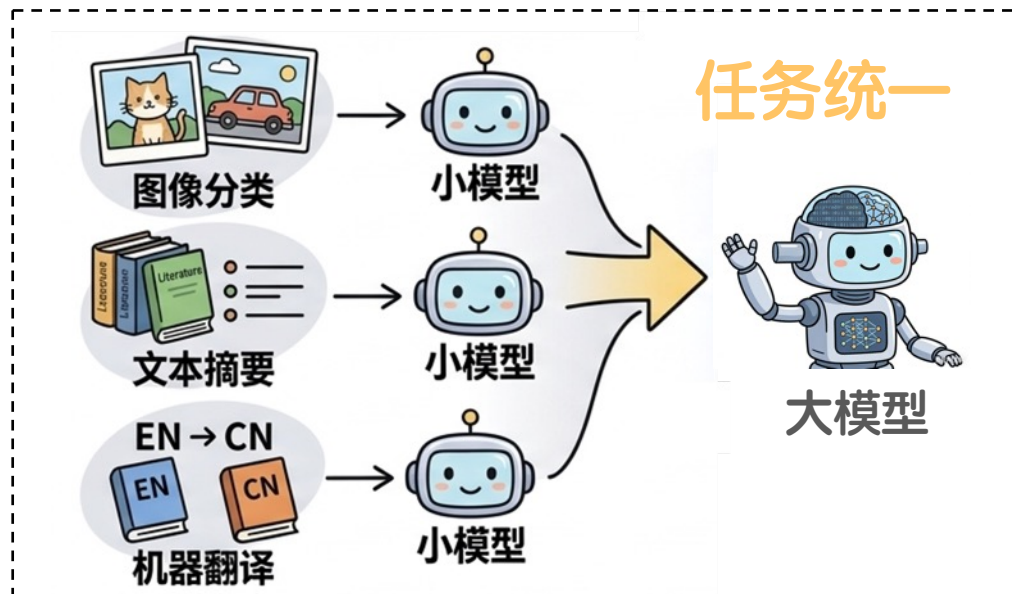
大模型发展历程

□ 近年来，以 ChatGPT 为代表的大模型快速发展，逐步迈向通用人工智能



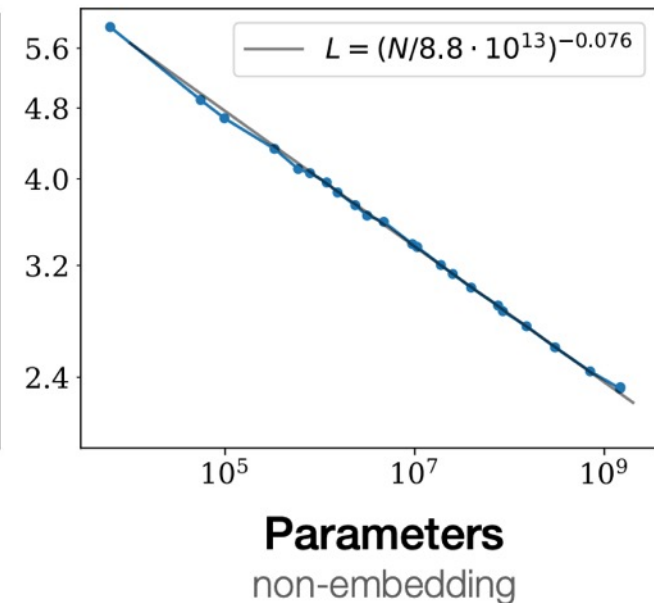
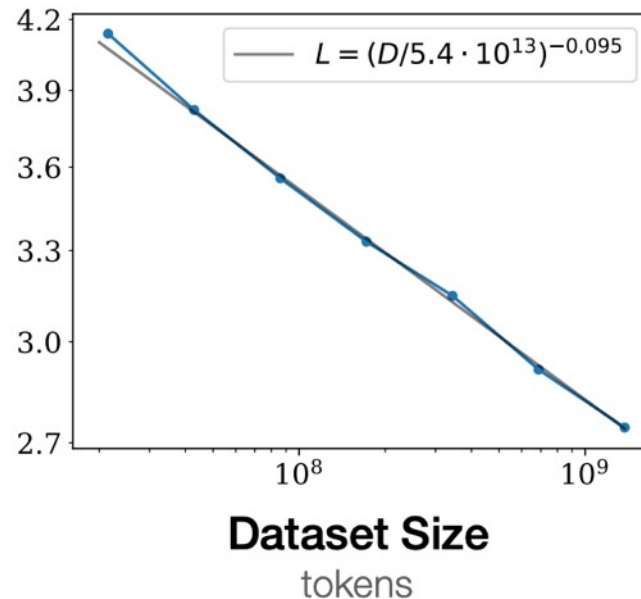
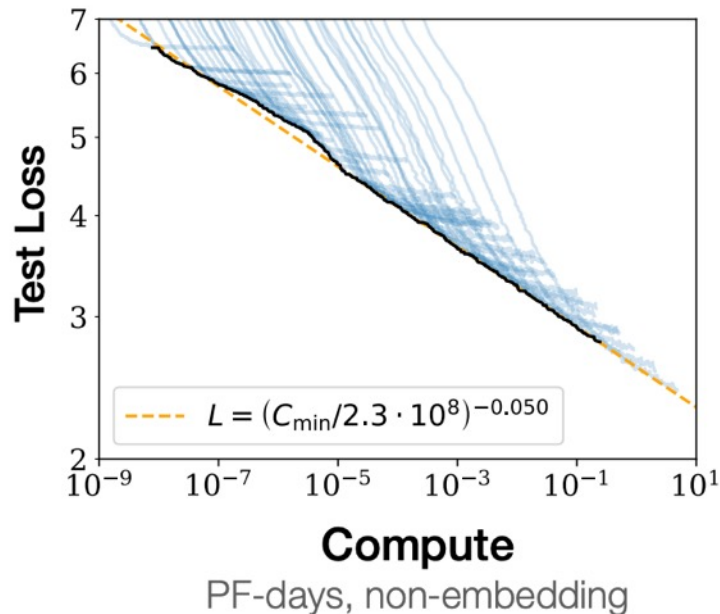
大模型范式变化

- 任务范式的统一：由独立任务转向统一序列建模框架
- 技术范式的迁移：由单一任务的专门训练，转向以“预训练 + 后训练”和“预训练 + 提示学习”



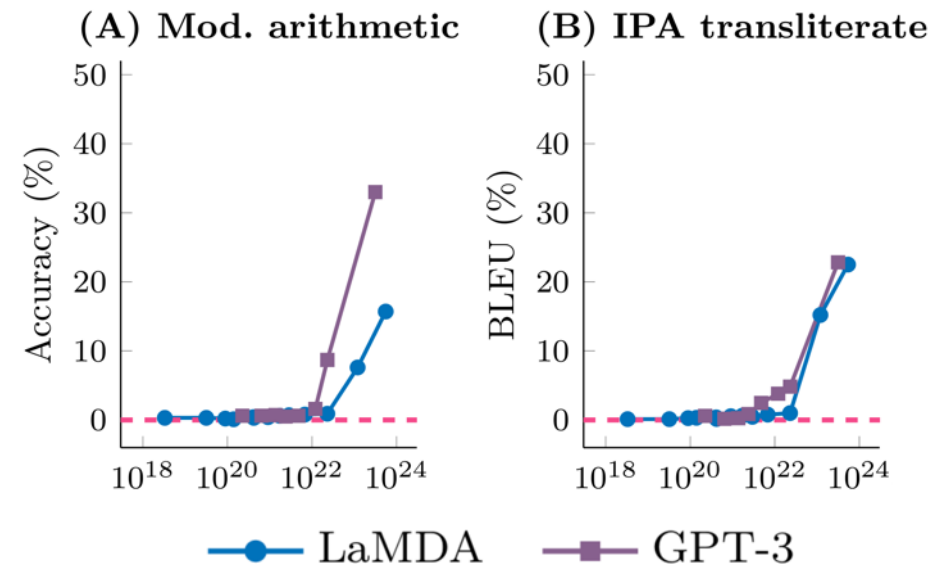
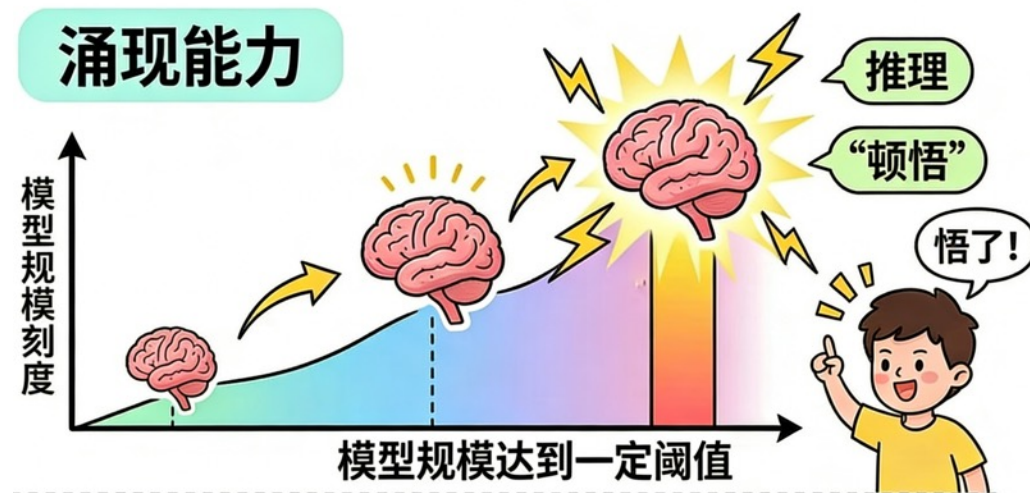
大模型扩展定律 (Scaling Law)

- 模型性能与模型大小、数据规模和算力三者之间的关系
- 随着模型参数和预训练数据规模的增加，模型性能持续提升；
Chinchilla建议参数与数据规模约为1:20 (如10B模型: 200B tokens)



大模型涌现能力 (Emergent Abilities)

