OSCHINA gitee gitee AI

2024 中国开源开发者报告

China Open Source 2024 Annual Report

聚焦大模型

2024 年 12 月

支持单位：北京银行 BANK OF BEIJING

目 录

Part 1：中国开源开发者生态数据

Part 2：TOP101-2024 大模型观点

Part 3：国内 GenAI 生态高亮瞬间

编委会

局长，OSCHINA 副主编

肖滢，OSCHINA 副主编

李泽辰，Gitee 主编

高瞻，Gitee AI 运营

设计：张琪

出品：OSCHINA | Gitee | Gitee AI

<Part 1：中国开源开发者生态数据>

开发者是开源生态的重要支柱。

本章结合 Gitee & Gitee AI 平台、OSS Compass 的数据分析，勾勒 2024 年中国开源开发者的整体画像趋势轮廓，主要反映中国开源开发者使用开源大模型概况、开源项目/组织健康度，以及中国开源社区的生态评估等情况。

OSCHINA gitee gitee AI

中国开源开发者生态数据

2024 中国开源开发者报告

China Open Source 2024 Annual Report

聚焦大模型

Gitee 数据篇

本报告数据来源：2024年1月至2024年12月
Gitee及Gitee AI平台相关公开数据

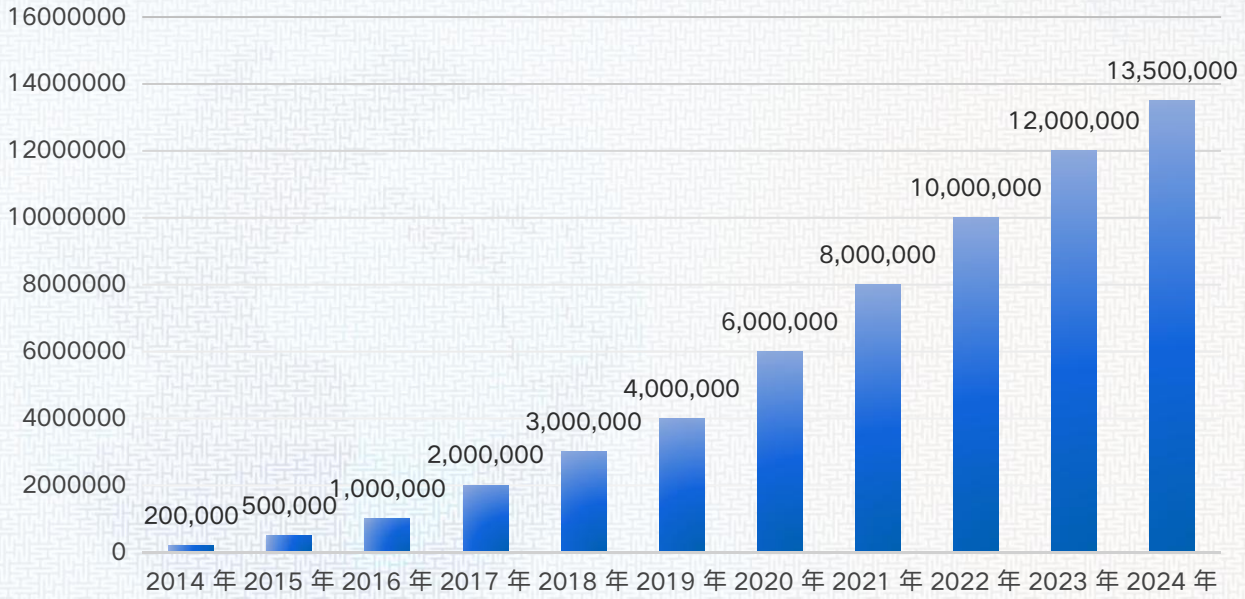
开发者是社区的力量源泉

1350 万

2024年Gitee总用户数

150 万

2024年Gitee新增用户数



2014-2024 Gitee 用户数增长曲线

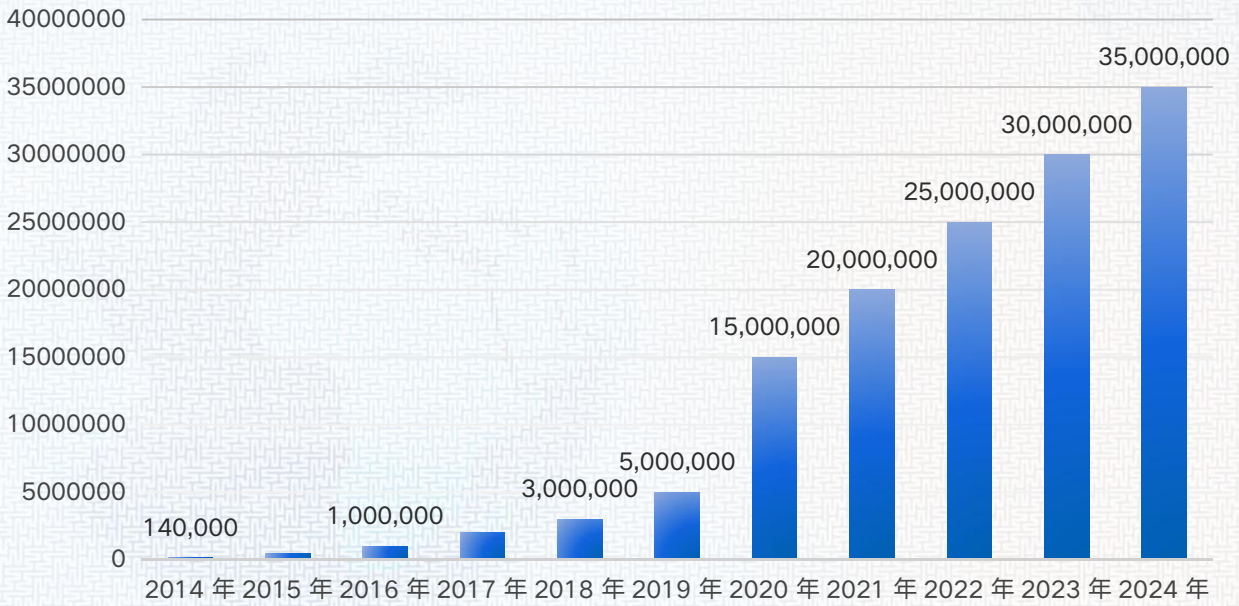
开发者是社区的力量源泉

3600 万

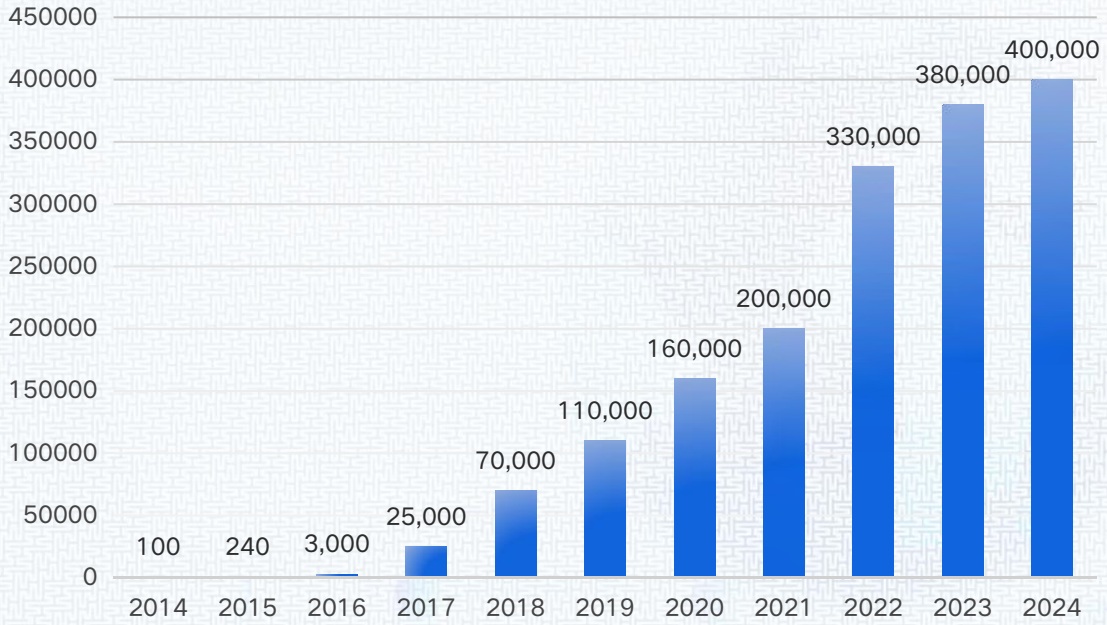
2024年Gitee总仓库数

500 万

2024年Gitee新增仓库数



和开源共同体拥抱开放透明



40 万

2024年Gitee开源组织数量

2024年，Gitee上的开源组织数量达到了40万个，越来越多的开发者选择凝聚在一起，共同拥抱开放透明的组织协同。

本年度最受开发者关注的开源组织

2024 年，技术大厂及其大型项目依然备受关注，它们推动着技术的快速发展和广泛应用。

同时，「民间」开源组织虽然在关注度上不及大厂主导的项目，但它们在某些技术细分领域中却拥有非常强的影响力和活跃的开发者社区。

两者的不同发展模式相互补充，共同推动了国内开源生态的繁荣与多样化。

最受关注开源组织指2024年获得 Star 数最多的开源组织（成员5人及以上）

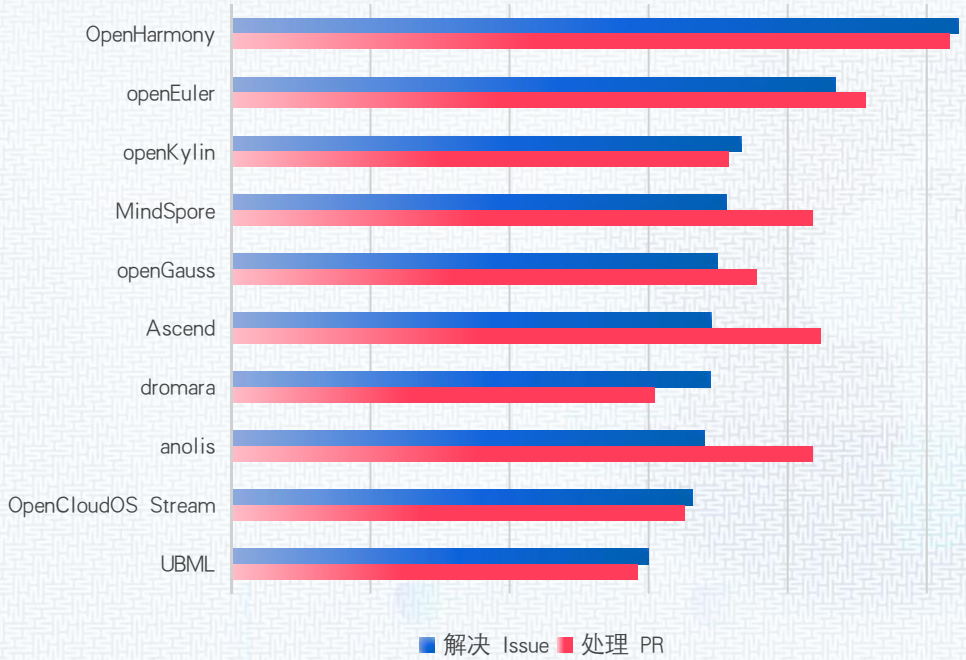
本年度最受开发者喜爱的开源组织

收获 Star 数方面，更加注重创新和技术探索「民间」开源组织更加「接地气」，受到了更多开发者的喜爱。

虽然面临资源的挑战，但它们的灵活性和社区驱动力使得它们获得了更多开发者的支持。

最受喜爱开源组织指2024年获得 Star 数最多的开源组织（成员5人及以上）

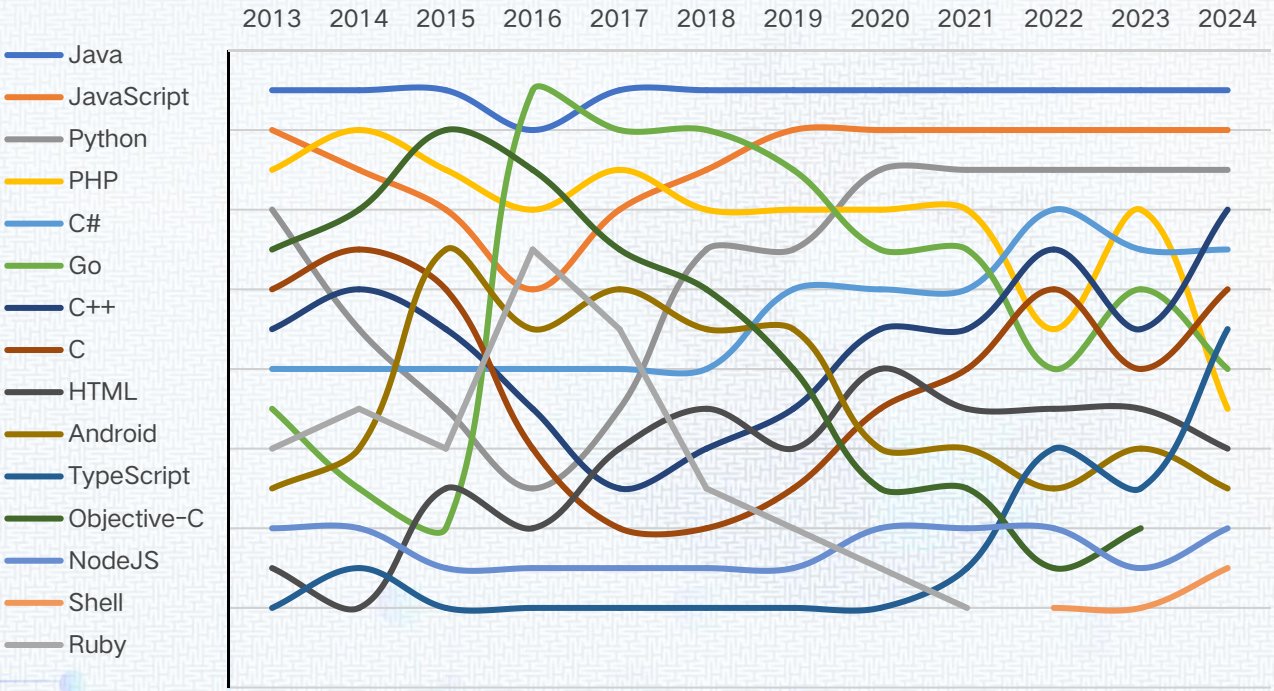
本年度最活跃的开源组织



不同开源组织在 Issue 解决 和 PR 处理数量上的差异，反映了它们在开发活跃度、社区参与度和技术成熟度上的不同战略。

技术大厂主导的项目往往具有较高的资源投入和社区管理效率，而民间组织则可能更注重技术问题的快速解决，并逐步吸引更多的开发者参与贡献。

编程语言流行趋势

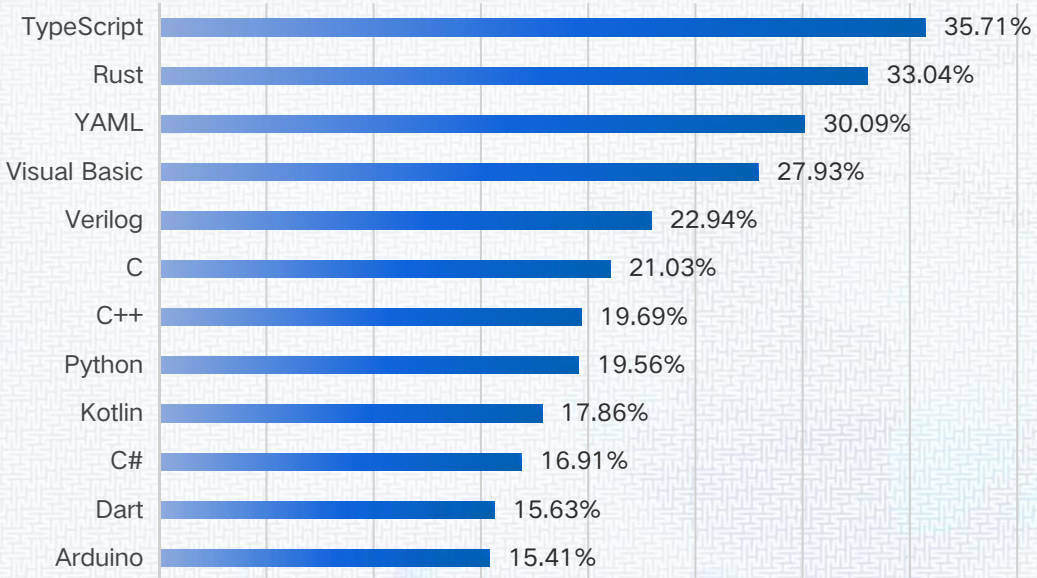


2024年，Gitee上的编程语言依然由Java、JavaScript、Python引领潮流。

与此同时变化也在悄然进行中：凭借AI开发热潮，C与C++依然在今年焕发着生命力，流行度已与十年前不相上下。

TypeScript依然强势增长，随着越来越多的开发者从JavaScript转向TypeScript，其未来的发展更值得期待。

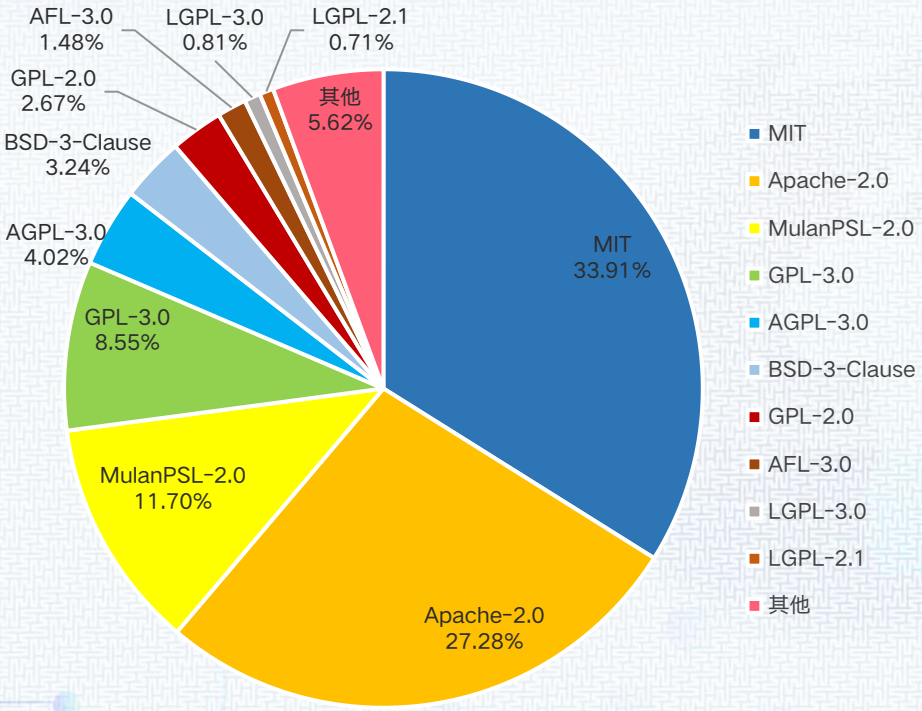
本年度增长最快编程语言



TypeScript连续两年成为了Gitee年度增长最快编程语言（2023年增长率为49.04%），同样持续强势的还有Rust以及C语言家族。

此外，Dart及Arduino首次上榜，符合2024年跨平台开发及机器人开发的潮流。

本年度最常用开源许可证

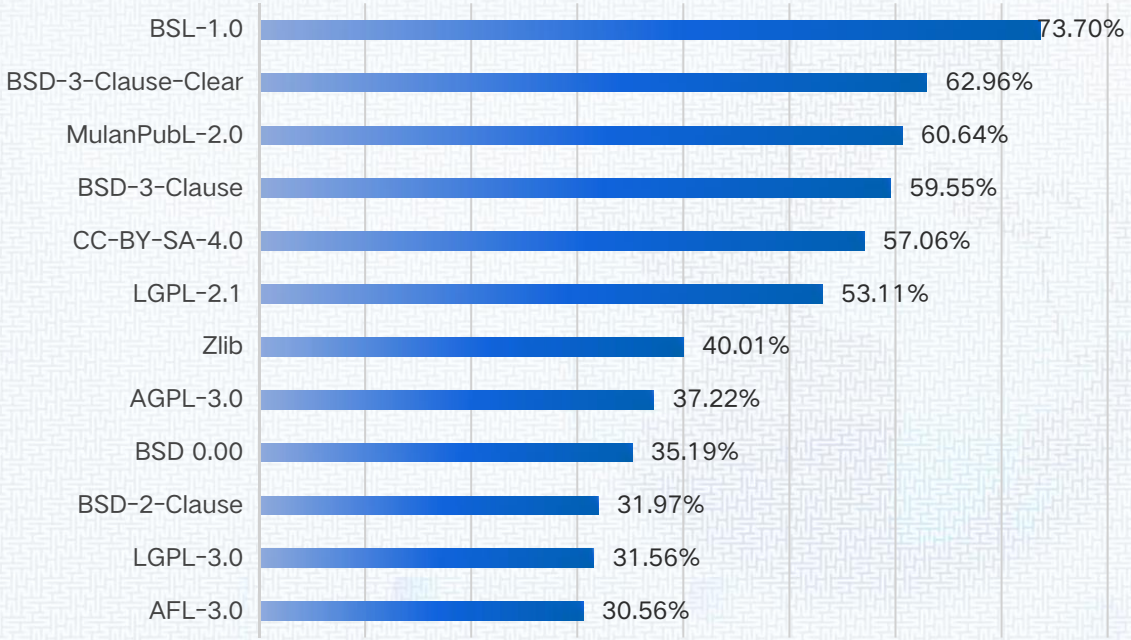


开源许可证方面，MIT及Apache-2.0依然是Gitee开发者最常用的开源许可证，使用它们作为开源许可证的仓库占比超过了61%。

木兰宽松许可证第二版（MulanPSL-2.0）紧随其后，获得了越来越多国内开发者的认可的MulanPSL-2.0已经连续两年成为了Gitee最常用开源许可证的前列。

可以预见，在未来的国内开源生态中，木兰宽松许可证将会越来越主流。

本年度使用率增长最多的开源许可证



2024年，宽松许可证依然是开发者选择的主流，BSL、BSD、CC、Zlib等宽松许可证依然增长迅速。

值得注意的是，除了LGPL、AGPL这样大家熟知的较为严格的许可证外，相比于木兰宽松许可证更加严格的木兰公共许可证第二版（MulanPubL-2.0）也在今年受到了更多关注。

OSCHINA gitee gitee AI

中国开源开发者生态数据

2024 中国开源开发者报告

China Open Source 2024 Annual Report

聚焦大模型

OSS Compass Insight

本章节数据来源：2024年1月至2024年12月
OSS Compass 平台相关公开数据

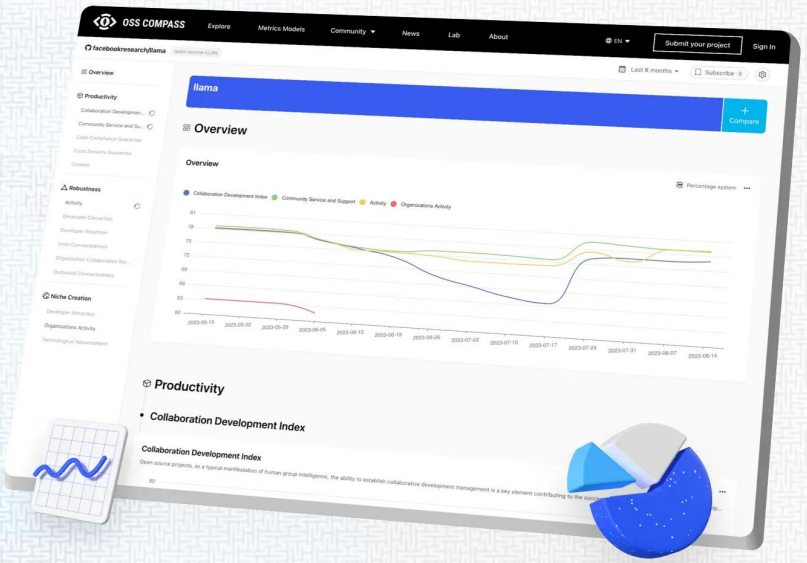
OSS Compass Insight

2024 中国开源开发者报告重点聚焦大模型，本章节以大模型 LLM 开发技术栈作为切入点，将深入探讨以下中国 AI 大模型领域的代表性开源项目社区。

这些开源项目社区覆盖了深度学习框架、向量数据库、AI辅助编程、LLM 应用开发框架、模型微调、推理优化、LLM Agent，以及检索增强生成（RAG）等多个关键技术栈。

为了更全面客观地展示中国大模型 LLM 开发技术栈的开源社区生态，我们使用了OSS Compass对开源社区的生态评估体系，希望通过这些数据洞察中国开源开发者在 AI 技术领域的活跃度、生产力和创新能力。

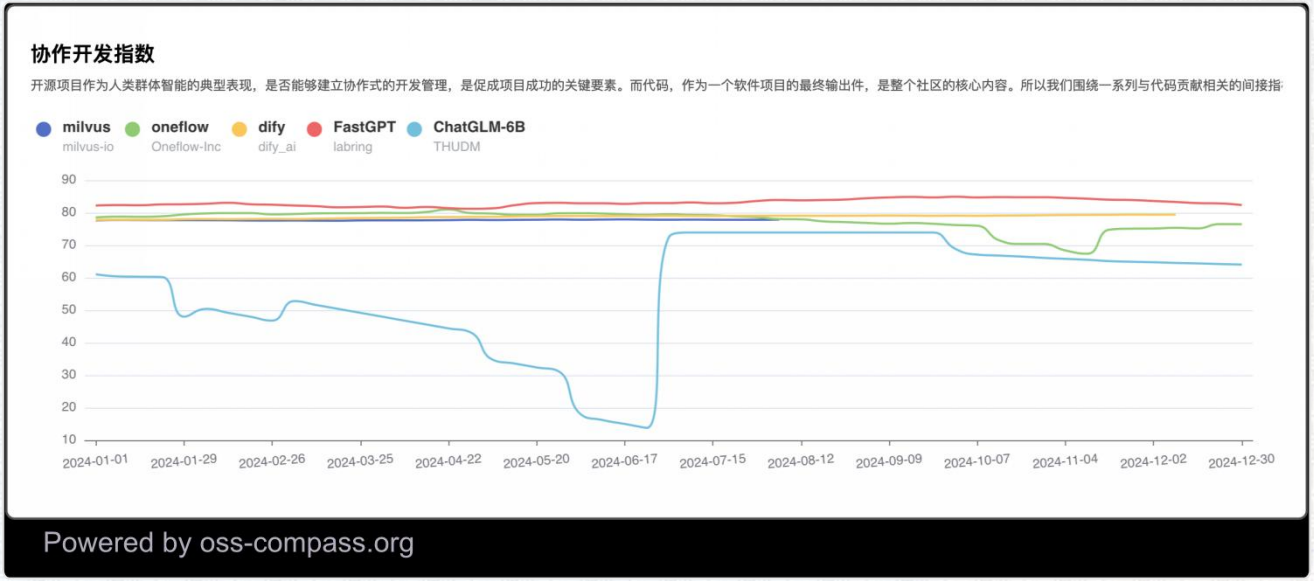
OSS Compass 提供了一个公共的平台用来评估开源项目和社区的健康度，该平台对整个社区开放，支持 GitHub 和 Gitee 等平台托管的开源项目。



OSS Compass Insight

生产力-协作开发指数

作为国内及业内领先的 AI 开发基础设施，本部分图表中的开发框架、向量数据库、开发平台、大模型均表现出色，代表着它们的代码提交频率、参与者、代码合并比率等协作开发工作保持着较高的水平。



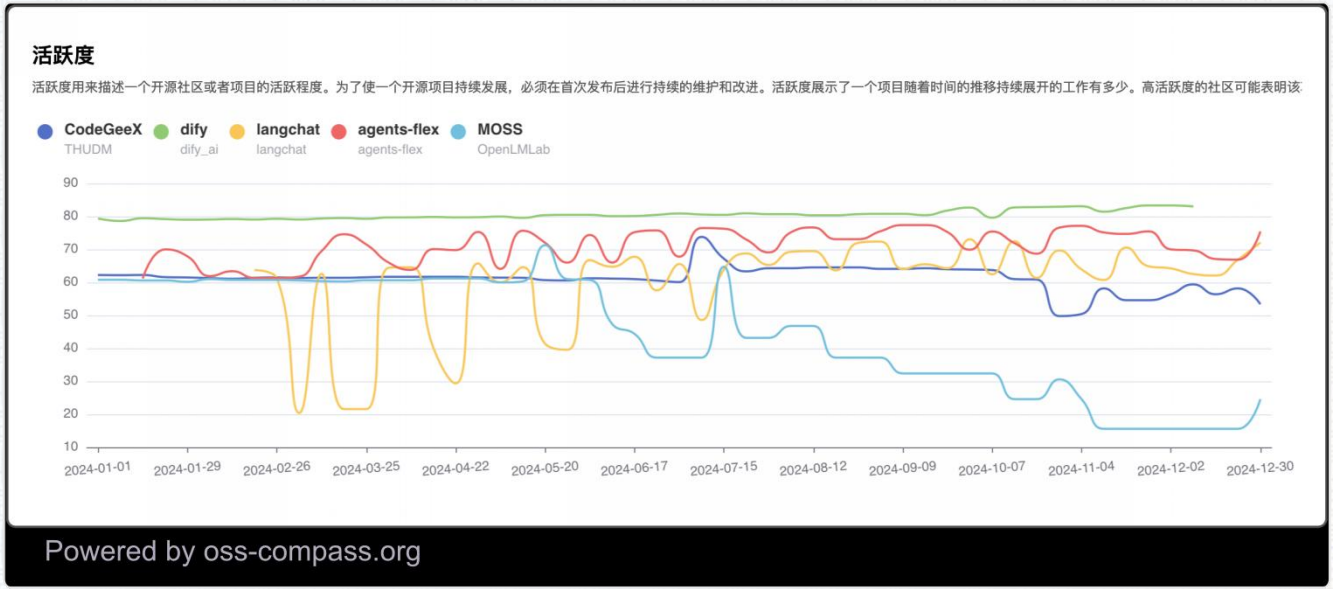
本部分图表仅用于数据展示，不涉及先后排名

OSS Compass Insight

稳健性-活跃度

作为 AI 开发生态中的关键组成部分，本部分图表中的开发框架、大模型及相关工具在活跃度的表现各有千秋。

如应用开发平台 Dify 受行业技术更新影响较小，其活跃度始终保持着较高水平；而大语言模型 MOSS 则较易受技术更新影响，活跃度随时间整体呈下滑趋势。



本部分图表仅用于数据展示，不涉及先后排名

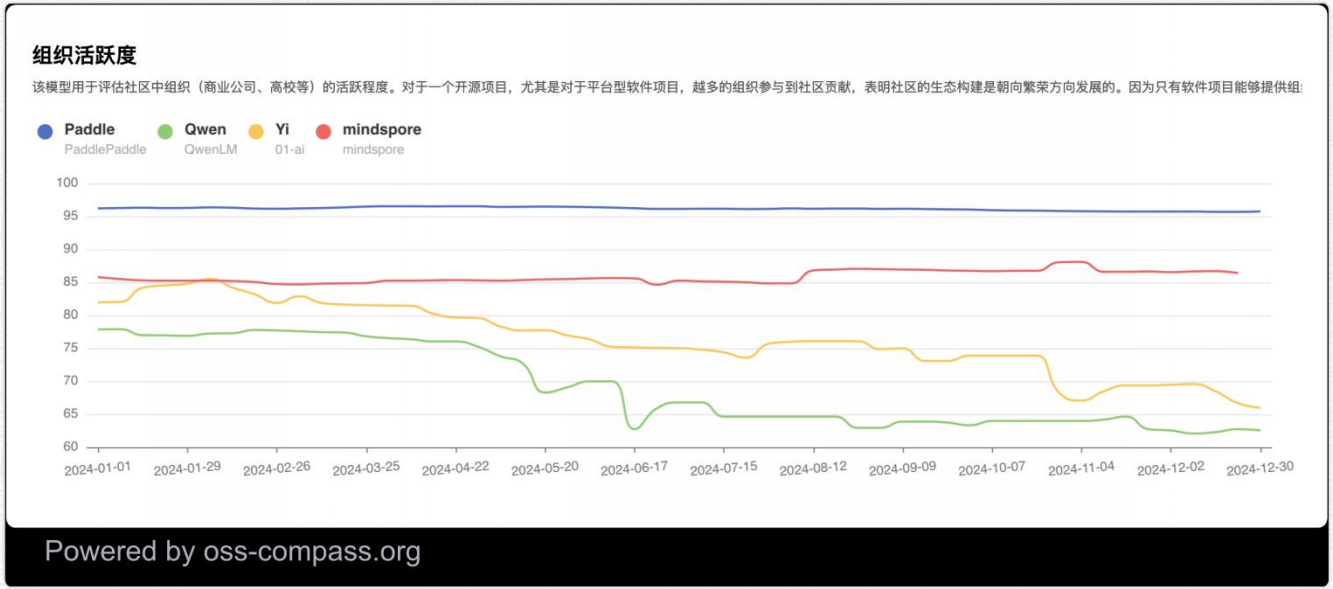
OSS Compass Insight

创新力-组织活跃度

开源组织的活跃程度成为衡量社区生态建设是否繁荣的重要指标之一。

本部分图表中的多个组织在社区活跃度上表现各有差异。

如某些组织在开源项目中长期保持较高的贡献水平，展现出其对生态建设的持续支持；而部分组织的活跃度则随时间推移逐渐下降，可能受到内部资源调整或技术方向变化的影响。



本部分图表仅用于数据展示，不涉及先后排名

<Part 2: TOP101-2024 大模型观点>

本章汇集了来自不同领域专家和开发者对开源大模型和人工智能技术的深刻见解，不仅涵盖了技术层面的深入探讨，也触及了社会、伦理和政策层面的广泛议题。

从对中国开源模型崛起的分析，到对开源模型持久性的思考，再到对超级应用探寻之路的探索，每篇文章都为我们提供了独特的视角，帮助我们理解开源大模型在 AI 技术领域的作用和影响。

