12. 专用模型协作与语音模型，数据与TTC

本章探讨了基础与专用模型的协作与进化，强调了高效持续预训练基础设施的重要性，旨在生成高质量的小型语言模型（SLM）和多模态SLM，推动AI民主化。回顾了AI的发展历程，讨论了大模型与实际应用的关系，指出小模型在实际应用中的优势；在语音部分，介绍了语音语言模型的定义和工作模式，讨论了端到端语音聊天机器人的优势和挑战，以及语音表示和预训练阶段的挑战。最后，介绍了GLM-4-Voice模型，通过向量量化层实现监督语音标记化，并采用语音-文本交叉预训练模式，表现出优越的性能。

2024版本：唐杰、杜晋华、白华君、高杨驰、徐家骏、高博文、赵若雯、罗开荣

# 1. 基础与专用模型的协作与进化

## 1.1 摘要

* 当前的计算资源垄断显著限制了AI的发展，将大规模语言模型（LLM）预训练阶段的参与者局限于少数研究人员。我们正在开发一种高效的持续预训练基础设施，旨在生成高质量的小型语言模型（SLM）和多模态SLM，特别强调增强推理能力。
* 我们还引入了一种创新系统，将数百个领域特定模型整合起来，构建一种需求最小化计算资源的人工通用智能（AGI）基础模型。通过采用更小、更高效模型，利用跨不同领域的顶尖模型通过稳健的排名算法，以及持续优化进化中的基础模型，这种方法旨在实现AI开发的民主化。它从传统的“数据之上的模型”或集中式LLM方法，转变为“模型之上的模型”或去中心化LLM策略，旨在减少对大量计算资源的依赖，促进更广泛的创新和包容性。

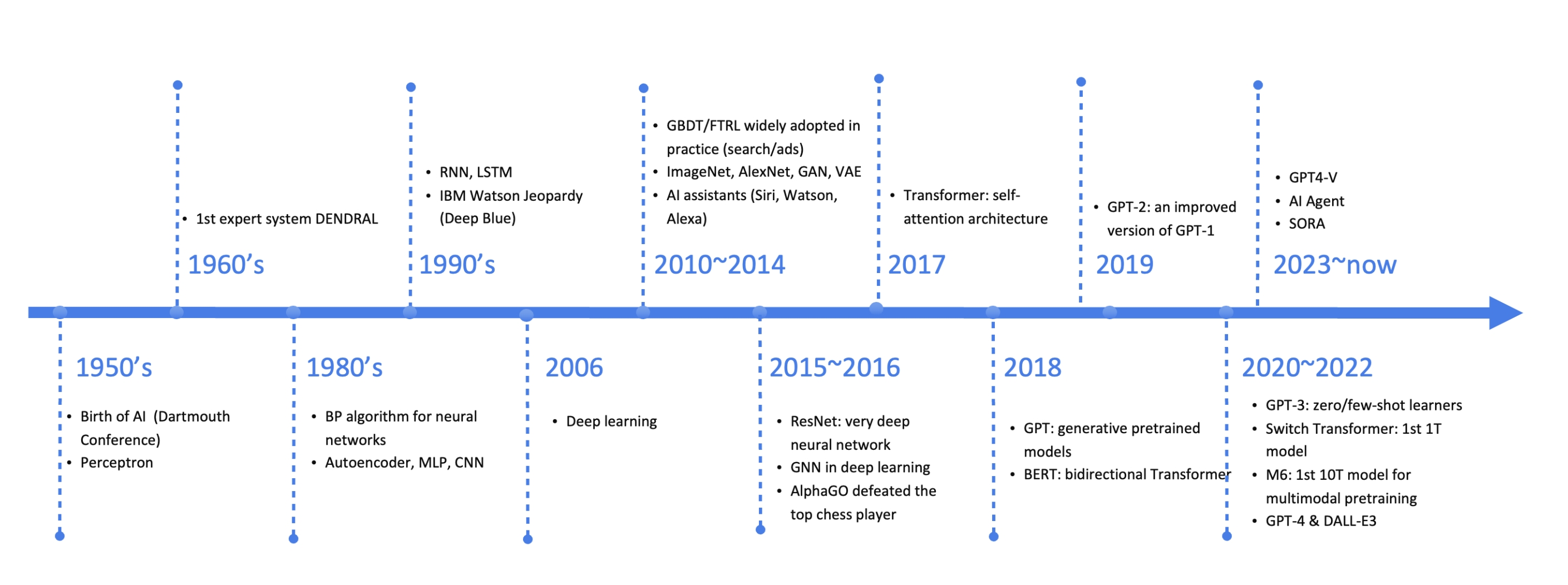
## 1.2 杨红霞教授简介

* 杨红霞教授拥有超过15年的AI科学家经验，专注于大规模机器学习、数据挖掘和深度学习。
* 在她的职业生涯中，她开发了10个重要的算法系统，提升了多个企业的运营效率。
* 她的研究包括预训练模型、大数据分析，以及大规模语言模型（LLM）系统在实际设置中的应用。
* 杨教授发表了超过100篇顶级论文，累计近1万次引用，H指数为46，并拥有超过50项专利。
* 她获得了多项奖励，包括2019年世界人工智能大会SAIL奖和2020年国家科学技术进步奖——中国最高科技奖项。她被评为2022年福布斯中国科技女性Top 50以及2023-2024年AI 2000最具影响力学者奖。
* 杨教授曾在字节跳动美国、阿里巴巴集团、雅虎公司和IBM T.J.沃森研究中心担任重要职位。她获得了杜克大学的博士学位和南开大学的学士学位。