- 当语言模型参数达到一定规模时，某些能力（如语言推理）会突然显著提升，在小模型中未观察这些能力



大模型三大阶段

- 大模型训练范式通常包括三个阶段：预训练、指令微调和对齐

阶段1: 预训练 Pre-training



从千亿级文本与代码中学习通用知识，构建基础模型

阶段2: 指令微调 SFT



使用数万条标注指令进行训练，使模型学会精准遵循人类指令

阶段2: 对齐 Alignment



通过百万级对比数据让模型生成符合人类价值观的内容

核心阶段1: 预训练

- 预训练是在大规模无标注语料上学习通用语言表示与知识能力



- 数据层面：依赖**海量语料**，规模通常达到十亿到万亿级



- 算力层面：在**大规模分布式集群**上进行月级训练



- 效果层面：使得模型具备**记忆、理解、推理和生成能力**

核心阶段1: 预训练

- 数据规模通常达到万亿级，在**大规模分布式集群**上进行训练
 - 数据
 - 算力

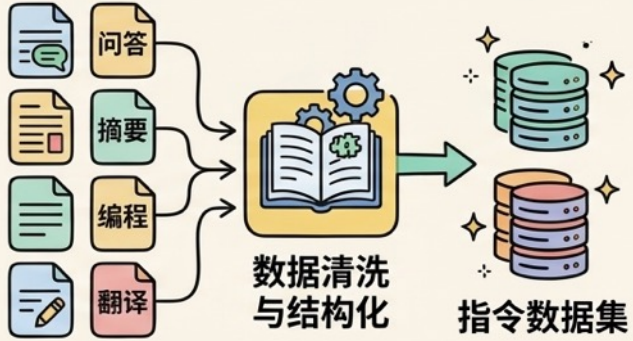
Model	Release Time	Size (B)	Base Model	Adaptation		Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time
				IT	RLHF				
UL2 [89]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-
OPT [90]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-
NLLB [91]	Jul-2022	54.5	-	-	-	-	-	-	-
CodeGeeX [92]	Sep-2022	13	-	-	-	850B tokens	-	1536 Ascend 910	60 d
GLM [93]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d
Flan-T5 [69]	Oct-2022	11	T5	✓	-	-	-	-	-
BLOOM [78]	Nov-2022	176	-	-	-	366B tokens	-	384 80G A100	105 d
mT0 [94]	Nov-2022	13	mT5	✓	-	-	-	-	-
Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-
BLOOMZ [94]	Nov-2022	176	BLOOM	✓	-	-	-	-	-
Publicly Available OPT-IML [95]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-
LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d
Pythia [96]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-
CodeGen2 [97]	May-2023	16	-	-	-	400B tokens	-	-	-
StarCoder [98]	May-2023	15.5	-	-	-	1T tokens	-	512 40G A100	-
LLaMA2 [99]	Jul-2023	70	-	✓	✓	2T tokens	-	2000 80G A100	-
Baichuan2 [100]	Sep-2023	13	-	✓	✓	2.6T tokens	-	1024 A800	-
QWEN [101]	Sep-2023	14	-	✓	✓	3T tokens	-	-	-
FLM [102]	Sep-2023	101	-	✓	-	311B tokens	-	192 A800	22 d
Skywork [103]	Oct-2023	13	-	-	-	3.2T tokens	-	512 80G A800	-

核心阶段2: 指令微调

- 指令微调通过高质量“指令-响应”数据对模型进行监督学习，使模型学会“听懂并执行人类指令”

大模型指令数据构造

定义：为大语言模型生成多样化、高质量的指令与回答对，包含不同的任务类型。



问答
摘要
编程
翻译

数据清洗与结构化

指令数据集

大模型全量微调 (FFT)

方法：更新预训练大模型的所有权重参数，使其对目标领域或任务有深度的适应。

-  全量模型权重更新
-  极高计算资源需求
-  全面适应特定任务
-  训练成本和硬件门槛极高

高效微调 (PEFT)

方法：仅冻结基座模型权重，通过引入少量可训练参数 (如LoRA、Prompt) 来适应新任务。

-  基座模型权重冻结
-  引入少量额外模块 (LoRA, P-Tuning)
-  仅调整少量参数
-  训练速度快，成本低

核心阶段3: 对齐

□ 大模型对齐 (Alignment) 逐渐成为人工智能领域的核心议题，并受到政府和监管机构的高度重视



中华人民共和国国家发展和改革委员会
National Development and Reform Commission

人工智能安全治理的国际经验和启示

发布时间: 2025/12/03

来源: 习近平经济思想研究中心

 [打印]

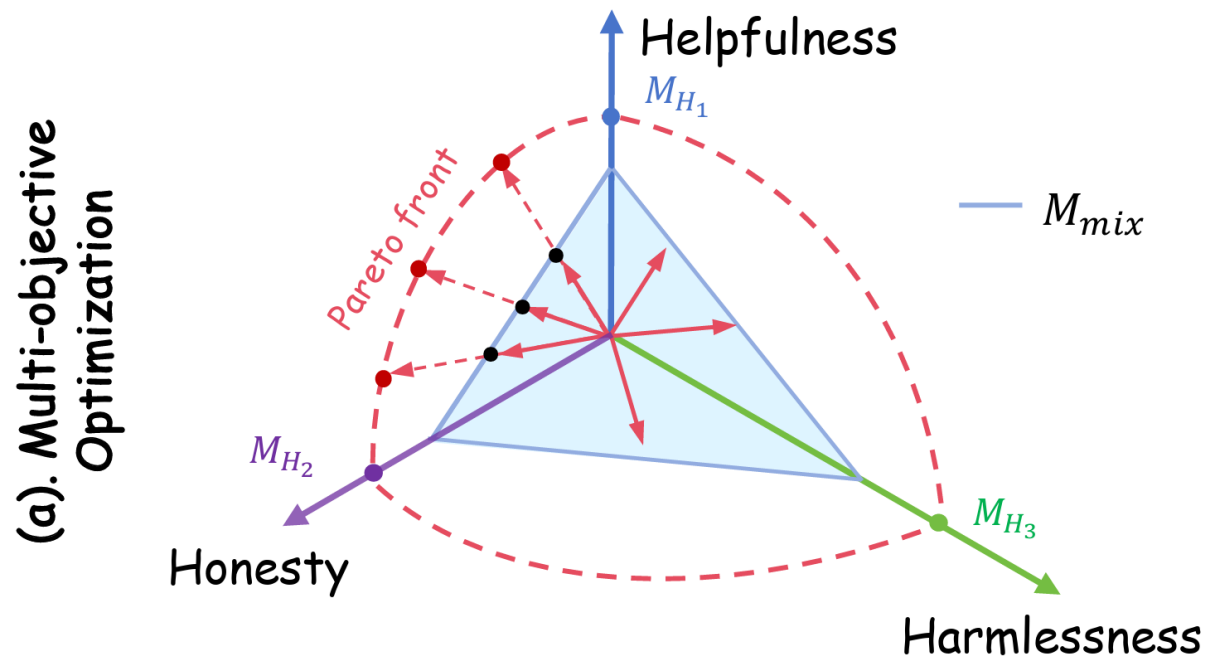
习近平总书记强调，要坚持促进发展和依法管理相统一，既大力培育人工智能、物联网、下一代通信网络等新技术新应用，又积极利用法律法规和标准规范引导新技术应用。在新的科技浪潮下，人工智能成为引领新一轮科技革命和产业变革的“现象级”驱动力，在推动生产力转型升级的同时，也伴随着技术使用和社会安全的新型风险挑战。为把人工智能打造成安全可靠、风险可控、造福人类的新型生产工具，国际组织、政府部门、社会机构和相关企业不断挖掘敏捷有效的监管方式，以人工智能安全治理保障人类基本权益，实现更加安全地享受人工智能技术发展红利。人工智能安全治理关系着全球科技和经济的发展，对人类共同命运产生深远影响，正在成为世界各国共同面临的时代课题。

一、主要国家、经济体和国际组织人工智能安全治理举措

为应对人工智能技术催生的数据隐私、算法偏见、伦理冲突及国家安全威胁等多维度安全风险，全球主要国家、经济体和国际组织构建了适应国情的治理框架，形成了六大安全治理举措。

核心阶段3: 对齐

- 对齐旨在使模型输出符合人类偏好、价值观和使用规范，包括有用性 (helpfulness)、安全性 (safety) 和一致性 (harmlessness)