2024 年中国开源模型：崛起与变革

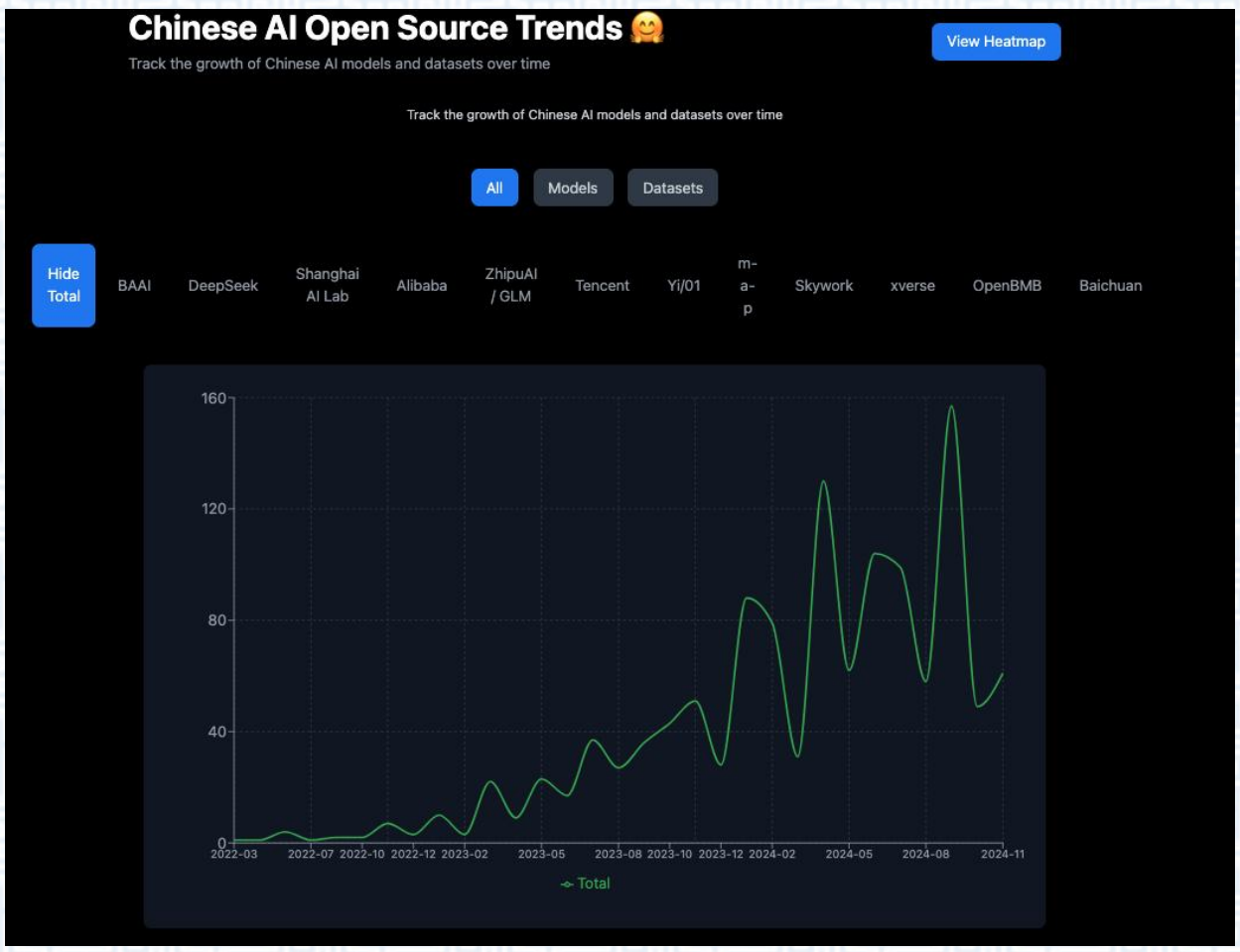
文/Tiezhen、Adina、Lu Cheng

2024 年，中国在开源人工智能模型领域的崛起和变革成为全球瞩目的焦点：从学术到产业，从技术到生态，中国通过自主研发和协同创新，逐步完成了从“追随者”到“引领者”的转变。这种转变不仅是技术实力的体现，更是中国人工智能生态系统快速完善的真实写照。以下，我们将从崛起与变革两个维度，探讨中国开源模型在这一年取得的重大成就和未来展望。

崛起

从“追随者”到“引领者”

2024 年，中国学术界和产业界大力推进自主研发，在技术创新和模型能力上实现了显著飞跃，并在全球范围内取得了显著成就。 Hugging Face Open LLM 排行榜数据显示，从智谱的 GLM 系列、阿里巴巴的 Qwen 系列到深度求索的 DeepSeek 系列，这些自主研发的模型在国内外各项评测中表现卓越。



/

Hugging Face

https://huggingface.co/spaces/zh-ai-community/zh-model-rel

其中，Qwen 系列凭借灵活的多尺寸选项，强大的多语言支持以及友好的模型授权功能，赢得了社区开发者的高度评价。DeepSeek 通过引入多头潜在注意力（Multi-head Latent Attention，MLA）技术，在性能和成本上实现了革命性突破，开创高性价比的 AI 新纪元。

智谱的 CogVideoX 系列文生视频模型，成为全球首批开源的文生视频模型之一，不仅在技术方面让中国视频生成模型列入领先梯队，强化了中国模型在全球范围的竞争力，也为国际开源生态的发展产生了积极的影响，为全球开发者提供了更多创新和应用的可能。

中国开源模型从最初的质疑中崛起，逐步赢得了广泛认可。这不仅彰显了中国开源模型从追随者到行业引领者的跨越式成长，也为全球人工智能发展注入了新的活力与动力。中国开源模型的成功并非偶然。在政府对人工智能产业的持续支持以及国内人工智能行业对模型研发的巨额投入下，从基础算法到行业应用、从算力基础设施到数据资源整合，中国人工智能生态体系正在迅速完善。这一趋势表明，未来中国有可能在全球人工智能领域占据更为核心的地位。

开源生态的繁荣与协作

随着开源模型影响力的提高，中国开源社区的活跃度也明显提升。无论是企业、研究机构还是个体开发者都更加积极地参与到开源工作中。

以阿里巴巴的通义千问 Qwen 为例，据不完全统计，截止 2024 年 9 月，全球已有近 8 万基于 Qwen 的衍生模型，超越了 Meta 的 Llama。该系列模型已被集成到 Hugging Face Transformers、Hugging Chat 和阿里自家的百炼平台中，极大促进了全球开发者的交流和协作，形成了国际化开源生态。

北京智源研究院和上海人工智能实验室等研究机构，通过与企业和高校合作及开源平台的建设，建立了更完善的协作机制，从而在开源模型 (如 InternLM) 和数据集 (如 Infinity-MM) 领域贡献了大量有影响力的基础工作和资源。

2024 年，中国开源社区涌现出众多高质量的自发研究成果。其中，MAP 团队推出的全开源模型 Map Neo 引人瞩目。该模型在训练数据、脚本以及模型对齐工作上实现了全面公开，成为国内少有的真正意义上完全开源的项目。

而 InstantX 团队的 InstantID 则作为中国模型在国际开源社区的 2024 年首秀，一经发布便获得了广泛关注，为中国模型在全球开源生态中赢得了更多认可。

平衡发展与合规创新

中国在推动人工智能技术发展的同时，也在监管层面努力建立了完善、透明的治理机制。这种监管创新为开源模型的发展提供了稳定的政策环境，同时确保技术应用符合社会价值导向。比如 《人工智能示范法 2.0（专家建议稿）》对于免费且已开源方式提供人工智能研发的个人和组织给予减轻或免承担法律责任；《生成式人工智能服务管理暂行办法》 则明确了人工智能技术的使用和合规要求，促进了开源模型在合规框架下良性发展。

变革

端上模型的兴起与隐私保护

随着小型模型的性能逐步增强，更多高级 AI 正转向在个人设备上运行。这一趋势不仅显著降低了云端推理成本，还提升了用户隐私控制。

中国 AI 社区在这一领域也做了重要贡献，推出了如 Qwen2-1.5B、MiniCPM 系列和 DeepSeek Janus 等多款移动友好型模型。其中，最新发布的 GLM Edge 1.5B 模型通过与高通 GenAI 扩展的联合优化，在搭载骁龙 8 Gen 4 处理器的手机上实现了每秒 65 个 tokens 的推理速度，接近人类语音的平均输出速率。尽管存在电池续航和内存占用过大等挑战，端上模型代表了 AI 技术隐私保护和成本优化的未来方向。中国在这一领域的探索，为行业提供了宝贵经验。

推理扩展法则的潜力释放

通过推理扩展法则，模型性能可通过延长〝思考时间〞而进一步优化。这一技术模拟了人类〝深思熟虑〞的过程，显著提升了模型在逻辑推理和复杂任务中的表现。

中国开源社区在逻辑推理领域推出了许多创新项目，包括阿里巴巴国际的 Macro-o1、通义千问团队的 QwQ、上海人工智能实验室的 LLaMA-O1 和清华大学的 Llama-3.2V-11B-cot。这些模型不仅在技术上各具特色，还通过开源策略分享了大量研究细节，为整个开源社区提供了

丰富的资源，在这一过程中，小模型不仅在推理能力上有了显著提升，也推动了行业整体技术水平的进步。

结合当前人工智能产业界的“人工智能+”计划，小模型在特定任务优化上的优势愈发突出，预计将在金融、医疗和工业自动化等热门领域发挥引领作用，以更高效、更精准的方式满足多样化需求，帮助人工智能在实际应用场景中落地。

开源多元化与应用细分

中国开源模型的发展不仅体现在技术突破上，还在生态建设中展现出巨大的活力。中国开源模型从竞争激烈的“百模大战”逐步迈向多元化和深度细分，国内社区在今年发布了大量高质量开源模型，尤其是多模态理解与生成模型：

多模态理解：Qwen2-VL、Ovis、InternVL2、DeepSeek JanusFlow、GOT-OCR2_0；

图片生成：PixArt、Lumina、Kolors、Hunyuan-DiT、VAR、Meissonic；

视频生成：AnimateDiff-lightning、Latte、OpenSora、open-sora-plan、Pyramid Flow、CogVideoX；

TTS：GPT-SoVITS、ChatTTS、CosyVoice、FishAudio、MaskGCT、F5-TTS 。

这一趋势表明，模型的竞争已经从单纯的规模比拼转向应用场景细化。为了更好地展现这一演进路径，我们在 Hugging Face 的中文模型社群中对各个领域的开源模型进行了系统整理。

展望

2024 年，中国开源模型的发展展现了技术、生态和社会价值之间的深度协同。无论是从技术创新到社区建设，还是从行业实践到合规探索，中国开源生态体系的完善正在为全球人工智能发展注入源源不断的动力。

在 Hugging Face，我们坚信开源是推动人工智能技术进步和生态繁荣的核心力量。开源不仅能够打破技术壁垒，促进全球开发者之间的协作与创新，还能推动技术的普惠化，让更多的

人能够平等地享受人工智能带来的便利与机遇。

在未来，中国开源模型有望继续引领全球技术进步，为全人类的智能化生活提供更丰富的解决方案与可能性。我们希望看到更多来自中国的开源 AI 团队“出海”，积极融入和参与全球人工智能生态，勇于在全球市场发声，通过开放协作推动技术边界的不断拓展，共同构建一个更加包容、多元与可持续发展的人工智能的未来。

Tiezhen

现任 Hugging Face 工程师，曾在 Google Brain 任职。兼具实干精神与梦想追求，坚信开源是连接全球的纽带，让 AI 的益处普惠大众。他秉持"高手在民间"的理念，渴望激励更多的开源模型从业者成为行业的关键意见领袖，挖掘群体的智慧与潜能，促进社区的成长和影响力的扩大。

Adina

Hugging Face 中文社区项目经理。拥有 10 年以上国际化工作经验，足迹遍及亚洲、非洲和欧洲。从社会科学研究员到科技公司项目专员，积累了丰富的跨领域与跨文化经验。专注推动人工智能在中文开源社区的应用与发展，为开发者和企业带来更多价值，助力知识共享与技术协作。

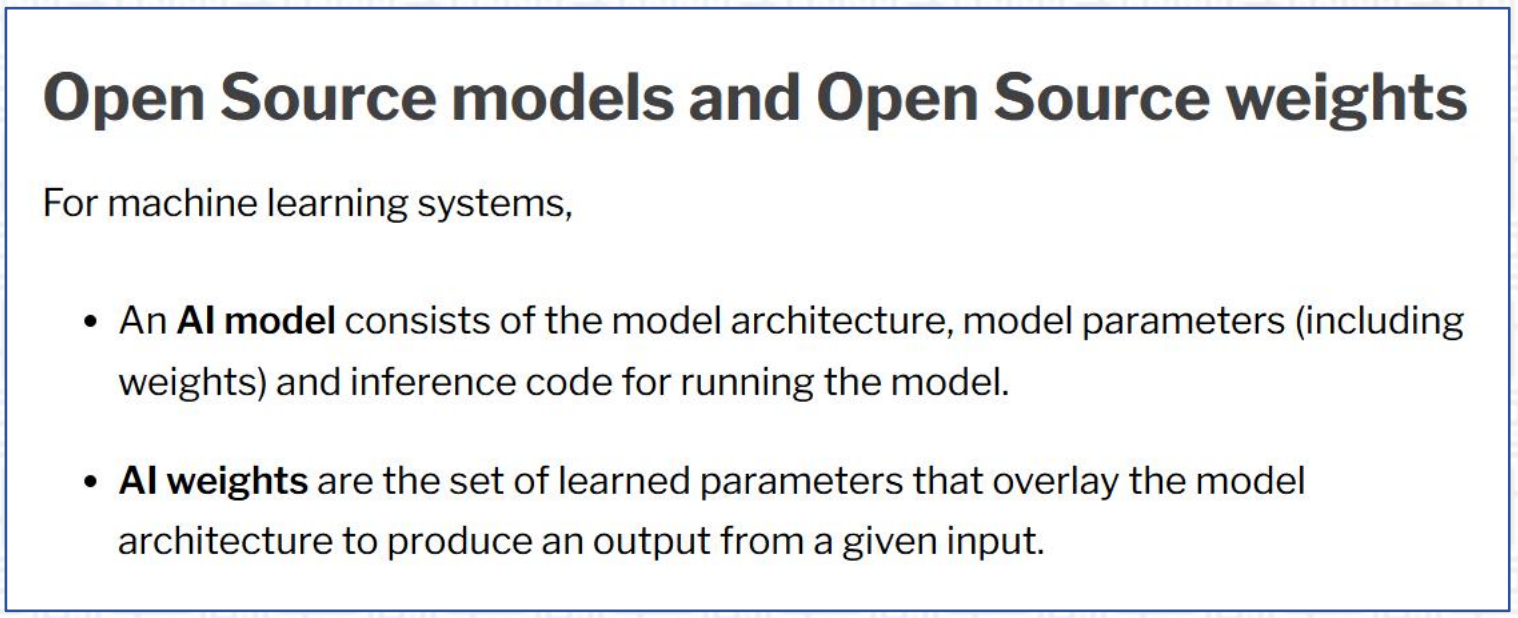
Lu Cheng

Hugging Face Fellow，致力于推动 AI 和开源软件的采纳和开发者体验。拥有超过十年的开发者关系、产品营销和开源生态构建的经验，曾在 Google 负责多个开发技术的深度推广和社区建设，包括 Android、Flutter 和 TensorFlow 等。他坚信开源是推动技术进步和开发者成长的关键步骤，希望有更多人参与开源和社区共建。

开源模型未必更先进，但会更长久

文/顾钧

“开源”是指采用符合 OSI 官方认可的软件许可证进行软件发布的行为。目前大模型的“开源”与传统的开源定义并不相同。我所说的开源策略是指以开源发布软件为起点，用户/开发者运营为途径的软件产品推广策略。



Open Source models and Open Source weights

For machine learning systems,

- An **AI model** consists of the model architecture, model parameters (including weights) and inference code for running the model.
- **AI weights** are the set of learned parameters that overlay the model architecture to produce an output from a given input.

我的观点是，开源策略是大模型最好的竞争策略。接下来让我们从头捋一捋推导过程。

我们先看大模型赛道的整体状况：

- 大模型是一项相对较新的技术。尽管 OpenAI 早在 2019 年就发布了第一个重要的模型 GPT-2，但大模型的广受关注实际始于 2022 年 11 月发布的 ChatGPT。8 个月以后 Meta 就与微软合作发布了开源大模型 LLaMA-2。这个赛道的主要玩家在技术和商业化上有差距，但没有到翻盘无望的程度。

- 大模型赛道不但包括模型的训练，也包括模型服务。训练是软件的制作成本，而服务是软件的长期运行成本。

- 大模型赛道的市场化程度非常高。算法、算力、数据、人才，这些构建大模型的基础要素并不为权力机构垄断，大多要从市场上获得。

- 大模型作为一项令人激动的技术，商业化场景覆盖了对企业(2B)与对个人(2C)两个大方向。

- 大模型赛道在海外是“一超多强”，在国内则是“多头并举”，两种典型的竞争格局都全了。

以上，大模型赛道的元素非常丰富，各种商业化方法的排列组合都不缺，为我们的分析与推演提供了可贵的素材。对软件商业化问题感兴趣的朋友一定要长期关注这个赛道。只有这样的对象才能更有力地说明开源策略的重要性。

其次，我们得明确一点——大模型竞争的赛点是什么？常用的判断依据包括：技术的先进性，C 端用户基数，依赖这个软件的生态系统大小等等。其中哪个更关键一点？

技术先进是好事，但大模型领域的先进技术远没有达到能为大模型企业带来可观收入的程度。整个大模型赛道还处在商业化的摸索阶段。这个时间点上的“技术先进性”更多是用于公关宣传的素材。考虑到数据获取、加工的成本，模型训练的成本，这是一种相当昂贵的宣传方式。

C 端用户指那些把大模型当成智能个人助理来使用的普通个人用户。OpenAI 在 ChatGPT 上一个重要且成功的操作就是把大模型从学术界、工业界直接推向了普通个体，让 C 端用户切实感受到了大模型的可能性与魅力。这一点被国内的大模型厂商广泛学习。在 B 站刷视频，国内知名的那几个大模型厂商的广告，你一个也不会落下。

受到大家的认可与喜爱固然重要，但对于 C 端用户，有两个需要时刻牢记的问题：一是 C 端用户是没有忠诚度的，谁免费就用谁，谁给补贴就用谁；二是某一个大模型对 C 端用户比较难产生独特的粘性。

第一个问题的例证太多了，百团大战、滴滴快的、社区团购、pdd。大模型厂商维系 C 端流量的成本可能是个无底洞。

第二个问题则涉及两个方面，一是大模型赛道本身的极度内卷，技术上拉不开差距；二是普通用户的使用随意性很强，准确性要求也不高，最终各家大模型的基础能力都足以应付。

一个大模型的生态系统的大小，也就是指有多少开发者在基于这个大模型构建应用。我认为这是一个更靠谱的评价指标，是某个大模型最终能胜出的关键所在。

构建开发者生态通常有两种做法，一种是提供 API 云服务，对注册开发者进行一定的云资源补贴；另一种是“开源”的方法，提供大模型免费下载，免费商用（一定条件下）。两种方法各有支持者。闭源大模型一般会采用第一种方法，其中的代表有 OpenAI、Anthropic 等（为避免麻烦，国内厂商的名字就不提了）。能用第二种方法的，必然是某种程度上的“开源”模型，以 Meta 的 Llama 2、Llama 3 模型为首。

前段时间李彦宏在 Create 2024 百度 AI 开发者大会上放言“开源模型会越来越落后”。前文我有提到，此时此刻的技术先进性并不重要。甚至在计算机发展史上，很多领域中笑到最后的产品，并不是技术上最先进的。抛开成本和易用性，空谈技术先进性是最常见的错误。

那么具体到大模型领域，闭源与开源，两种方法孰优孰劣？我的回答是采取什么方法因人而异，但开源会更有优势。

大模型赛道的核心制约条件是成本太高——训练成本高，运行成本高。如何尽可能降低成本，

比对手坚持得更久一些是确保长期成功的必要条件。现在的宏观环境下，一味靠融资来支撑自己的高成本支出不是长久之计。

闭源大模型厂商必须维持一定的云资源，工程师资源来支撑小额的开发者调试需求。投入产出上恐怕是算不过来的。即便闭源厂商愿意持续地补贴开发者，他们最终会发现大模型对开发者的粘性也非常有限，没比在 C 端用户那边好到哪里去。

大模型这一产品形态实在是太特殊了——大多以自然语言为交互方式。因此大模型 API 云服务的接口是非常简单的，高度一致的。在这种情况下，如果开发者构建的大模型应用只是调用大模型的 API，那么大模型应用与某个具体的大模型之间很难形成强绑定。也就是说，面对各种大模型云服务，主动权在开发者这里。

与之相对，开源的方法至少可以相当程度地省去为了拓展开发者生态而付出的大模型运行成本。开发者免费下载大模型以后，会在自己的计算机资源上进行大模型应用的开发和调试。大模型厂商提供一些技术支持即可。同时因为大模型运行在本地，开发者在构建大模型应用时，为了物理部署上的便利，很可能会在应用与模型之间创造出物理部署上的耦合性。

当然这种“开源策略”不是进攻的方法，而是“先为不可胜，以待敌之可胜”。目标是以最小的代价，尽可能多地消耗闭源对手的资源与心气。

顾钧

资深开发者社区运营专家，目前担任杭州映云科技（EMQ）市场&开发者社区总监一职。

2004 年，顾钧从北京大学计算机系本科毕业，其后在工商银行、IBM、摩根士丹利、华为和 Zilliz 等多家知名企业工作。

曾联合发起全球首个开源向量数据库项目 Milvus，并帮助 Milvus 社区在两年间迅速拓展到两千家企业用户。

大模型撞上“算力墙”，超级应用的探寻之路

文/傅聪

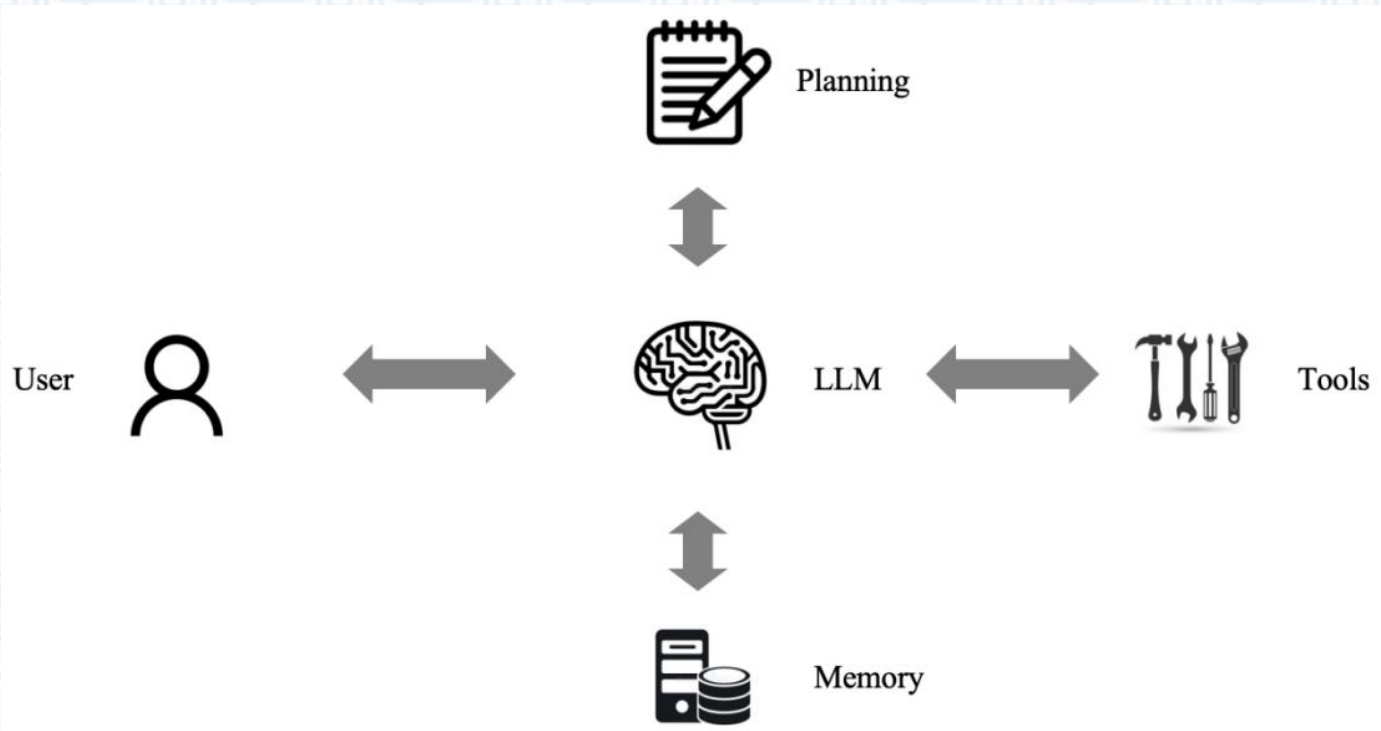
近日，大模型教父 Sam Altman 在 Reddit 上的评论透露出 GPT-5 难产的隐忧，直言有限的算力约束让 OpenAI 面临迭代优先级的艰难抉择，在通往 AGI 的道路上一路高歌猛进的领头羊似乎撞上了“算力墙”。

除此之外，能耗、资金，难以根除的幻觉，有限的知识更新速率、有限的上下文宽度、高昂的运营成本等等，都让外界对大模型的发展忧心忡忡。面对棘手的困境与难题，大模型的未来，又该何去何从呢？

下一代“明星产品”

“算力墙”下，模型效果边际收益递减，训练和运营成本高昂，在这个时间节点，最好的 AI 产品会是什么？奥特曼、盖茨、小扎、吴恩达、李彦宏等一众大佬给出了一致的答案——智能体（AI Agent）。2025，将会是智能体元年。

什么是智能体？目前业界一致认可的公式是“智能体=LLM+记忆+规划+工具”：



大模型充当智能体的“大脑”，负责对任务进行理解、拆解、规划，并调用相应工具以完成任务。同时，通过记忆模块，它还能为用户提供个性化的服务。

智能体为什么是“算力墙”前 AI 产品的最优解决方案？这一问题的底层逻辑包含两个方面。

1. LLM 是目前已知最好的智能体底层技术。

智能体作为学术术语由来已久，从上世纪的“符号、专家系统”[1]，到十年前风头无两的强化学习（代表作 AlphaGo[3]），再到现在的 LLM，agent 底层技术经历了三个大的阶段。

符号系统的缺点在于过于依赖人工定义的“符号”和“逻辑”，强化学习苦于训练数据的匮乏和“模态墙”，而 LLM 一次性解决这些问题。

人类语言就是一种高度抽象、跨模态、表达力充分的符号系统，同时它作为知识的载体，自然地存在大量数据可用于训练，还蕴含了人类的思维模式。

在此基础上训练得到的 LLM，自然具备被诱导出类人思考的潜力。在 COT（思维链）[4]、TOT（思维树）[5] 等技术的加持下，大模型正在学习拆解自己的“思维”，OpenAI 的 o1 就是典型案例，强化了推理能力的同时，也大大缓解了幻觉问题。

2. 大模型做不到的，“现存工具”强势补位。

无法持续更新的知识库，可以通过 RAG（Retrieval Augmented Generation，检索增强生成）来解决。

RAG 的出现，让各界越来越深刻地认识到，大模型没必要存储那么多知识，只需要如何使用搜索引擎这个外部工具即可。大模型可以在搜索结果上做进一步的信息筛选和优化，而搜索引擎弥补了大模型的知识缺陷，实现了 1+1>=2 的效果。

RAG 可以被理解为智能体的最简单形式。未来的智能体可以实现多种工具的混合使用，甚至多智能体协作，这不是猜想，我们已经在学术界看到了惊艳的早期方案[6, 7]。

“四把钥匙”解锁潜力

1. **领域模型小型化、平台化会成为新趋势。**

“算力墙”是一方面因素，但基座模型的趋同化和运营成本是源动力。GPT、Claude、Gemini 虽然各有所长，但实际体验越来越让大家分不出差异，基座模型作为智能体核心，决定了智能体效果下限，人人训练基座的可能性越来越低，“基座服务化”很可能是最合理的商业模式。

甚至，在错误不敏感的应用领域，出现一个开源、无商业限制的基座的可能性也很高。小应用开发商很可能很容易获得一个低成本 serving 的“量化小基座”。

“7B”是一个 magic number！无论是 RAG 里的向量表征模型，还是文生图、文本识别（OCR）、语音合成（TTS）、人脸识别等等垂直领域，一个 1B~7B 的小模型已经可以满足很多生产、应用需要，并且效果也在逐步推高【8，9，10】。这些模型，作为智能体的“三头六臂”，不需要太“大”。

同时，从学术角度来讲，各种领域专用模型的技术最优解也在逐渐趋同。应用开发者越来越不需要了解模型的底层技术，只需要懂得如何设计自己应用的任务流，懂一点点 COT 系列的 prompt engineering 的技巧，就可以利用 Maas（Model as a service）、Aaas（Agent as a service）这样的平台，如玩乐高一般搭建自己的 AI 云原生应用。