## 1.3 具体内容

### 1.3.1 回顾了AI的发展历史

* 20世纪50年代，AI名词被提出；
* 20世纪60年代，第一个专家系统DENDRAL面世；
* 20世纪80年代，神经网络的BP算法以及Autenencoder、MLP、CNN被提出；
* 20世纪90年代，RNN、LSTM以及IBM Watson Jeopardy被提出；
* 2006年起，深度学习开始涌现；
* 2010年至2014年，GBDT/FTRL被广泛用于实践中，比如用于搜索和广告，ImageNet、AlexNet、GAN、VAE也陆续被提出，同时还出现AI助手，比如Siri、Watson、Alexa等；
* 2015-2016年，一种深度神经网络ResNet被提出了，同时深度学习中的GNN诞生，另外，AlphaGO引起了世人的广泛关注，它打败了世界上最顶尖的围棋选手；
* 2017年，基于自注意机制框架的Transformer出现，自此一大批基于Transformer的模型出现；
* 2018年，GPT和BERT出现；2019年，GPT-1的改进版本GPT-2出现；2020-2022年，主要的一些代表模型有GPT-3（zero/few-shot learners）、Switch Transformer（第一个1T模型）、M6（第一个10T多模态预训练模型）、GPT-4以及DALL-E3；
* 2023至现在，GPT4-V、AI agent、SORA等出现。

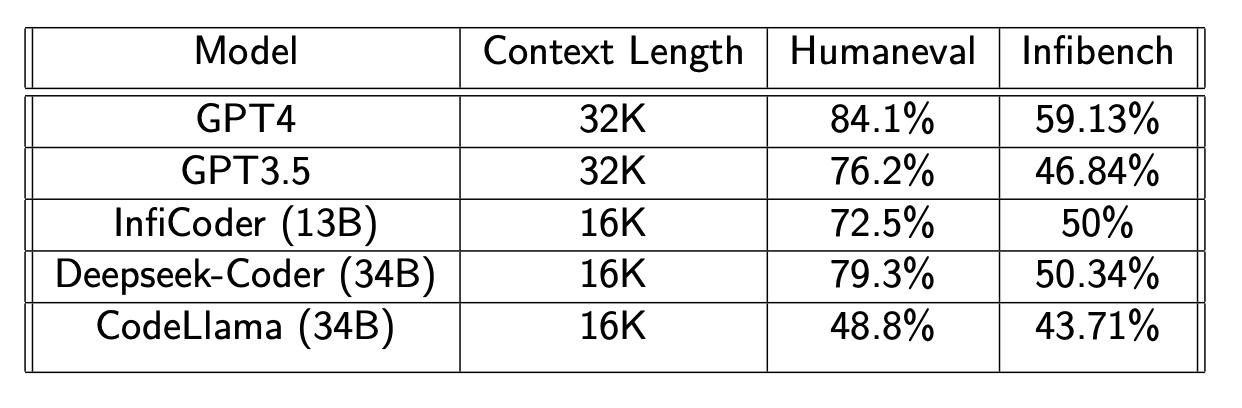


### 1.3.2. 大模型与实际应用

* 模型很大，不一定可以改动传统的行业，比如检索（M6、OFA）
* 目前很小的模型会更有实际应用场景，而且小模型更容易做增量预训练和微调，并优于GPT4（specific model / task-base LLM）

### 1.3.3 回顾了全流程训练的一些经验

* 1. **全流程训练入门**：初学者可以通过构建evaluation benchmark datasets ，可以用小算力理解各种模型：比如把gsm8k中的常量改为变量。***example:*** 针对freeformQA类问题（需要人参与给反馈）提出infibench——infocoder



* 1. **预训练阶段 PT**：

预训练阶段首先需要解决数据质量问题。

1. 经验一：使用高质量代码数据集or数据质量分类起。使用来自 BigCode the Stack V1 的 43B 高质量 Python 代码；使用有限数据训练低质量和高质量分类器。两者均实现了80% 以上的 AUC。
2. 经验二：构建预训练“教科书”。教科书生成流程 (10B)：

（1）教科书是使用 phi-1和 phi-1.5 论文中提到的方法创建的；

（2）从维基百科中过滤主题并生成相关主题；

（3）使用 Python、Java、C、C++、JavaScript、PHP 和 C 语言为每个主题提供 45 个提示的多样化语言目录；

（4）为每个主题生成 90 本教科书，其中各部分采用随机选择的语气和包使用组合；

（5）GPT-4 API 评估产生了高质量的教科书（平均分数：4.02）和代码（分数：3.97）

c. **监督微调阶段SFT**：主要目标是教会模型如何做对。其具有以下几个特性：

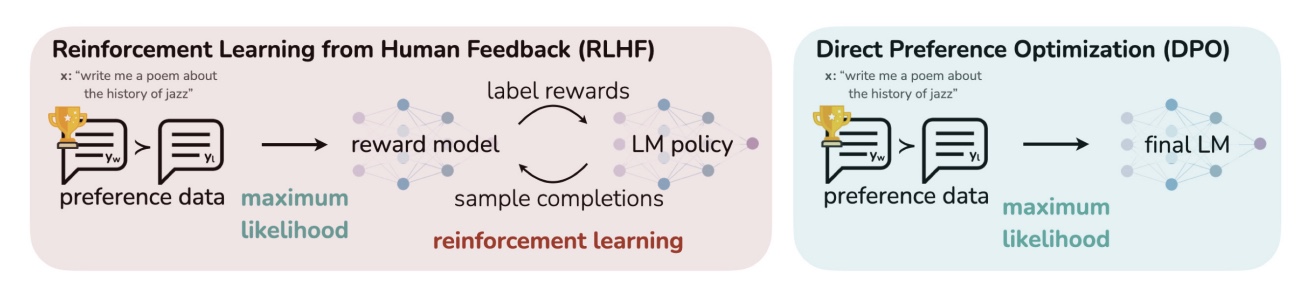
（1）多任务：利用来自各种来源的监督微调 (SFT) 数据来增强模型在基于文本的代码生成、自由格式 QA 和代码完成问题中的指令遵循能力；