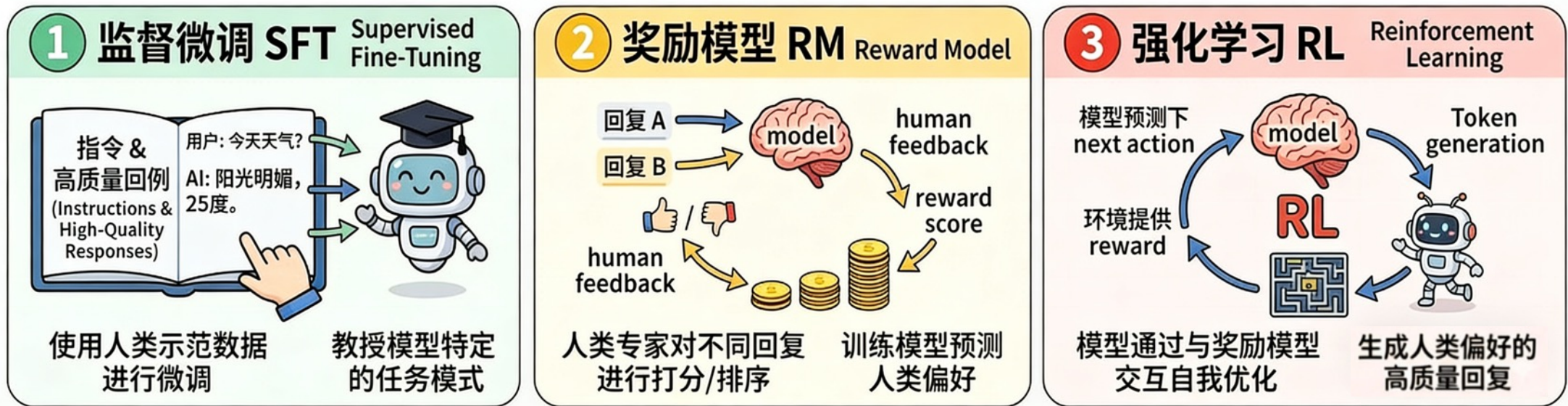


多目标优化问题

Pareto Optimality
帕累托最优解中的一个
折中点

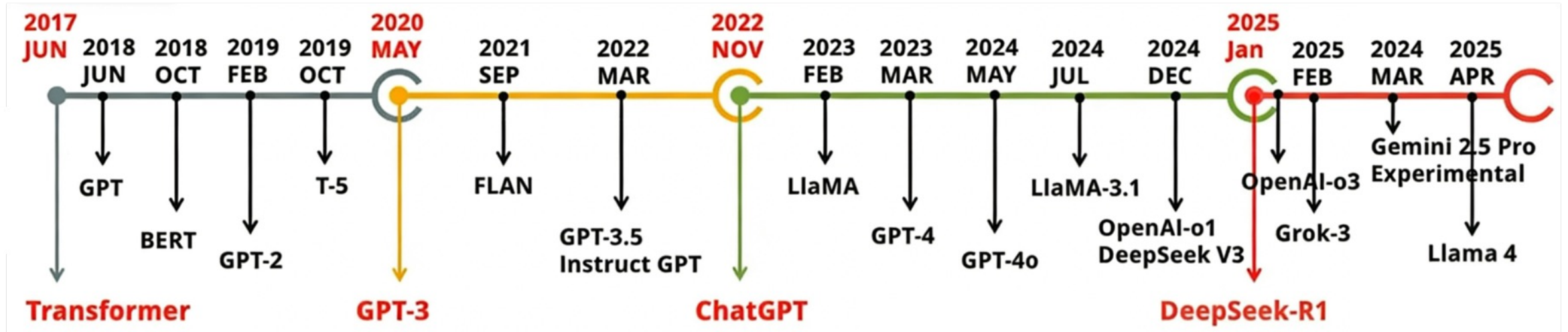
核心阶段3: 对齐

□ 目前，常见的对齐技术是RLHF（Reinforcement Learning from Human Feedback），通常分为三个核心步骤：



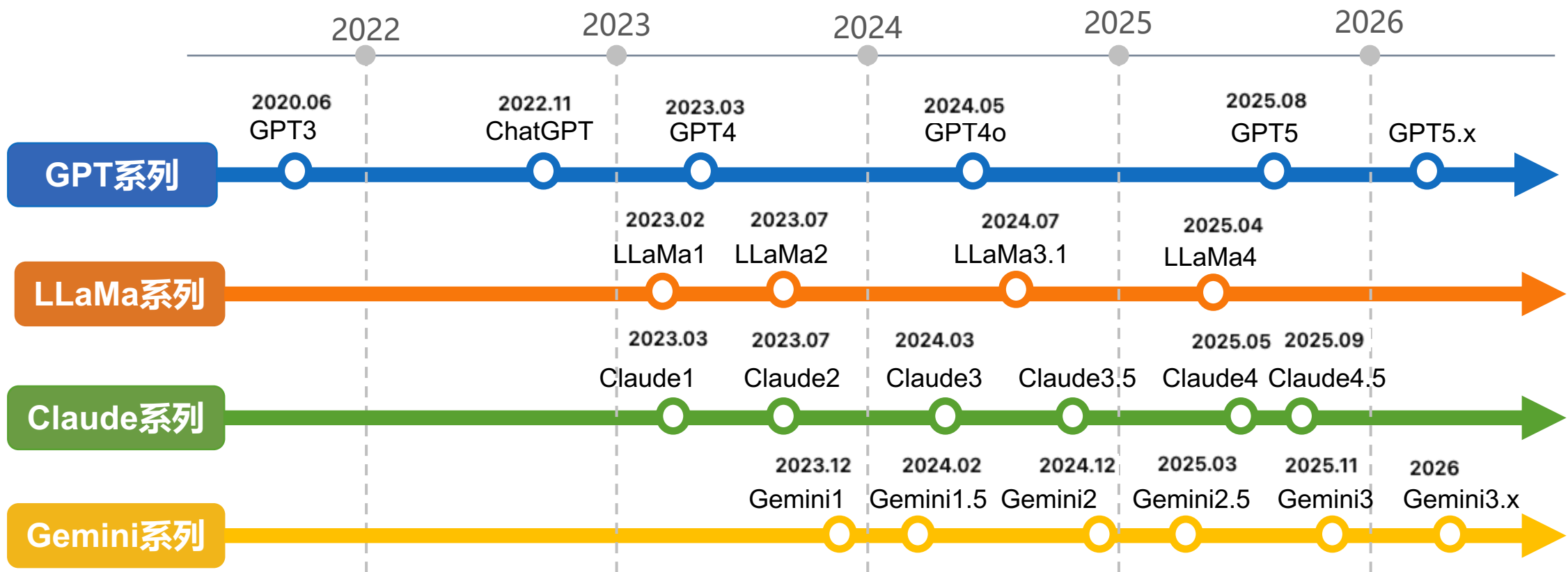
大模型主要代表

- 自2017年Transformer问世以来，从预训练模型到大模型的快速发展，在人工智能领域掀起了新的技术浪潮



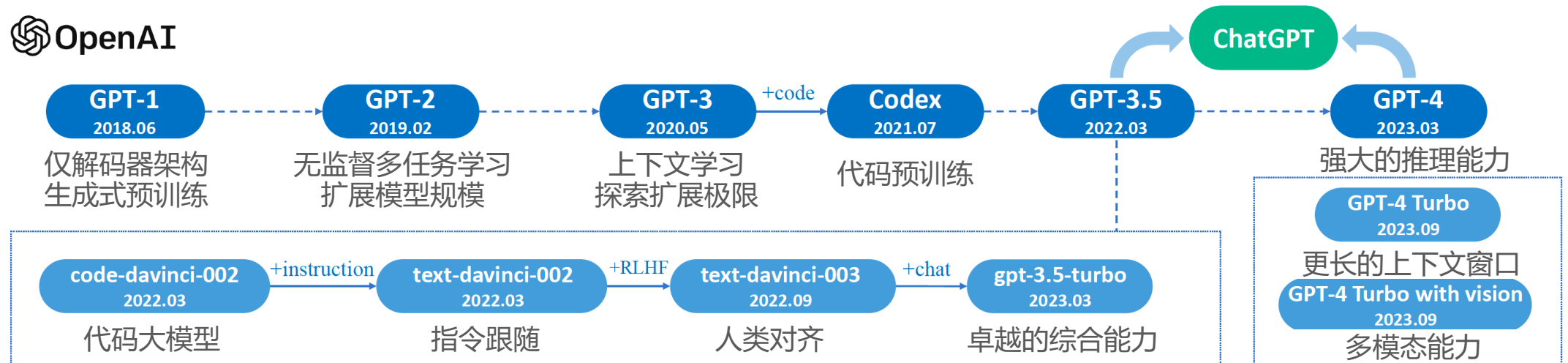
国外大模型发展

□ GPT、LLaMA、Claude 和 Gemini 等系列大模型发展时间线



GPT系列大模型

□ GPT 系列是由 OpenAI 提出的生成式语言模型，代表了当前大语言模型发展的重要方向



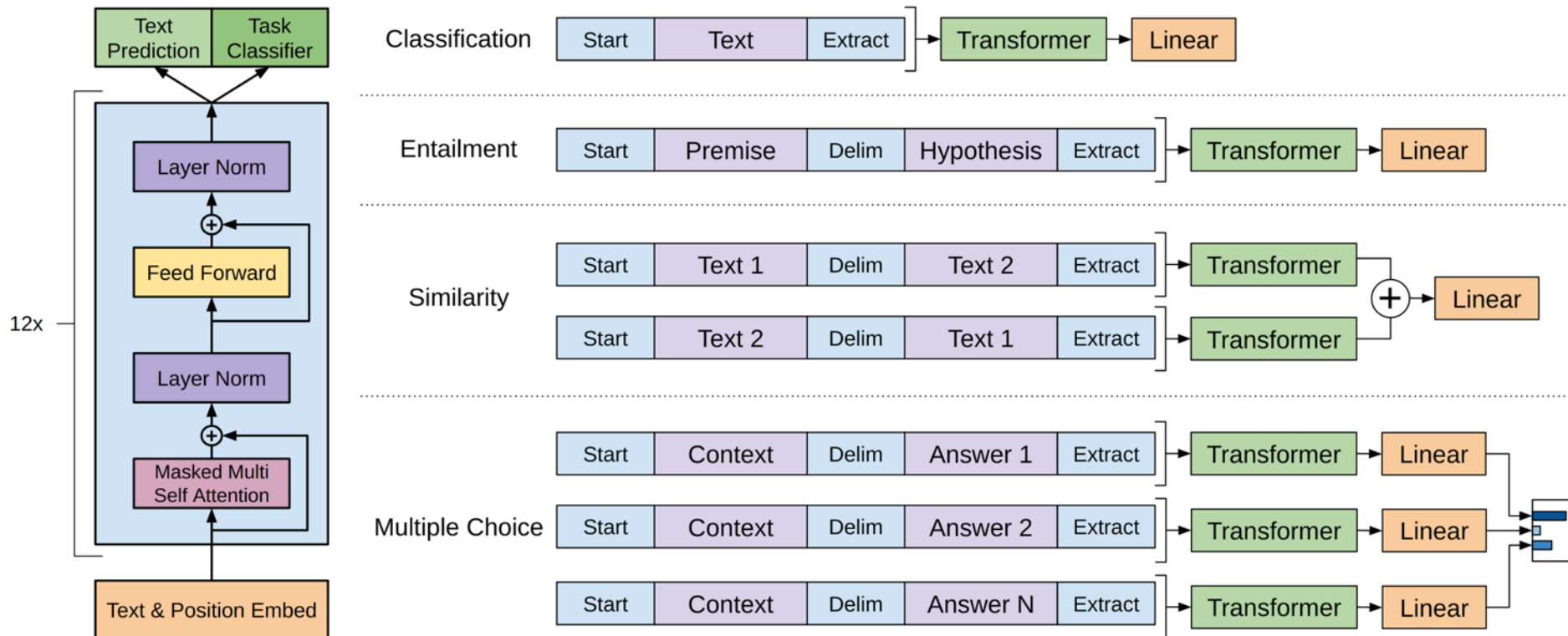
GPT系列大模型

□ GPT 系列解析：训练数据、模型参数、核心架构以及关键能力

模型版本	发布时间	参数量	核心架构	关键能力突破
GPT-1	2018.06	1.17亿	Decoder-only	开创“预训练+微调”范式，验证潜力
GPT-2	2019.02	15亿	大规模Transformer	验证“规模即能力”，展现零样本学习
GPT-3	2020.05	1750亿	超大规模Transformer	涌现“上下文学习”，开启API商业化
ChatGPT	2022.11	~2000亿	RLHF + 对话微调	流畅多轮对话，引爆全球AI应用热潮
GPT-4	2023.03	~1.8万亿	MoE 混合专家架构	引入多模态能力，逻辑推理大幅提升
GPT-4o	2024.05	未公开	原生多模态统一架构	统一处理图文音，实现实时交互

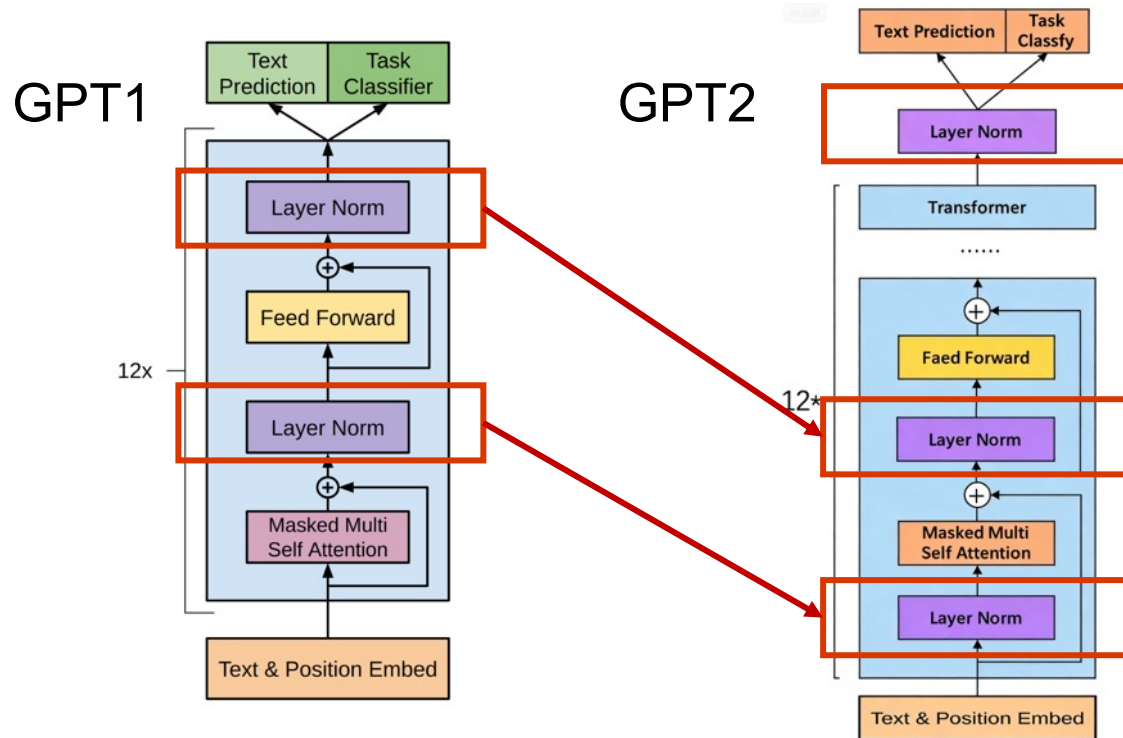
GPT-1

- Pre-training: 基于语言模型在BooksCorpus进行预训练
- Fine-tuning: 有监督微调, 看作NLU任务, 不是NLG



GPT-2

- Pre-training: 数据集更大WebText, 参数规模更大110M→1.5B
- Fine-tuning: 放弃微调, 利用zero-shot learning, NLU → NLG