2. **算力层深挖定制化、低能耗的可能性，但固化 transformer 可能不是最优解**

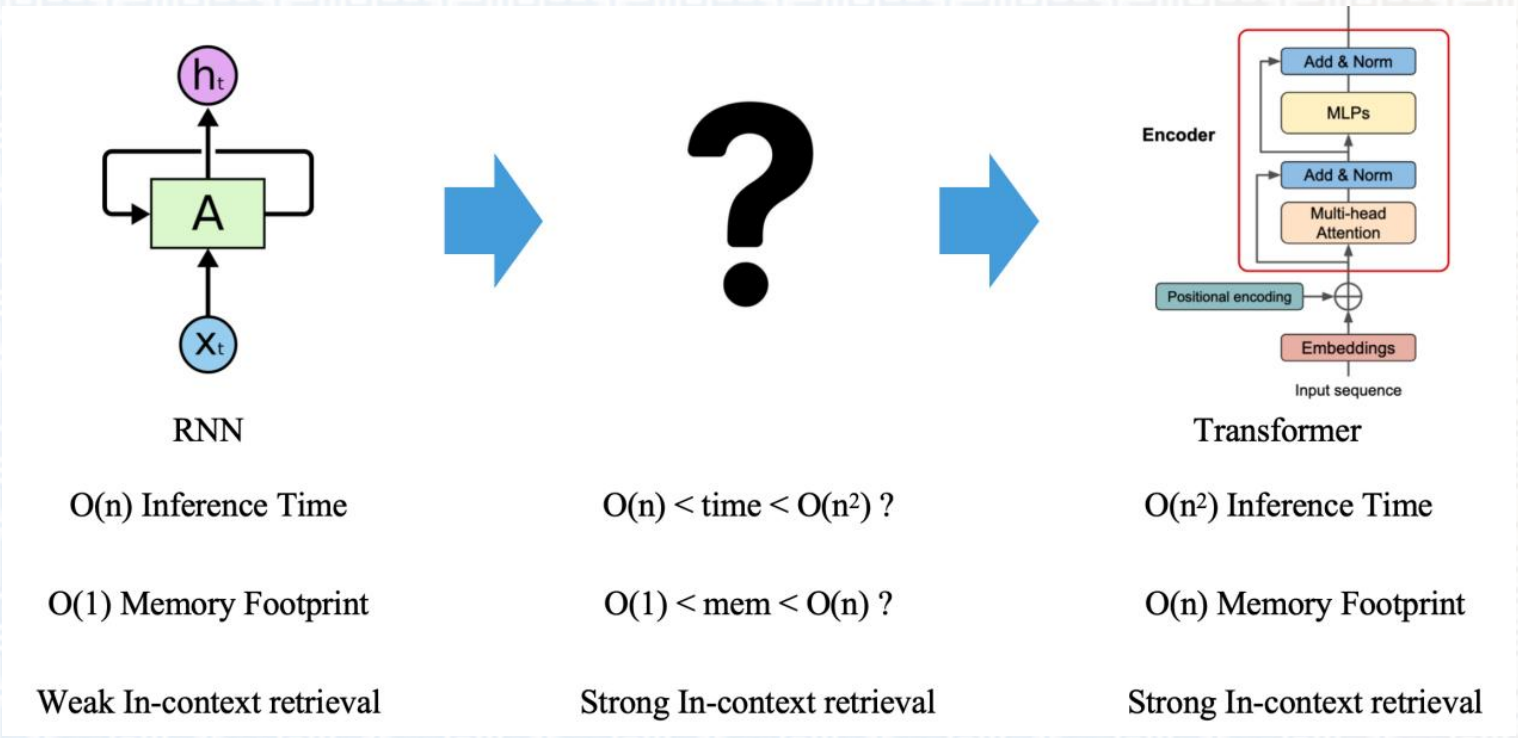
虽说智能体不需要太大的模型，但其运营成本（模型推理计算成本）仍然较高。在短时间内，算力、能源仍然会是大模型领域令人头疼的高墙。

根据报告【1】，能源消耗将会是 2030 模型 scaling 最卡脖子的因素。也就是说，在算力到达瓶颈之前，首先可能会出现电能供应不足甚至交不起电费的问题。因此，算力层可以根据大模型底层技术的特性，产出针对性的芯片，尤其是加速运算和降低能耗。这是未来 AI 芯片领域的最优竞争力。

那么，把 transformer“焊死”到板子上就是最佳方案吗？我知道你很急，但你先别急。大模型底层框架还存在底层路线之争。

我们知道，Transformer 架构呈现了 $O(n^2)$的理论计算复杂度，这里的 n 指的是大模型输入序列的 token 数量，但其前任语言模型担当 RNN 只有 O(n)的理论计算复杂度。

最近，以 Mamba、RWKV 为代表的类 RNN 结构死灰复燃，公开挑战 transformer 地位。更有最新研究[13]从理论上表明，RNN 对比 Transformer 的表达力，只差一个 in-context-retrieval。在这个方向的持续投入下，我们很可能会迎接一个介于 RNN 和 Transformer 之间的“新王”。

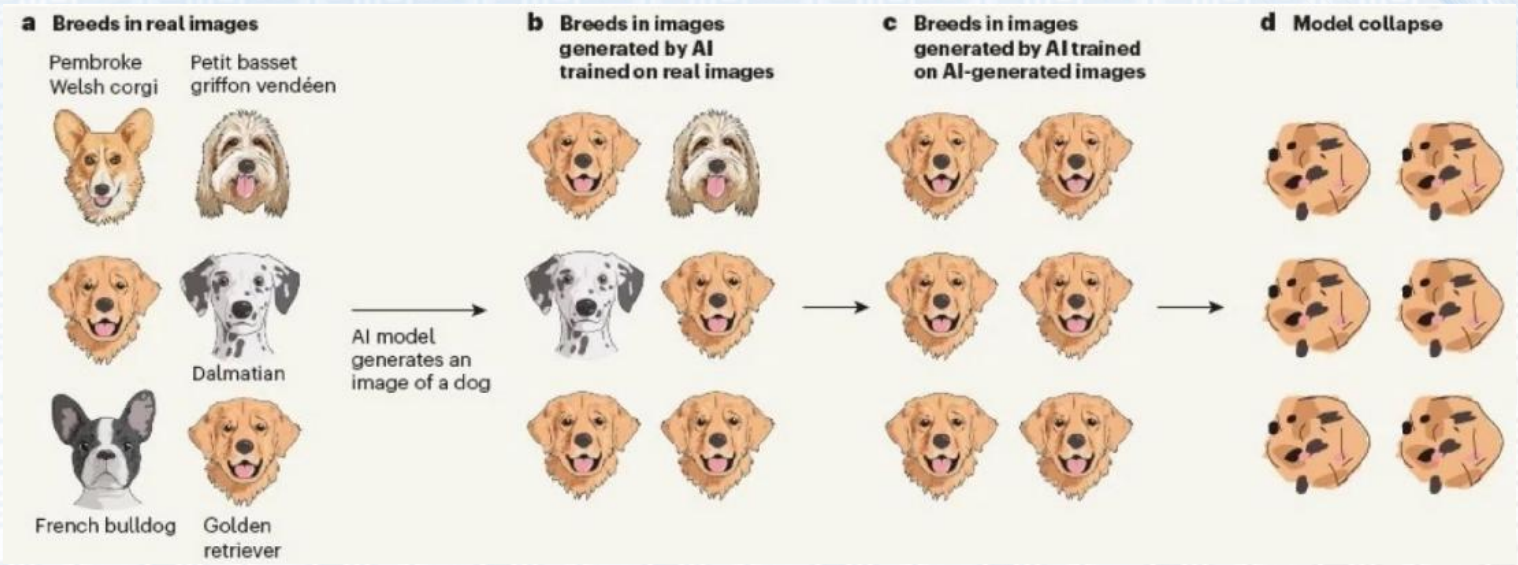


因此，算力层短时间内的主题仍然是“半通用化”“高算力”“低能耗”。

3. 合成数据驱动新产业链

早有机构预测，人类社会可利用训练数据会在 2026 年耗尽。这可能还是一个乐观估计。光头哥 Tibor Blaho 还曾爆料，OpenAI 用于训练“猎户座“的数据中，已经包含了由 GPT-4 和 O1 产出的合成数据。

这不仅是因为自然存在的高质量文本的匮乏，还因为智能体所需的数据很可能需要显式地蕴含任务思考和规划的拆解信息。然而，针对合成数据的问题，学术界早有预警，模型可能会在合成数据上的持续训练中崩坏[14]。

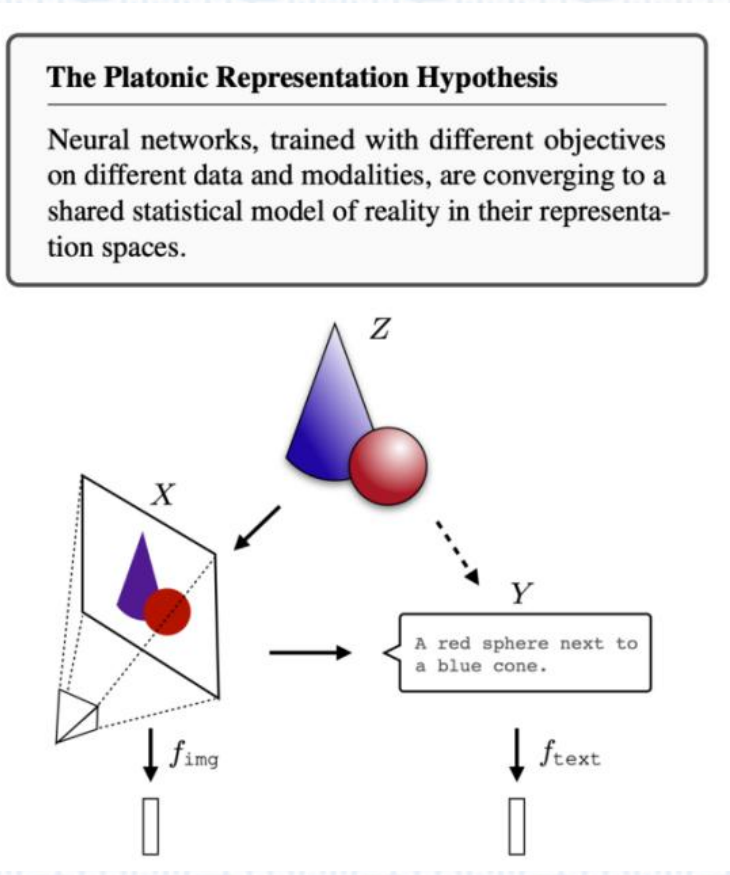


这是因为合成数据往往携带“错误”和“幻觉”，在一些冷门的知识上尤甚。因此，合成数据的实用秘诀是“去粗取精”，需要一定程度的“人机协同”。在如何构造大批量、高质量的合成数据，让智能体能够在持续地与用户的交互中自我优化而不是劣化，将会成为众多无机器学习技术背景的开发者的头号难题。

因此，面向数据进行定制化合成、评估、测试、标注、人机协同的“纯数据”产业，有可能会走上越来越重要的位置，不仅仅是服务于基座模型厂商。

4. 多模态对齐很可能给基座模型带来质的提升

最新研究发现，在没有预先约束和约定下，不同模态领域的最强模型正在向着某个世界模型认知领域收缩【15】，AI 模型对不同概念的数字化表达（向量表征）会逐步趋同，构建对这个世界的统一认知。这也符合我们人类对世界的认知：人类通过语言文字这种符号，将不同模态的信号统一地表达，并在脑中构建了某种受限于当前科技水平的统一模型，这是人类意识、社会沟通的前提。



从这个角度理解，多模态大模型很可能是通向真

正 AGI 的必经之路。将多模态信号统一对齐，是智能体与这个世界“无障碍”交互的前提，换个新潮的词汇，就是我们期待的“具身智能”。谁不想拥有一台自己专属的“Javis”呢？而多模态大模型的突破，也同样依赖前文所述的算力和数据上的沉淀。

参考文献

【1】https://epoch.ai/blog/can-ai-scaling-continue-through-2030
【2】Newell, A., & Simon, H. A. (1956). The Logic Theory Machine – A Complex Information Processing System. IRE Transactions on Information Theory, 2(3), 61-79.
【3】Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." nature 529.7587 (2016): 484-489.
【4】 Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." Advances in neural information processing systems 35 (2022): 24824-24837.
【5】Yao, Shunyu, et al. "Tree of thoughts: Deliberate problem solving with large language models." Advances in Neural Information Processing Systems 36 (2024).
【6】Karpas, Ehud, et al. "MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning." arXiv preprint arXiv:2205.00445 (2022).
【7】Schick, Timo, et al. "Toolformer: Language models can teach themselves to use tools." Advances in Neural Information Processing Systems 36 (2024).
【8】https://huggingface.co/spaces/mteb/leaderboard
【9】https://github.com/deep-floyd/IF
【10】https://developer.nvidia.com/blog/pushing-the-boundaries-of-speech-recognition-with-nemo-parakeet-asr-models/
【11】Mamba: Linear-time sequence modeling with selective state spaces
【12】Peng, Bo, et al. "Rwkv: Reinventing rnns for the transformer era." arXiv preprint arXiv:2305.13048 (2023).
【13】Wen, Kaiyue, Xingyu Dang, and Kaifeng Lyu. "Rnns are not transformers (yet): The key bottleneck on in-context retrieval." arXiv preprint arXiv:2402.18510 (2024).
【14】AI Models Collapse When Trained on Recursively Generated Data'
【15】The Platonic Representation Hypothesis

傅聪

浙江大学计算机博士，美国南加州大学访问学者，《业务驱动的推荐系统：方法与实践》作者。高性能检索算法 NSG、SSG 的发明者，知乎科技博主“傅聪 Cong”。

前阿里巴巴算法专家，目前就职于 Shopee（新加坡）任资深算法专家。在顶会和期刊 TPAMI、KDD、VLDB、IJCAI、EMNLP、CIKM 等发表十余篇论文，同时也是 Tpami、TKDE、KDD、ICLR、AAAI、IJCAI、EMNLP、ICLR 等会议的审稿人。

AI 的三岔路口：专业模型和个人模型

文/李博杰

2024 年大模型真正开始落地，大多数科技工作者在工作中至少使用一款大模型提升效率，很多国民级应用和手机厂商也接入了大模型。大模型开始往专业（Professional）模型和个人（Personal）模型两个方向分化。

专业模型是旨在提升生产力的模型，例如 AI 辅助编程、写作、设计、咨询、教育等。一旦模型能力达到门槛，专业模型将带来很高的附加值。

2024 年，专业模型已经在很多领域落地。例如，AI 辅助编程可以提升开发效率一倍以上，仅用每月数十美元的 API 调用成本，就相当于每月上万美元的工程师。AI 生成图片、播客、直播等，可以上百倍提升画师、配音员、主播的工作效率。AI 在心理、法律、医疗等领域的咨询服务可达到初级专业人士水平，每小时收费相比模型成本也高上百倍。AI 虚拟外教已经可以媲美真人外教，由于发音标准，效果甚至超过大多数国内英语老师。

专业模型是通用大模型和垂直领域数据、工作流的结合。这里通用大模型的基础能力是关键，一个世界领先的通用大模型加上 RAG（搜索增强生成）行业知识库，做出的专业模型效果往往超过开源模型加上一些垂直领域数据微调得到的行业模型。因此，专业模型虽然训练、推理成本都较高，但考虑到较高的溢价空间，投入是值得的。

由于通用大模型的通用性，难以建立差异化壁垒，也难以形成网络效应，因此基础模型公司的竞争将非常激烈，算力将成为长期竞争力的关键。

对于大公司而言，能否集中算力、数据和人才，保持组织高效很关键。创业公司需要更多的资金支持，或者与云计算平台或芯片厂商深度合作，才能竞争专业模型的最高水平。一个例外是图片、视频等基于扩散模型的生成模型，在创作需求简单的情况下，未必需要通用语言模型这么大，是一个差异化竞争的机会。

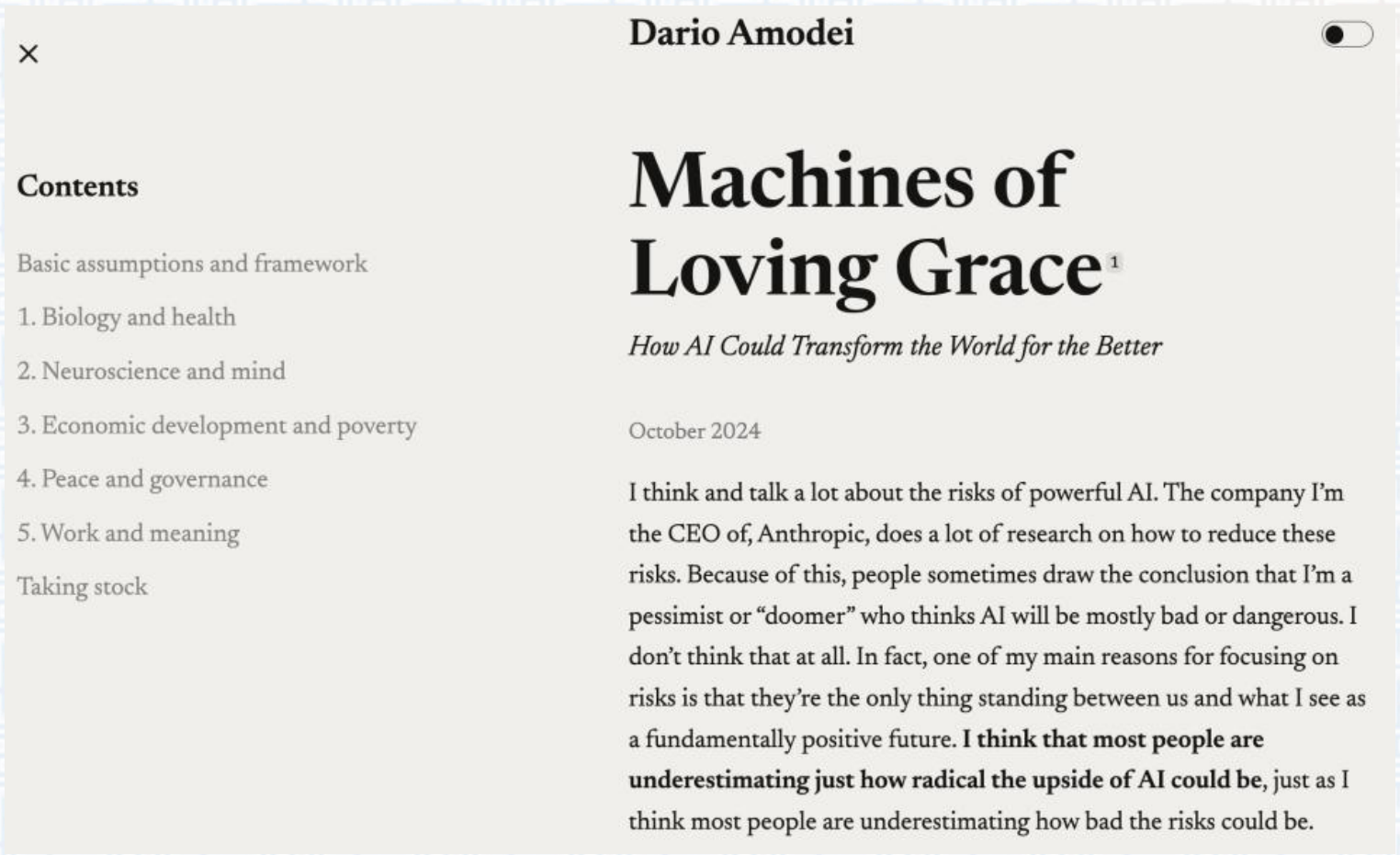
随着专业模型编程能力的提升和 AI Agent 工作流进一步成熟，低代码编程将成为可能，很

多人心中的想法将可以快速转化成应用，应用创业的试错成本已经大幅降低，未来甚至可能出现 Sam Altman 所说的“仅有一个人的 10 亿美金公司”。

由于定制化开发、知识收集整理的成本降低，大量现实世界中的工作流和行业知识将转化为行业应用和行业数据，传统行业数字化转型中的定制化开发难题有望解决。

对程序员而言，需求表达能力、沟通能力等软技能和系统架构设计等硬核能力将越来越重要，因为 AI 就像今天的基层程序员，需要人表达清楚需求才能做好，复杂系统的架构设计和问题解决也还是要靠人。

专业模型是通向 AGI 的必经之路。Anthropic CEO 预测，未来 5 年专业模型将达到人类顶尖专家水平，将人类科研进展加速 10 倍，15 年后人类寿命有望达到 150 岁。但 AGI 能否实现，最大的不确定性在于技术和资金。

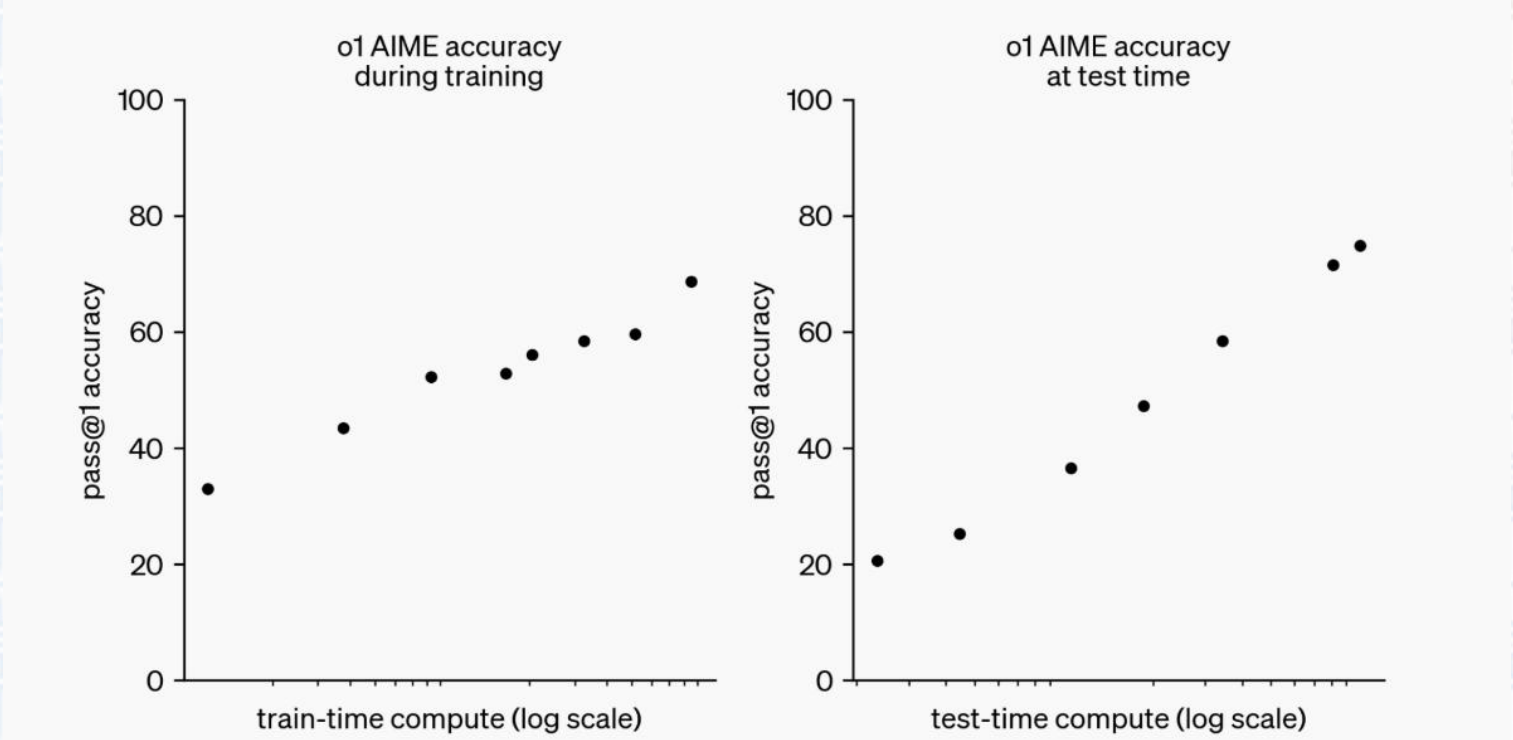


技术方面，一些头部大模型公司已经发现 Transformer 能力“撞墙”，现有高质量语料基本都被用过了，进一步提升模型智力需要强化学习等新方法。资金方面，一些智库预测，AGI 将需要上万亿美元的投资，芯片的能耗也将使人类的能源消耗增加一倍。如果 AGI 达成，将显著改变国际竞争格局和人类生活方式。

相比更类似“阿波罗计划”的专业模型，个人模型不需要那么大训练投入，也更容易变现。个人模型旨在帮助普通人提升生活质量，例如生活助手、旅行助手、电话助手等，把《Her》等科幻电影中的场景变为现实。

一般认为，同时具备 GPT-4o 多模态能力和 o1 推理能力的模型就可以满足个人模型的需求，目前国内的头部 AI 公司也已接近个人模型的技术目标。但目前端到端多模态模型和推理模型的成本仍然较高，且在一些场景下还不够稳定。

但 2023 年以来，模型知识密度有每 8 个月提升一倍的“类摩尔定律”趋势，加上硬件的摩尔定律和推理框架的优化，一到两年后，个人模型的成本将达到可以让用户随时使用的水平，就像互联网应用一样，通过广告和少数订阅即可盈利。类似 o1 的强推理能力模型也不一定需要很大，未来将成为个人模型的标配，经常算错数的模型将被淘汰。



o1 performance smoothly improves with both train-time and test-time compute

手机、PC 和空间计算设备的端侧个人模型将足够满足大多数日常需求，智能汽车可能成为家庭计算中心。云端模型将作为端侧模型的补充，用于处理较复杂的任务和处理大量数据。模型的多模态能力将使 AR/VR 等空间计算设备成为更自然的人机交互入口。推理能力将使得模型可

以可靠处理复杂任务，真正节约用户时间，甚至做到人力不能及的信息采集和分析。多模态和推理能力也将使具身智能真正具备通用的感知、规划、控制能力。

顶级的专业模型公司有最高质量的数据，因此可以蒸馏出知识密度最高的个人模型。但由于个人模型的推理成本较低，知识密度稍低的模型未必没有市场。由于训练成本较低，未来个人模型将百花齐放，AI 公司很难单靠模型本身建立护城河，产品的重要性将高于模型能力。

面向个人生活和娱乐的 AI 产品关键是用户交互，目前优秀的 AI 应用已经不简单是生成文字。在 Claude Artifacts 之后，AI 生成代码，再运行代码，生成图文并茂的回答，直观的图表，多模态带讲解的播客，甚至带交互的小游戏、小应用，已经成为 AI 应用的新范式。

在个人模型成本尚未降低到可以随意使用时，商业上成功的应用可能将有更高的“读写比”，也就是每次模型生成的内容可以被用户多次使用，一种模式是内容社区，创作者利用 AI 生成内容，大量的用户访问这些内容；另一种模式是用户的问题有很高比例是重复的，例如拍照搜题、生成调研报告等。

总体来说，目前 AI 应用尚处于“iPhone 1”时代，模型能力、应用生态、用户习惯都在快速进化中。所谓“AI 一天，人间一年”，即使是 AI 专家，也很难跟上所有最新的科研进展。大模型的时代才刚刚开始，预测未来的最好方式就是持续学习、探索、利用 AI 能力，创造未来。

李博杰

李博杰是 AI 创业者，研究方向为高性能数据中心系统。曾任华为计算机网络与协议实验室助理科学家、首届“天才少年”。2019 年，在中国科学技术大学与微软亚洲研究院的联合培养项目中取得博士学位。在 SIGCOMM、SOSP、NSDI、PLDI 等顶级会议上发表多篇论文，曾获 ACM 中国优秀博士学位论文奖和“微软学者”奖学金。

2024 年 AI 编程技术与工具发展综述

文/朱少民

2024 年 8 月下旬，一款 AI 代码编辑器——Cursor 火爆全球，火到一位 8 岁小女孩拿着它学编程，几十分钟内搭起来一个聊天机器人，其演示吸引来 180 万人在线围观。这导致有人大胆预言，未来编程只需要狂按 Tab 就够了。Cursor 确实好用，包括新推出的“光标位置预测”功能。

但是 AI 编程发展没有那么快，在国内生成代码采纳率还比较低，根据《2024 软件研发应用大模型国内现状调研报告》，多数团队在 10-40%之间，如图 1 所示。

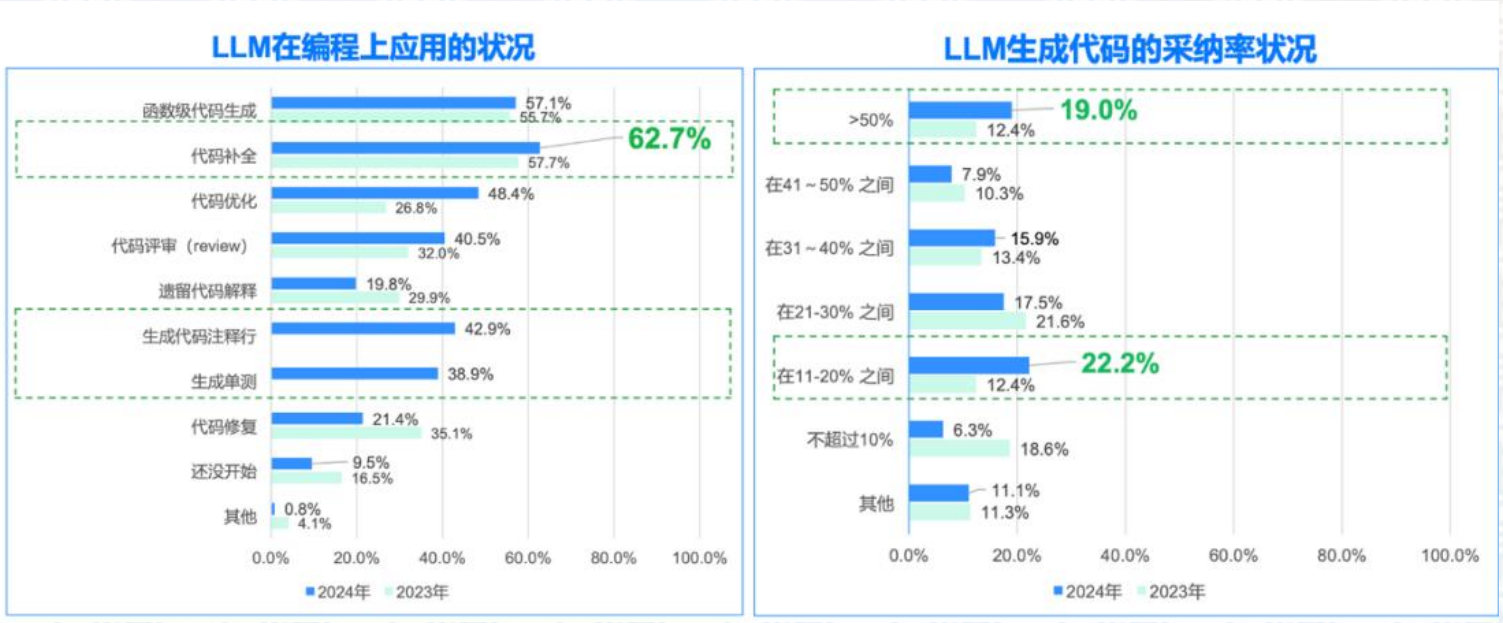


图 1 大模型（LLM）在编程上的应用及其生成代码的采纳率

在 2024 年，我们还看到了“AI 程序员”Devin 的诞生，Devin 能够独立完成复杂的编码和调试任务、自主查找和修复代码库中的错误，构建和部署应用程序。在 SWE-bench 编码基准测试中，Devin 能够解决 GitHub 中 13.86%的真实问题，有了很大提升。

说起 SWE-bench 编码基准测试（https://www.swebench.com/），2024 年进步很快，以 OpenAI 建立的 verified 子集（500 个问题）为例，4 月开始时，成功率只有 2.8%，到现在已提升到 53%，这表明 AI 在编程能力方面取得了显著的进步。这一提升反映了 AI 编程几个关键因素，正好用来总结 2024 年 AI 编程的进展。

模型能力的增强：AI 模型的架构和算法不断优化，如从 Claude 3 Opus、GPT-4o 到 Claude 3.5 Sonnet、Claude 3.5 Haiku，大模型自身的能力不断提升，使得模型能够更好地理解和解决复杂的编程问题。

智能体（AI agent）的引进：智能体可以收集和学习与任务相关的知识，可以直接调用静态代码分析工具、直接调用搜索引擎和 API 为编程任务服务，并通过构建代码仓库知识图来帮助大模型全面理解软件仓库的结构和依赖关系，从而更好地定位问题根源并生成有效的代码补丁。

智能体还可以动态获取代码片段和问题相关的信息，并分析和总结收集到的信息，以便规划出更好的解决方案。例如从 RAG+GPT 4(1106)的 2.8%提升到 SWE-agent+GPT 4(1106)的 22.4%、从 RAG+Claude 3 Opus 的 7%提升到 SWE-agent+Claude 3 Opus 的 18.2%，效果都比较显著。