（2）模型自适应：模型的进化或自我指导提示效果良好；

（3）多样性：实际时采用多种策略混合微调；

（4）效率高：可以利用LoRA微调方法，减少微调的权重。

d. **DPO**：主要目标是教会模型识别错的方面。传统的 RLHF包括三个阶段：监督微调 (SFT)、偏好采样和奖励学习以及 RL 优化。DPO将奖励函数的损失函数转换为策略的损失函数。



e. **智能体Agent**：

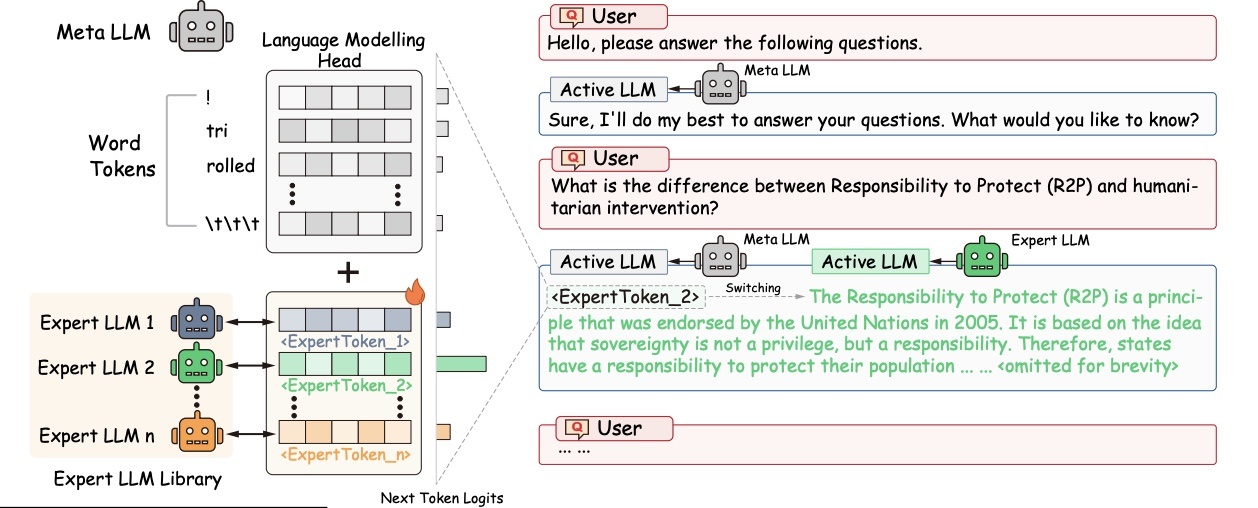
1、单代理单工具：高级数据分析（ADA，以前称为代码解释器）

2、单代理多工具：插件（基于推理）

3、多代理多工具：GPT（如GPT4）和 基于学习的GPT 存储

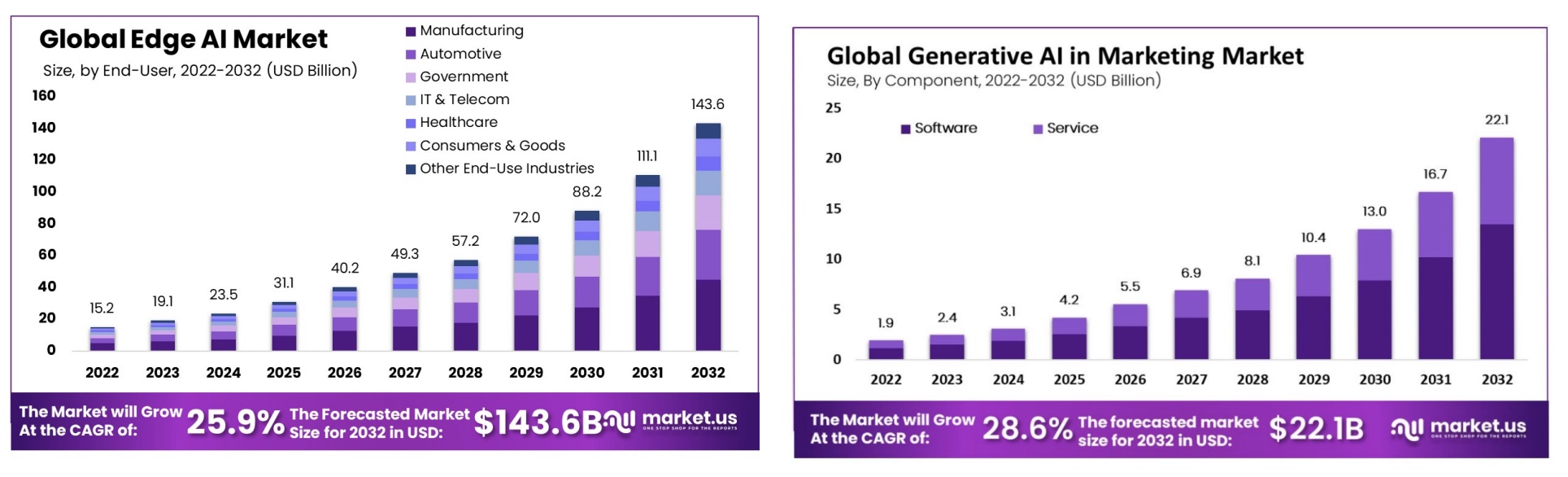
f. **智能体通信Agent Communication**：经过微调的 LLM（expert）可能会经历某些能力的下降（灾难性遗忘），一个潜在的解决方案是将general LLM 与expert LLM 协作，例如通过多代理对话进行协同。