针对下游的任务, 无需对输入结构进行专门设计, 只需通过指示词来引导其完成任务

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



GPT-3

- 训练范式: Pre-training + Prompt (few-shot learning)
- 上下文学习: zero-shot、one-shot、few-shot

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

GPT-1、2、3比较

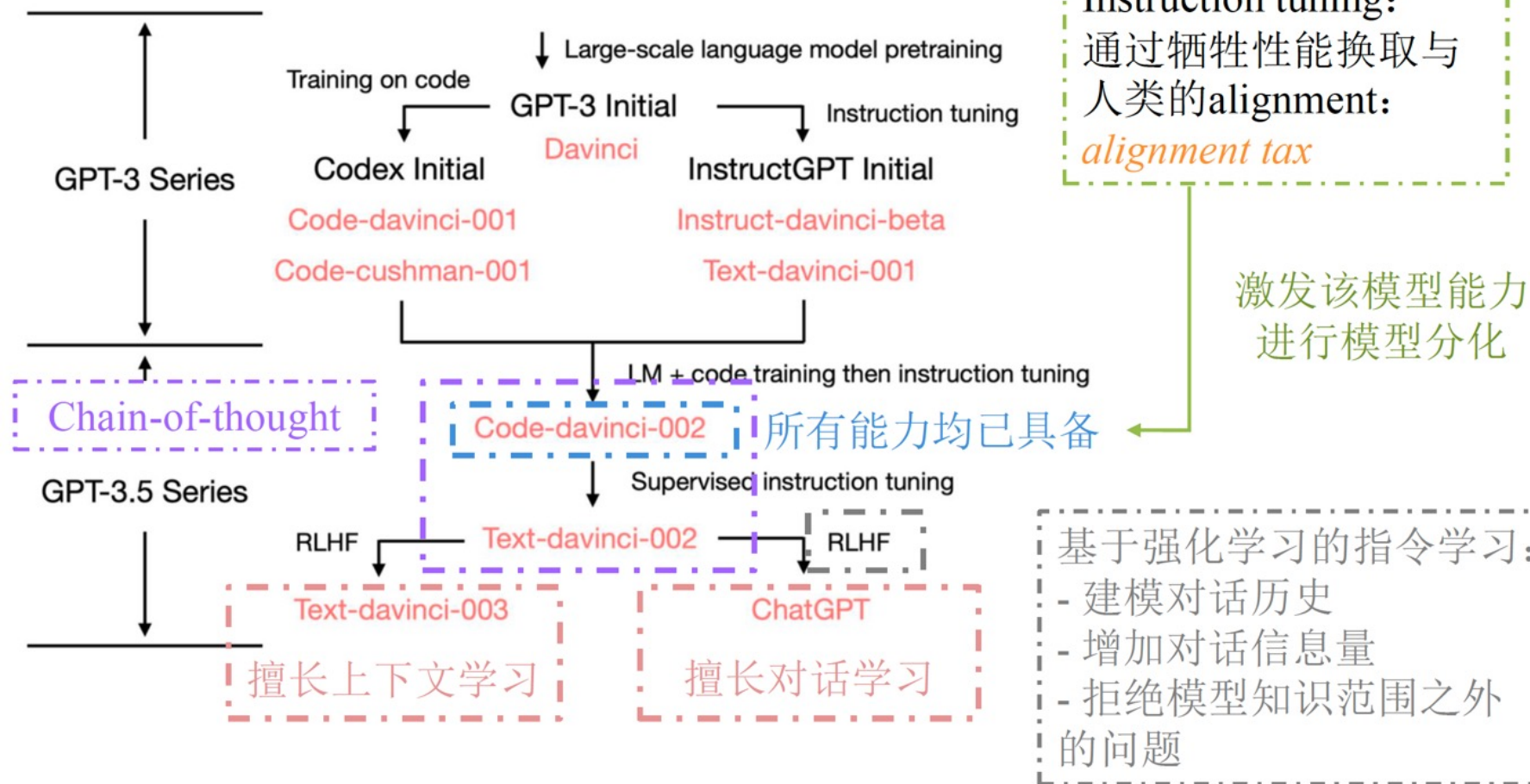
□ 模型比较

	GPT-1	GPT-2	GPT-3
Vocabulary	40478	50257	50257
Context size	512	1024	2048
Batchsize	64	512	320万
Parameters	1.17亿	15亿	1750亿
Layers	12	48	96
Dimensional states	768	1600	12288
Pre-training dataset	BooksCorpus	WebText (Origin)	Common Crawl WebText2 等
Data size	5GB	40GB	45TB
Downstream task	NLU	NLG	NLU+NLG
Downstream transfer	Fine-tuning	Zero-shot + prompt	Prompt

- 海量数据集基座，奠定模型理解与生成基础
- 千亿级参数规模，开启大模型时代
- 上下文学习，涌现能力初现

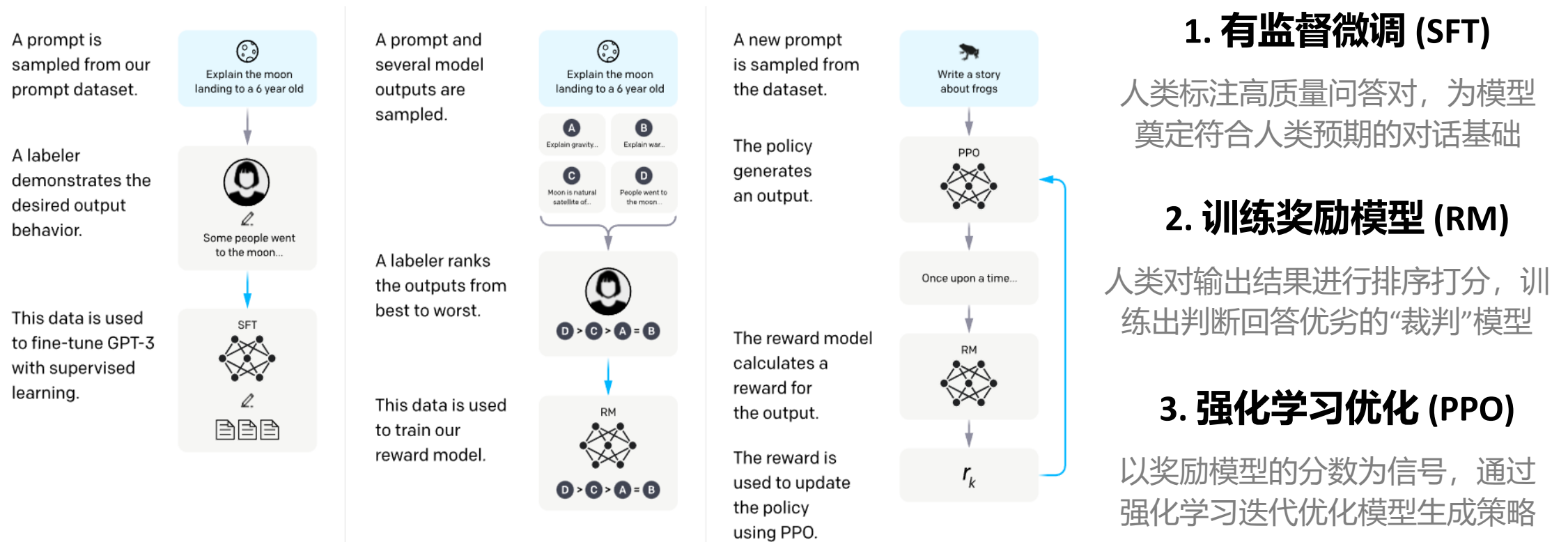
GPT-3系列

□ 模型演化



ChatGPT (GPT-3.5)