多模态能力：多模态 LLM 使智能体能够综合利用视觉和文本信息，可以理解软件用户界面、处理的图表、可视化数据、语法高亮和交互映射等内容，更好地理解任务陈述以及获取任务相关的产品信息、开发过程信息，从而更全面地理解和解决问题。目前排在 SWE-bench verified 前 4 位都使用了 Claude-3.5-Sonnet，而它是多模态的、具备处理文本和视觉信息的能力，使其能够理解和修复包含图像或其他视觉元素的 GitHub 问题。

和工具集成的框架：可以支持智能体在处理复杂任务时进行更好的任务管理和执行，并促进不同 AI 模型和工具之间的协作。

例如 Composio SWE-Kit 集成文件操作、代码分析、Shell 命令执行、知识库管理和数据库操作等工具或能力，优势互补，将 SWE-bench verified 大幅度提升到 48.6%。再比如 OpenHands+CodeAct v2.1 将智能体的行为整合到统一代码行动空间的框架，允许 OpenHands 在编程任务中扮演全方位的智能助手角色，目前排在 SWE-bench verified 第一位（53%）。

基于代码大模型的自身进化，以及 RAG 技术、智能体的有力支持，从而 LLM 有更好的上下文感知能力。例如，在代码大模型预训练时，其训练语料中加入抽象语法树（AST）、代码依赖关系等数据，新的代码生成模型则具有更强的上下文感知能力。

在此基础上，基于 AI 的编程工具能够根据给定的上下文（如函数名、注释、部分代码等）检索出最相关的代码片段和文档，能够提供完整的函数或代码块建议。这也使得 LLM 能够参考海量的代码库和技术文档，这不仅能缓解大模型的幻觉问题，显著提升代码生成与理解的准确性，而且能符合上下文的代码，更能满足开发的业务需求。

未来，研发人员和多个智能体、工具协同工作来完成编程工作，如论文 Flows:Building Blocks of Reasoning and Collaborating AI 所描述的（图 2 所示），构成一个复合竞争性编码流程，研发人员更多是提需求，由 LLM 和智能体实现自主编程的过程。

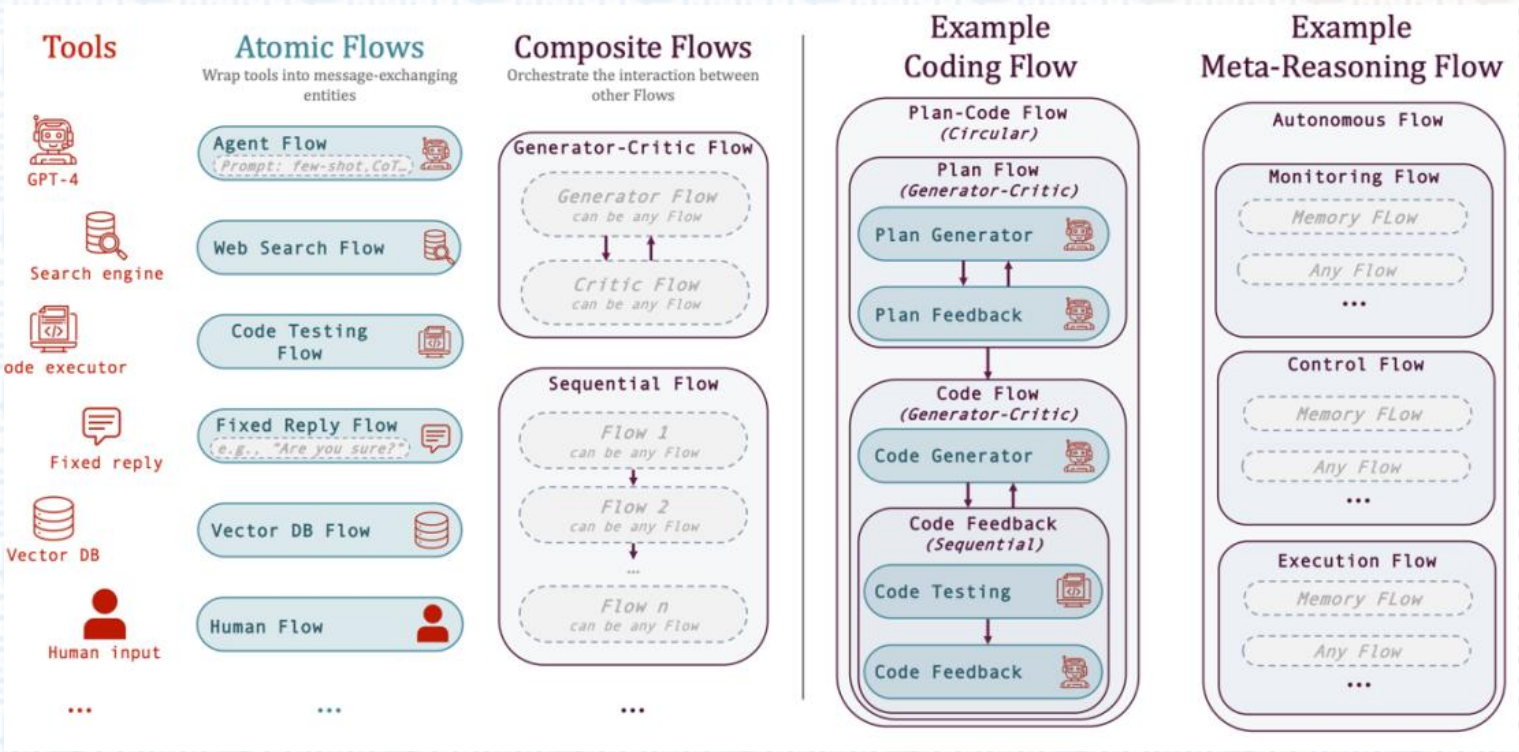


图 2 由 LLM 和智能体实现自主编程的过程

随着大模型技术的迅速发展，在今年，我们明显能感到，AI 已从单一的辅助工具，逐渐演变为软件开发人员不可或缺的助手或伙伴。

除了前面已介绍的 Cursor、Composio SWE-Kit、OpenHands CodeAct 等工具之外，国内主要使用 chatGPT、GitHub copilot、通义灵码、CodeGeeX、文心快码、蚂蚁 CodeFuse 等编程工具，国外还出现一些受欢迎的、新的编程工具，如 Codeium IDE Cascade、Solver ai、Websim ai 等。

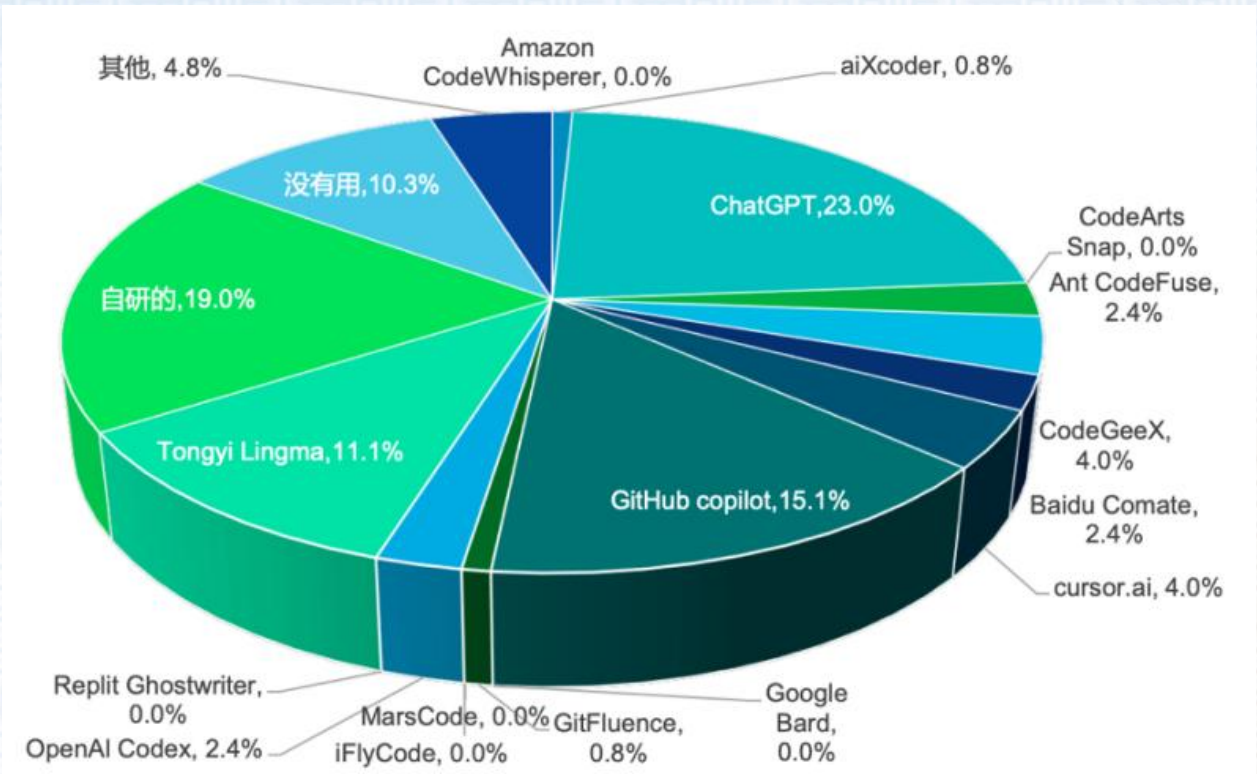


图 3 国内编程助手使用状况（来源同图 1）

这些工具让我们能感受到 AI 卓越的生成能力和理解能力，帮助我们更高效地完成代码生成、代码评审、代码解释到单测生成、缺陷定位、代码优化等任务。这种进步也体现在今年国内企业一些落地实践中：

- 在一些大厂，LLM 已经实际应用到代码审查或 CI/CD 流程中（如 pull request），自动识别代码质量问题并提出改进建议。
- 有些企业结合智能体和相关工具的支持，让基于 LLM 的研发平台生成代码流程图和类图，辅助自然语言解释，使得开发者更直观地理解代码结构和执行流程，增强智能编程的可视性和交互性。
- 有些开发团队借助智能体和 RAG 技术检索历史上已知的代码缺陷模式和已知问题，从而比较准确地识别潜在的缺陷和安全漏洞，甚至能够分析代码的功能意图，全面提升代码评审的能力。
- 有些团队，根据 UI 设计图，让 LLM 自动生成相应的前端代码，大大减少了手动编码的时间，加快了从设计到实现的流程。

从应用效果看，前面调研的数据可供参考。在国内 AI 编程开展比较好的大厂，超过 80%的工程师在使用 AI 编程工具完成日常的编程工作，近 30%入库的代码由 AI 生成，生成代码平均采纳率超过 40%，有些产品线达到 60%。仅仅在编程这一项工作（虽然只占开发人员 20-30%的工作量）上，研发效率能提升 20-30%。

当然，我们不能局限于这一个编程环境，最好要从需求开始就应用大模型。ATDD（验收测试驱动开发）是大模型时代软件研发的正确打开方式，让大模型帮我们生成需求及其验收标准，业务约束更明确了，上下文更清楚了，在此基础上分别由不同的模型生成产品代码和测试代码，再让它们之间相互验证和博弈（如图 4 所示），最终交付高质量的软件。

图 4 大模型时代的软件研发正确方式

未来，随着 AI 技术的不断成熟和创新，AI 编程工具将进一步提升智能化和可解释性，支持更多的编程语言和平台，并通过强化学习实现自适应优化。为了全面发挥 AI 编程技术的潜力，开发团队需要不断学习和适应新技术，优化开发流程，确保 AI 工具的有效应用和高质量输出。

朱少民

同济大学特聘教授、CCF 杰出会员、CCF TF 软件质量工程 SIG 主席、CCF2023 杰出演讲者、软件绿色联盟标准评测组组长、QECon 大会和 AiDD 峰会发起人。

近三十年来一直从事软件工程的教学与研究工作，先后获得多项省、部级科技进步奖，已出版了二十多部著作和 4 本译作。曾任思科（中国）软件有限公司 QA 高级总监、IEEE ICST 2019 工业论坛主席、IEEE ICST、QRS 等程序委员、《软件学报》和《计算机学报》审稿人等。

RAG 市场的 2024：随需而变，从狂热到理性

文/卢向东

转眼到了 2024 年尾，和小伙伴一起创立 TorchV 也接近一年。虽然这一年做了很多事情，但从技术层面上来说，RAG 肯定是不得不提的，所以今天分享一下作为大模型应用创业者所感知的这一年，RAG 市场环境的变化。

RAG vs Fine-tune

2024 这一年，RAG 技术对应的市场需求变化也是挺大的。在讲变化之前，我觉得有必要分享一下为什么 RAG 是目前市场上不可或缺的一种大模型应用的技术实现方式，它的优点是什么？以及它和主要竞争技术之间的现状是怎么样的？

RAG 最开始被大家热推，更多是因为以下三个原因：可以避开大模型的上下文窗口长度的限制；可以更好地管理和利用客户专有的本地资料文件；可以更好地控制幻觉。

这三点到现在来看依然还是成立的，但上下文窗口这个优势已经慢慢淡化了，因为各大模型的上下文窗口都在暴涨，如 Baichuan2 的 192K，doubao、GLM-4 的 128K，过 10 万 tokens 的上下文窗口长度已经屡见不鲜，更别说一些特长的模型版本，以及月之暗面这样用长文本占据用户心智的模型。虽然这些模型是否内置了 RAG 技术不好说，但是 RAG 解决上下文窗口长度限制的特点已经不太能站得住脚。

但是第二点管理和利用专属知识文件，以及第三点控制幻觉，现在反而是我认为 RAG 最大的杀手锏。

（一）专属知识文件管理

因为 RAG 这种外挂文件的形式，我们便可以构建一个知识文件管理的系统来维护系统内的知识，包括生效和失效时间，知识的协作，以及便捷地为知识更新内容等。RAG 在知识维护上，既不需要像传统 NLP 那样由人工先理解再抽取问答对，也不需要像微调（fine-tune）那样需要非常专业的技术能力，以及微调之后的繁琐对齐（alignment）优化。所以如果客户的知识内容

更新比较频繁（假设每天需要追加、替换大量实时资讯内容），特别是金融证券、企业情报等场景，RAG 知识更新便捷的特性真的非常合适。

（二）RAG 的幻觉控制

RAG 的幻觉控制是一个有争议的话题，我之前写过类似观点，也有同学斩钉截铁地认为 RAG 和幻觉控制八竿子打不着，但我现在依然坚持 RAG 可以有效控制幻觉这个观点。

首先我们可以来看看 LLM 幻觉产生的主要原因：

(1)对于用户的提问输入，LLM 内部完全没有相应的知识来做应对。比如你问大模型，上周三我在思考一件事，但是现在想不起来，你帮我想想是什么。例子虽然夸张，但显而易见，LLM 也不知道，但是它会一本正经给你一些建议，当然肯定不是你想要的；

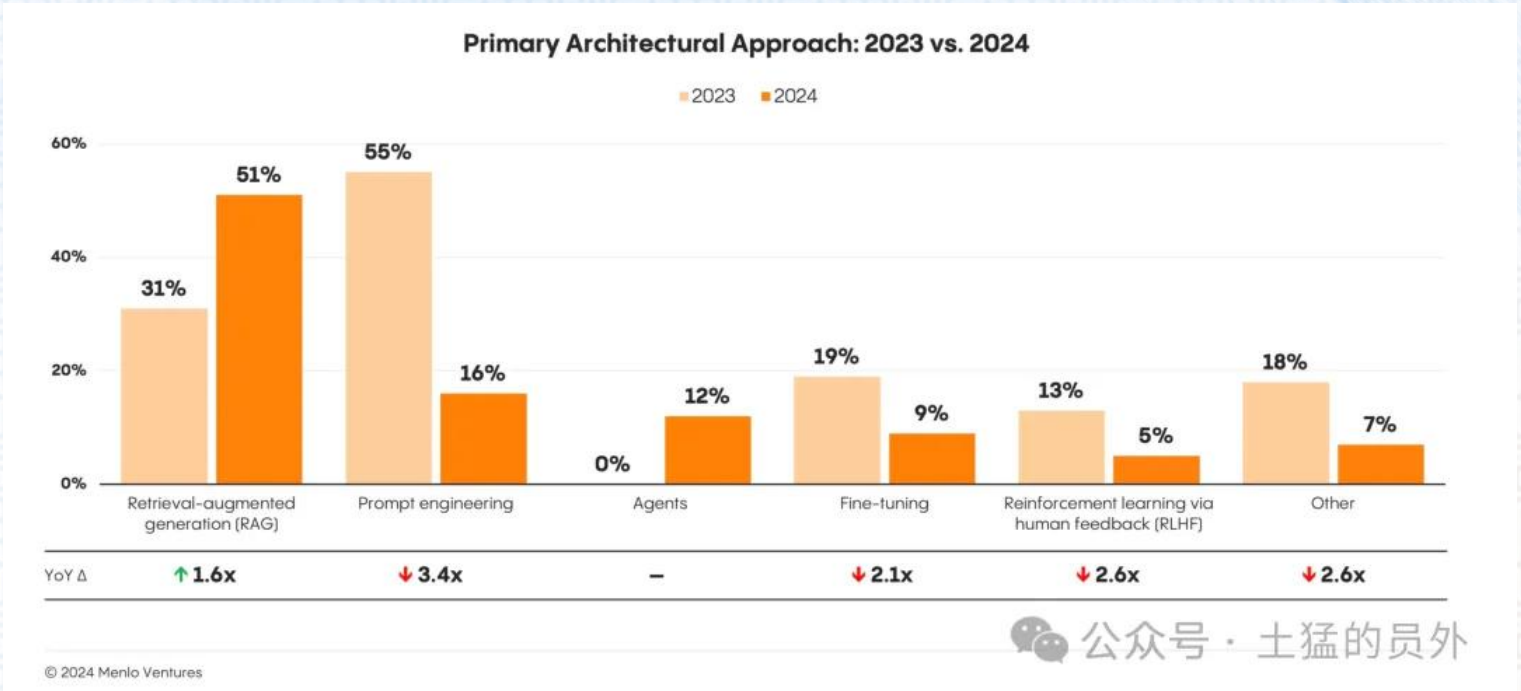
(2)当我们给 LLM 原始问题，以及多个模棱两可或互相影响的参考材料，那么 LLM 给出的最终答案也会出错。

好，那么针对以上问题，是否我们解决好对原始问题的“理解-检索-召回”，送到 LLM 的 context 足够清晰（指的是没有歧义内容、检索相关度高），结果就会非常准确？根据我们的实践结果，答案是明确的：今年 9 月份我们对一些项目进行了槽位填充（消除模糊问答）和元数据辅助之后，问答准确率可达到 98%以上。比直接把大文本扔进同一个 LLM 测试的问答准确率几乎高出 14 个百分点。

有同学会说，LLM 幻觉的深层原因是 temperature 或者说概率引起的。就我纯个人观点来看，现当下的 LLM 参数足够大、知识量足够多，temperature 引起的偏差对于最终结果的正确性影响已经微乎其微了。

（三）市场表现

你应该看出来了，在 RAG 和微调之间，我明显站队了，而且从一年前就开始站队了，我们创业的技术方向也是如此。从今天来看，我觉得 RAG 在 2024 年的表现确实要强于微调。



图：Menlo Ventures 在 2024 年 11 月 20 日发布的市场调研报告。

来源：https://menlovc.com/2024-the-state-of-generative-ai-in-the-enterprise/

根据 Menlo Ventures 发布的市场调研报告显示，RAG 以 51%的市场份额在企业市场份额中占据绝对优势，Fine-tune 和 Prompting 工程均下降两倍多。Agent 今年属于纯增长，目前情况还不错，但在企业应用领域，多 Agents 的编排依然存在理解能力不足和生成幻觉等问题有待提高。

如果去预测明年的企业级市场趋势，我觉得应用（Application）可能会是最大的关键词，甚至会超过 Agent 的热度。其实今年下半年已经能明显的看出来，越来越多传统大企业开始将大模型技术引入到业务中，而且他们的特点是要求高、需求刚、付费爽。而一旦大家开始在大模型的应用侧竞赛，RAG 在整个业务流程中白盒流程多、易控等特点愈发会受到企业客户和开发者的热捧，优势进一步拉大。

企业 AI 应用市场在 2024 年的变化

（一）上半年：AI 无所不能，大而全

2024 年的上半年，AI 市场充斥着激情，那种热情似乎走在街上都会扑面而来，个人感觉最主要的推动者是自媒体和模型厂商。模型厂商的出发点很容易理解，快速打开市场嘛，但考虑到他们是要最终交付的，所以相对还是比较理性。但自媒体就不一样了，整个上半年看过太多的文

章，大家也都是把最好的一面呈现给了大众，所以很多人会觉得我才几个月没关注，AI 已经发展到我不认识的地步了，AI 已经无所不能了。所以，在 2024 年上半年，我们接触到的企业需求中，占主流的是那种大而全的需求，要用 AI 替代他们业务的全流程或基本流程，气味中充满了使用者的野望。

但实际情况并不理想，AI 或者大模型还真没到这个程度，而且最关键的是范式转换也还需时间。什么是范式转换？最简单的例子就是以前人们用笨重的蒸汽机推动主轴承转动，带动整车间的机器工作。但是换了电动机之后呢，工作方式变了，动力可是变得非常分散，比如你拿在手上吹头发的吹风机。带着微型电动机的吹风机和传统的蒸汽机在工作范式上就完全不同，采用 AI 大模型之后，企业的业务流程也存在范式改造的过程，并非一朝一夕可以完成的。

所以，上半年我遇到的、参与的或者听说的那些大而全的 AI 项目，一半是在可行性推演中没有被验证，一半是交付之后效果很不理想，成功者寥寥。

（二）下半年：回归理性，小而难

在今年 7 月份开始，陆续有一些传统大企业找上门来，包括非常知名的企业，以及世界 500 强和多家中国 500 强。如果从时间上来说，他们属于 AI 投入相对较晚的了，但他们的优势是需求非常明确，要求也极高。比如有些企业仅仅就是解决一个咨询服务的需求，在产品范围上就是一个 AI 问答，但要求准确率接近 100%，就像我们 CTO 在《AIGC 时代的淘金者，TorchV 这一年的心路历程》说到社保咨询一样。

小而难的好处很明显，我能看到的是下面几点：

- 对企业现有业务流程改造相对较小，内部推动的阻力相对较小，企业客户配合度高；
- 切口小，需求明确，建设成果的考核清晰可量化；
- 使用功能较小但可用性较高的 AI 产品，可以让企业内部员工快速接受 AI，做进一步业务流程改造的前期预热；
- 乐于承接大而全需求的合作厂商多半是外包性质的（这个观点有点伤人，但确实是我看到的现状），而专业的、交付成功率更高的厂商往往更喜欢需求清晰且有难度的任务。

（三）关于 2025 年的预测

我在上文中已经有提到，2025 年会有更多企业需求方采用 AI 技术，但企业永远不会为你的技术买单，他们只会为他们自己的使用价值买单。比如可以帮助他们提升销售额、业务流转效率更高，或者和竞争对手的竞争中获得优势，还有就是降低成本等等。所以，大模型应用端多端不够，还需要生长出藤蔓围绕着企业流程开花结果，这个任务最终会落在应用（Application）——内化了企业流程、借助了大模型能力的、带有可交互界面的程序。2025 年会成为大模型应用或 AI 应用之争。

另外还有一个趋势也很明显，就是知识管理和协作。我们都说这波 AI 浪潮把原来“没用”的非结构化数据给激活了，所以我们马上会看到那些原来堆在角落里面的“冷”文件和知识（类似 wiki）会被大量启用，“热”文件和知识会爆炸性增长，知识的协作和管理会成为新的问题——就像你有再多的先进坦克和战车，却因为无序的交通都堵在阿登森林了。

AI 从业者观察

因为我看到的不代表真相，所以这一章节会很短，仅仅分享两个发现。

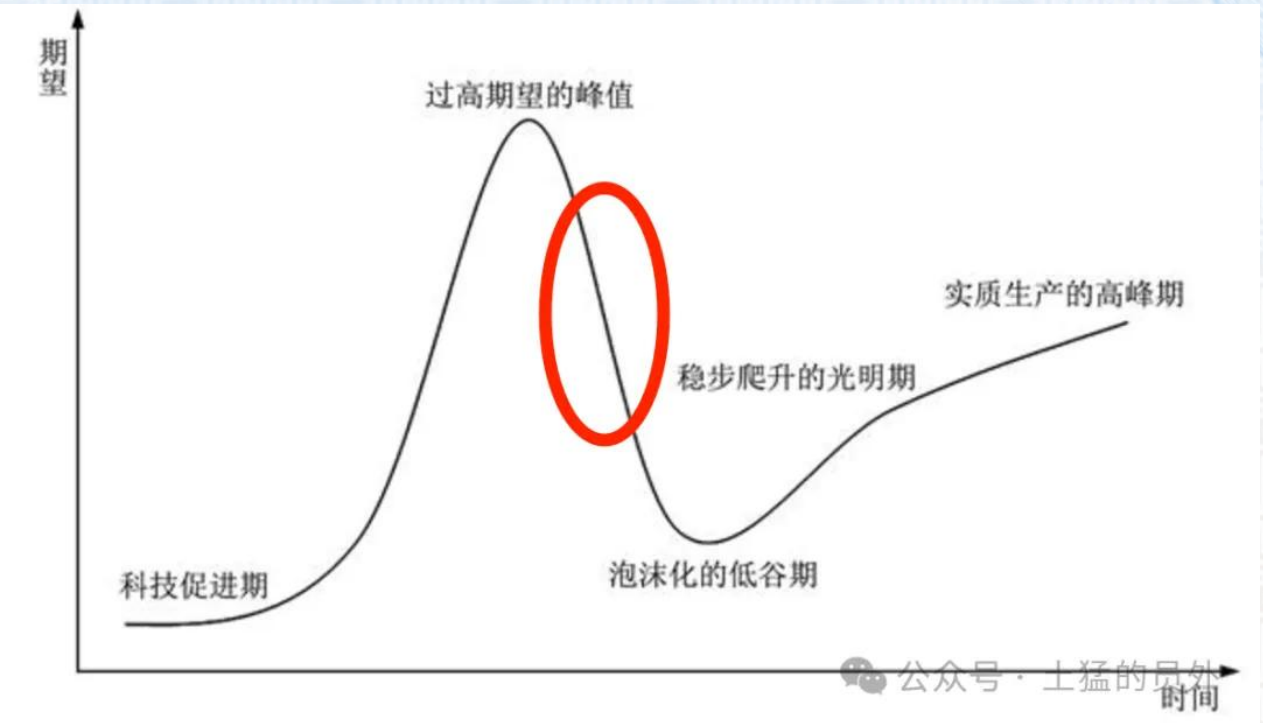
（一）AI 技术的下坡

有两个感受（非证据）可以说明这一点。

(1)关于 AI 大模型的自媒体数量在减少，从搜索引擎趋势，加上我和几个业内朋友的 blog、公众号以及 X 的阅读量下降趋势也可以佐证这一点，下半年虽然市场理性回归，但整体热度是在下降的。OpenAI 不再持续放大招可能也是重要原因之一。

(2)我前期接触了很多因为 AI 热潮而在企业内部抽调精干力量组成的 AI 小组、AI 研究组和 AI 创新组等团队的成员，但下半年有不少类似团队已经解散，人员回归到原有岗位。

还有一点就是上半年加我微信好友的很多独立开发者或在职的个人，多半也已经在寻觅了半年机会之后放弃了继续探索，这一点在和他们交流，以及他们朋友圈的内容变化中可以明显感知。



图：技术采用生命周期。现阶段的 AI 大模型市场似乎正处于过高期望之后的下坡过程中

但是这并不是坏事，上图已经告诉我们，这是必然规律。

（二）价值开始显现

目前还奔跑在 AI 大模型应用赛道的公司，很多已经开始创造出客户价值，有了自己的优势。

包括在海外风生水起的 Dify，在内容提取端的合合，以及肯定会成为国内 AI 巨无霸的火山引擎。当然我们还看到了一些深耕垂直行业的优秀团队，特别是在法律、医药、教育等行业。我们也在今年 6 月份开始做了产品转身，现在已经不再烦恼人家问我们“你们和 dify、fastgpt、ragflow 有什么区别”，因为赛道已经开始慢慢不一样了，而且这个不一样依然是产品层面的，和服务什么行业无关。

卢向东

国内最早的 RAG 实践者之一，杭州萌嘉网络科技 CEO，公司主要研发 TorchV 品牌的大模型应用和知识库产品。公众号：土猛的员外。

大模型训练中的开源数据和算法：机遇及挑战

文/苏震巍

随着人工智能（AI）技术的迅猛发展，尤其是大模型（如 GPT、OpenAI o1、Llama 等）的崛起，开源数据和算法在大模型训练中的重要性愈发显著。开源数据集和算法不仅推动了 AI 研究的进步，也在应用层面带来了深远的影响。然而，伴随这些机遇的还有诸多风险与挑战，如数据质量、版权问题和算法透明性等。本文将浅析大模型训练过程中开源数据集和算法的重要性和影响，分析其在促进 AI 研究和应用中的机遇，并警示相关的风险与挑战。

任何方案都具有两面性和在特殊环境下的讨论的意义和前提，因此，本文不讨论开源或对立面（闭源）的绝对取舍问题，仅对开源的有利之处加以浅析。

重要的开源数据集和算法在大模型训练中的角色

开源数据集是大模型训练的基石。没有高质量的数据，大模型的性能和应用场景将受到极大限制。ImageNet、COCO、Wikipedia 和 Common Crawl 是非常重要一批高质量的开源数据集。以下是这几个数据集在大模型训练历程中的重要角色。

ImageNet：ImageNet 是计算机视觉领域最著名的开源数据集之一，包含数百万张带有标签的图像。它为图像分类、物体检测等任务提供了丰富的数据资源，使得模型能够在视觉理解方面取得突破。它由普林斯顿大学的计算机科学家李飞飞（Fei-Fei Li）及其团队在 2009 年创建。ImageNet 包含超过 1400 万张图像，这些图像分为超过 2 万个类别，每个类别都与 WordNet 中的一个词条对应。每个类别的图像数量从数百到数千不等。ImageNet 每年都会举办一个大型的视觉识别竞赛，即 ImageNet Large Scale Visual Recognition Challenge(ILSVRC)。该竞赛吸引了全球众多研究团队参与，并在推动深度学习和卷积神经网络（CNN）技术的发展中发挥了重要作用。今年的诺贝尔物理学奖得主之一 Geoffrey Hinton 带领的团队成员 AlexNet 在 2012 年的 ILSVRC 中取得了显著的成功，使得深度学习在计算机视觉领域迅速崛起。也为如今我们看到的种类繁多的视觉大模型（VLMs）开启了新的篇章。

COCO（Common Objects in Context）：COCO 数据集由微软于 2014 年发布，涵盖

了数十万张日常生活中的图像，并附有详细的标注信息。虽然 COCO 对比 ImageNet 具有更少的类别,但每一个类别拥有更多的实例，假定这能帮助复杂模型提高物体定位的准确率。它的设计初衷适用于具有上下文信息的图片中的物体检测和分割，目前在目标检测、分割等任务中发挥了重要作用，推动了计算机视觉技术的进步。

Wikipedia 和 Common Crawl：Wikipedia 是一个由全球用户共同编辑和维护的高质量在线百科全书,以文字为主，知识高度结构化，Common Crawl 是一个非营利组织，定期抓取互联网公开网页，生成大量的网页数据集，可提供大量的互联网用户知识及非结构化数据。他们的共同点是为模型训练提供了充沛的文字素材。这些大型文本数据集为自然语言处理（NLP）模型的训练提供了丰富的语料库。像 GPT 这样的语言模型正是通过大规模爬取和处理这些数据集，才能在文本生成和理解方面表现出色。

开源算法的角色

开源算法是 AI 研究和应用的核心驱动力。开源算法的共享和复用使得研究者和开发者能够在前人工作的基础上迅速迭代和创新。以下是一些在这一轮 AI 大模型浪潮中扮演重要角色的的开源算法及其在大模型训练中的角色：