我们的解决方案：**Expert-Token-Routing**：用于解决精调LLM（专家模型）能力下降的问题。该方法通过将专家模型以特殊令牌的形式嵌入到元模型（Meta LLM）的词汇表中，实现通用模型与专家模型的高效协作。用户的问题首先由元模型解析，随后动态路由到最适合的专家模型完成任务，并返回精准答案。这种机制有效避免了灾难性遗忘，同时增强了模型的灵活性与多任务处理能力，为构建更强大的协作式生成式AI提供了新的方向。



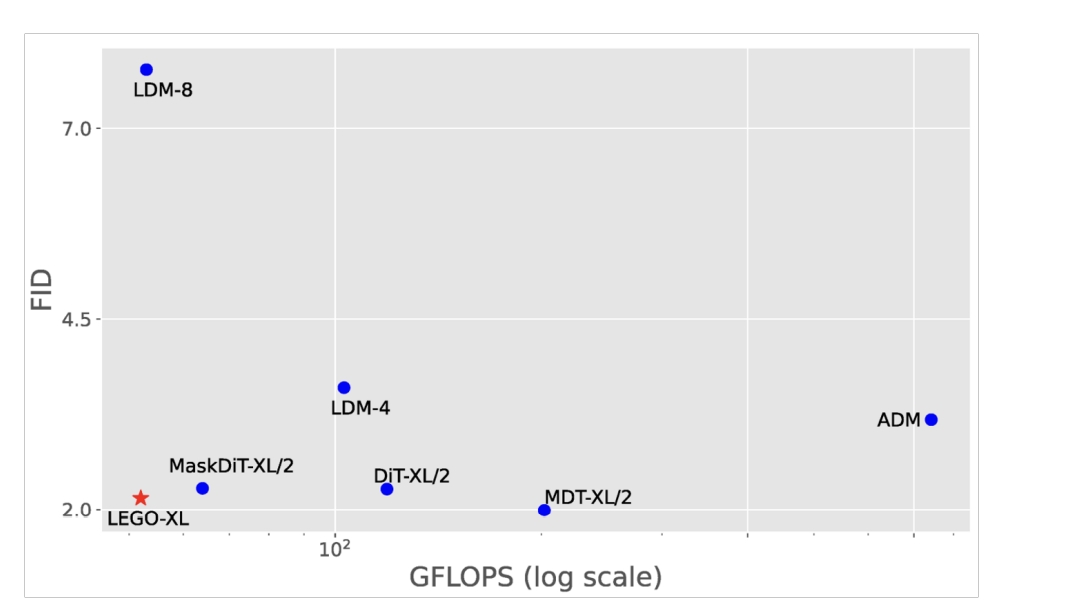
### 1.3.4. 小模型的必要性分析

预计到 2032 年，边缘 AI 市场规模将达到 1436 亿美元，复合年增长率为 25.9%，这主要得益于对低延迟处理、能源效率、隐私和 AI 去中心化的需求不断增长。



模型蒸馏：通过引入一种无需数据的新方法SiD，可将预训练的扩散模型提炼为单步生成器，从而改进 FID 和 IS 指标，同时将计算成本降低 30% 以上。SiD 通过最小化分数匹配损失进行提炼，在噪声水平上对齐真实分数和生成分数，有望实现统一的 GenAI 提炼范式。案例：GPT-4v是基于goole 的Flamingo结构来的，InfiMM在多模态推理中排名第一，使用 Flamingo 架构和 Vision Language 连接器中的自回归，实现多种模态。

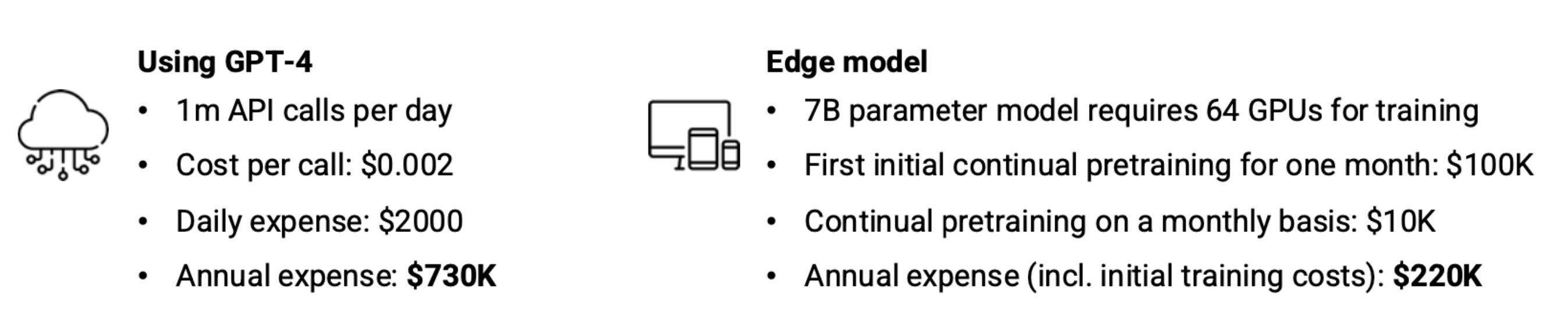
* 低资源推理与学习：采用LEGO式模块架构，通过模块堆叠与跳跃实现高效推理，显著降低60%的GPU资源消耗。案例：Learning Stackable and Skippable LEGO Bricks for Efficient, Reconfigurable, and Variable-Resolution Diffusion Modeling, ICLR 2024