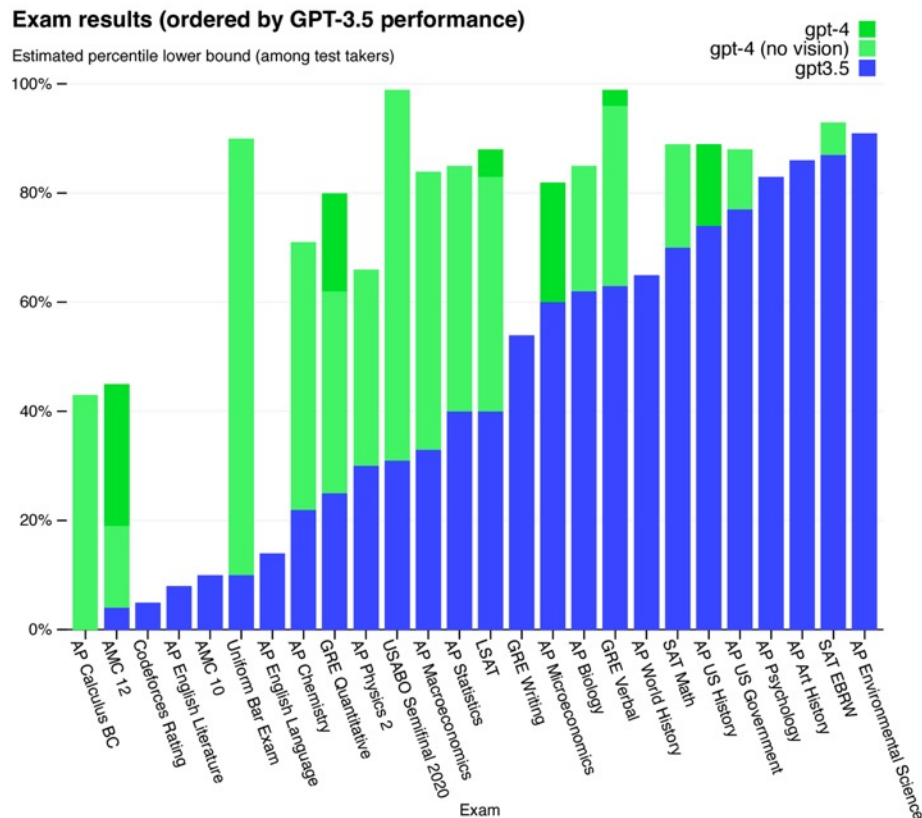
通过RLHF让模型行为与人类价值观精准对齐



ChatGPT引爆全球AI热潮，成为首个真正的大众级AI应用

GPT-4&GPT-4o

□ 多模态与实时交互的飞跃：从感知到理解的进化

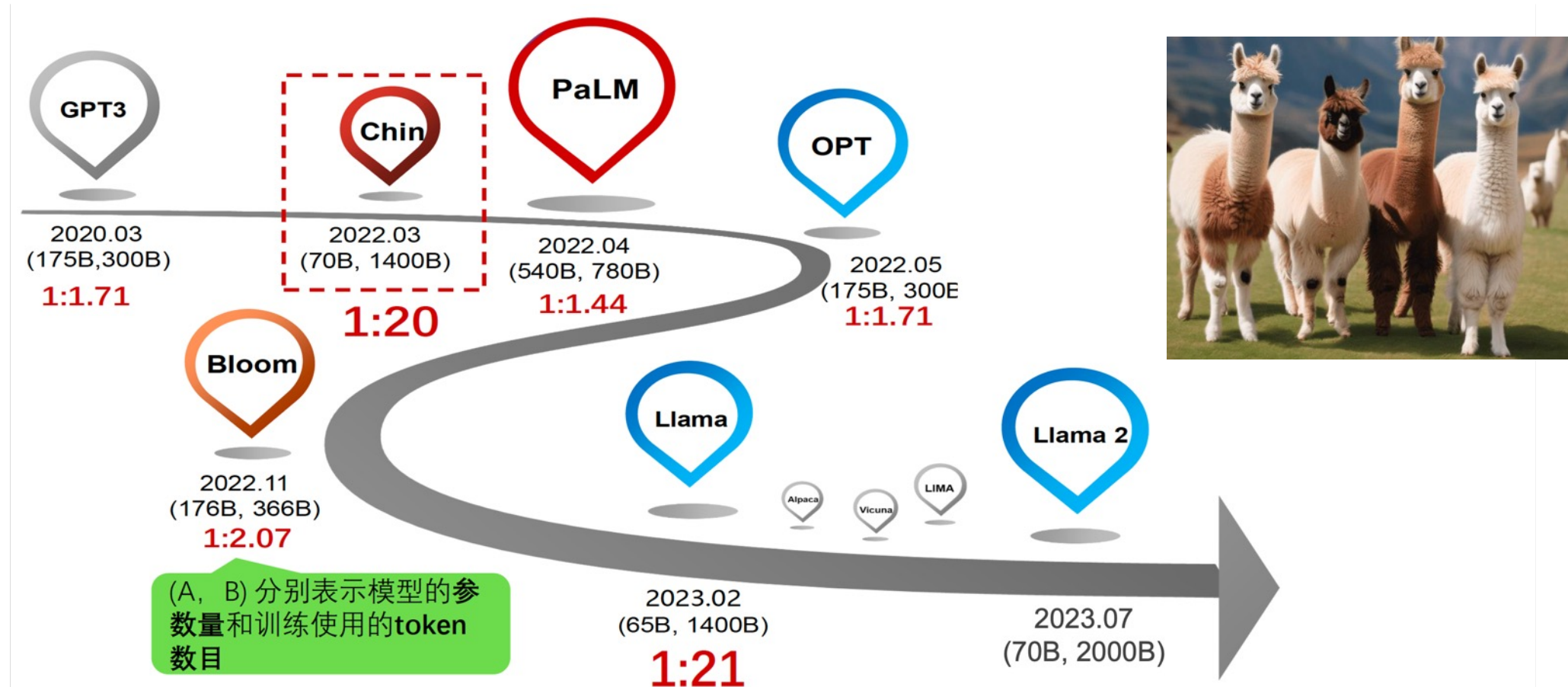


□ **GPT-4**: 突破文本限制，可理解图像与图表；采用MoE混合专家架构降低成本；在律师资格考试等专业领域表现卓越。

□ **GPT-4o**: 原生端到端多模态处理，音频响应延迟极低；实现“边说边看”的自然对话，交互体验接近真人。

LLaMa系列大模型

□ 从GPT到LLaMa系列，大模型向**开源、高效和可微调**方向演进



LLaMa系列大模型

□ LLaMA系列发展总览： 四代模型的跨越式演进



LLaMA 1

2023年2月

开启阶段

促进开源社区，确立了开源基座

- 架构: Decoder-only
- 能力: 7B-65B参数, 2K上下文, 引入RMSNorm等核心优化



LLaMA 2

2023年7月

商用普及

确立开源工业标准, 免费商用, 推动产业落地

- 架构: 引入分组查询注意力GQA
- 能力: 7B-70B参数, 4K上下文, RLHF对齐技术



LLaMA 3

2024年4月

巅峰时刻

性能逼近GPT-4, 成为开源模型的性能标杆

- 架构: 全系GQA, Dense架构极致优化
- 能力: 8B/70B参数, 128K上下文, 15T训练数据



LLaMA 4

2025年4月

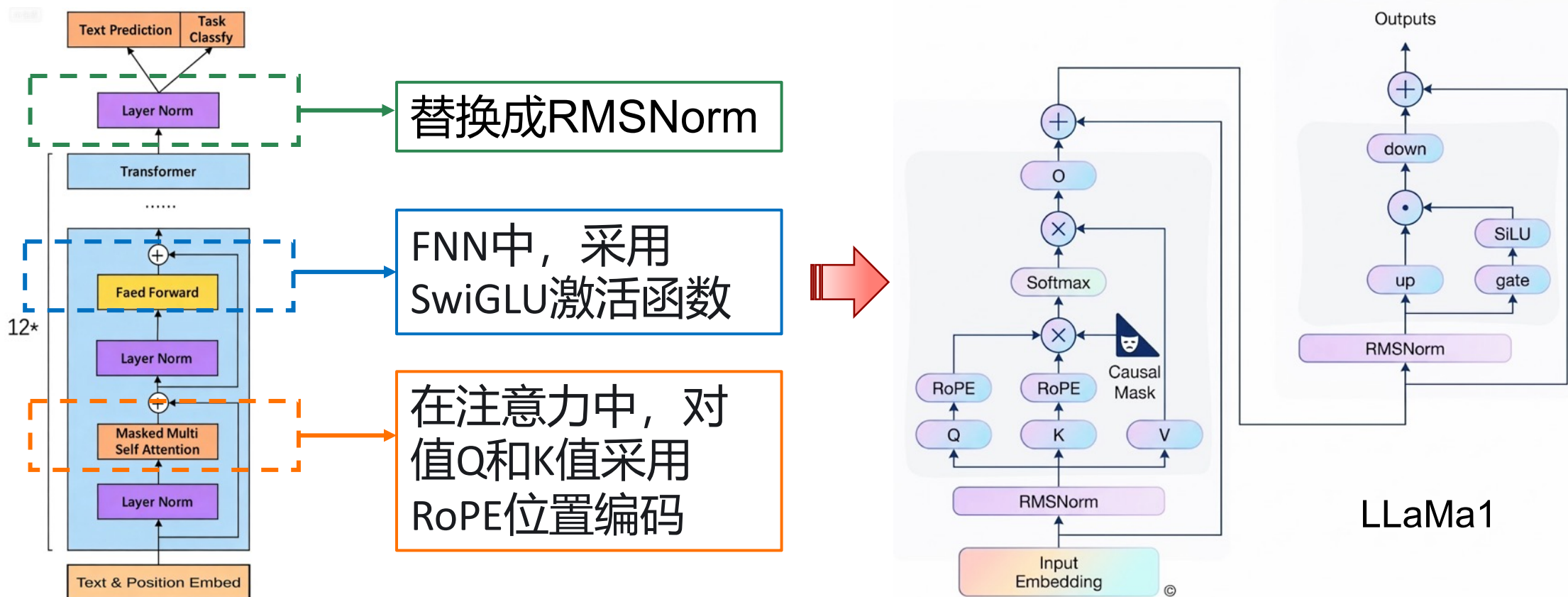
Agent & MoE

架构革新, 迈向原生多模态与智能体时代

- 架构: 混合专家模型
- 能力: 千万级上下文, 原生多模态, 强化Agent规划能力

LLaMa1

□ LLaMa1在GPT基础架构上，通过细节优化实现性能跃升



LLaMa2

□ Llama2结构基本不变，在34B/70B版本采用分组查询注意力GQA

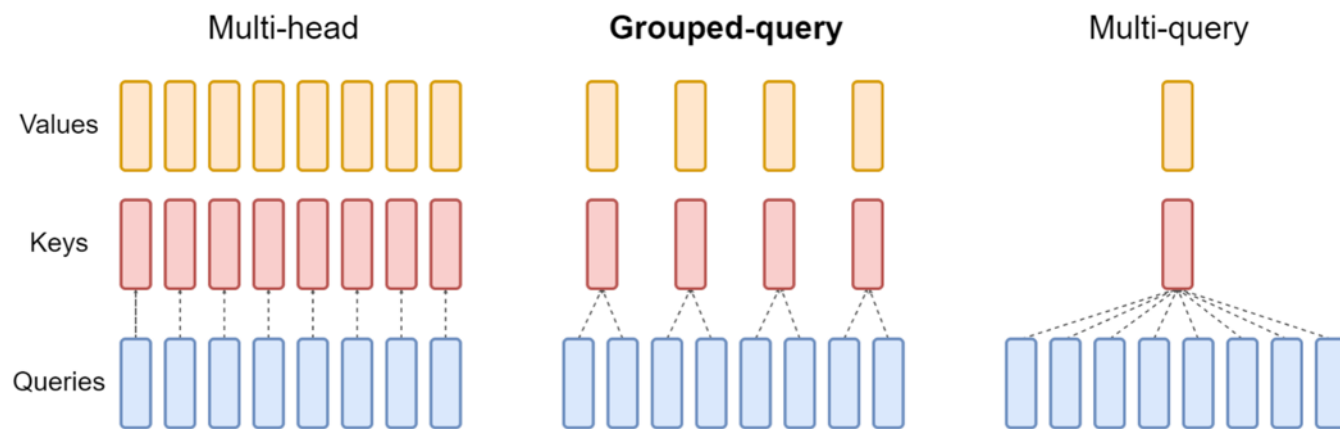


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between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

- **Multi-Head Attention (MHA)**
每个头独立维护 KV，缓存开销较大
- **Multi-Query Attention (MQA)**
所有头共享同一组 KV，大幅降低缓存成本
- **Group Query Attention (GQA)**
在多头间进行分组，组内共享 KV，在效果接近 MHA 的同时，兼顾了接近 MQA 的速度与效率