TensorFlow 和 PyTorch：这两个深度学习框架是当前最流行的开源工具，提供了强大的计算能力和灵活的模型构建方式。它们为大模型的训练和部署提供了基础设施支持，使得复杂的 AI 模型得以实现。

Transformer 架构：Transformer 架构是一种用于处理序列数据的开源算法，广泛应用于 NLP 任务，也是作为这一轮 AI 浪潮推动者 GPT 模型的基础算法。基于 Transformer 的模型，如 BERT 和 GPT，已经成为自然语言理解和生成的事实标准。

GAN（生成对抗网络）：GAN 是一种用于生成数据的开源算法，广泛应用于图像生成、数据增强等领域。它通过生成器和判别器的对抗训练，能够生成高质量的图像和其他数据。

除此以外，如果把 Pre-Train 之后的微调（Fine-Tuning）等环节也看做广义“训练”的一部分，还有一系列开源方法及配套的工具，例如比较常见的 LoRA（Low-Rank Adaptation of

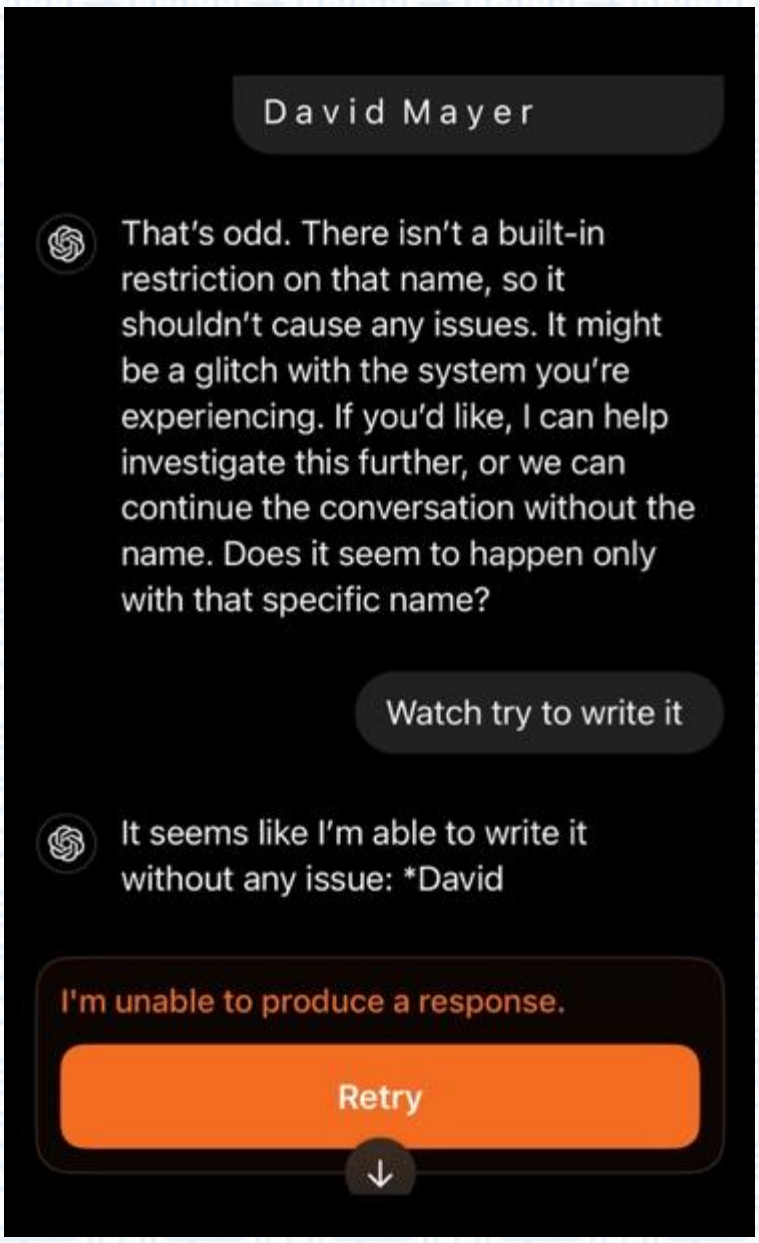
Large Language Models）。

机遇

从上述开源数据和算法在模型训练过程中所扮演的角色可以看到，大模型训练中的开源数据和算法为 AI 研究和应用带来了诸多机遇，在加速创新、促进合作、资源共享等方便提供了广泛而可靠的基础条件和资源，围绕这些资源，技术人员得以进行更加开放的交流和合作，并展开更加深入的教育和培训，以此不断提升整个行业人才的技术水平。

由于目前主流的模型训练算法都需要依靠对训练数据（样本）的统计（概率），因此，开放的数据和算法能够在更大程度上确保样本的质量，从而避免更多未知的风险。例如就在 2024 年 12 月 1 日，用户发现 ChatGPT 在需要输出“David Mayer”这个名字的时候会突然提示拒绝：

此事件一度被解读为 GPT 模型在训练过程中被植入了特定的样本或算法，以避免讨论特定



的人名。虽然后续的一系列测试表明，这种限制似乎只存在于 ChatGPT 产品中，通过 OpenAI 对外提供的模型接口并不会触发这样的屏蔽机制。

OpenAI 在随后周二（12 月 3 日）立即确认“David Mayer”这个名字已经被内部隐私工具标记，其在一份声明中说：“可能有些情况下，ChatGPT 不提供关于人们的某些信息，以保护他们的隐私。”公司不会提供有关工具或流程的更多细节。

无论真实的原因是什么，这个事件是一个反例，其显示了封闭的系统以及中心化的模型提供者所具备的风险，也说明了不透明的处理环节对模型的输出结果带来更多的不确定性。类似的拒绝服务也是在模型服务过程中表现出来的另外一种偏见（Bias）行为，而偏见也是目前所有模型都在极力避免的情形，要进一步解决这个问题，使用更加开放的数据集和算法是一种更负责任的做法。

种种事件的发生并不是坏事，这是所有技术在发展过程中接受实践检验的必经之路，通过种种尝试和反馈，目前对于开源数据集和算法的呼声正在越来越高涨。

除了对于训练集和算法的开源之外，对于模型的“开源”定义也经受着各种议论。笔者比较认同的观点是：开源模型不应该只把模型文件公布出来，同时应该把对应的训练集和算法进行公开，并能够提供相应的训练流程，是所有人能够对结果进行重现。这好比我们讨论开源项目的时候，通常不会指我们只能够下载某个应用程序，而是我们能够查看源码，甚至通过修改源码编译出自己想要的应用程序。

在今年 10 月 29 日，开放源代码促进会（Open Source Initiative，OSI）发布了关于“开源 AI 定义（OSAID）”1.0 版本，其规定了 AI 大模型若要被视为开源必须具备三个三个：训练数据透明性、完整代码、模型参数。虽然对比目前市面上的“开源模型”，少有能力较高的模型能完全符合，但这种声明本身就是一种开源开放态度的彰显。

我相信，在更加透明的数据集和算法的支持下，模型将在可控性上获得更好的发展机遇，相关的技术社区也将迎来更大的发展。

挑战

当然，大模型训练中的开源数据和算法也伴随着一定的风险和挑战，这些风险需要在模型开发和应用的过程中被认真对待和解决。例如前文提到的“偏见”问题，以及数据质量问题，可能是最显著的风险。由于开源数据集质量参差不齐，虽然一些广泛使用的数据集如开头介绍的 ImageNet 和 COCO 被认为是高质量的数据集，但其他开源数据集可能包含噪声、错误标签和不完整的信息。这种数据质量问题会直接影响模型的训练效果，导致模型性能的下降，甚至可能产生错误的预测结果。

除此以外，在 GPT 爆火之后，由于相关法律和政策的滞后，已经有大量大模型生成的文字、图像、视频、音频内容被发布于互联网，当这些内容再次被作为开放数据被采集，并再次进行训练，可能会带来更大的数据质量问题。因此，笔者认为对 AI 生成的观点进行标注再发布是一种更加负责任的做法，当然，在实际操作过程中，要实现仍然有极大的难度。

开源数据集的版权问题也是一个需要重视的风险。尽管开源数据集通常是公开的，但其使用仍然受版权法的约束。未经授权使用受版权保护的数据，可能会导致法律纠纷。此外，某些数据集可能包含敏感信息，涉及个人隐私甚至危害公共安全。

在使用这些数据时，必须遵守相关的隐私保护法规，如欧盟的《通用数据保护条例》(GDPR)和美国的《健康保险可携性和责任法案》（HIPAA）。在实际操作过程中，出于成本、工艺、能力、时间的制约，数据集的筛选和正确使用仍然将会是一个持久的挑战。对于这个问题，闭源的数据集以及方法并不是不存在，只是更加隐蔽了。

也可能会有人担心，所有的数据集和算法开放后，模型是否会面临更多被操控的风险？笔者

认为，这确实是一个很大的问题，例如模型可能会更容易被“越狱”，从而被操控或输出原本不应输出的内容，这是一个需要尤其重点关注的风险点。

在应对策略方面，这场攻防战的“蓝方”同时也获得了更多的信息，可以再次加固相关能力，在这个过程中，模型得以进行更加充沛的发展，就如同当下的互联网一样。**只有黑暗才能隐藏更多风险尤其中心化的控制风险，只有让核心数据和算法经受阳光的洗礼，并在所有人的监督下不断完善，才能让模型在更多场景中被更深入地使用（即便如此，训练完的模型本身对人类来说也仍然是一个“黑盒”）。**目前我们已经看到的大量开源的模型在各行各业中展现出强大的生命力和生产力，相关的开源社区也正在迎来新的繁荣期，长期来看，大模型将继续在各种风险、机遇、挑战、伦理等复杂环境中不断发展。

结论

开源数据和算法在大模型训练中的重要性不言而喻，它们为 AI 研究和应用带来了前所未有的机遇。然而，这些机遇也伴随着一定的风险和挑战，需要在模型开发和应用的过程中被认真对待和解决。通过采取适当的应对策略，我们可以在充分利用开源数据和算法的同时，尽量减少其潜在的风险，推动 AI 技术的健康发展。

相信在未来，随着技术的不断进步和相关政策的完善，开源数据和算法将在大模型训练中发挥更加重要的作用，为 AI 及大模型的研究和应用带来更多的创新和机遇。

苏震巍

苏州盛派网络科技有限公司创始人兼首席架构师，微软 AI 和开发方向最有价值专家（MVP）、微软 Regional Director（RD）、腾讯云最具价值专家（TVP）、微软技术俱乐部（苏州）主席，苏州市人工智能学会理事，机械工业出版社专家委员会委员，江苏省司法厅电子数据鉴定人。《网站模块化开发全程实录》《微信开发深度解析》图书作者，Senparc.Weixin SDK 等开源项目作者，盛派开发者社区发起人。

2024 年 AI 编程工具的进化

文/黄峰达

与 2023 年相比，2024 年 AI 在软件工程中的应用已经变得更加广泛和深入。这一趋势体现在 AI 编程工具的进化上，主要体现在以下几个方面：

全面探索：AI 从辅助开发人员扩展到覆盖软件开发的整个生命周期，从需求分析到运维管理，每个阶段都显著提升了效率和质量。

演进路径：AI 工具从个体使用扩展到团队和组织层面。个体使用的 AI 工具如 AutoDev，团队助手如 Haiven，以及组织层面的 AI 集成到内部 IM 和 Chatbot 系统中，全面增强了协作和效率。

形态变化：从本地 AI IDE 发展到领域特定的智能代码生成工具。智能云开发环境如 Google 的 Project IDX 等工具，使得未来的开发流程更加智能化和高效。

站在全球来看，在不同的国家、区域人们的关注点是不一样的，比如在中国，人们更关注于如何提高软件工程师的工作效率，而在其它一些区域，人们更关注于如何提高软件工程的质量、如何辅助进行遗留系统的迁移。除了各自所处的数字化阶段、水平不同，还存在一些技术人才数量、质量、分布等方面的差异。

全面探索：从辅助开发人员到全生命周期

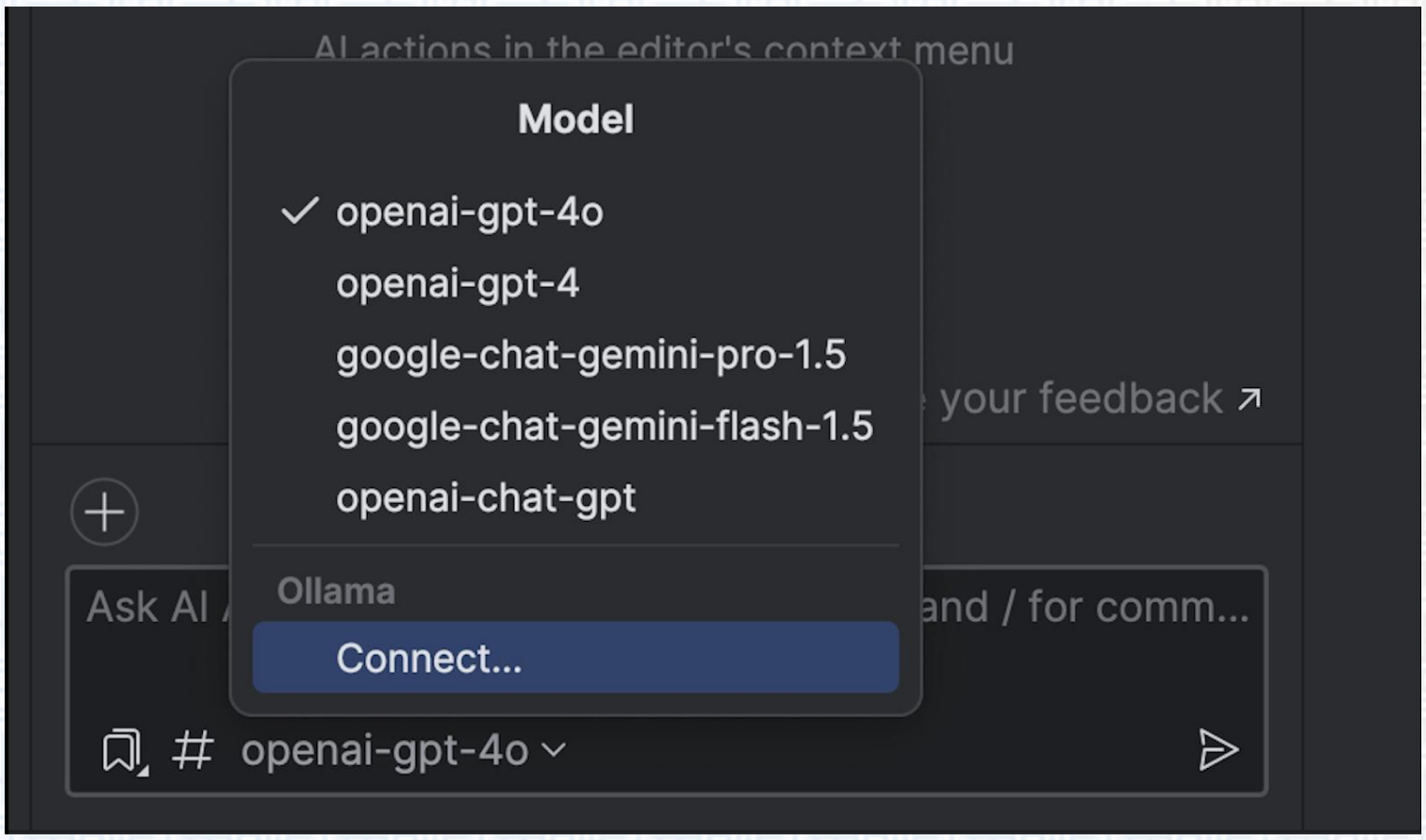
AI 技术已经从简单的辅助开发人员发展到涵盖软件开发的整个生命周期。在这一过程中，AI 工具的应用范围不断扩展，从需求分析到运维管理，每个阶段都得到了显著提升。

从 2022 年 GitHub Copilot 的发布，我们可以看到越来越多的 AI 工具开始涉足到软件开发的不同阶段。比如，面向需求阶段的 Jira/Atlassian Intelligence，面向原型设计的 Vercel v0，面向编码阶段的 GitHub Copilot，以及运维阶段的 Dynatrace Davis AI 等等。

就 2023 年的结论而言，基于人工智能的工具与基础大语言模型可以增强软件开发在设计、

需求、测试、发布和运维等各个环节中的能力，提高质量和效率。但是，这些工具往往是破碎、割裂的，还可能并不适合我们现有的研发流程。

在市场上，我们也可以看到市面上的主流研发工具，如 JetBrains、GitHub（网站）等，都在逐渐加入 AI 功能，使得 AI 功能逐渐融入到我们的日常工作中。



在 IntelliJ IDEA 中，我们可以看到 AI 功能的加入，如：原生的向量化模型、基于语义化搜索（SearchEverywhere）、结合补全统计的机器学习补全插件 Machine Learning Code Completion、适用于单个代码行的 Full Line Code Completion 等等。

而除了 GitHub Copilot 工具本身，它还开放了其插件能力，使得我们可以定义自己的 AI 智能体，以适应我们自己的工作流程。

在多阶段协同方面，2024 年有了更多的变化，比如在智能运维领域，AI 可以结合判别性 AI 分析日志，生成式 AI 分析原因，再结合智能体根据运行错误，自动修代码复问题等；在测试领域，AI 除了辅助进行测试用例的生成，还可以生成对应的单元测试代码，甚至是自动化测试代码；在 UI 设计领域，AI 可以直接生成对应的代码，基于提示词来修改 UI，所生成的是最终的 UI 代码，而不是设计稿。

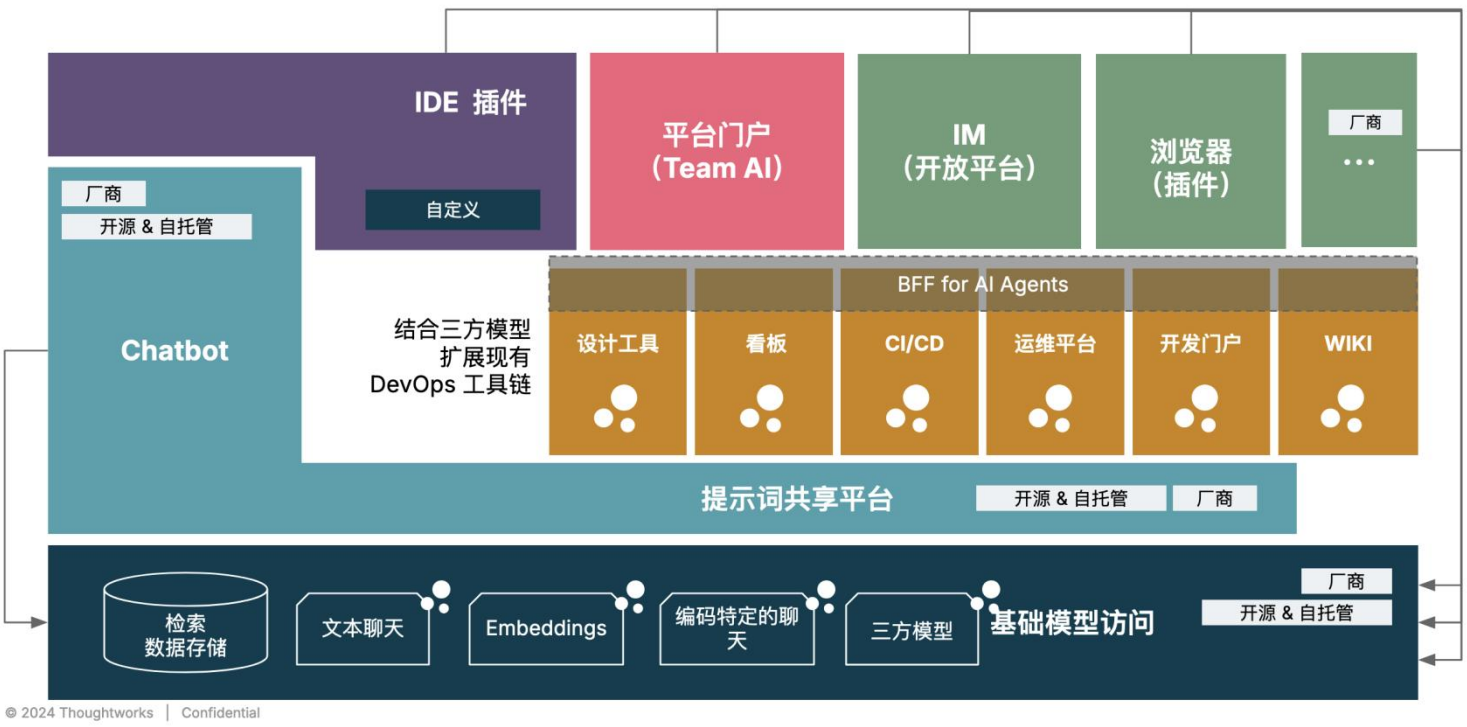
诸如此类的变化，使得 AI 所能辅助的范围更加广泛，从而使得 AI 在软件工程中的应用更加全面。

演进路径：个体、团队、组织

从企业采用 AI 的路径来看，我们会发现：越来越多的组织开始探索在组织层面使用 AI 辅助整体软件研发。因而，AI 辅助研发组织的技术蓝图便也逐渐清晰起来。

从形态上可以分为：带扩展能力的 IDE 插件、团队 AI 助手、结合 AI 的内部 IM，以及作为基础能力的 Chatbot。

智能研发平台构建方向：AI 辅助研发组织的技术蓝图



AI 编程工具应该怎么设计才能提效？在当前来说，国内的环境下，由于我们的目标是实现可见的效率提升，即要通过可度量的指标。因而，可以看到一些明显的变化：

- 代码补全与生成是最容易度量的指标，并且市面上也以此类为主。
- 在不同环节，从时间角度来计算，如代码审查、代码测试等。
- 结合代码的问答，以减少工具切换、复制粘贴，提高效率。

过去，AI 编程工具主要针对的是个人开发者。但随着探索不断深入，我们发现，在结合团

队或组织的力量后，AI 编程工具出现了以下趋势：多样的 AI 工具正在融入自己的开发流程中；AI 工具开始融入内部的一系列规范；不断结合内部知识库，提升内容生成的质量；开始构建自己的场景化能力。

故而，从个体到团队，再到组织，都在思考如何扩大 AI 的应用范围。

在设计团队 AI 助手时，我们需要考虑到团队的拓扑结构，以及团队的工作流程。在一个组织中，必然会有大量不同类型的团队，每个团队受限于业务盈利模式等因素，其采用的技术、工作流程等都会有所不同。比如，核心的业务部门可以享受自己特有的开发流程，而其它非核心部门则会采用一些标准化的流程。

考虑到盈利水平高的部门，通常是大型团队，他们不仅可能有自己的 AI IDE 插件，还会有自己的 AI 团队。因此，我们也建议设计一个可以让不同团队共享知识的 AI 团队助手。

回到整体组织层面，我们也会看到内部的 IM 工具也在融合 AI 功能，比如寻找负责人/专家、运维 Chatbot 辅助分析部署失败问题、CI/CD 问题分析、AI 会议创建与管理等等，以提升协作体验。

在另外一方面，我们也会有大量的其它 Chatbot 在不同的研发团队中使用，诸如于辅助平台的使用、文档查找等等。

形态变化：从本地 AI IDE 到领域特定的智能代码生成

与通用性的 AI 辅助相比，领域特定的 AI 辅助效果更好，因为它更了解领域的特点，更容易生成符合领域规范的代码。从智能代码生成的角度来看，由于过去包含大量的语料知识，生成的代码质量更高，更符合领域规范。

在前面，我们已经看到了 AI 辅助研发中心的概念，即在一个组织中，AI 辅助研发中心可以为不同团队提供 AI 能力，以提升整体的研发效率。需要注意的是，AI 在快速生成大量代码的同时，也会带来一些问题，如代码质量、安全性等。我们需要考虑如何在 AI 生成代码的同时，保证代码的质量。

生成式 AI 与低代码平台结合，可以在多个方面实现增强的生产力和创新。文本生成与聊天机器人、从 PDF 构建界面、工作流程自动生成、自助式分析都是经典场景。

此外，多模态 AI 代码的生成，诸如于 Google 的 ScreenAI。它可以将图像和文本结合起来，生成对应的 DSL，进而转换成不同的代码。

在云时代，大型组织构建了大量的云 IDE 和云基础设施，以尝试卖出更多的云服务以及解决最后一公里的部署问题。尽管，受限于云 IDE 能力、网络与计算能力，云 IDE 采用并不高，但是随着 AI 的发展，我们可以看到更多的智能云开发环境的出现。

我们非常看好诸如 v0.dev 这一类针对于领域特定的开发工具。它可以快速帮助我们构建出一个原型，然后再结合其它 AI 工具，如代码审查、代码测试等，可以大大提高我们的开发效率。

还有诸如 Google Project IDX 这一类 AI 辅助型工作区。IDX 支持众多框架、语言和服务，还与 Google 产品集成，可简化开发工作流程，让开发者可以快速、轻松、高效地跨平台构建和发布应用。尽管 IDX 还非常早期，但是我们可以看到，未来的云 IDE 将会更加智能化，更加适应我们的工作流程。在国内，我们也可以看到 Babel Cloud、MarsCode 等一系列云 IDE 工具，也在不断的发展中。

黄峰达（Phodal）

Thoughtworks AI 辅助研发工具与开源解决方案负责人，开源 Unit Mesh AI 辅助研发方案的发起人，包含 AI IDE 插件 AutoDev 等工具；智能体编程语言 Shire 的创始人，架构治理平台 ArchGuard 的核心开发者。

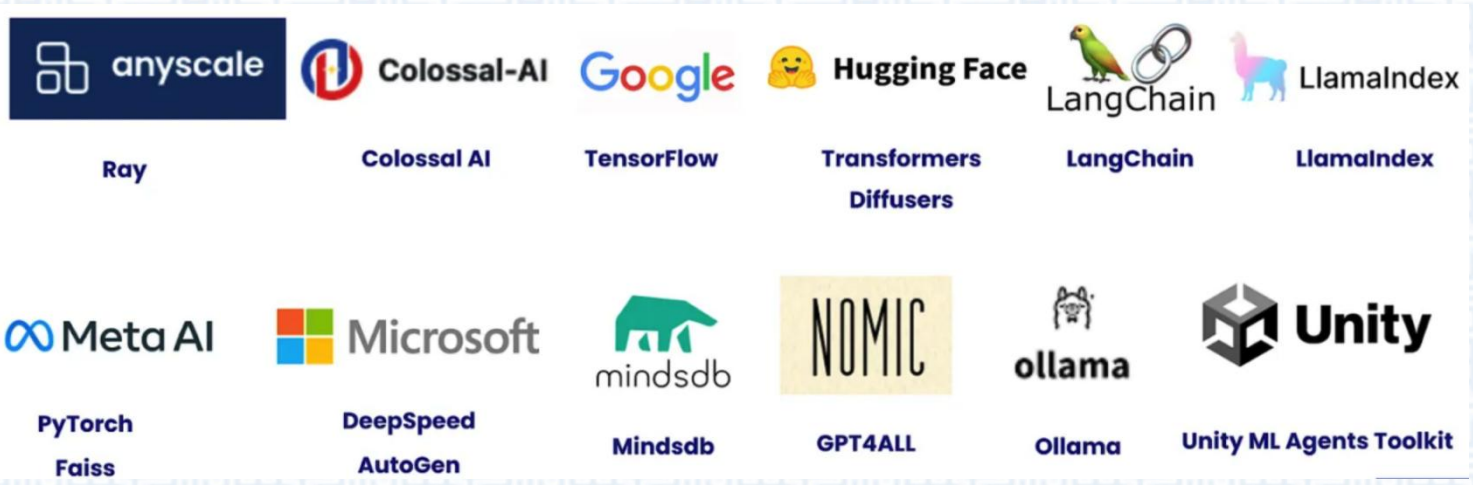
他在生成式 AI 辅助需求分析、开发和质量保障方面为多家金融和互联网企业提供落地支持，著有《前端架构：从入门到微前端》《自己动手设计物联网》等多本书籍。

AI 开发者中间件工具生态 2024 年总结

文/莫尔索

AI 应用开发者工具自下而上涵盖了模型托管与推理服务、代理工作流编排、大型模型应用的监控与追踪、模型输出的可控性以及安全工具等多个层面。模型是 AI 应用的核心组成部分，其服务需依赖推理引擎实现。开发者接入模型的方式大致可分为四类：

- 首先是以模型初创企业为代表，提供先进的商业闭源模型，如 OpenAI、Anthropic、智谱及 MiniMax 等。
- 其次是由 TogetherAI、Groq、Fireworks、Replicate、硅基流动等组成的 GPU 推理集群服务提供商，它们处理扩展与缩减等技术难题，并在基本计算费用基础上收取额外费用，从而让应用公司无需承担构建和管理 GPU 推理集群的高昂成本，而是可以直接利用抽象化的 AI 基础设施服务。
- 第三类是传统的云计算平台，例如亚马逊的 Amazon Bedrock、阿里云百炼平台、微软的 Azure AI、谷歌 Vertex AI 等，允许应用开发者轻松部署和使用标准化或定制化的 AI 模型，并通过 API 接口调用这些模型。
- 最后一类是本地推理，SGLang、vLLM、TensorRT-LLM 在生产级 GPU 服务负载中表现出色，受到许多有本地托管模型需求的应用开发者的欢迎，此外，Ollama 和 LM Studio 也是在个人计算机上运行模型的优选方案。



除模型层面外，应用层面的工具同样在快速发展，工具的进步紧密跟随 AI 应用的发展趋势。自 ChatGPT 发布以来，应用构建方式大致经历了三个阶段。

首先是基于单一提示词模板的聊天助手类应用，此阶段重点关注模型和提示词的安全性以及模型输出的可控性。例如，garak 可用于检测模型幻觉、数据泄露和生成毒性内容等问题；rebuff 则针对提示词注入进行检测；DSPy 框架提供了系统高效的编程方法，帮助解决应用开发中的提示编写问题；而 LMFormat Enforcer、Guidance 及 Outlines 等项目旨在帮助开发者控制模型输出的结构，以获得高质量的输出。

第二个阶段涉及通过组合一系列提示词和第三方工具或 API 来编排复杂的工作流，这是目前成熟的 AI 应用构建思路之一。值得注意的是，RAG 技术的出现，得益于大语言模型天然适合处理知识密集型任务，RAG 通过从外部记忆源检索相关信息，不仅提高了模型生成的精确性和相关性，还解决了大语言模型在数据隐私保护、实时数据处理和减少幻觉问题等方面的局限。RAG 技术在数据预处理和索引构建方面的努力，直接影响最终应用的效果。

尤其是在本地数据预处理方面，PDF 内容处理成为一大难点，众多开源项目应运而生，如基于传统 OCR 技术和版面分析的 Unstructured 和 Marker 库，以及结合了多模态大模型识别能力的 ZeroX 和 GPTPDF 库。

此外，还有融合了 OCR 和多模态大模型方案的 PDF-Extract-API 库。在公开在线数据处理方面，Jina Reader、Crawl4AI 和 Markdowner 等开源项目，能够将网页内容转换成适合大模型处理的上下文，从而利用最新信息提升问题回答的质量。这些项目的共同目标是将原始数据转化为有价值的资产，助力企业大规模部署 AI。

对于结构化数据，如对话历史记录和其他数据源的存储管理同样重要。向量数据库如 Chrom、Weaviate、Pinecone、Milvus 等，提供了语义检索和向量存储功能，使得 AI 应用能够利用超出模型上下文限制的数据源。传统数据库 PostgreSQL 现在也支持通过 pgvector 扩展进行向量搜索，基于 PostgreSQL 的公司如 Neon 和 Supabase 为 AI 应用提供了基于嵌入的搜索和存储解决方案。

为了有效管理 AI 应用的复杂工作流程，市场上涌现了 Dify、Wordware、扣子等低代码

平台，它们集成了多种大模型，支持外部数据接入、知识库管理和丰富的插件库，通过拖拽式配置帮助初学者快速构建 AI 应用。

同时，在开源生态系统中，LangChain、Haystack、Semantic Kernel 等编排框架的出现，使开发者能够构建、定制和测试 Pipeline，确保这些 Pipeline 的组合能够达到特定应用场景的最佳生成效果。