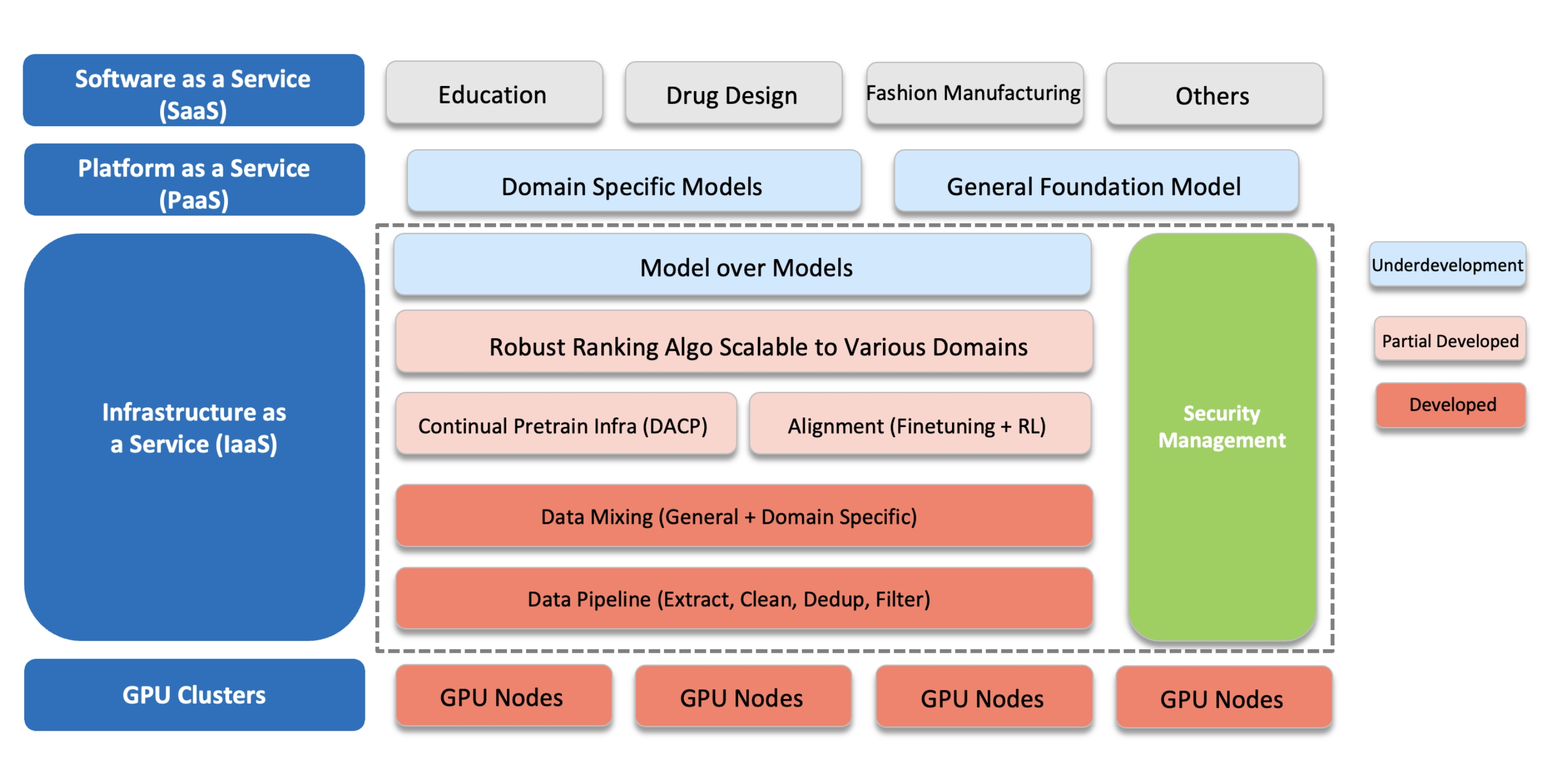


* 统一生成式AI蒸馏范式：得分匹配技术（SiD）大幅提升生成质量，并降低计算成本超过30%。案例：Score identity Distillation: Exponentially Fast Distillation of Pretrained Diffusion Models for One-Step Generation, ICML 2024
* 多模态推理与规划。案例：InfiMM模型采用Flamingo架构，通过小规模模型实现高效多模态理解，尤其适用于视觉与语言的结合任务。InfiMM: Advancing Multimodal Understanding with an Open-Sourced Visual Language Model, ACL 2024

### 1.3.5 新范式：从model over data到model over models

杨老师预测：未来的大模型范式不会知识通过训练数据训练得到模型，而是每个领域都有sota model，有一个“新范式模型”从这些模型中学习——避免了处理繁杂的数据，而且很容易学学好，可能的实现：API ordatabase-RAG。案例：bladeAI。





# 2. 语音

## 2.1背景

在与人类的交互过程中，像ChatGPT和Claude这些基于大型语言模型（LLM）的聊天应用主要处理文本的输入和输出，更多关注语义信息和逻辑信息。然而，真实的交互过程并不只有文本。为了实现更加自然的人机交互，我们需要考虑许多其他因素，例如视觉和音频输入的处理，语音输出的实现，对情感的理解和表达，交互延迟的降低等等。