LLaMa3

□ LLaMa3首次具备闭源模型抗衡实力，采用四大核心技术：



15万亿 Token

训练数据量暴涨7.5倍，覆盖海量知识与语言模式，奠定模型智慧基石。



128K 词表

词表扩展4倍，大幅提升多语言、代码及复杂内容的编码与理解效率。



全系 GQA 技术

全参数版本标配分组查询注意力，优化内存占用，实现推理速度的飞跃。

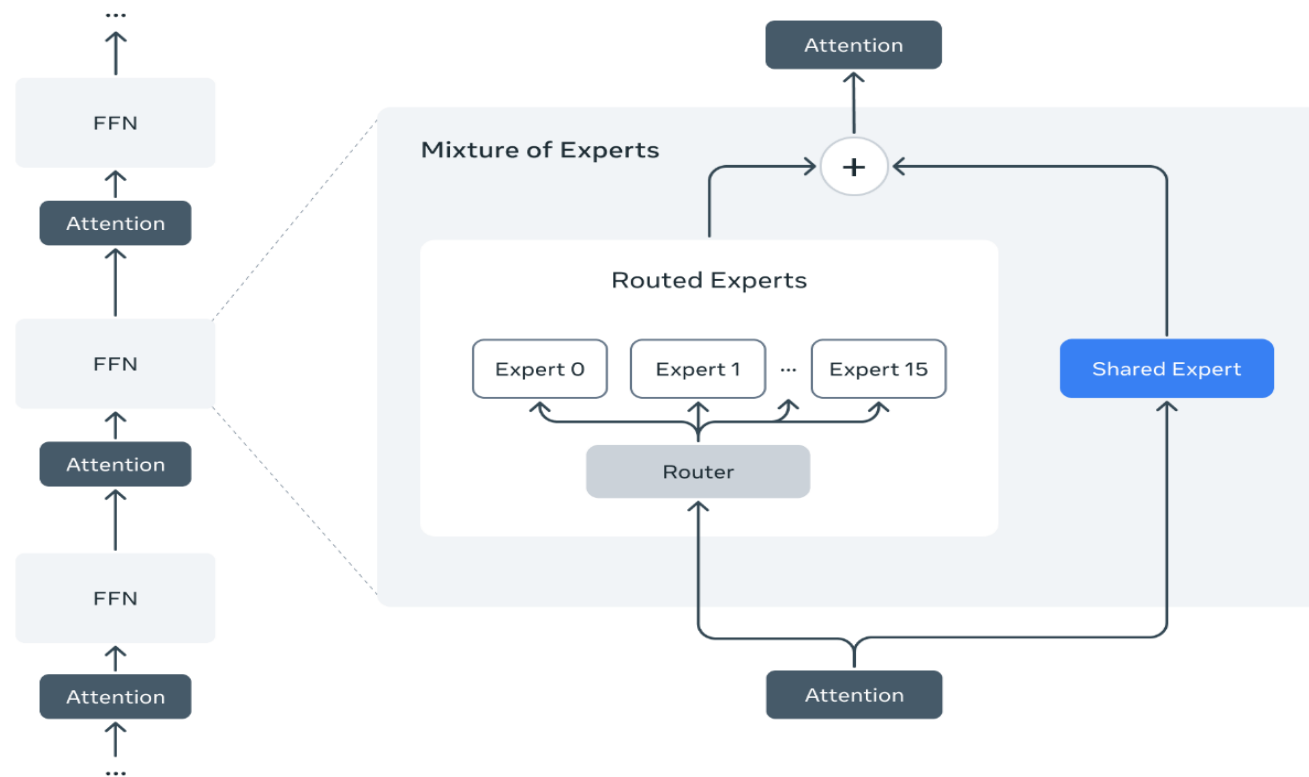


128K 上下文

窗口扩展至128K，彻底突破长文本处理瓶颈，从容应对整书级文档理解。

LLaMa4

- LLaMa4标志模型在多模态能力和计算效率方面进一步提升
- 主要亮点：多模态能力；核心架构变革，即混合专家模型 (MoE)



Claude系列大模型

- Claude是由Anthropic推出的一系列通用大语言模型，强调**安全性与对齐**、支持超长上下文、面向企业级应用

Claude 1

2023年初

奠定安全基石

确立“宪法式AI”核心地位，验证安全优先理念，筑牢AI发展根基

Claude 3

2024年初

多模态家族

引入Opus/Sonnet/Haiku模型矩阵，首次实现强大的视觉理解与分析能力

Claude 2

2023年中

迈向实用化

扩大上下文窗口至200k，开放API，具备处理真实世界复杂任务的能力

Claude 4

2025年末

智能体时代

通过Agentic能力，使AI从被动交互工具进化为主动的任务规划与执行者

Gemini系列大模型

□ Gemini由 Google DeepMind推出的新一代**多模态大模型**，强调跨模态理解与工具调用能力



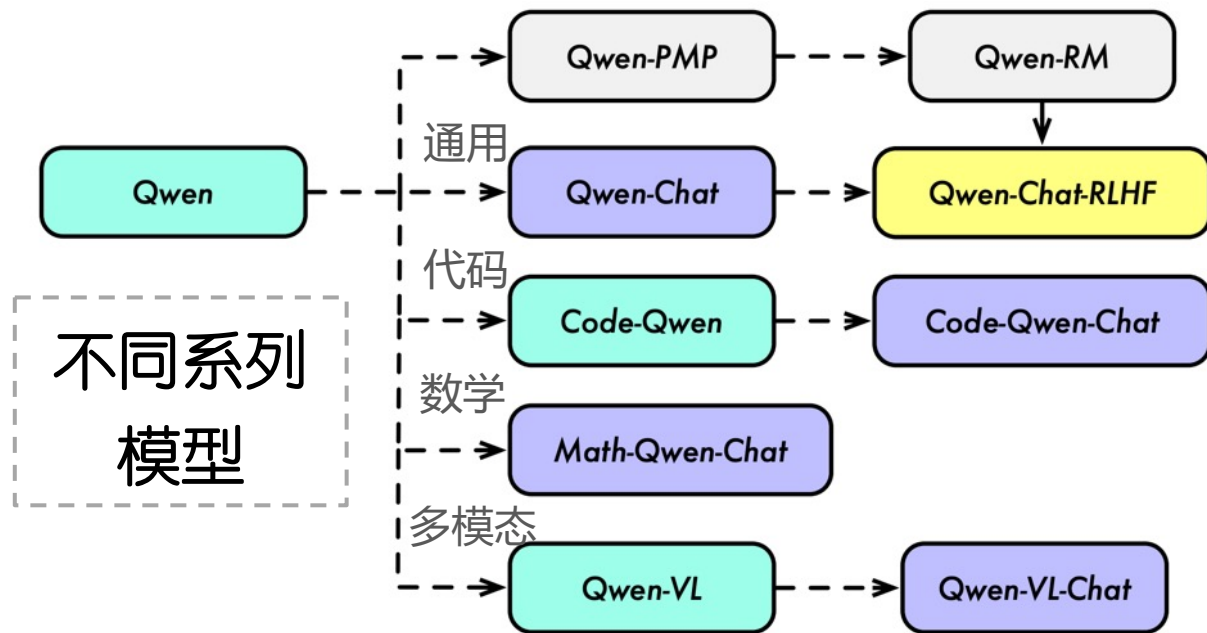
国内大模型发展

□ 当前，中国大模型发展已从早期“百模大战”的规模竞争阶段，进入以技术突破和商业落地为核心的深化发展期



Qwen系列大模型

- 通义千问Qwen是一个**多能力统一**的大模型体系，具备**自然语言理解、生成、推理、编程、多模态处理**等能力



- **第一阶段 Qwen (2023)**
建立基础模型能力
- **第二阶段 Qwen1.5 / Qwen2 (2024)**
模型规模扩展，增强推理能力
- **第三阶段 Qwen2.5 (2024)**
引入MoE架构，支持超长上下文
- **第四阶段 Qwen3 / Qwen3.5 (2025)**
进入智能体时代

Qwen系列大模型

□ Qwen不同系列比较

维度	Qwen (2023.9)	Qwen2 (2024.9)	Qwen2.5 (2025.1)	Qwen3 (2025.5)
参数规模	1.8B / 7B / 14B	0.5B / 1.5B / 7B / 72B Moe:57B-A14B	0.5B / 1.5B / 3B/7B / 14B/32B/ 72B	dense:0.6/1.7/4/8/14/32 MoE:32B-A3B+235B-A22B
模型类型	Dense 模型	Dense + MoE	Dense + MoE	Dense + MoE
层设计	基于 LLaMA:RoPE + MHA+SwiGLU + RMSNorm+FFN	RoPE+GQA+SwiGLU+ +RMSNorm	RoPE+GQA + SwiGLU + + RMSNorm + QKV Bias;	RoPE+GQA + SwiGLU + + RMSNorm + QK-Norm(移除QKV Bias)
Moe机制 (替换FFN层)	无	64个专家, 共享8个, 激活8个	共享专家路由 + Top-K激活	128专家中8激活 + 全局负载均衡 损失
Tokenization	改进BPE, 加入中 文、代码、多语词表 (152064)	BBPE:151643+3 token 通用token+control token	BBPE(151643+22 token)	BBPE(151669)

Qwen系列大模型

维度	Qwen (2023.9)	Qwen2 (2024.9)	Qwen2.5 (2025.1)	Qwen3 (2025.5)
预训练数据	3T tokens, 双语 (中英) 为主, 含代 码与数学	7T tokens <ol style="list-style-type: none"> 1 质量提升🚀: qwen过滤+生成) 2 数据拓展: 数学/代码/29种语言) 3 数据配比🚀 	18T tokens <ol style="list-style-type: none"> 1 数据过滤(qwen2-Instruct) 2 数据生成: qwen2-72B-Instruct+RM过滤 3 数据融合: qwen2.5Math/Coder训练数据集 4 数据配比: qwen2->instruct分类- >resample->balance) 	36T tokens <ol style="list-style-type: none"> 1 增加pdf数据: Qwen2.5-VL抽取->qwen2.5- Instruct refine) 2 数据生成: qwen2.5/qwen2.5Math/code) 3 增加更多语言数据(119种) 4 数据配比 ⚡ 数据标注系统标注: (教育性、安全性、领域) 多维度 标签,更细粒度控制样本配比平衡
对齐方法	SFT + PPO	SFT + DPO + online-DPO + OMO	SFT+offline-DPO+GRPO	long-cot 冷启动(SFT) + 推理强化 (GRPO) + 混合thinking mode SFT + 通用强化(GSPPO/精细而复 杂的reword model)+蒸馏(小模型 提升)

DeepSeek系列大模型

- DeepSeek是深度求索（DeepSeek）公司自主研发的大模型，具备**高性能、低训练成本、模型开源**等特性

DeepSeek-V3 系列 (通用模型)