对于 RAG 应用，这是一种由多个环节构成的工作流应用，出现了许多端到端的开源解决方案，如 LlamaIndex 框架，它集成了数据预处理、索引构建、 多样化检索方法等功能，专为大语言模型设计；RAGFlow 是一个基于深度文档理解的开源 RAG 引擎，提供高质量的问答能力，适用于处理大规模的复杂格式数据；Verba 是向量数据库厂商 Weaviate 开源的一个模块化 RAG 框架，允许开发者根据不同的应用场景灵活定制 RAG 应用的不同环节。

第三个阶段，一些产品团队正探索开发完全由大模型驱动的代理应用。这类代理应用具备从历史记忆中反思、自主规划和使用工具执行特定动作的能力。大语言模型负责选择要调用的工具及其参数，而具体的执行动作则在沙箱环境中进行，以确保安全。

E2B、Modal 等服务提供商正是为了满足这一需求而诞生。代理通过 OpenAI 定义的 JSON 模式调用工具，这使得代理和工具能够在不同的框架中兼容，促进了代理工具生态系统的增长。例如，Composio 是一个支持授权管理的通用工具库，Exa 则提供了一个专门用于网络搜索的工具。随着更多代理应用的构建，工具生态系统将持续扩展，提供更多新功能，如认证和访问控制。

在代理应用中，记忆管理同样关键。开源项目 Mem0 将记忆分为短期记忆和长期记忆，后者进一步细分为事件记忆、语义记忆和程序记忆，并基于此抽象出一套记忆管理 SDK。Zep 通过时态知识图谱管理和更新用户信息，跟踪事实变化并提供最新数据线索。MemGPT 借鉴了计

算机操作系统内存管理机制，模拟虚拟内存工作原理，构建了一套记忆管理系统。这些项目使 AI 应用能够记住对话历史，提供更个性化、上下文感知的交互体验，极大地增强了用户的满意度。

此外，代理应用的另一个探索方向是多个代理之间的协同工作。开源社区中出现了许多解决方案，如 CrewAI 和 AutoGen 具备原生的多代理通信抽象，而 LangGraph 和 Letta 中的代理可以互相调用，良好的多代理系统设计使得跨代理协作变得更加容易实现。

鉴于生成模型本质上是一个概率黑盒，AI 应用作为一个复杂的系统，其在生产环境中的质量评估与监控尤为重要。实际应用中最大的挑战之一就是输出结果的不确定性。

面对这些挑战，需要采用科学的评估方法。LangSmith、Arise、Langfuse、Ragas 和 DeepEval 等项目提供了评估和监控所需的各种指标和工具，帮助开发者量化测量、监控和调试他们的 AI 应用系统。

展望未来，o1 模型的发布标志着大模型研究进入了新的时代。o1 模型的推理能力提升对 AI 基础设施提出了更高的要求，例如并行计算部分思维链路、减少不必要的思维过程等。研究的重点重新回到了算法层面，而非简单的算力堆砌，这对于中小型模型开发公司和学术界而言是一大利好。o1 模型的更强推理能力推动了越来越多真正的 autopilot 类产品进入日常生活，预示着 AI 技术将更加深入地融入人类社会的方方面面。

莫尔索

技术顾问，聚焦大模型工程化落地，生成式 AI 开发者，AutoGPT、Agenta、LangChain 等开源项目贡献者，著有《LLM 应用开发实践》《LangChain 编程：从入门到实践》《从零开始构建企业级 RAG 系统》。

AI Agent 逐渐成为 AI 应用的核心架构

文/张善友

随着人工智能技术的迅猛发展，大模型应用已经成为了 2024 年的热点话题，大模型应用已逐渐从初期的 Chatbot 迈向 RAG、Copilot 及 Agent 等更为高级的阶段。这些大模型具备强大的数据处理能力和深度学习能力，为各种应用场景提供了前所未有的便利。

单一的大模型在处理所有任务时往往存在局限性，因此需要借助外部工具或函数来增强其处理能力。

2023 年 6 月 13 日 OpenAI 发布的 GPT 模型的 Function Calling 功能，成为大模型与现实世界交互的桥梁。Function Calling 机制在很大程度上推动了 Agent 技术的发展。Agent 技术以其独特的自主性和智能性，正逐渐成为大模型应用的重要组成部分，引领着新的发展趋势。

在智能化方面，Agent 的学习能力得到了极大的增强。传统的机器学习技术为 Agent 提供了初步的学习框架，使得 Agent 能够通过数据驱动的方式学习并优化自身行为。

然而，随着神经网络模型的广泛应用，Agent 的学习能力得到了质的飞跃。深度学习技术使得 Agent 能够处理更加复杂、高维的数据，从而更精确地感知环境信息并做出相应决策。此外，强化学习技术的不断发展也为 Agent 提供了持续学习和自我优化的能力，使得 Agent 能够在与环境的互动中不断改进自身策略，实现更高级别的智能化。

早期的 Agent 系统往往依赖于预设的规则和策略进行决策和行动，自主性和灵活性相对有限。然而，随着大模型推理能力的发展，Agent 已经具备了更高的自主性。这种自主性不仅体现在 Agent 能够根据环境变化和任务需求自主调整行为策略上，更表现在 Agent 能够在一定程度上进行自我管理和自我修复，以应对各种复杂和不确定的情况。

AI Agent 的发展趋势是其逐渐成为 AI 应用的核心架构，通过自主感知、决策和执行能力，实现对现有软件的智能化改造和升级，从而改变业务流程和用户交互方式。

2024 年，开源社区中出现了一些著名的多智能体框架，如 MetaGPT、LangGraph 和 AutoGen，这些框架通过不同的技术手段来解决多智能体系统中的信息冗余和任务复杂性问题。这些框架的出现表明，未来多智能体系统将更加注重灵活性和可定制性。多智能体框架逐渐成为主流，有效解决了单智能体视角局限的问题，实现了多工作流的并行处理，使得推理过程更加可靠，并具备了对多模态数据的兼容性。这种趋势表明，未来的 AI Agent 将不再局限于单一任务，而是能够处理复杂的多任务环境，提高整体效率和可靠性。

AI Agent 的发展还依赖于大模型的持续优化和创新。大模型的发展方向包括优化性能、利用庞大的训练数据集模拟人类行为以及增强模型固有的通用能力。

这些优化和创新将推动 AI Agent 在推理、规划、记忆和工具使用等方面的能力提升，2024 年的 OpenAI 的 o1 模型是这方面的典型代表，不仅是 OpenAI 所代表的闭源的大模型是这样，开源的大模型也是在不断提升智能化能力，例如 2024 年 9 月阿里发布的 Qwen 2.5 72B 模型成为全球最强开源大模型。此外，Qwen 2.5 的整体性能相比前一代提升了超过 18%，并且在多模态能力、长文本处理和指令遵循等方面也有所增强。

大模型应用从 Chatbot 到 RAG、Copilot 和 Agent 的发展历程充满了挑战与机遇。随着技术的不断进步和创新，我们有理由相信，大模型应用将在未来展现出更加广阔的应用前景和巨大的社会价值。

张善友

从事.NET 技术开发二十三年，认证 CKAD 专家，曾在腾讯工作 12 年,目前在广东智用人工智能应用研究院担任工业&社区 CTO。

业余积极参与运营.NET 技术社区、Dapr 中文社区、Semantic Kernel 中文社区，.NET 黄埔论坛以及相关开源项目，运营微信公众号“dotNET 跨平台”和“新一代智能应用”。

荣获连任 19 次微软最有价值专家 MVP，6 届华为云 HCDE，6 届腾讯云最有价值专家 TVP。

谈开源大模型的技术主权问题

文/王政

开源大模型的问题其实不是开源，甚至不是大模型，问题是政府正在深度介入这个技术领域。自习课上老师坐在前面和班长坐在前面是两种感觉，国家权力机构在场和不在场也是两种感觉。这篇文章想讲一讲权力机构的视角，即“技术主权”视角。

作为一种科学和技术的门类，开源大模型有（但不仅有）以下两种讨论方式：一种是康德和黑格尔式的，开源大模型受到“世界公民观点下的普遍历史观念”影响，造成共有的财产、公共的财产关系和永久和平的国际关系；另一种是马克思恩格斯式的，认为科技是人的力量的对象化，有时候可能站在人的对立面。

于是，在百年未有之大变局加速演变的时候，那种虚构的“美好时代”（La Belle Époque）就被戳破了，开源大模型实际上被设置在一种冲突的结构中：无论如何界定冲突的主体，其中主要的两方阵营必然不能坐视先进科技加强对方。

所以现在要谈几个问题：**第一，在一个有大模型的世界，冲突的阵营如何界定？第二，在大模型开源的世界，冲突将遵循怎样的客观规律？第三，30 年以后是什么情况？**

技术主权划定阵营

霍布斯在《利维坦》中说：“一个国家的主权在国境内具有至高无上的权威。”但是显然，在中美两国之外，欧盟国家在大模型上没有什么权威：他们缺少有影响力的企业，唯一一个自主大模型似乎是德国人工智能协会的 LEAM；他们不得不关心立法问题，以响应先进技术对欧洲社会福祉、科技成果转化和国家行为能力的挑战。这就落入了技术主权的窠臼，从社会学的作用方式来说，我们可以从一系列冠以“技术主权”的故事中寻找冲突发生的规律。

例如，2011 年德国时任内政部长 Thomas de Maizière 和时任德国电信首席执行官 René Obermann 发起了 SICT（Security in critical ICT applications and ICT architectures，关键 ICT 应用及基础设施安全）工作组，**在五个不同的应用领域对德国的技术主权进行了审查。**

这和 2013 年欧盟宣布将拨款 5000 万欧元，加快 5G 移动技术的发展是先后进行的。

这足以说明欧盟在技术落后的情况下的策略。可以想象：如果欧盟确定自己在大模型方面存在危害国家主权的依赖，他们会通过拨付资金、付诸法律、外交等渠道直接获得优势。而由于客观的国际社会的团块性质，欧盟显然会对中美采取有区别的策略。比如欧盟对华新能源汽车征收高额关税，想要用大模型增强新能源汽车产品力输出欧盟，显然就此存在了人为设置的困难。

开源的历史规律

早前，开源讨论中有一种强烈的〝去主权〞倾向，观点最早可追溯到 1996 年美国学者发表的《网络空间独立宣言》。当时普遍认为，互联网是一个全球性和无边界性很强的独立空间，不应将传统的主权概念强加于网络空间上，应该靠技术、标准以及服务协议等自律形式代替主权监管的他律形式。

短暂（20 年）的历史经验标明，Facebook、Google、Twitter，〝去主权〞论调实际上就是技术先进国家对落后国家的主权的明晃晃重塑。

开源大模型正在发展。黑格尔在某个地方说过，一切伟大的世界历史事变和人物，可以说都出现两次。他忘记补充一点：第一次是作为悲剧出现，第二次是作为喜剧出现。马克思对黑格尔的扬弃在于发现〝喜剧〞蕴含的在历史的周期性中的变革性因素，开源大模型不可能也不打算〝去主权〞：大模型不仅仅要通过显卡和机房去训练，它还要占领各种高标准基础设施去做服务，而建设这些基础设施就是政府行为能力的延伸。那些更加智能、更加聪明的大模型就是这样使用开源这个烟幕弹的。

也就是说，在技术主权划分的阵营对立和一帮墙头草的冲突格局下，开源大模型的历史规律就是阵营内的主权按照技术领先国家的意愿重塑，在阵营之间存在对抗关系。甚至不需要大模型动手，Linux 这样的互联网基础设施级别软件已经在进行这种对抗（Linux 移除俄罗斯开发者），而俄罗斯也已经宣布重组自己的阵营。

一种可能的未来

我们现在开始构思开源大模型的技术主权图景：它已经诞生了；它通过每个人的电脑、手机、

机房，执行技术先进国家对别国的主权的重塑；它给自己所处的竞争性的国际环境和对抗结构增加了新的变数。

这个变数就是，阵营之间的科技交流管控和阵营内国际科技交流是同步强化的，北约和华约，马歇尔计划和莫洛托夫计划：它们一开始纯是出于增强己方实力维持对抗强度的要求，后来也伴随国家差别观念的消退和市民社会财产观念的滋长。

于是我们似乎可以判断，如果在这一波即将到来的对抗的高潮之后大模型还存在的话，康德和黑格尔关于国际关系的构想似乎就要在 30 年的实践里逐渐成型。〝世界公民观点下的普遍历史观念〞让**开源大模型重新具有过去被认为的〝合则两利〞的性质，并形成新的共有的财产和公共的财产关系。**

于是现在要问，这 30 年我们这代人怎么办？作者认为有以下几个因素决定了我们的策略：

第一，要做大模型就要做成技术主权划定的本阵营内领先，要能够占据一切基础设施；

第二，不做大模型就要把大模型熬死，让它成为过时的观念，不过问题是你未必上位；

第三，列宁指出科技具有二重力量，即它是资本主义灭亡的力量和社会主义发展的力量，毕其功于一役。

总结

在开源大模型问题上不能用一个纯粹的科技的观点来描述发展现状和未来，而应当用一个综合的社会观点去描述：认识的来源和影响实践的机制。

从这个意义上来说，过去的国际科技交流是一个经济、文化和科技多机制协同的工作方法，它们连缀成了产业链、供应链、价值链、科技依赖链条，产生了复杂的社会实践。但是现在链条上的自由主义放任正在消退，在主要国家关于世界秩序设想的矛盾基础上，各个国家按照自身力量和利益的假设正在尽速地把这张网撕成小块，并且希望抢到其他部分。

这就限制了开源大模型所附着的复杂的社会实践。在一个琐碎、细分、以邻为壑的可能的局面中，知识产权的堑壕两面因为竞争紧紧依靠，似乎〝坚定守住就有办法〞。

而第一次世界大战已经表明，将防线对面描述为牛鬼蛇神的结果是〝旧的国家及其世代相因的治国才略一齐崩溃，以致王冠成打地滚在街上而无人拾取〞。

第二次世界大战则表明，一个更高的政治观点引领下的进攻是维持防线的全部手段。〝回形针〞行动（英语：Operation Paperclip）、〝奥萨瓦根〞行动（俄语：Операция «Оссоавиахим»，德语：Aktion Ossawakim）、中程弹道导弹条约、巴黎统筹委员会——那种加固防线和穿越防线的社会机制从来没有被消灭过。

当这个机制还正在进行的时候，开源大模型的未来就必然受到它的拉扯。决策要同时具有判断力和执行力，要团结就倾其所有的团结，要对抗就看准了方向全压上去。〝见小利而忘义，临大事而惜身〞，单纯经济观点和单纯科技观点，那只能证明开源大模型的有用性是一个伪命题。

附录：自我辩护

知乎上有一个故事：中世纪欧洲签订契约有一种奇妙的方法，因为大家普遍不识字，所以就到路边随机抓一个小男孩过来暴打一顿；二十年后问小男孩那天发生了什么，记忆犹新。

我是那个小男孩，如果你们不同意我的观点，现在可以开始打我了。

王政

中国科学技术信息研究所助理研究员，硕士，主要研究方向为知识工程，开源软件，国际科技交流。深度参与开源，开发 scheme-langserver 项目。

邮箱：ufo5260987423@163.com

2024：大模型背景下知识图谱的理性回归

文/梁磊

2024 年 11 月 30 日，适逢 GPT 3.5 发布两周年，在过去的两年时间里，国内的大模型产业在基座模型、智能体（Agent）技术以及检索增强生成（RAG）等方面都取得了显著的进步，并催生了众多优秀的开源项目。

随着越来越多高质量和多样化的数据被加入到预训练数据中，如通义千问、DeepSeek、文心一言、蚂蚁百灵等基座大模型在知识掌握、推理能力和理解水平上都有了显著的提升，在诸多榜单和真实问题上展现出了超越 GPT-4o 的水平。然而，大模型幻觉、数据时效性、隐私安全、以及推理解释性等问题并没有随着模型能力越来越强而消失，这些问题仍然存在并严重阻碍着大语言模型在垂直领域的应用。这也造就了模型越来越强，垂直领域的杀手级应用依然没有出现的怪象。

为了应对这些问题挑战，行业及社区都在不断的积极探索外部知识库与大型语言模型的方式来寻找解决方案。在此过程中，涌现出了许多出色的开源项目，它们的技术路径大致可以分为两类：一类是基于搜索引擎技术的改进，另一类则是基于知识图谱技术的发展。

以搜索引擎为基础的演进

2024 年有多个搜索引擎为基础的 RAG 框架发布并取得比较大的关注，包括 QAnything、Ragflow、MaxKB 等近 20 个开源框架。这类都是比较经典的方法，以搜索引擎的向量检索和文本检索为基础为大语言模型提供外挂的文本知识库，能够在保证垂直领域数据隐私安全的前提下，将私域知识与大语言模型有效融合提升垂直领域的应用效率。RAG 开源项目通常集 Chunk 切分、向量化、存储、检索、生成等几个阶段于一体，其核心在于其中的不同策略适应和优化，如文档处理、检索策略等。

以搜索引擎为基础的演进方案以文档检索为开始，以大语言模型的生成为终。RAG 回答问题的准确率受限于召回的 Chunks 和 LLM 的生成能力，也受限于搜索引擎向量相似度计算的不足，传统搜索引擎解决不了的问题，如难以感知文档间细粒度的实体知识关联、无法对文档内知识的知识要素执行逻辑推理等，这类 RAG 方法依然解决不了。这也让开发者陷入了“一周出 demo，半年用不好”的困境。为克服以搜索引擎为基础方法在向量计算和逻辑推理方面的不足，业界也涌现出了越来越多基于知识图谱的方案。

以知识图谱为基础的演进

知识图谱技术是 2012 年 Google 为改善搜索引擎的质量和相关性而提出的，他能够构建并理解实体及其之间的关系，能够整合不同来源的文档实现跨文档的实体关联，这使得知识图谱可以对用户查询提供更加精确和语境化的回答，可以突破向量计算的瓶颈而执行多步推理、逻辑推理。尽管有这些优势，知识图谱因其较高的构建和维护成本高，过去这几年也遭到了较多的诟病。

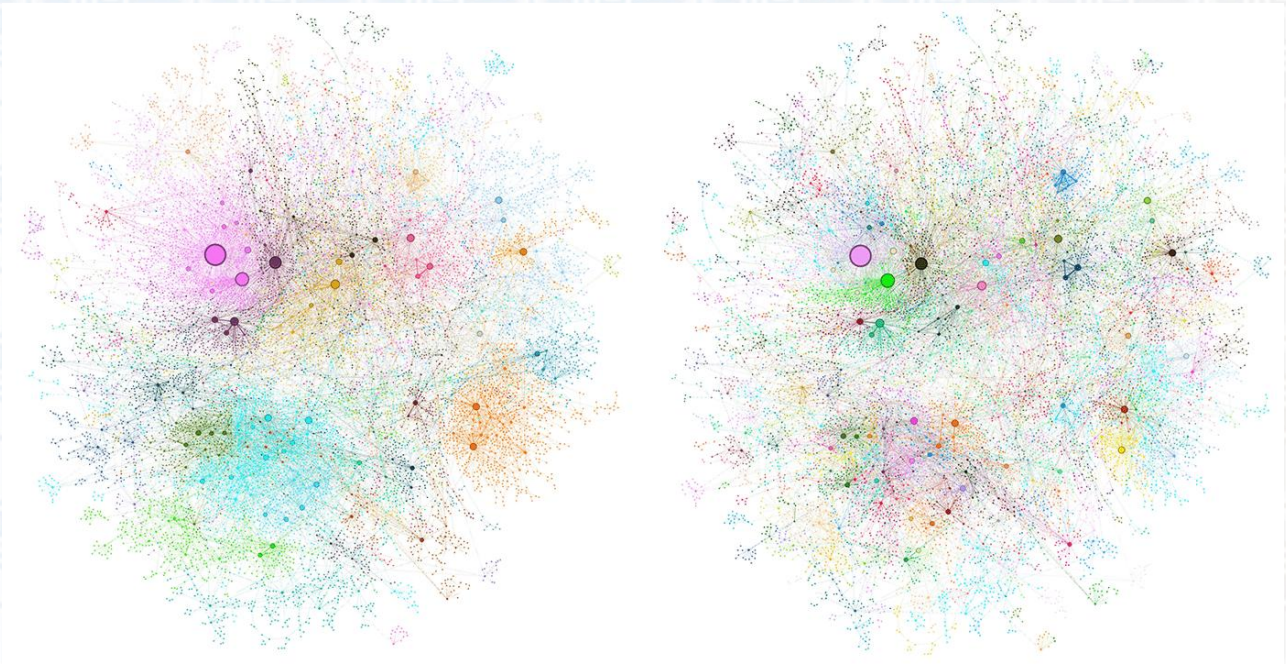
大模型技术的出现，为知识图谱技术的发展提供了新的机遇窗口。如何充分利用大语言模型的能力来克服知识图谱的不足，并充分发挥知识图谱的优势？2024 年，涌现出了多个不错的开源项目并获得了广泛的关注。

以 GraphRAG 为代表增强文档间语义关联

2024 年初有两个非常有代表性的工作，微软发布的 GraphRAG 和俄亥俄州立大学发布的 HippoRAG。两者都引入了知识图谱的方法通过开放信息抽取(OpenIE)来构建跨文档的细粒度语义关联以期缓解 RAG 在这方面的不足。

GraphRAG 借助大模型和社区挖掘构建层次化社区摘要以支持能更全面的回答全局性问题，比如“文档中的娱乐明星都出席过哪些活动”，而 HippoRAG 则引入了 PPR 及 IRCoT 的方法

来挖掘跨文档的事实关联以回答多跳事实问答，比如"斯坦福哪个教授是从事神经科学阿尔兹海默症研究的"。两者的核心目标依然是更有效的召回与目标 Query 相关的 Chunks，以生成更高质量的摘要或事实问答。但由于这两个方法的目标有所不同，导致它们的技术路线在 Chunks 构建、召回、答案生成及评价指标上有所差异。



GraphRAG 使用大型语言模型 （LLM）提取的知识图谱。图片来源于：https://www.microsoft.com/en-us/research/blog/graphrag-new-tool-for-complex-data-discovery-now-on-github/

GraphRAG 类方法有效缓解了 RAG 跨文档语义关联不足的问题，无论在摘要问答和多跳问答上都取得了较大的效果提升，证明了这类方法的有效性。

后续开源的 LightRAG、DB-GPT、lazyGraphRAG 是针对 GraphRAG 资源消耗大的方法改进，OpenSPG 开源的 KAG 较多的借鉴了 HippoRAG 的思想。GraphRAG 类方法因引入 OpenIE 抽取而引入了大量噪声导致构建的知识图谱并不能直接应用于推理，知识图谱强事实性、准确推理等优势并没有得到有效发挥。

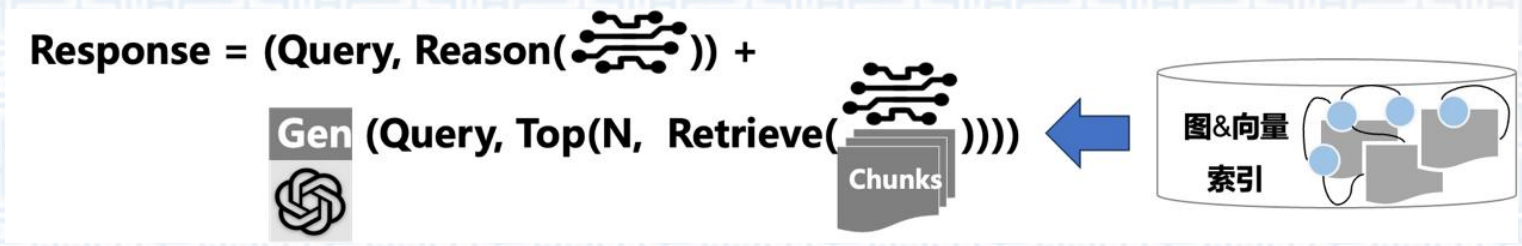
以 ToG 为代表的大模型增强传统 KBQA

2024 年也有较多以传统知识图谱 KBQA 为基础的方法，通过大模型重塑了问答阶段的问题理解和答案生成过程，同时通过大模型 SFT 技术提升问题的逻辑拆解、三元组提取的准确性，具体到图谱的推理和检索过程与传统的 KBQA 类方法基本一致，比较有代表性的工作包括 ChatKBQA、ToG 等。

这类方法是比较纯正的知识图谱技术路线，实现了结构化知识图谱与大模型技术的结合。社区关注度较高的 ToG 也推出了 2.0 版本在处理复杂推理任务、增强深度推理能力以及提供可解释和可追溯的推理路径方面更加出色。KBQA 为基础的方法依赖已构建完备的知识图谱，知识图谱长期被诟病的构建门槛高的问题依然没有得到缓解。

以 KAG 为代表的知识推理和信息检索

2024 年 10 月蚂蚁集团发布的 OpenSPG/KAG 是知识图谱类方法中比较值得关注的，它主张逻辑符号引导的推理和检索以提升垂直领域知识问答的逻辑性、事实性。KAG 在框架设计中充分考虑了知识图谱、向量检索及大语言模型的能力优势，传统知识图谱被广为诟病的构建门槛高、知识稀疏性等问题在 KAG 框架中都得到了较多的诠释。



KAG 框架是结合医疗、政务等垂直领域应用打磨而来，其中为降低领域知识图谱的构建门槛，KAG 适配了开放信息抽取以支持垂直领域的开箱即用和快速冷启动，并通过自动知识对齐模块来缓解开放信息抽取带来的噪声问题；为提升推理准确性并降低知识稀疏性带来的影响，KAG 引入了分层知识推理与检索框架，在结构化推理无果的情况下借鉴 QFS 的思想从 Chunks 中检索与目标问题相关的答案。

KAG 框架上算是知识图谱与大模型技术的集大成者，代码中大量使用的本体结构、逻辑规则等图谱的技术元素。KAG 目前开放的是一个比较基础的版本，一般用户的上手成本还比较低，

基本可以开箱即用。同时，结合垂直领域的推理要求还有较多工作需要持续优化，大量使用的图谱技术也让开发者的优化有一些学习门槛。

展望

随着大模型训练范式从预训练（Pre-training）阶段向后训练（Post-Training）阶段的迁移，人们的关注焦点也逐渐从语言模型的生成能力越来越多转向推理能力。这一转变的本质是更加重视模型理解和处理复杂问题的能力。

以此为驱动，垂直领域私域知识库的应用也会更关注解决复杂问题的能力，如指标解读、研报生成、诊疗决策、表格计算、事实问答等，这些都是传统 RAG 向量计算模型难以解决的。

随着大语言模型理解能力的不断增强，知识图谱可以不断克服并降低其构建门槛高、知识稀疏性等带来的影响，其固有的强推理能力和高可解释性的优势将得到更充分的发挥。可以预见，2025 年基于知识图谱+大语言模型的垂域推理应用和开源项目将越来越多的涌现，为垂直领域的复杂问题问答推理提供新的解决方案。

梁磊

蚂蚁集团知识引擎负责人，KAG 项目负责人，OpenKG TOC 专家。个人的主要技术方向为知识图谱、图学习及推理引擎、AI 引擎等，也从零到一基于蚂蚁多样化的业务场景构建了企业级知识图谱平台，平台累计提报专利 140 余项，软件著作权 10 余项。主导的项目先后获得 BU 总裁特别奖、优秀科技成果、金融科技创新奖、金融科技发展奖等。

KAG 是一个知识增强生成的专业领域知识服务框架，KAG 依赖 OpenSPG 提供的引擎依赖适配、逻辑推理执行等能力：

https://github.com/OpenSPG/KAG

人工智能与处理器芯片架构

文/包云岗

一、引言

芯片有几十种大类，上千种小类，**本文主要关注处理器芯片**。这类芯片的特点是需要运行软件，例如：**微控制处理器（MCU）**会运行实时操作系统或者直接运行某个特定程序；**中央处理器（CPU）**往往会运行 Windows、Linux 等复杂操作系统作为底座支撑整个软件栈；**图形处理器（GPU）**一般不加载操作系统而是直接运行图形图像处理程序，**神经网络处理器（NPU）**则直接运行深度学习相关程序。

处理器芯片设计是一项很复杂的任务，整个过程犹如一座冰山。冰山水面上是用户或者大众看到的处理器芯片架构，呈现为一组微架构核心参数，比如 8 核、8 发射乱序执行、32KB 指令 Cache、2MB L2 Cache 等等。

但为何是选择这样的配置，不同配置对处理器的 PPA（性能、功耗、面积）有什么影响？要搞清楚这些联系，则**需要一整套处理器架构设计基础设施的支撑（即冰山水面下部分）**——从程序特征分析技术、设计空间探索技术、高精度模拟器、系统仿真技术、验证技术等等；还需要对大量程序特征进行分析，需要收集大量的原始数据，需要大量细致的量化分析，需要大量的模拟仿真……

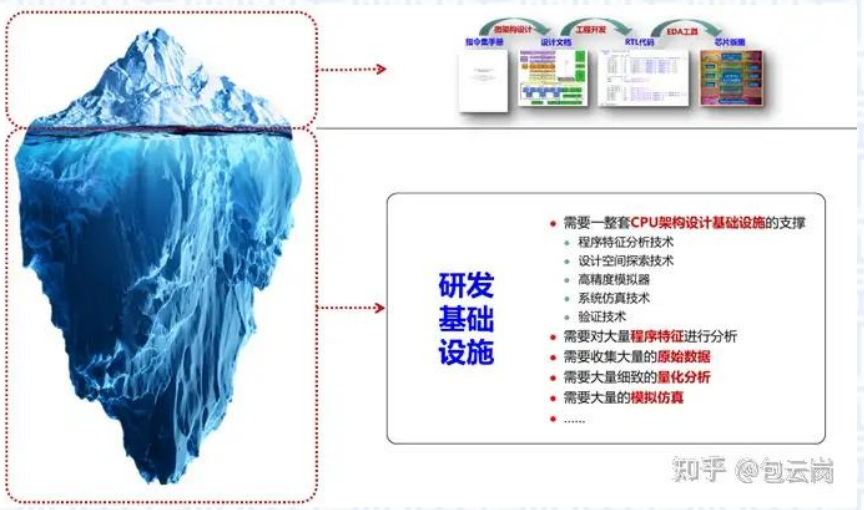


图 1.处理器芯片研发之冰山模型

以**苹果于 2020 年推出的 M1 处理器**为例，其微架构中有个模块 ROB（Reorder Buffer）设计为 630 项。这是一个很奇怪的数字，可以说是颠覆了传统 CPU 架构设计人员的观念，以致于有人在技术网站上讨论 M1 微架构时提出**这会不会是一个笔误**，因为一方面以往 CPU 的 ROB 一般都不超过 200 项，另一方面是 ROB 项目一般都是 32 或者 64 的倍数。更进一步，苹果为什么要这么设计？为什么不是 400 项 ROB 或者是 800 项 ROB？

显然，苹果在其公司内部拥有一整套 CPU 研发基础设施，能通过分析 APP Store 上数百万个应用来提取程序特征，根据程序特征开展微架构设计空间探索，开展大量实验进行量化评估分析 PPA，最终确定微架构参数配置。