## 2.2语音语言模型

语音语言模型(Speech Language Model, SLM)的定义是：

* 输入和输出至少有一个是语音；
* 不经过微调就能处理多种语音任务；
* 能够遵从以自然语言给出的指令。

音频分析是SLMs的一种可能的工作模式。在音频分析任务中，模型根据指令处理输入的音频数据，其中指令输入和音频输入是相互独立的。也就是说，模型会根据文本指令对音频信息进行分析，这里的音频包括人类语音、音乐乃至任何可能的声音。

SLMs另一种可能的工作模式是语音聊天应用，在这种模式下，模型会依据用户的输入语音生成语音回复。下文列出了几个语音聊天应用的例子：

* GPT-4o Advanced Voice Mode：<https://openai.com/index/hello-gpt-4o/>
* Moshi：<https://moshi.chat/>
* GLM-4-Voice：<https://github.com/THUDM/GLM-4-Voice>

在本节内容中，我们将主要聚焦于语音聊天应用。

除了输入输出形式的不同外，语音聊天应用（以语音为输入和输出）相对于文本聊天应用（以文本为输入和输出）还有一个重要的区别，那就是前者不仅需要关注回复内容的正确性，还需要关注生成的语音是否能够满足使用者的需要。

语音聊天应用的一个baseline是语音识别（ASR），文本语言模型（Text LLM）和语音合成（TTS）三种模型的级联系统。其中ASR模型负责将用户的语音输入转化成文本输入，随后Text LLM根据文本输入生成文本回复，最后TTS模型根据文本回复生成相应的语音回复。

然而，在实际应用中，这种简单的级联系统存在一些问题。

首先，在语音识别的过程中存在语义信息的丢失，包括可能发生的语义错误，例如ASR模型将"I know what you mean"识别成"I know what you man"；另外语音识别还可能造成情感信息的丢失，例如用户以讽刺口吻说出"Oh, great"表达沮丧时，将语音转化成文本会丢失沮丧的情感信息。

第二个问题是语音合成过程中存在的副语言信息的丢失（如语速、情感等）。例如，当用户输入“快速地阅读一个绕口令” 时，语音合成模型可能只会以正常的语调和速度输出。

另外，级联方案还存在较高的延迟问题。假设三个模型分别存在1s的延迟，级联之后的延迟将达到3s，这在实际的人机交互过程中可能带来较为糟糕的用户体验。

端到端语音聊天机器人是一种能够同时处理语音输入和输出的模型，它避免了模态转换过程中的信息丢失，并能以较低的延迟完成交互。目前，实现这样的端到端语音模型存在的困难主要包括语音表征的形式、预训练过程中的挑战，以及人机交互的自然性。

### 2.2.1 语音的表示

在文本的标记化（Tokenization）过程中，我们会使用字符对合并算法（BPE）对文本进行分词，并将词表与标记进行对应，从而将原本的字符序列转化为标记序列。但是，当输入是音频的波形时，我们应该如何将其转化成标记序列呢？

在语音的表示中，关键特征可以分为语义信息、副语言信息（如韵律、情感），以及非语音信息（如音乐、背景声）。通常而言，对于处理人类语音的模型而言，语义信息的重要性大于情感信息，远大于非语音信息。

对语音模型而言，语音表示的目标是得到离散化的标记序列。对于某个固定的输入，存在着多种不同的可能的输出，离散化的标记对于应用LLM已有的建模算法进行序列生成是十分关键的。

那么，我们应该如何得到语音的离散化标记呢？我们可以利用向量量化（Vector Quantization，VQ）算法将连续的向量表示转换为离散的变量。

在VQ算法中，码本（codebook）是一组预定义的向量集合，每个向量称为码字。码本e是大小为K×D的实数矩阵，其中K是码本中码字的个数，而D是每个码字的维数。量化就是将输入向量映射到最接近的码字的过程。对于前置的神经网络输出的一个D维向量z，VQ算法会在码本中找到与之距离最近的码字ei，并在之后的运算中用这个码字代替原来的向量z（通常用zq表示）。

VQ算法可以用于任意神经网络，一个常见的应用是向量量化变分自编码器（VQVAE）。离散化表示的优点包括更适合自回归生成任务和更高的存储效率（存储整数而非连续值）。

利用VQ算法进行语音标记化的方法分为无监督学习方案和监督学习方案两种。

无监督学习的方案是将向量量化模块插入一个音频编码器中，以重建语音为训练目标。音频在输入编码器（Encoder）后，通过量化转化成离散的标记，随后经过解码器（Decoder）输出重建后的语音。这种方法的优点是能够实现最大程度的信息保留，但缺点是可能导致较高的比特率，信息的集中程度较低，编码效率不高。

监督学习的方案则是将向量量化模块插入一个语音识别（ASR）模型，以ASR任务作为训练目标。音频输入由分块因果自注意力模块、前馈神经网络、池化层和向量量化模块组成的Tokenizer后转化成离散的标记，随后再次经历自注意力和前馈网络，最终由语音识别解码器输出识别的文本。监督学习方案的优点是能够实现低比特率的信息编码和最大程度的语义信息保留，缺点则是会丢失所有的非语音信息，副语言信息的保留也无法保证。