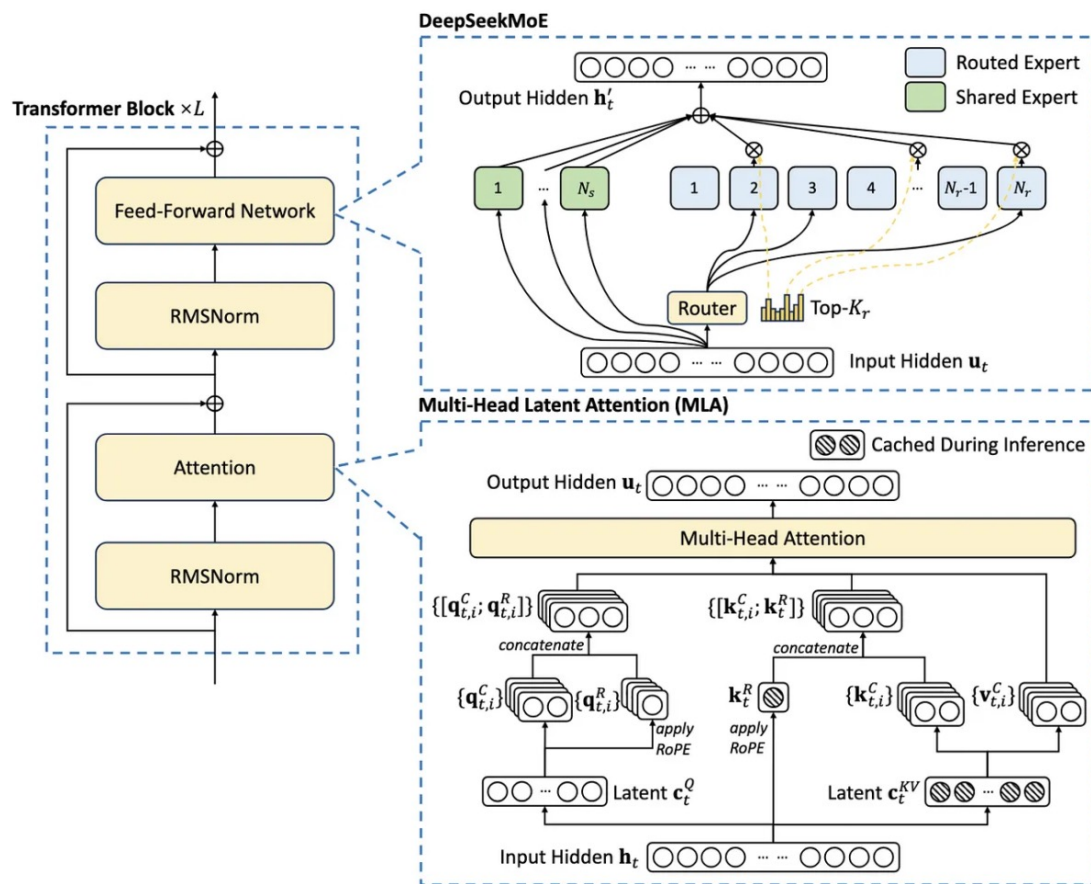
- **架构**: 采用MoE架构，引入多头潜注意力和无辅助损失的负载均衡
- **优势**: 极致性价比，训练成本极低，但性能可比肩GPT-4o，目前最强的开源模型之一
- **应用**: 擅长日常对话、内容生成、知识问答

DeepSeek-R1 系列 (推理模型)

- **架构**: 利用RL训练，在不经过SFT，进行冷启动训练和多阶段训练
- **优势**: 强推理能力，对标OpenAI o1系列，低成本高性能、高可解释性、支持蒸馏、微调、私有部署
- **应用**: 专注于复杂逻辑推理、数学、编程和科学问题