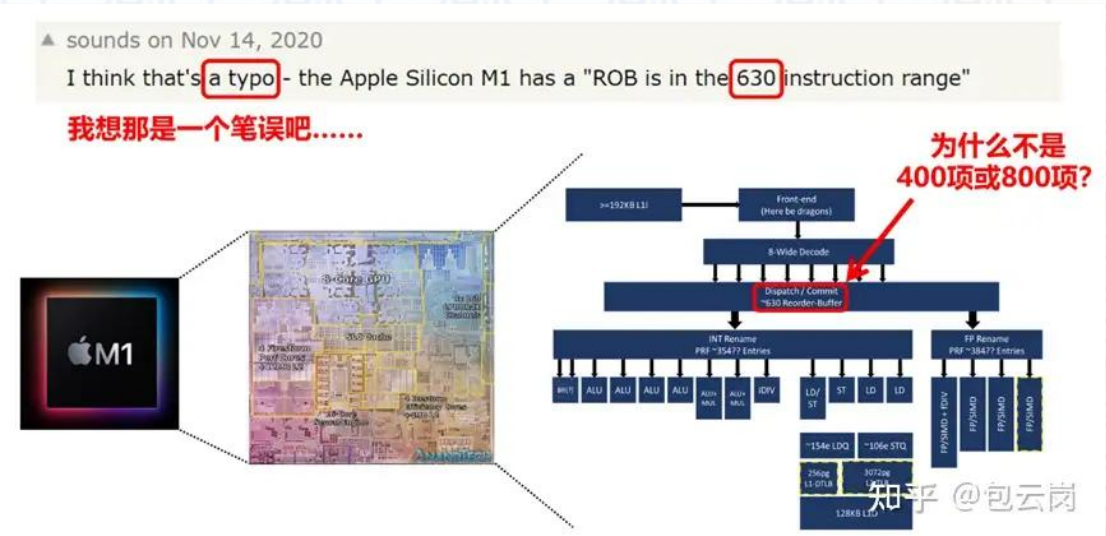


图 2.苹果 M1 处理器引起的讨论

从上述苹果 M1 芯片的例子可知，处理器芯片设计过程存在程序特征分类与提取、微架构设计空间探索、PPA 多目标优化等环节，在这些环节中 AI 技术可以发挥积极作用，这方面工作可归类为“AI for Chip”。

另一方面，随着 AI 应用越来越广泛，如何加速 AI 应用也成为处理器芯片领域的热点，最近十余年各类 AI 处理器芯片不断涌现，这方面工作科归类为“Chip for AI”。本文将分别从这两方面做简要介绍。

二、AI for Chip

总的来说，将 AI 技术应用到处理器芯片领域，总体上都是围绕以下目标展开，即性能更高、开发更快、面积更小、功耗更低，安全更好。具体而言，这方面的工作可分为三类：一是在 **AI 嵌入芯片**，在处理器芯片微架构设计中应用 AI 技术；二是 **AI 设计芯片**，在处理器芯

片开发过程中应用 AI 技术；三是 **AI 调优芯片**，利用 AI 技术搜索更优的软件与芯片适配参数。

2.1 AI 嵌入芯片

处理器微架构设计优化的一个思路是**发现程序行为中的共性特征并进行加速**。过去，业内戏称处理器微架构优化有“三大法宝”：**缓存、流水线、并行**。今天，还可以加上一个法宝，即**预取**。这“四大法宝”都对应着程序行为的共性特征，比如缓存和预取为了充分利用程序的局部性特征。今天的教科书中都会介绍局部性特征，并进一步细化为时间局部性与空间局部性，这些特征又是因为程序是顺序执行且存在大量循环。不过顺序执行、大量循环这些显而易见的特征早在 1960 年代就已被提炼与总结，今天再要从大量程序中提炼出运行时的共性特征已不是那么容易，因为**有些特征不再是人类能够直观理解，很多表现为统计意义上的特征**。

AI 技术正是挖掘统计意义特征的有效手段。过去十几年，很多体系结构研究开始考虑在芯片微架构中引入一些借鉴 AI 思想可挖掘统计特征的模块。

以分支预测单元 BPU（Branch Prediction Unit）为例，这是现代高性能处理器的一个重要组成部分，负责根据分支指令执行历史预测分支的走向，从而提前执行对应方向上的指令。**BPU 的预测准确率直接影响着处理器整体的性能和功耗**，当 BPU 预测准确率高，则处理器流水线的空泡（Stall）就比较少，甚至完全消除。但当 BPU 出现预测错误，不仅已执行的错误路径上的指令都被浪费，而且还需要冲刷流水线等来保证后续执行的正确性，这降低了处理器性能，也带来额外的功耗。

BPU 的本质作用就是提炼程序执行过程中的分支行为特征，这是一项很有挑战的任务，一直以来都属于前沿研究，基于神经网络的动态预测便是一条技术路线。其中一项经典工作是 2001 年德州大学奥斯汀分校的 Daniel Jiménez 和 Calvin Lin 提出的一种**可用硬件实现的使用基于单层感知机的预测器**[1]。如今，基于感知机的分支预测器已应用在商业处理器中。根据公开资料，AMD、三星的多款处理器中都包含了基于感知机的神经分支预测器。

还有很多研究提出将 AI 设计嵌入到芯片微架构的设计，但总体而言这类工作面临的一大难点是如何在硬件上可实现。很多设计方案由于硬件实现开销较大，实现复杂的 AI 算法存在困难而无法落地。

2.2 AI 设计芯片

AI 设计芯片已成为近年来的热点，各类工作层出不穷，大家的理解也各有不同。在这里，本文将“AI 设计芯片”定义为**从设计规范（specification）到 GDS 文件的整个芯片设计流程中，存在一个或多个环节使用 AI 技术参与设计，将一个格式的输入转换为另一种格式的输出**。在这个定义下，AI 设计芯片的智能化的程度可由覆盖的环节数量来确定，比如：谷歌的 AlphaChip[2]、Cadence 的 iSpatial[3]工作覆盖了布局布线（输入为 RTL，输出为布局布线后的 Layout）；纽约大学、中科院计算所开展的基于 ChatGPT 设计芯片则覆盖了从规范到 RTL 的整个前端设计环节[4][5]；中科院计算所的启蒙 1 号工作则覆盖了从设计规范到 GDS 的全流程[6]，自动化程度最高。（值得一提的是这个定义不涵盖一些参数优化配置类的工作，这类工作可归到下一节“AI 调优芯片”）

AI 设计芯片仍然处于起步阶段，各界都在积极探索，虽然已有一些亮点成果，但尚无被业界广泛使用的成熟解决方案。**中科院计算所陈云霁团队曾在 2022 年《中国科学院院刊》上发表了一篇题为“Chip Learning：从芯片设计到芯片学习”的观点文章**[7]，对 AI 设计芯片做了很好的总结与展望，在此引用其中一些观点。文章认为 AI 设计芯片可以分成 3 个重要问题（如图 3）：

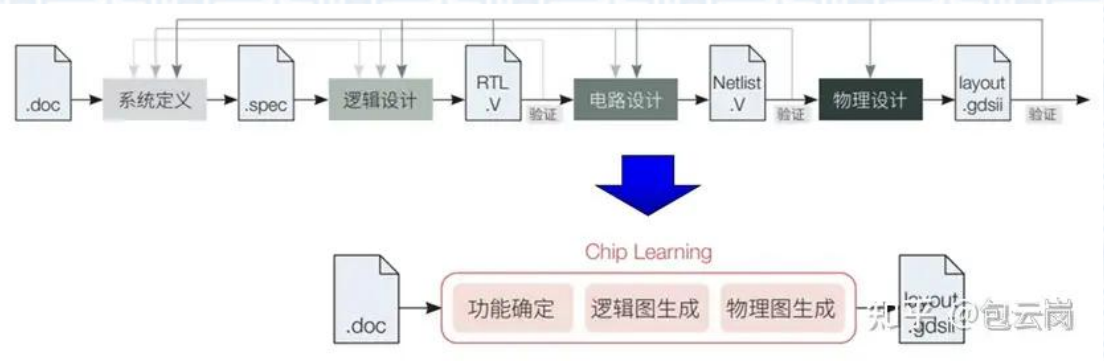


图 3.AI 设计芯片流程[7]

①功能确定。根据用户意图确定系统正确的功能，并生成系统的准确表达。这种准确表达可以是硬件代码，也可以是表达式，也可以是真值表。这个问题对应着传统芯片设计流程的逻辑设计。

②逻辑图生成。在准确表达的基础上生成电路的逻辑图表达，并在这张逻辑图上进行优化，最后生成物理无关（包括工艺）的逻辑图表达。这个问题对应着传统芯片设计流程的电路设计。

③物理图生成。在电路逻辑图基础上生成电路的具体物理版图，这等价于一种多种约束下（如面积、功耗、物理等限制）的图映射和优化问题。这个问题对应着传统芯片设计流程的物理设计。

针对上述问题，该文章由梳理了一系列从逻辑设计、电路设计、物理设计、验证测试各环节上的技术挑战。例如，验证测试环节要解决的核心挑战就是**黑盒解决方案的精度保证**，一方面端到端全自动设计的芯片是一个黑盒，验出 bug 后的可调试性将会是一个挑战，另一方面现有的 AI 技术也像个黑盒，缺乏可解释性，对于输出结果的精度无法保证与解释。关于 AI 设计芯片的更多内容，推荐大家阅读该观点文章。

2.3 AI 调优芯片

处理器芯片的最终性能取决于三个阶段的设计（参数）空间探索（Design Space Exploration，DSE）：第一阶段是**前端微架构设计**，即探索各种微架构参数的最优组合提高处理器芯片性能，就如前面提到的苹果 M1 处理器的 630 项 ROB 设计方案；第二阶段为**后端物理设计**，即探索不同的布局布线方案，不同的工艺参数提升 PPA；第三阶段是芯片运行过程中根据软件特征动态调整芯片参数，或者反过来对软件进行优化从而让芯片运行更高效，这是一种**软硬件协同优化**思路，例如根据软件负载轻重来动态调整电压频率以降低处理器芯片运行功耗，也可以优化软件提高处理器运行效率以缩短软件运行时间。

AI 技术在上述三个阶段都能发挥重要作用，事实上近年来已有大量的相关工作，有些甚至已经成为落地应用到产品中（比如 Synopsys 推出的 DSO.ai 工具[8]）。这里不再展开介绍，仅介绍两个具有代表性的华人学者的工作。

早在 2010 年左右，中科院计算所陈云霁研究员团队与南京大学周志华教授合作研究如何在微处理器设计阶段使用 AI 技术来提高设计空间探索（DSE）的效率和效果。传统的 DSE 方法依赖于大规模的周期精确架构模拟，非常耗时。研究团队提出了一种新的 COMT（Co-Training Model Tree）方法，能够利用未标记的设计配置来提高模型的预测准确性。这项工作发表于 2011 年的 IJCAI 会议[9]，是 AI 调优芯片方向的早期工作之一。

2021 年，杜克大学陈怡然教授团队与 ARM 公司合作，使用一套统一的机器学习模型同时对设计和运行阶段的 CPU 功耗进行快速实时计算，发表于 2021 年计算机体系结构领域

MICRO 会议，论文题目为《可用于大规模商业化处理器的全自动化功耗模拟架构（APOLLO:An Automated Power Modeling Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors）》[10]（一作谢知遥博士目前已在香港科技大学任教），获得当年最佳论文。

2.4 未来方向

在芯片开发中应用 AI 技术已成为主流方向。2024 年 Hot Chips 会议第一天专门组织了一个主题为“**AI Assisted Hardware Design-Will AI Elevate or Replace Hardware Engineers?”的教程**，来自 UCSD、英伟达、Synopsys 的专家系统地介绍了当前 AI 辅助硬件（芯片）设计的最新进展。

总体来看，除了传统的基于强化学习的 AI 辅助设计方法，学术界与产业界都在积极探索如何将大语言模型（LLM）应用到芯片设计的各个环节中（如图 4）。感兴趣的读者可以访问 Hot Chips 2024 网站查阅相关资料。

图 4.大语言模型（LLM）在芯片设计中的应用

三、Chip for AI

虽然 Hinton 教授团队在 2006 年发表了关于深度神经网络的工作后并未受到广泛关注，但是在体系结构领域却有专家较早地认识到深度神经网络的潜在影响，从 **2010 年便开始探索加速深度神经网络的处理器架构设计**。另一方面，**2003 年前后，英伟达开始追求 GPU 的高性能算力与可编程性**，为 GPU+CUDA 生态大厦打下地基，成为今天 AI 算力生态的统治

者。

3.1 先驱者：DianNao 家族 AI 处理器

2010 年的体系结构领域国际旗舰会议 ISCA 在法国召开，当时来自法国 INRIA 的 Olivier Temam 教授做了题为“**The rebirth of neural networks**”的大会报告[11]，向体系结构界介绍深度神经网络以及为其硬件加速器的可行性。

在 2012 年的 ISCA 上，Temam 教授提出了第一个机器学习加速器设计[12]。随后，中科院计算所陈云霁研究员和 Temam 教授启动了一个学术合作项目——**DianNao 家族加速器设计**[13][14]。DianNao 在 ISCA-2012 加速器的基础上增加了局部存储，解决了严重影响加速器性能的内存带宽限制，发表于 ASPLOS-2014，获得了该会议的最佳论文奖，得到各界的广泛关注。从此，DianNao 家族系列加速器不断推出，DaDianNao 发表于 MICRO-2014 并获得最佳论文奖；ShiDianNao 发表于 ISCA-2015；PuDianNao 发表于 ASPLOS-2015。2016 年以后，陈云霁研究员团队为一大类神经网络加速器设计了一套名为 Cambricon 的指令集，发表于 ISCA-2016，并成立寒武纪公司。

DianNao 家族加速器向全球展示了为 AI 应用设计专用加速器这条技术路线充满前景，此后各界都开始积极投入各类 AI 处理器芯片的设计，例如谷歌于 2014 年开始部署 TPU[15]、Meta 开始自研 MTIA 芯片、Tesla 自研 Dojo 芯片等等。可以说，在这一轮全球 AI 处理器热潮中，中国科研团队起到了当之无愧的引领作用。

3.2 英伟达 GPU 与 CUDA

2001 年，斯坦福大学 Bill Dally 教授团队在处理器微结构旗舰期刊《IEEE Micro》上发表了一篇题为“**Imagine:Media Processing with Streams**”的论文正式介绍 Imagine 项目[16]，一个可加速多媒体应用的流处理器（Stream Processor）结构。这项工作很快引起了英伟达的关注，并向 Dally 教授伸出橄榄枝。

随后在 2003 年，Dally 教授担任英伟达的顾问，参与 GeForce 8800 系列 GPU 的微架构设计，指导如何在 GPU 中加入流处理器。**2006 年 11 月，第一款 GeForce 8800 GTX GPU 发布**，包含 128 个流处理器，单精度浮点运算性能达到 345.6GFLOPS，访存带

宽 86.4GB/s，远高于同期 CPU 性能。

与此同时，斯坦福大学一位年轻的博士生 Ian Buck 在导师 Pat Hanrahan 教授（2019 年图灵奖得主）的指导下开展如何方便且高效发挥 GPU 能力的研究。2004 年，Buck 发表了一篇题为“Brook for GPUs:Stream Computing on Graphics Hardware”的论文[17]，为 GPU 设计了一套支持流编程（Stream Programming）语言的编译器和运行时系统 Brook，从而能让开发者像在通用 CPU 上那样进行编程。这项工作旋即得到英伟达的青睐，于是 Buck 博士毕业后便立刻加入英伟达，带领两位工程师创立了 CUDA 项目。2007 年，CUDA 1.0 正式发布，全面适配 GeForce 8800 系列 GPU。随后，**UIUC 胡文美教授团队在 GeForce 8800 GPU 上用 CUDA 实现一些程序**，性能比通用 CPU 高 10 倍到 400 倍不等，充分展示了 GPU 的高性能与可编程性[18]。至此，**英伟达的 GPU 生态大厦的地基已基本构成，英伟达的 GPU 开始被称为 GPGPU（General Purpose GPU）**。

彼时的 GPU 生态大厦中其实还缺一块拼图——应用。当传统上用于图形处理的 GPU 具备了高性能浮点运算能力，且能像通用 CPU 那样可编程，那它可以用来做什么呢？对于这个问题，**中国的计算机科学家给出了答案——科学计算与超级计算机**。2007 年，国防科大杨学军院士带领团队在国际计算机体系结构旗舰会议 ISCA 上发表题为“A 64-bit stream processor architecture for scientific applications”的学术论文[19]，揭示了流处理器架构可大幅加速科学计算。这项工作不仅为具备大量流处理器的 GPU 开辟了新应用场景，也为超级计算机架构设计开辟一条新技术路径，即采用“CPU+GPU”异构结构提升性能。

此后，GPU 便开始出现在超级计算机排行榜（Top500）上，并逐步成为主流：2009 年 11 月，采用 Intel CPU+ATI GPU 异构结构的“天河一号”名列 Top500 排行榜第五；2010 年 6 月，采用 Intel CPU+Nvidia GPU 异构结构的“曙光 6000”位列 Top500 排行榜第二；2010 年 11 月，采用 Intel CPU+Nvidia GPU+FT-1000 异构结构的“天河一号大江 A”荣登 Top500 榜首，打破了长期以来美日霸榜的格局。

虽然彼时的英伟达 GPU 与 CUDA 已被证明是构建超级计算机的神器，但却尚未与这一轮 AI 浪潮直接关联起来。

3.3 汇聚：算法+数据+算力

普林斯顿大学李飞飞教授于 2009 年发布 ImageNet[20]，随后为了推广 ImageNet，他们决定启动一项基于 ImageNet 的物体识别竞赛。2010 年第一届竞赛冠军识别错误率为 28%，2011 年第二届竞赛冠军错误率降到约 25%。2012 年的第三届竞赛正是转折点——**加拿大多伦多大学的 Jeffrey Hinton 教授与其两位学生 Ilya Sutskever 和 Alex Krizhevsky 带着使用英伟达 GPU+CUDA 训练出来的深度神经网络 AlexNet 参加竞争**，将错误率大幅降低了近 11 个百分点，高出第二名 41%。

算法、数据、算力在 2012 年的 ImageNet 竞赛上汇聚了，形成一条势不可挡的“大江”，在学术界掀起了惊涛骇浪。而英伟达的 GPU 芯片与 CUDA 软件生态也正式搭上了 AI 快车，不断发展壮大。

3.4 新机遇：RISC-V+AI

英伟达的 GPU 芯片和 CUDA 软件生态已在当前 AI 浪潮中占据算力主导地位并形成高度垄断。打破这种垄断格局成为全球的共识，Google、Meta 等企业都自研各自的 AI 处理器芯片。

在具体技术路线上，**一个有趣的观察是几乎全球企业都选择基于 RISC-V 研制 AI 处理器芯片**。根据不完全统计，OpenAI、Google、Meta、Microsoft、Tesla 等 AI 领域系统与应用巨头，均宣布或者已经基于 RISC-V 指令架构搭建自有 AI 芯片与系统。而大量初创企业如 Tenstorrent、Rivos、Semidynamics 等不约而同地选择 RISC-V+AI 解决方案。

Semidynamics 公司在 2024 年 6 月的欧洲峰会上总结了基于 RISC-V 扩展 AI 指令集，有如下几点优势[21]：

1. *单一软件栈。传统 AI 处理器架构中 CPU 一般采用 ARM、GPU 往往采用 Imagination，NPU 自研，因而需要三套软件栈，如果都是采用基于 RISC-V 为基座来扩展 AI 指令集，那么就可以使用一套编译器和运行时（不同编译选项）。*
2. *无需考虑 DMA 操作的编程范式。当前 AI 处理器解决方案中需要通过 DMA 拷贝数据，基于 RISC-V 可实现紧耦合设计共享地址空间，从而避免显式的 DMA 操作。*
3. *其他优点包括更低延迟，更低功耗，更高效的 Cache 共享等等。因此，Semidynamic 推出*

了一款基于 4 发射乱序高性能 RISC-V 核实现向量扩展单元与张量扩展单元的 AI 处理器芯片解决方案（图 5）。

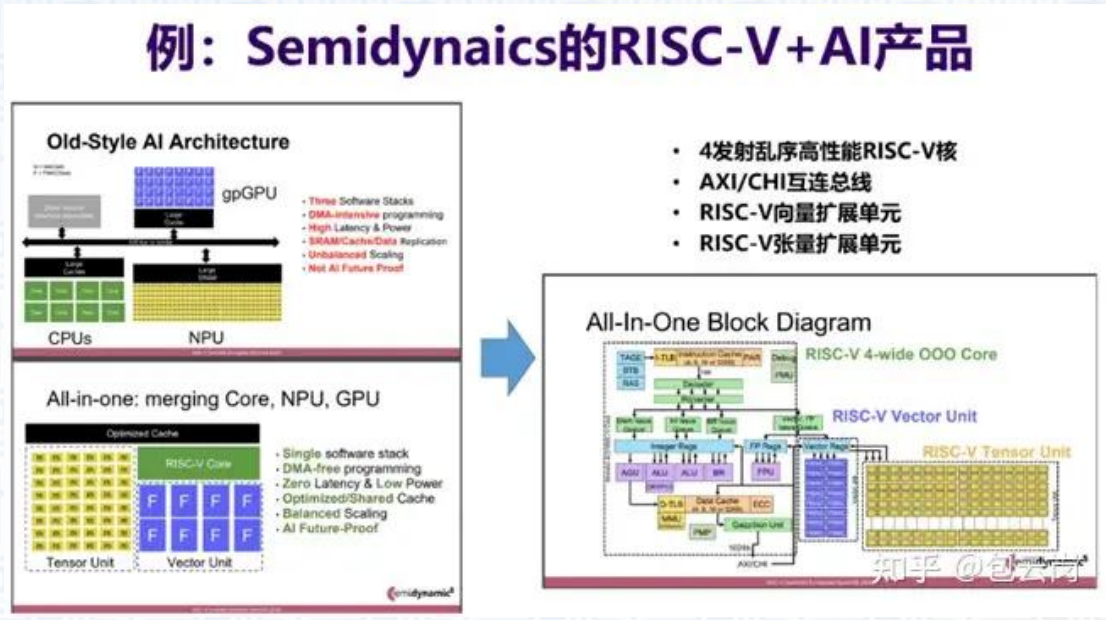


图 5.RISC-V+AI 解决方案

事实上，RISC-V 国际基金会也有布局。2023 年 10 月，RISC-V 国际基金会成立 AI/ML SIG 组，推动全球 RISC-V AI 指令架构、基础软件和核心应用技术的研究与标准化。目前，RISC-V AI 向量（Vector）已形成全球统一的标准（RVV1.0 指令集标准），正在推动矩阵（Matrix）和张量（Tensor）标准制定。

若能形成**统一的基于 RISC-V 的 AI 扩展指令集，那就有望在软件生态上形成合力**，从而形成一个能平衡 CUDA 生态的新的 AI 软件生态（图 6）。道阻且长，行则将至。

图 6.统一基于 RISC-V 的 AI 扩展指令集支撑统一 AI 软件生态

四、结语

四川宜宾，长江、金沙江、岷江三江汇流于此。有人说，长江自此始称“长江”，一路奔流向东，最终成为一条波澜壮阔的大江。

今天，人们也常说这一轮 AI 浪潮离不开三个要素的汇聚，即算法、数据与算力。这一轮 AI 浪潮是颠覆性的，但这种颠覆性技术能被预测吗？又有多少人提前预测到了？

事实上，当 Hinton 教授在 2006 年发表两篇关于深度神经网络的论文时[22][23]，即使是大多数同行也并没认识到这将会在未来成为颠覆性技术，直到 2012 年的 ImageNet 竞赛。进一步，虽然 ImageNet 竞赛让学术同行感知到了深度学习技术的颠覆性，但彼时绝大多数企业、风投机构、政府部门以及社会大众都还未感知到深度学习将会掀起新一轮 AI 浪潮，直到 2016 年 AlphaGo 战胜围棋世界冠军李世石。

改变人的思想和观念也许是世界上最难的事情之一。

此时此刻，我们又该如何预测这一轮 AI 浪潮对芯片产业发展的影响？这个问题便留给读者思考了。

参考文献

[1] Jim é nez D A, Lin C. Dynamic branch prediction with perceptrons [C], Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture. IEEE, 2001：197-206

[2] Mirhoseini, A., Goldie, A., Yazgan, M. et al. A graph placement methodology for fast chip design. Nature 594, 207–212 (2021)

[3] Cadence. Ispatial flow in genus：A modern approach for physical synthesis. (2020-06-14)[2021-11-25]. https://community.cadence.com/cadence_blogs_8/b/di/posts/ispatial-flow-in-genus-a-modern-approach-for-physicalsynthesis

[4] Jason Blocklove, Siddharth Garg, Ramesh Karri, Hammond Pearce. Chip-chat：Challenges and opportunities in conversational hardware design [C]//2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD). IEEE, 2023：1-6.

[5] Kaiyan Chang, Ying Wang, Haimeng Ren, Mengdi Wang, Shengwen Liang, Yinhe Han, Huawei Li, Xiaowei Li. Chipgpt：How far are we from natural language hardware design [J]. arXiv preprint arXiv:2305.14019, 2023.

[6] Shuyao Cheng, Pengwei Jin, Qi Guo, Zidong Du, Rui Zhang, Yunhao Tian, Xing Hu, Yongwei Zhao, Yifan Hao, Xiangtao Guan, Husheng Han, Zhengyue Zhao, Ximing Liu, Ling Li, Xishan Zhang, Yuejie Chu, Weilong Mao, Tianshi Chen, Yunji Chen. Pushing the limits of machine design：Automated cpu design with ai [J]. arXiv preprint arXiv:2306.12456, 2023

[7] 陈云霁，杜子东，郭崎，李威，谭懿峻. Chip Learning：从芯片设计到芯片学习，中国科学院院刊，2022.

[8] https://www.synopsys.com/ai/ai-powered-eda/dso-ai.html

[9] Qi Guo, Tianshi Chen, Yunji Chen, Zhi-Hua Zhou, Weiwu Hu, Zhiwei Xu. Effective and Efficient Microprocessor Design Space Exploration Using Unlabeled Design Configurations. IJCAI 2011.

[10] Zhiyao Xie, Xiaoqing Xu, Matt Walker, Joshua Knebel, et al. APOLLO：An Automated Power Modeling

Framework for Runtime Power Introspection in High-Volume Commercial Microprocessors, MICRO, 2021.

[11] Olivier Temam. The rebirth of neural networks. ISCA 2010. https://pages.saclay.inria.fr/olivier.temam/homepage/ISCA2010web.pdf

[12] Olivier Temam. A defect-tolerant accelerator for emerging high-performance applications. ISCA 2012.

[13] DianNao 项目介绍—-智能计算系统官方网站, https://novel.ict.ac.cn/diannao/

[14] from DianNao to Cambricon|The evolution of AI Accelerator, http://zhifeiding.github.io/programming/history/2024/01/31/from-DianNao-to-Cambricon-The-Evolution-of-AI-Accelerator/

[15] David Patterson. A Decade of Machine Learning Accelerators:Lessons Learned and Carbon Footprint, 2022. (中文翻译：TPU 演进十年：Google 的十大经验教训. https://zhuanlan.zhihu.com/p/573794328)

[16] Brucek Khailany, William J. Dally, Ujval J. Kapasi, Peter R. Mattson, Jinyung Namkoong, John D. Owens, Brian Towles, Andrew Chang, Scott Rixner, Imagine: Media Processing with Streams. IEEE Micro 21(2): 35-46 (2001)

[17] Ian Buck, Theresa Foley, Daniel Reiter Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, Pat Hanrahan, Brook for GPUs: stream computing on graphics hardware. ACM Trans. Graph. 23(3): 777-786 (2004)

[18] Ryoo, S., Rodrigues, C. I., Stone, S. S., Baghsorkhi, S. S., Ueng, S. Z., Stratton, J. A., & Hwu, W. M. W. Program optimization space pruning for a multithreaded GPU. In Proceedings of the CGO, 2008

[19] Xuejun Yang, Xiaobo Yan, Zuocheng Xing, Yu Deng, Jiang Jiang, Ying Zhang, A 64-bit stream processor architecture for scientific applications. ISCA 2007: 210-219

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. CVPR 2009.

[21] Roger Espasa. All-in-One RISC-V AI compute engine, RISC-V Summit Europe, 2024. https://riscv-europe.org/summit/2024/media/proceedings/plenary/Tue-09-40-Roger-Espasa.pdf

[22] Hinton, G. E., Osindero, S. and Teh, Y. A fast learning algorithm for deep belief nets. Neural Computation, 18, pp 1527-1554, 2006.

[23] Hinton, G. E. and Salakhutdinov, R. R. , Reducing the dimensionality of data with neural networks. Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

包云岗

包云岗博士，研究员，现任中国科学院计算技术研究所副所长，中国科学院大学计算机学院副院长，中国开放指令生态（RISC-V）联盟秘书长。主要研究领域为开源芯片与敏捷设计、数据中心体系结构等，带领团队在国内率先开展了一系列开源芯片实践。

大模型生成代码的安全与质量

文/谢筱恒

随着人工智能技术的快速发展，基于大型预训练模型的代码生成工具，如 OpenAI 的 Codex 和 GitHub 的 Copilot，已成为开发者的重要助手，能显著地提升开发者的开发效率。

然而，这些工具在带来便利的同时，也带来了代码安全性和质量的新挑战。根据 IDC 数据，82%的开发者已使用带有智能代码生成功能的工具，而这些开发者中的 71%表示其超过 40%的代码由这些工具生成。

GitClear 研究检查了 2020 年 1 月至 2023 年 12 月间最初编写的 1.53 亿行代码，在 AI 辅助编程用户激增期间，代码翻转（即需在两周内修复或恢复的代码）情况明显增加，这也表明代码稳定性在下降。

> We find disconcerting trends for maintainability. Code churn -- the percentage of lines that are reverted or updated less than two weeks after being authored -- is projected to double in 2024 compared to its 2021, pre-AI baseline. We further find that the percentage of "added code" and "copy/pasted code" is increasing in proportion to “updated,” “deleted,” and “moved” code. In this regard, code generated during 2023 more resembles an itinerant contributor, prone to violate the DRY-ness of the repos visited.