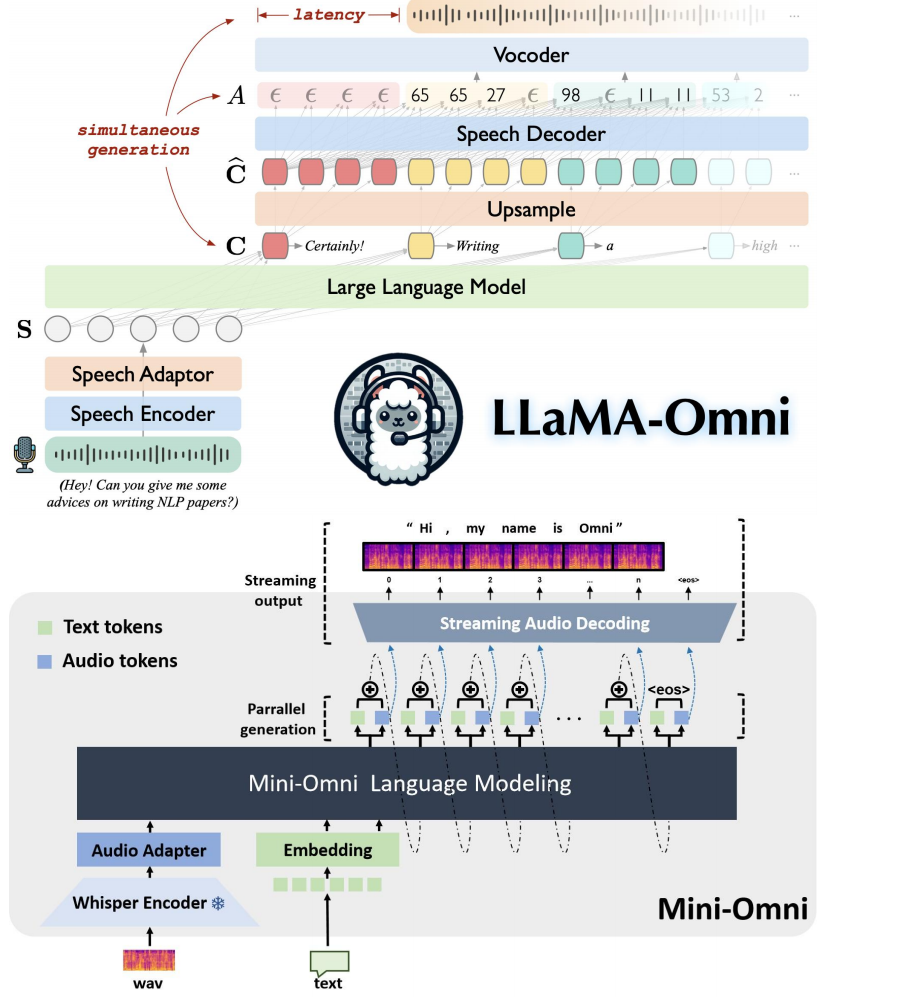
### 2.2.2 预训练阶段的挑战

在文本语言模型的预训练中，我们的目标是让模型在文本语料库中预测下一个文本标签（text token），于是，在语音模型的预训练中，一个自然的想法是让模型在语音语料库中预测下一个语音标签（speech token）。然而，想让预训练的语音模型生成语义连贯的句子实际上是十分困难的。

预训练语音模型的一个核心问题是语音模态的知识密度有限。与文本预训练相比，语音预训练需要处理的数据量更大。以LLaMA 3为例，LLaMA 3在拥有15T文本token的数据上进行预训练，以每分钟100个token计算，这相当于2.5B小时的语音（约285k年），语音token数可以达到450T。

为了解决语音预训练效率低下的问题，我们可以考虑将文本语言模型的知识迁移到语音模型中。一种可能的方法是轻量级对齐（类Llava架构），它通过语音编码器将语音转换为连续嵌入（embedding），并利用一个Adapter将生成的嵌入映射到文本语言模型的特征空间，结合文本指令获得文本模型的输出。该架构使用ASR数据和语音问答进行训练，在语音语言模型中是一种流行而简单的解决方案（例如Qwen-Audio就采用了这种方法）。

不过，这种方案并没有解决如何生成语音的问题。为此，Llama-Omni / Mini-Omni的设计在上述架构中增加了文本到语音的转换模块。



与前文类似，训练阶段，模型利用自动语音识别 (ASR) 数据对齐语音和文本的特征，同时支持语音输入和文本输入两种模态的训练。模型工作时，输入的语音信号经过编码器编码并通过Adapter 转化为音频token，文本输入则转化为文本token，这些token经过嵌入处理后被输入到大语言模型中。特征对齐后，模型的以并行方式进行生成，输出经过上采样和流式音频解码，最终通过声码器 (Vocoder) 输出为语音信号。