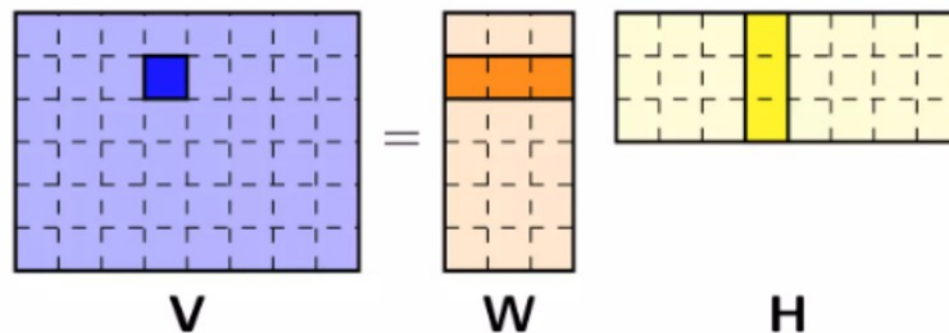
DeepSeek-V3

DeepSeek-V3创新：多头潜注意力 (MLA)和DeepSeekMoE架构



● 多头潜注意力MLA

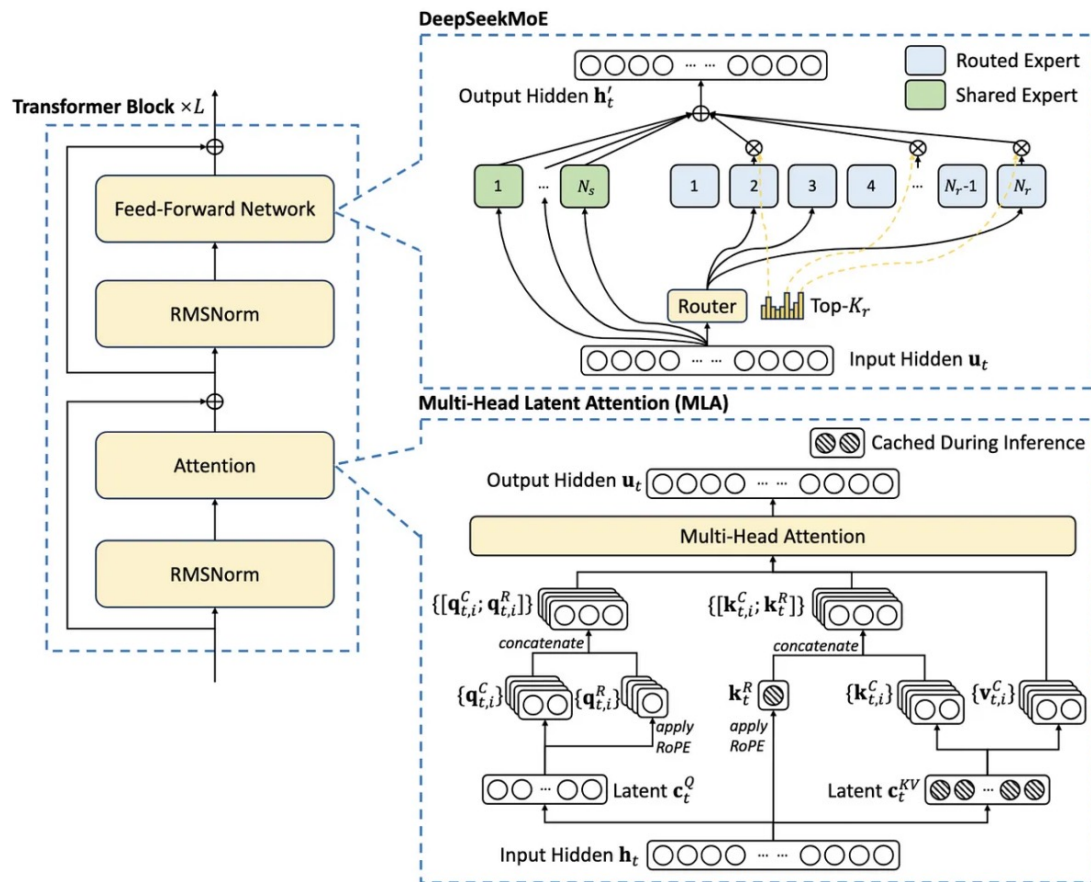
本质是对KV的有损压缩，提高存储信息密度，保留关键细节



KV矩阵转换为低秩形式：两个较小矩阵的乘积

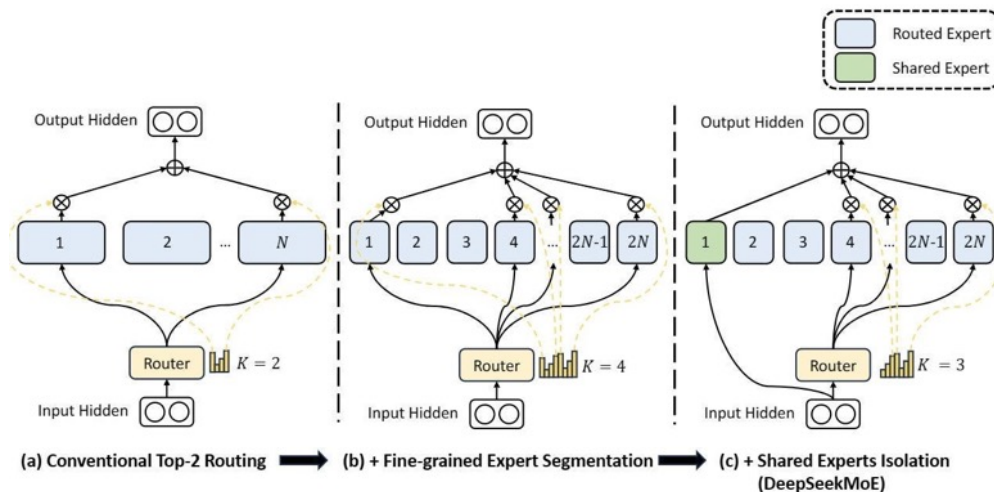
DeepSeek-V3

DeepSeek-V3创新：多头潜注意力 (MLA)和DeepSeekMoE架构



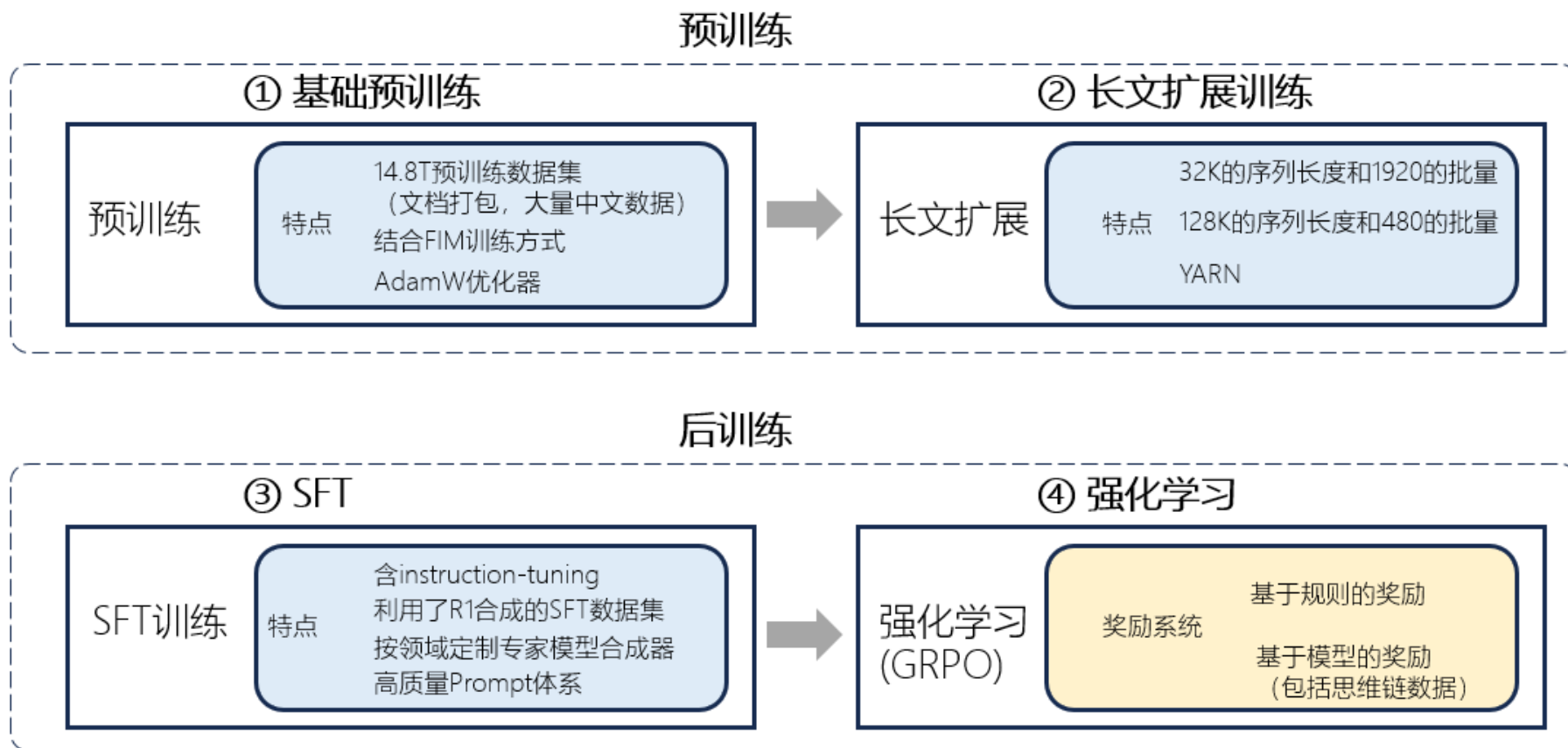
DeepSeekMoE架构

将专家分为两类：共享专家始终会被路由，路由专家需路由均衡



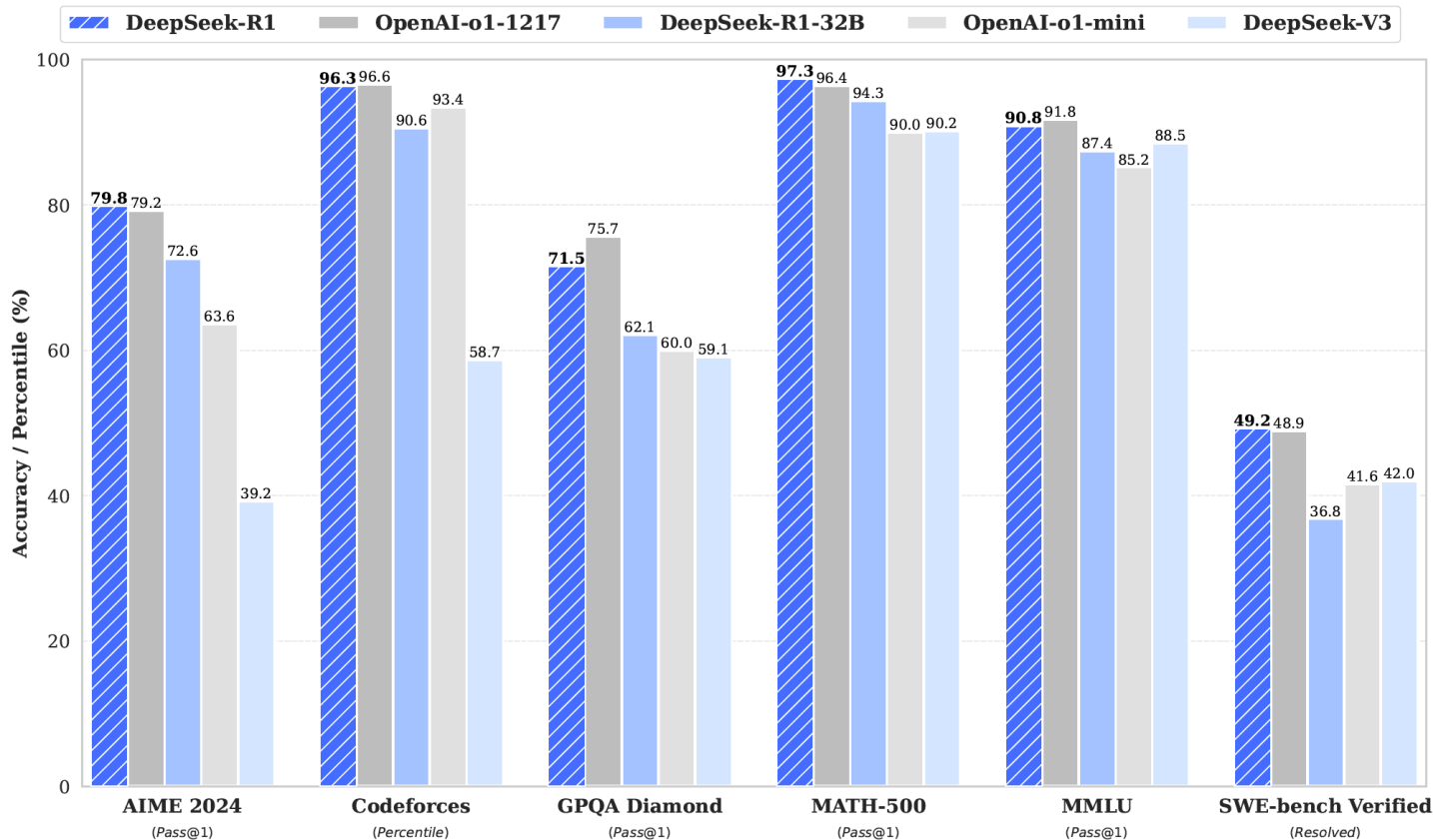
DeepSeek-V3

□ DeepSeek-V3训练流程：预训练和后训练两个阶段



DeepSeek-R1

□ DeepSeek-R1的模型架构来自V3，推理能力赶超OpenAI-o1



DeepSeek 回答示例 1

问题：

如果函数 $f(x) = 2x^2 - \ln x$ 在其定义域内的一个子区间 $(k-2, k+1)$ 上不是单调的，那么实数 k 的范围是？
请逐步思考，并始终在 □ 中回答。

关于原问题的 DeepSeek R1 回答 (Token 数量: 5602) :

为了确定实数 k 的范围，使得函数 $f(x) = 2x^2 - \ln x$ 在其定义域内的区间 $(k-2, k+1)$ 上不是单调的，我们需要考虑该函数的驻点，并确保它们位于该区间内。

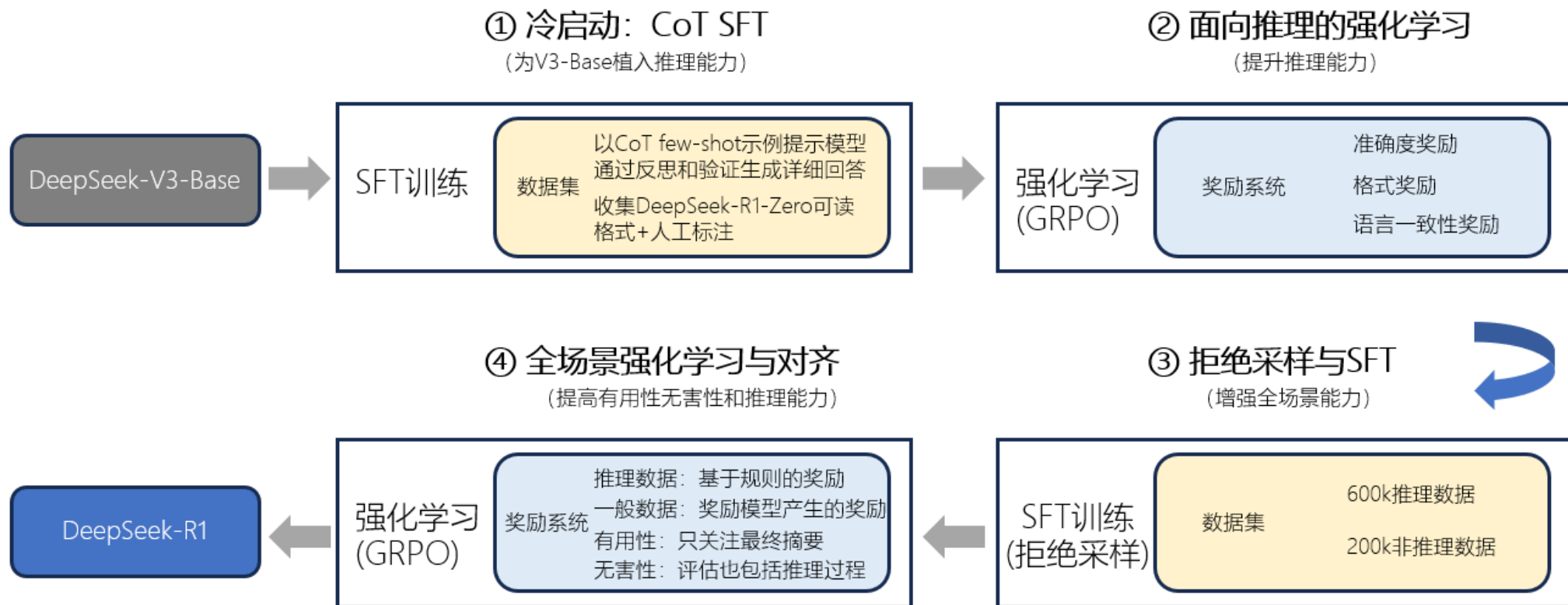
- 寻找驻点**:** $f(x)$ 的导数为 $f'(x) = 4x - \frac{1}{x}$ 。令 $f'(x) = 0$ ，得到 $4x - \frac{1}{x} = 0$ ，简化为 $4x^2 = 1$ 。解此方程，得到 $x = \frac{1}{2}$ (因为 $x > 0$)。
- 检查区间**:** 函数 $f(x)$ 在 $x > 0$ 时有定义。区间 $(k-2, k+1)$ 必须在定义域 $(0, \infty)$ 内。为了使区间 $(k-2, k+1)$ 在定义域内，我们需要 $k-2 > 0$ ，这意味着 $k > 2$ 。
- 包含驻点**:** 驻点 $x = \frac{1}{2}$ 必须位于区间 $(k-2, k+1)$ 内。这需要 $k-2 < \frac{1}{2} < k+1$ 。
- 解不等式**:** 从 $k-2 < \frac{1}{2}$ ，得到 $k < \frac{5}{2}$ 。从 $\frac{1}{2} < k+1$ ，得到 $k > -\frac{1}{2}$ 。
- 合并条件**:** 由于 $k > 2$ 比 $k > -\frac{1}{2}$ 具有更强的约束力，我们将条件合并为 $2 < k < \frac{5}{2}$ 。
- 检查端点**:** 当 $k = 2$ 时，区间为 $(0, 3)$ ，包含驻点 $\frac{1}{2}$ 。当 $k = \frac{5}{2}$ 时，区间为 $(\frac{1}{2}, \frac{7}{2})$ ，不包含驻点 $\frac{1}{2}$ 。

因此， k 的范围是 $[2, \frac{5}{2})$ 。

□ $[2, \frac{5}{2})$

DeepSeek-R1

□ DeepSeek-R1训练流程包括四个阶段：



总结

□ 大模型正在迅猛发展，未来是否有望通向通用人工智能？



本节复习

- 预训练模型范式: Pre-training、Fine-tuning
- 预训练模型家族, 如BERT、GPT、T5
- 大模型范式: Pre-training、SFT、Alignment
- 大模型主要代表, 如ChatGPT、Qwen、Deepseek

参考文献

- ❑ Zhao, Wayne Xin, et al. A Survey of Large Language Models. arXivpreprint arXiv:2303.18223.
- ❑ Han, Xu, et al. Pre-Trained Models: Past, Present and Future. arXiv preprint arXiv:2106.07139.
- ❑ Wei, Jason, et al. Emergent Abilities of Large Language Models. TMLR, 2022.
- ❑ <https://jalammar.github.io/illustrated-transformer/>
- ❑ <https://jalammar.github.io/illustrated-bert/>
- ❑ <https://zhuanlan.zhihu.com/p/21208287743>

致谢

- 胡玥、曹亚男、方芳：国科大《自然语言处理基础》
- 曹亚男、任昱冰：国科大《深度学习与自然语言处理概述》





THANKS