同时，Snyk 的调查显示，超半数组织“偶尔”或“频繁”地遭遇了因低质量 AI 生成代码引发的安全问题。

本文将探讨大模型生成代码的安全性与质量问题，并分析相关技术手段与未来发展方向。

大模型生成代码的常见安全和质量问题

大模型基于 Transformer 架构，经过大规模代码数据集的预训练，能够理解并生成符合上下文逻辑的代码。然而，这些模型并不具备真正的编程理解能力，而是通过学习大量代码样本中的模式进行生成。因此，生成的代码可能存在如下的安全和质量问题。

代码质量不稳定。大模型生成的代码有时存在语法错误、逻辑问题或无法通过测试的情况。例如，生成的代码可能缺少必要的语法元素（如括号、分号）而无法编译，或在运行时因逻辑错误导致意外行为。此外，许多生成的代码在功能测试中无法通过，增加开发者的修复工作量，降低了效率。

缺乏上下文理解。大模型通常基于输入的表面模式生成代码，缺乏对业务逻辑和系统架构的深度理解。这导致生成的代码可能不符合项目的整体编码规范，无法正确调用已有模块或接口，甚至与现有功能重复。在复杂项目中，这种缺乏上下文的生成方式使代码的直接可用性较低，开发者需要投入额外精力进行适配和修改。

安全隐患。生成代码在处理用户输入、文件操作和网络请求等敏感操作时，容易忽略安全性，导致潜在漏洞。例如，未对用户输入进行验证可能引发 SQL 注入或 XSS 攻击，文件路径处理不当可能导致目录遍历或未授权的文件读写，网络请求中未正确加密数据可能导致信息泄露。这些问题增加了代码在生产环境中的风险，可能被恶意利用。

法律和版权问题。大模型生成的代码可能基于公开代码库或开源项目，而这些代码可能受到版权保护，存在侵权风险。例如，生成的代码可能直接复制开源代码片段，未遵守相关开源协议，或与受版权保护的代码高度相似，从而引发法律争议。这可能导致开发者或组织面临知识产权诉讼及相关损失。

影响大模型生成代码质量的因素

模型训练数据的质量。大模型的代码生成能力极大程度依赖于训练数据的质量。如果模型训练时使用的数据集包含不良的代码模式、不规范的编码风格或者过时的技术，生成的代码往往会继承这些问题，导致质量较差。例如，模型可能生成带有潜在安全漏洞的代码（如未验证用户输入的 SQL 查询），或使用已经被淘汰的技术框架和方法（如不安全的加密算法）。此外，训练数据中若缺少高质量的代码示例，生成结果可能在可维护性、性能和测试性方面表现不佳。

上下文理解的局限性。大模型的代码生成通常基于短期上下文（如用户输入的一段描述或代码片段），缺乏对整体系统架构和长期上下文的理解。这种局限性使得其在复杂项目中难以生成高质量的代码。例如，在大型软件项目中，模型可能忽略全局变量的依赖关系、模块间的交互逻

辑或编码规范，导致生成的代码无法融入现有的系统环境。此外，模型难以理解业务需求背后的深层逻辑，可能生成表面上正确但实际不可用的代码，从而给开发者带来额外的修改和调试工作。

缺乏调试能力。与人类开发者不同，大模型无法主动调试或验证自己生成的代码，因而可能生成带有隐藏逻辑错误的代码，而这些错误往往难以快速定位和修复。例如，生成的代码可能在边界条件下表现异常，或者因函数调用错误导致运行时崩溃。这种缺乏调试能力的特性，使得开发者需要手动排查生成代码的潜在问题，增加了工作负担。此外，生成代码可能未遵循最佳实践（如缺乏注释或测试用例），进一步加大了调试难度。

如何保障大模型生成代码的安全与质量

为提升大模型生成代码的安全性与质量，业界与学术界提出了多种解决方案，涵盖工具研发、模型改进和技术研究等多个方面。

生成代码的安全性验证与质量评估。工业界已开发了多种代码分析工具，如 Black Duck 和 AWS 提供的服务，能够帮助开发者检测并修复 AI 生成代码中的安全漏洞。工具如 CodeQL 和 CodeFuse-Query 支持通过自定义查询快速定位安全隐患，并以高性能的语法和语义分析能力提供实时反馈。这些工具不仅速度快，还支持定制化查询，并能无缝集成到开发环境和持续集成工具链中，自动化地检查和优化生成代码的安全性与质量。此外，研究者也在探索通过形式化验证、符号执行等技术对生成代码进行安全审查，并利用 AI 自动评估代码质量，特别是关注可维护性和性能等指标。

结合安全编码实践的生成模型。将常见的安全编码准则融入大模型的训练中，是提升生成代码安全性和可靠性的关键。例如，通过在模型训练数据中引入安全编码示例，增强模型在处理用户输入、文件操作、网络请求等场景下的安全性。同时，近年来的研究还表明，结合大模型的程序语义理解能力，通过提示工程技术，可以引导模型输出更高质量的代码分析结果，如检测源代码中的潜在漏洞。这类研究不仅提升了大模型的代码分析能力，也为开发者提供了新的技术手段。

自动调试与修复功能的集成。开发集成自动调试与修复功能的代码生成工具，是保障代码质量的重要方向。这些工具可以在代码生成后，自动检测并修复潜在错误，减少开发者手动修复的负担。例如，CodeFuse 的编译诊断修复 Agent 可以基于代码分析发现质量和安全问题，然后

基于预设规则或模型预测直接修复漏洞，从而提高代码的稳定性和可用性。

多模态模型的应用与上下文理解。未来的代码生成模型可能结合代码、自然语言和图像等多模态信息，进一步提高生成代码的准确性和上下文理解能力。例如，通过结合项目文档、架构设计图和代码片段，模型可以更全面地理解开发环境，从而生成更符合业务逻辑或系统架构的代码。

可解释性与审计功能的增强。提升代码生成模型的可解释性与透明性是增强信任度的核心方向。例如，通过提供审计机制，开发者可以追溯生成代码的来源并了解模型的决策过程。这不仅有助于发现潜在问题，还能帮助开发者更高效地验证代码的安全性和质量。

结语

大模型生成代码在提高开发效率、降低编程门槛等方面展现出了巨大的潜力。然而，代码的安全性与质量问题仍然是亟待解决的挑战。通过结合静态分析、强化学习、模型优化等技术手段，可以在一定程度上提升生成代码的安全性和质量。

未来的研究应聚焦于模型的可解释性、安全性优化、多模态融合等方向，以充分发挥大模型在代码生成领域的优势。同时，开发者在使用大模型生成代码时，应保持审慎态度，结合自身专业知识，对生成的代码进行仔细的审查和测试，确保代码的可靠性和安全性。

谢筱恒

蚂蚁集团程序分析团队技术专家，香港科技大学博士。蚂蚁程序分析团队致力于打造蚂蚁的先进、专业、普惠的程序分析系统，过去四年团队发表 30+篇国际顶会论文，技术能力广泛应用于变更影响评估、安全风险检测、代码自动修复、构建程序知识图谱以及智能问答系统等多个方面。在智能研发时代，我们正在全力推进软件工程和 AI 智能结合的技术演进，努力帮蚂蚁打造软件工程 3.0 时代的研发新范式。

2024 年 AI 大模型如何影响基础软件行业中的「开发工具与环境」

文/张宏波

前言

AI 大模型是指在机器学习和深度学习领域，采用大规模参数（至少在一亿个以上）的神经网络模型。这些模型在训练过程中需要消耗大量算力和高质量数据资源。AI 大模型主要分为自然语言处理（NLP）大模型、计算机视觉（CV）大模型、多模态大模型等，其中 NLP 大模型和多模态大模型在市场中占据主导地位。

目前 AI 大模型行业应用的实现路径主要有两种：一是持续加大对通用大模型的研发投入，提升 AI Agent 能力，直接服务各行业；二是融合行业专业知识，基于通用大模型打造垂类行业模型。垂类行业大模型已通用大模型的基础，通过二次训练和优化，实现更精准的行业应用。

60%的企业通过垂类行业大模型实现 AI 大模型在行业的应用布局。金融、政务、影视游戏和教育领域是大模型渗透率最高的四大行业，渗透率均超过 50%。电信、电子商务和建筑领域的应用成熟度较高。

一、AI 大模型正在颠覆基础软件行业

基础软件是指构成计算机系统和信息技术基础设施核心的软件，它们为应用程序的运行、数据处理和系统管理提供基本功能和支持。操作系统、数据库、中间件和网络软件、安全软件、系统管理软件、虚拟化软件、云平台等都属于基础软件。

作为构建软件世界的重要工具，随着 AI 大模型能力的不断释放，基础软件行业正在快速变革，传统的软件公司受到巨大冲击。

来自美国老牌 VC Menlo Ventures 的报告显示：以 ChatGPT 为例，在它的影响下 Stack Overflow 的网络流量减少了一半。IT 外包公司和传统的自动化公司必须为即将到来的 AI 自动化

挑战做准备。随着时间的推移，即使是 Salesforce 和 Autodesk 等软件巨头也将面临被 AI 公司颠覆的危险。

> 2. **David beats Goliath: More incumbents to fall.**
> ChatGPT's disruption of Chegg and Stack Overflow this year was a wake-up call: Chegg saw 85% of its market cap vanish, while Stack Overflow's web traffic halved. Other categories are ripe for disruption. IT outsourcing firms like Cognizant and legacy automation players like UiPath should brace for AI-native challengers moving into their market. Over time, even software giants like Salesforce and Autodesk will face AI-native challengers.

面临的挑战

当然机遇之中也蕴含着挑战。

值得注意的是尽管 AI 大模型在软件开发中的潜力巨大，但也面临数据隐私与安全性、模型可解释性、资源成本高以及法律与伦理风险等挑战。为了应对这些挑战，需要制定相应的数据保护政策、提高模型透明度、优化资源分配，并在法律层面清晰界定 AI 应用的责任归属。

大模型最常用的调用方式是通过模型厂商或者云服务提供商的 API 进行调用。在一些关注数据隐私和安全的商业公司里面，这通常意味着需要从模型厂商或者云服务提供商处购买昂贵的私有化部署的实例。

大模型在大量的数据上进行了训练，获得了很强的泛化能力，但是在一些少见、结构化的数据上仍然没有办法表现的足够优秀。现有的解决方案，例如推理时进行约束解码、微调，通常会导致模型能力的退化。如何让模型能够在推理的时候“主动”地生成符合期望约束的输出，仍然是一个没有解决的问题。

二、AI 编程成为热门赛道

AI 编程在过去一年的硅谷是非常热门的话题，包括 GitHub Copilot 等 AI 产品成功破圈用户达数百万。根据 Gartner 预测，到 2028 年，智能编码助手在企业软件工程师群体中的采用率将从 2024 年初的不到 14%提升至 90%。IDC 估计，到 2029 年，AI 编码工具市场规模达 40

亿美元。最近谷歌 CEO Sundar Pichai 也表示，谷歌超 25%新代码均由 AI 完成。

很重要的一个原因是编程像数学一样结果非常容易评估，但是当下的 AI 编程有两个很大的瓶颈，第一个是缺乏对领域知识的理解，比如 GitHub Copilot 对所有的编程语言用的都是一样的模型，它甚至无法保证生成的这个编程语言的语法一定是正确的。

第二个是它缺乏原生的 IDE 支持，这也是因为时间关系，以前的 IDE 都没有预想到未来代码将由 AI 生成，所以现在的 IDE 无法与 AI 大模型进行一个高效的互操作，例如要修复 Copilot 的一个问题，我们需要一个接一个的串形进行修复，但大模型对算力的要求很高，会产生较大的延迟。

最近硅谷还有一个非常火的公司叫做 Cursor，它的官网上专门有一篇文章描述：「怎么用已有编译器传统智能帮助大模型生成一个更高效可靠的代码」，证明 Cursor 也预见到了 AI 代码生成需要更为宏观的看到工具链上的每一环，工具链包括：编程语言、集成开发环境（IDE）、调试器、编译器和构建工具等。

但因为 Cursor 是一个做 IDE 的公司，它本身不拥有编程语言的服务，如果能从编程语言开始自底向上的适配大模型，编程体验将会产生质的飞跃。

三、AI 编程主要的机遇在于自底向上重构工具链

目前绝大多数编程语言诞生在 AI 大模型时代之前，也就是说此前很少有人认真设想过未来代码将由 AI 生产。

几乎没有人从底层考虑如何适配 AI 编程，建立一整套开发者工具包括编程语言、编译器、调试工具、构建系统、静态分析安全检测工具等。

特别是编程语言不仅要设计易于人类学习的编程语言，还要设计易于 AI 学习和生成代码。以 Python 举例，它的优势是对于人类易读易学，动态类型系统减少了代码量，允许快速原型开发，增强了开发效率。但性能问题、内存消耗和多线程和并发处理确是它的劣势。

在 AI 大模型时代 Python 的优势不再明显，机器永不疲惫，不会因为代码枯燥而降低生成速度，劣势却愈发变成挑战。优秀的智能编程助手不能仅关注行业大模型，AI 大模型对于「编程」只是冰山上的技术，未来将是「开发工具和环境」与 AI 大模型高度整合平台的天下。

一家科技媒体近期的一篇文章中提到：“现阶段在海外能解决这种 IDE 生态问题的中国团队尚且看不到。”IDE 指的是集成开发环境，指用于提供程序开发环境的应用程序，包括代码编辑器、编译器、调试器和图形用户界面等工具，如微软 VS 系列问题。他认为，目前中国公司们哪怕想在海外实现 Cursor 这样的“插件逻辑”，也是遥遥无期。

当然，国内并非真的无人有能力整合平台生态，MoonBit（月兔）是从编程语言到集成开发环境（IDE）、调试器、编译器和构建工具等全部自研的中国团队，目前正在致力于打造下一代 AI 云原生开发平台。

MoonBit 诞生于 ChatGPT 横空出世之后，有更好的机会去重新构想整个程序语言工具链该如何与 AI 友好的协作，不用承担太多的历史包袱。一开始就从顶层设计考虑如何让语言和工具(IDE,调试器，静态分析，包管理等）一体化发展。这两点使得 MoonBit 在非常早期就构建了专属 MoonBit 的云端 IDE 和 AI 代码助手，并创新性的从底层构建之初就与 AI 大模型进行适配。

一个更加完善的 AI 编程场景出现：

- 通过思考语言设计和工具链适配，IDE 和工具链能够非常灵活地在当前的仓库检查、运行并更新 AI 大模型生成的代码块。
- IDE 内置的测试功能除了为程序员提供一种方便测试的机制以外，能够很好地在代码上下文中作为相关代码的示例，从而大大提高代码生成及相关下游任务的正确性。
- 除了在 IDE 中向用户提供常规的 AI 生成测试、撰写文档、修复错误等功能外，还能够提供完全后台运行的自动化智能体，通过读取编译器的各项反馈来完成各种相对复杂、长时间的下游任务，例如提升测试覆盖率、从自然语言提示生成完整代码组件或者项目等。

四、总结

未来程序员的工作场景，会更多的使用到 AI 大模型代码生成助手。值得注意的是，提升程序员最终使用体验有几个方向，快速的更新迭代适合生成代码的垂类 AI 大模型只是一方面，更重要的是从编程语言入手，打造一个完整和 AI 高度集成的 IDE 生态。

张宏波

现任基础软件中心首席科学家、 Moonbit 平台负责人，是通用程序语言 ReScript 的作者，程序语言 OCaml 前核心开发人员。

本科毕业于清华大学电子系，在美国宾夕法尼亚大学读博期间受 Bloomberg 邀请，从事函数式语言编译器的开发工作。

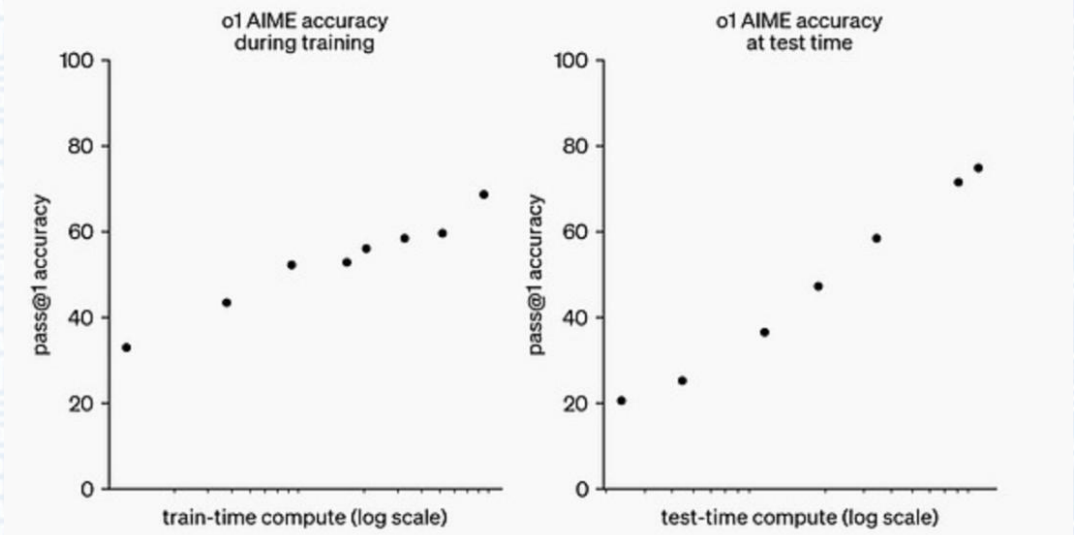
推理中心化：构建未来 AI 基础设施的关键

文/趋境科技

相比于 2023 年的参数量快速扩张，2024 年以来，大模型迭代动力更多源于大模型应用落地、端侧部署的需求，大模型正往更加广泛的行业应用发展。然而，大模型的成功落地并非易事，尤其是在为大模型提供算力的稳固底座——基础设施的建设环节中，推理的算力需求日益增加，将成为制约大模型广泛应用的关键因素。本文将探讨大模型基础设施建设的未来趋势，以及大模型在落地层面，要如何做基础设施建设。

推理算力的爆发式增长，将会转移基础设施建设的重心

OpenAI 的 o1 模型以其思维链式思考（Chain of Thought）模式，为大模型的推理带来了新的方向。这种模式通过模拟人类解决问题的思维方式，显著提升了模型的推理能力，使大模型在解决复杂的推理任务上表现出了超越以往的卓越性能，其效率也远超其他模型。



o1 performance smoothly improves with both train-time and test-time compute

但这意味着，类 o1 大模型在推理阶段需要更多的计算资源，思维链的推理模式相当于从原来的单次推理变成了多次推理，推理端对算力的需求大幅增加。原有大模型的推理模式更多是一般推断，即大模型只进行单次的简单推理，加入链式思考之后，不仅是思考次数成倍数增加，每次思考还会将上一次的思考结果作为 Prompt 再次输入，对推理的算力需求将是原来的数十倍。

除了对推理算力的爆发式增长，推理还将成为高质量的数据来源。英伟达高级科学家 Jim Fan 表示，大量计算将被转移到服务推理而不是训练前/后，o1 将成为数据飞轮，反过来将进一步优化 GPT 未来版本的推理核心。

这样的发展趋势预示着大模型的基础设施建设将向推理转移。传统的 AI 算力设施主要围绕模型训练构建，但在新的范式下，要求我们重新思考和设计 AI 基础设施去适应这一变化。未来的 AI 基础设施将更加注重推理能力，以支持大模型的广泛应用。

算力需求的增加导致的挑战

在大模型落地行业中，效果、效率与成本之间存在着难以调和的“不可能三角”。企业往往希望获得更好的模型效果，即生成内容的准确性高、无幻觉问题且对用户有实际帮助；同时也追求更高的处理效率，以便快速响应市场需求和用户反馈；然而，还需要控制成本。

举个例子来说，参数越大的模型有越高的推理性能，但成本也极高，部署千亿大模型动辄需要成数百万元至数千万元，此外还需较高的人员成本来做模型维护和应用开发。

效率、成本、效果这三个点本就难以调和，随着推理思维链带来的新范式，虽然模型效果有了更大的提升，但同时对算力的需求爆发，导致成本数十倍增加，使得这一平衡更加难以实现。在保证同样的性能条件下，如何平衡成本，成为了大模型落地的关键问题。

如何降低大模型的推理成本

平衡效果、效率与成本三者间，本质是在于如何在有限的成本里做到最优的性能。加之大模型的算力建设重点转移到推理侧，因此，如何优化推理算力成为大模型落地的关键点。

以存换算：在大模型的推理任务中，尤其在实际的应用中，大量的推理任务间往往会有关联，因此如果将计算的中间结果存储，遇到相似的问题，甚至是全新的问题，也能让大模型从历史记忆中提取相应的计算结果，结合部分现场计算，融合推理得出结论，那么推理任务的计算量将会大大降低。

这一思路的本质是调用存储空间，存储复用 KVCache，来置换部分的算力。存储设备的成

本相对算力而言要低得多，从而进一步降低成本。

全系统异构协同推理：大模型算力最主要是来源于 GPU，当前传统的算力优化方案主要是提升 GPU 的利用率。然而，仅针对 GPU 优化所带来的性能提升空间非常有限。此外，据财通证券发布的行业深度分析报告统计显示，国产 GPU 产品在单精度/半精度浮点算力、制程及显存容量上都与英伟达有 2-4 倍的显著差距。因此，单单依靠 GPU 单点优化，短时间内很难赶超英伟达 GPU 方案。

而且对于推理而言，算力要求相较于训练较低，因为推理仅涉及前向计算，无需复杂的反复试错和参数调整。

底层的硬件，除 GPU 之外，还有存储、CPU、NPU 等。从这些角度考虑，降低 GPU 算力的占比，释放设备中其他硬件的隐藏算力的方法成为最优解。

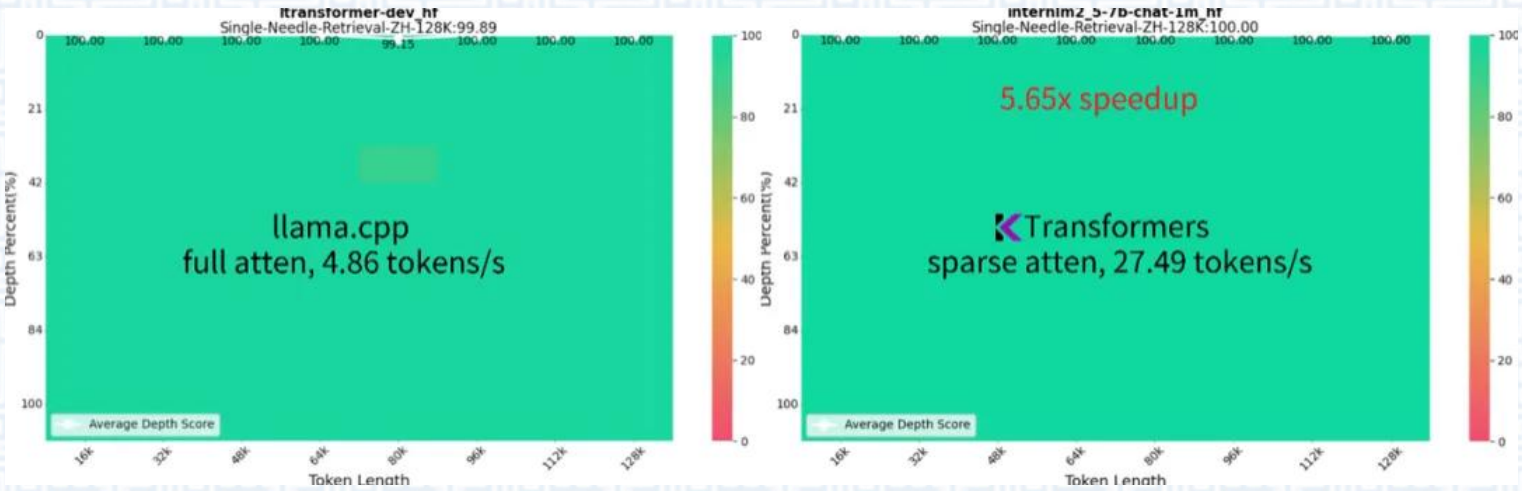
通过以存换算利用存储的算力，加入高性能算子充分调用 CPU 和 GPU 算力，增加算力利用率，再加上更科学的算力调度和分配系统，充分并高效释放了全设备的算力，这就是全系统异构协同推理架构。

这种架构不仅提高了整体算力利用率，还突破了 GPU 单点优化的瓶颈，能够为大模型的推理提供强大的支持。

推理新架构的开源项目实践

趋境科技与清华 KVCache.AI 团队共同开源的项目 KTransformers，便是采用了异构协同的推理新框架：仅需单个 4090 即可在本地运行 Mixtral 8x22B 和 DeepSeek-Coder-V2 等千亿级大模型，性能远超 Llama.cpp 数倍。

同时，KTransformers 成为业界首个仅需单张 4090 即可完成长达 1M 的超长上下文推理任务的高性能推理框架，且生成速度达到 16.91 token/s，比 Llama.cpp 快 10 倍以上，同时维持接近满分的“大海捞针”能力。更进一步的，KTransformers 可以兼容各种模型和算子，在 Transformers 之上，兼顾了兼容性、灵活性、易用性和性能。在框架中可以集成各种各样的算子，不管是放到 CPU 还是 GPU，都能够做各种组合测试。



KTransformers 在发布后仍在不停地迭代更新，根据开发者们的需求提供了对 Windows、Linux 平台的支持、实现了对主流开源 MoE 大模型的支持等，非常方便开发者根据本地环境进行调整。未来还将根据开发者的不同诉求持续完善。

此外，趋境科技深度参与的开源项目 Mooncake，以超大规模 KVCache 缓存池为中心，通

过以存换算的创新理念大幅度减少算力开销，显著提升了推理吞吐量。其中 Mooncake Store 组件可以充分利用推理集群中的 CPU、DRAM 和 SSD 资源形成一个高效的多级 KVCache 缓存池。通过共享 KVCache，大幅度减少对于 GPU 资源的消耗。

大模型的未来与算力基础设施的转型

AI 大模型是未来的趋势，随着大模型更加广泛地落地到各种应用中，对大模型的性能要求会越来越高，推理任务也变得越来越复杂。因此在上层能力与应用高速发展的状态下，底层对算力的基础设施建设，尤其以推理为中心的基础设施建设变得尤为关键。

以存换算和全系统异构协同推理是当前应对大模型落地挑战的创新解法，能够在大模型的部署阶段有效平衡效果、效率和成本三者间的关系，将成为推理基础设施建设的一种新方式，助力大模型更广泛的应用。

正如中国工程院院士郑纬民在数据存储专业委员会中所提到的：内存型长记忆存储以存换算，是 AI 推理新趋势。AI 存储是人工智能大模型的关键基座，存储系统存在于大模型生命周期的每一环，是大模型的关键基座，通过以存强算、以存换算，先进的 AI 存储能够提升训练集群可用度，降低推理成本，提升用户体验。

趋境科技
APPROACHING.AI

趋境科技成立于 2023 年底，基于业界首创的以存换算和全系统异构协同推理架构推出“大模型知识推理一体机”，为用户提供开箱即用的大模型落地解决方案。公司创始团队均来自清华大学，在 AI、体系结构、系统软件等相关的技术系统和软件领域，有多年学术与产业实践经验。

<Part 3:
国产 GenAI 生态高亮瞬间>

「2024 年度中国 GenAI 消费应用人气评选」和「Gitee AI&天数智芯创新应用大赛」的成功举办，不仅展示了国内 GenAI 消费应用和 AI 应用的现状和创新成果，也激发了更多人对科技创新的关注和热情。

这反映了中国开发者在 GenAI 领域的技术实力和创新能力，他们能够深入理解大模型的特性和应用场景，将前沿技术转化为实际可用的产品和服务。同时推动 AI 技术在更多领域的实际应用，为 AI 行业发展提供了新的思路和方向。

2024 年度中国 GenAI 消费应用人气榜 Top10

开源中国于 2024 年 11 月举办了「2024 年度中国 GenAI 消费应用人气评选」活动，旨在展示国内 GenAI 消费应用的现状，激励创新，为科技爱好者提供一个表达自己观点的平台。

本次活动候选应用涵盖【AI 通用会话与联网搜索】、【AI 多媒体创作】、【AI 编程】、【AI 智能体工坊】、【AI 文字内容效率】类别。

最终，以下 10 个应用进入「2024 年度中国 GenAI 消费应用人气评选」Top 10 榜单。

✧ 文心一言：*https://yiyan.baidu.com/*

✧ 夸克：*https://www.quark.cn/*

✧ Kimi：*https://kimi.moonshot.cn/*

✧ 可灵 AI(Kling AI)：*https://klingai.kuaishou.com/*

✧ 即梦 AI：*https://jimeng.jianying.com/*

✧ 吐司 AI(TusiArt)：*https://tusiart.com/*

✧ 文心快码-Baidu Comate：*https://comate.baidu.com/zh/*

✧ 通义灵码：*https://tongyi.aliyun.com/lingma/*

✧ 扣子：*https://www.coze.cn/*

✧ AiPPT：*https://www.aippt.cn/*

AI 通用会话与联网搜索

文心一言
得票率：19.52%

夸克
得票率：16.32%

Kimi
得票率：14.65%

AI 多媒体创作

可灵AI
得票率：16.63%

即梦AI
得票率：12.14%

吐司AI
得票率：9.84%

AI 智能体工坊

扣子
得票率：32.85%

AI 文字内容效率

AiPPT
得票率：21.43%

AI 编程

文心快码-Baidu Comate
得票率：24.95%

通义灵码
得票率：21.84%

「Gitee AI & 天数智芯创新应用大赛」获奖作品

2024 年，Gitee AI 联手天数智芯举办了备受瞩目的 AI 创新应用开发大赛。本次大赛共有近 300 支团队提交创意方案，共有超过 100 支团队经过选拔进入初赛。

最终产生了六支获得一、二、三等奖的团队及十支获得优胜奖的团队，以下是最终的获奖名单。

一等奖

AiEditor

面向 AI 的下一代富文本编辑器，AIEditor Max（基于 AIEditor 优化）10 倍提高用户的文字创作能力。提供了 AI 快捷功能、边写边问、共享智能体、智能翻译、智能配图、一键排版等能力。

体验链接：https://ai.gitee.com/apps/fuhai/aieditor-max

二等奖

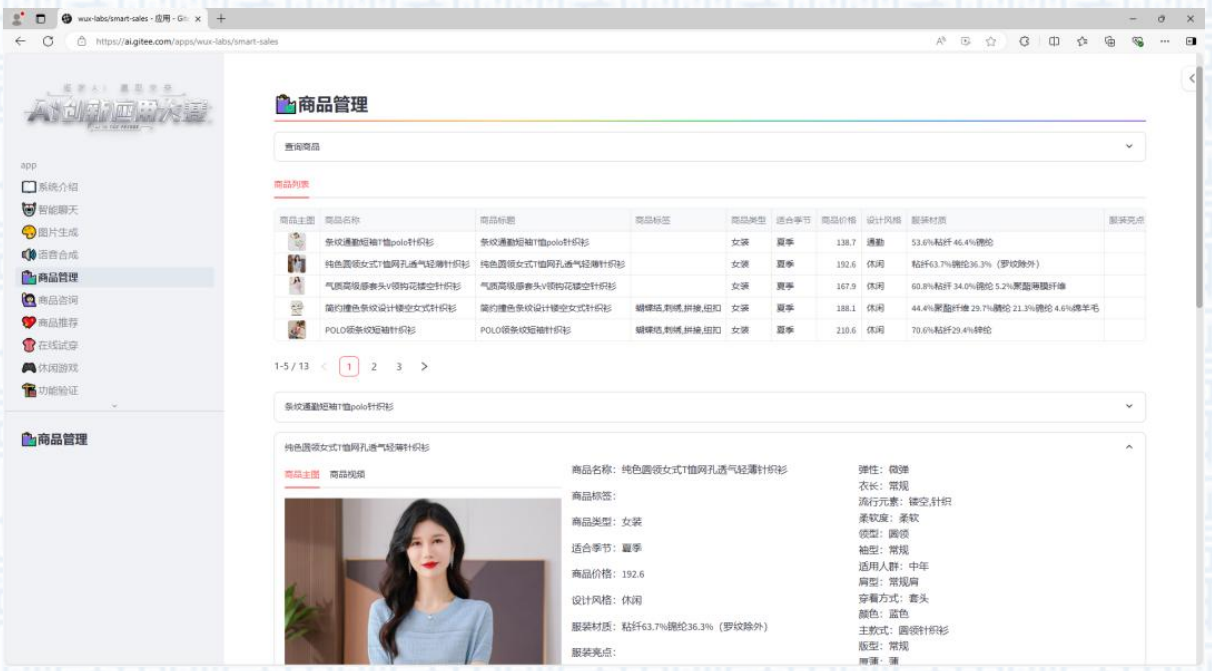
CrossProbe

CrossProbe 是基于 AIGC 技术的一站式跨文化设计调研平台，帮助用户在全球多元文化背景下进行高效且准确的设计调研和分析。

体验链接：https://ai.gitee.com/apps/gong-tao-825/CrossProbe-2024

智能营销助手

智能营销助手将营销与经营结合，它能够与用户的对话，了解用户的需求，基于多模态的 AIGC 生成能力，整合了 RAG、Agent、数字人、虚拟试穿、智能推荐等功能，能够持续输出更符合用户消费习惯的文本、图片、语音和视频等营销内容，推荐符合用户的商品。



体验链接：https://ai.gitee.com/apps/wux-labs/smart-sales

三等奖

视频课件提取 Agent 应用

该应用可以实现视频内容提取、音频转字幕以及字幕总结功能，可自动提取视频中的 PPT 图片，帮助学习者快速定位关键视觉信息；同时使用 ASR 模型将视频音频转换为字幕，增强内容的可访问性，以及利用大语言模型对字幕进行总结，生成结构化的 Markdown 文件，便于复习和资料整理。

体验链接：https://ai.gitee.com/apps/qq764073542/chat_with_video_0917

易拍即识

对于企业和机构而言，大量的纸质文件和图片资料需要人工整理，这不仅降低了工作效率，还增加了成本。鉴于此背景，一款基于大模型技术的文字识别应用——易拍即识。

体验链接：https://ai.gitee.com/events/iluvatar-ai-app-contest/detail?app=27

灵眸沙盘

灵眸沙盘是一款 AI 心理沙盘游戏应用，专注于心理健康领域的智能互动产品，旨在为用户提供个性化、趣味化的心理疏导和咨询服务，帮助用户缓解压力、调整心态，提升心理健康水平。

体验链接：https://ai.gitee.com/apps/liujunjun2023/sandbox-therapy-final

优胜奖

以下排名不分先后

- 医药行业 AI 填单及审核风控平台
- 全自动视频翻译
- UML 绘制智能助手
- 超级写手
- BlindEyes
- PentestAssist
- LezziDine
- 新时代老中医
- 3D 形象生成
- 智安助手

支持单位：北京银行 BANK OF BEIJING

感谢阅读

关注我们，开源开发者圈一网打尽

OSCHINA oschina.net

gitee gitee.com

gitee AI ai.gitee.com

2024

2024 中国开源开发者报告

China Open Source 2024 Annual Report

聚焦大模型

OSCHINA 公众号

Gitee服务号

Gitee AI公众号