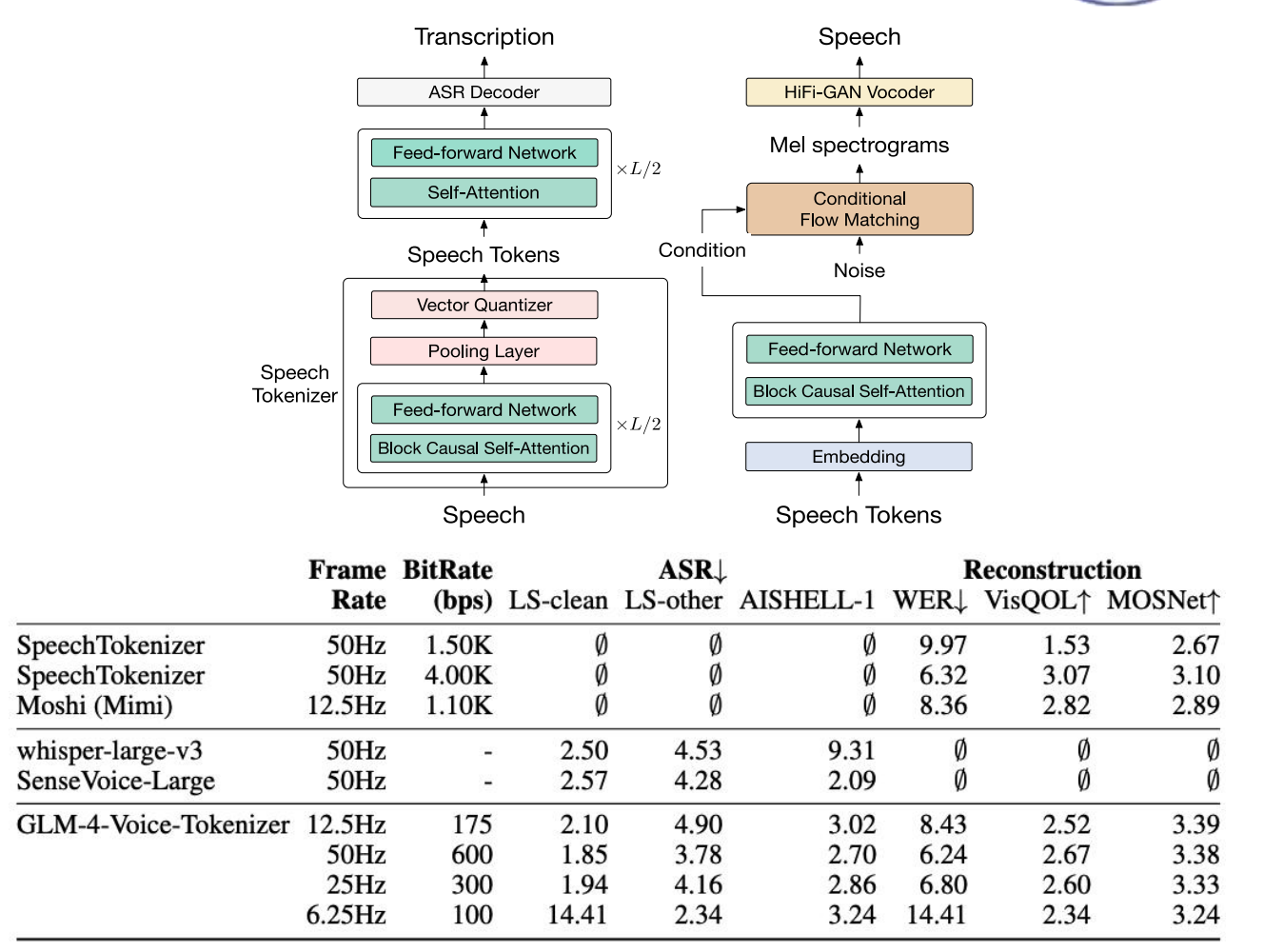
Llama-Omni / Mini-Omni架构的优点是实现了端到端的语音转换，降低了输出延迟；缺点是由于缺乏针对语音的预训练阶段，生成的语音质量较低，缺乏情感表达。

如何通过指令生成具有丰富情感的语音呢？在探讨这个问题时，我们可以先回顾一下大语言模型是如何发展其遵循指令的能力的。这个过程可以分为两个主要步骤：一是大规模互联网语料库的预训练——模型通过学习大量的互联网文本数据，学会了如何继续撰写文档，这意味着模型能够理解语言的结构和语义，从而为生成文本打下基础；二是在特定任务上使用高质量数据进行微调，这使得模型能够更好地遵循指令来执行任务。

然而，在语音模型领域，现有的训练架构缺少了在多种音频数据上进行预训练的步骤，这使得模型难以实现音频的泛化能力。所以，生成富有情感语音的关键仍然在于音频数据上的预训练。

### 2.2.3 GLM-4-Voice——迈向接近人类的端到端智能语音聊天应用

GLM-4-Voice通过在Whisper模型中插入向量量化层来实现监督语音标记化，并使用基于流匹配的解码器。它的特点包括单码本、超低比特率（每秒12.5个标记），准确保留语义信息，并保留了语速和情感等副语言信息。



GLM-4-Voice结构及效果示意

GLM-4-Voice使用了语音-文本交叉预训练的模式，迫使模型对齐语音和文本模态。其中交叉预训练的数据是利用SpiritLM从语音-文本并行数据构建的（不过，这样做的缺点是语音-文本并行数据收集难度较大）。通过在语音和文本混合序列上的预训练，语音模型同时从语音和文本输入中学到知识，对两种模态进行了更好的对齐。

在预训练阶段，研究人员从文本语料库中合成了600B的语音-文本交叉标记，并进行了总计1T标记的预训练。预训练的数据主要包含三种形式：语音-文本交叉数据，无监督语音数据，和语音-文本并行数据（ASR, TTS）。

在合成语音-文本交叉标记时，研究人员首先训练了一个将文本转化为音频标记的语音合成（TTS）模型，随后从文本语料库的样本中选择要转换为语音的文本跨度，再利用训练好的TTS模型将选定的文本跨度转换为语音标记，从而得到交叉数据。

在微调阶段，由于直接从语音提问到语音回答的任务较为困难，为了提高准确性，GLM-4-Voice采用了语音输入到文本输出再到语音输出的方法，并应用流式思维来降低延迟。微调过程使用了高质量的对话数据，最终延迟达到了13个文本标记+10个语音标记，并在聊天应用的各项评估指标上得到了优越的性能。

