电子

云端之下，苹果端侧 AI 布局加速

强于大市（维持评级）

投资要点：

- **苹果多层次布局 AI 产品，成果不断发布**

苹果在 AI 布局相对低调，但是近期随着相关成果的对外发布，表明了其对 AI 领域的高度重视。从相关布局看，苹果布局涵盖芯片硬件层、开发者框架和工具、各类针对端侧 AI 的优化策略以及最上层的大模型和入口，正在全方位多层次的进行 AI 布局。随着 6 月份 WWDC 及后续发布会的进行，相关产品有望面世。

- **硬件：ARM 和统一内存架构业内领先，Flash-LLM 突围端侧 AI 内存瓶颈，AI 任务中硬件优势进一步扩大**

在 AI 浪潮前，苹果手机和电脑芯片综合性能就业内领先，M 系列芯片更是将 ARM 架构和统一内存方案推向大众消费市场。M 系列芯片由于采用统一内存架构，无需在多个内存池之间复制数据，进而实现了笔记本端最高 128G 内存和 400GB/s 的带宽。作为对比，英伟达 RTX 4090 显存为 16GB，英特尔 Meteor Lake 最高支持 120GB/s 的带宽。随着 AI 负载的增加，苹果高内存和高带宽的优势将进一步扩大。同时我们看到统一内存架构并非苹果率先提出，苹果作为芯片设计商、操作系统开发商、PC 品牌的多重角色，凭借对生态的强力把控，以 M1 芯片为硬件基础才得到了开发者的支持，实现了 CPU、GPU 协作的高效性，硬件优势壁垒极高。23 年 12 月苹果发布 Flash-LLM 方案，结合存储的硬件特性，创新性提出利用闪存解决大模型运行的内存瓶颈，通过该方案端侧设备能运行的模型参数量达到了原来的 2 倍。同时大模型在 CPU 上的推理速度提高了 4-5 倍，GPU 上推理速度提高了 20-25 倍，合计可将 1Token 的 I/O 延迟从 2130ms 降低至 87ms。

- **软件和模型：端侧模型新思路，结合使用场景的模型路线**

除了 30B 参数的 MM1 模型外，苹果更多在端侧模型上进行布局。2024 年以来陆续发布了 ReALM、Ferret-UI 以及开源模型 OpenELM。ReALM 从智能终端与用户交互中常见的指代消解问题出发，结合 AI 新技术和特征工程为该类任务提出了全新的研究范式，8000 万参数模型性能与 GPT-4.0 相当，大幅提升端侧 AI 的可用性。Ferret-UI 更是从“如何让 AI 更好的理解屏幕”这一问题出发研发模型，初级任务显著优于 GPT-4V。同时高级任务上 GPT-4V 更好的表现也展示了云端 AI 的优势所在，混合 AI 将会是 AI 的未来。OpenELM 模型提出了“分层缩放”策略，有效分配 Transformer 模型每一层参数从而提高准确率，并降低约 50%训练数据量，从实验结果看苹果提出的架构非常有效，为后续苹果端侧 AI 模型的推出打下了坚实的基础。

- **投资建议**

苹果在 AI 领域的相关成果展现了其在端侧 AI 领域强大的技术储备和领先性，后续有望落地到硬件产品。建议关注：

苹果供应链：立讯精密、歌尔股份、鹏鼎控股、东山精密、长盈精密、国光电器、领益智造

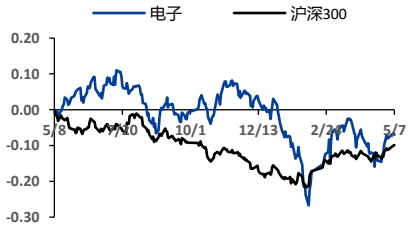
端侧 IC：恒玄科技、晶晨股份、乐鑫科技、中科蓝讯、瑞芯微、矩芯科技

其它整机品牌：传音控股、漫步者

- **风险提示**

技术发展不及预期、场景落地不及预期、市场竞争加剧

一年内行业相对大盘走势



团队成员

分析师： 任志强(S0210524030001)
rzq30466@hfzq.com.cn
联系人： 陈妙杨(S0210124040080)

相关报告

正文目录

1 苹果 AI：硬件/框架/模型全面布局 3
2 硬件端优势明显，在 AI 浪潮下进一步扩大 4
2.1 芯片 AI 相关能力持续进化 4
2.2 M 系列芯片将 ARM 和统一内存架构推向大众消费市场 4
2.3 Flash-LLM：利用闪存解决运行大模型遇到的内存瓶颈 6
3 苹果在模型端针对端侧场景进行深度布局 9
3.1 ReALM：屏幕交互新思路-端侧场景化小模型 9
3.1.1 ReALM 为端侧 AI 交互打开新思路 9
3.1.2 指代消解问题的解决将大幅提升端侧 AI 的可用性 9
3.1.3 Apple 新路线致力于解决 AI 模型在端侧的应用瓶颈 10
3.1.4 ReALM 效果突出，小参数实现更高效果 10
3.2 Ferret UI：让多模态 LLM 更好的理解屏幕 11
3.2.1 现有多模态大模型在屏幕内容理解上表现不佳 11
3.2.2 Ferret-UI 结合 UI 特点和需求改进模型 12
3.3 OpenELM：多款可在端侧运行的开源小模型 14
4 行业重点公司 15
5 风险提示 16

图表目录

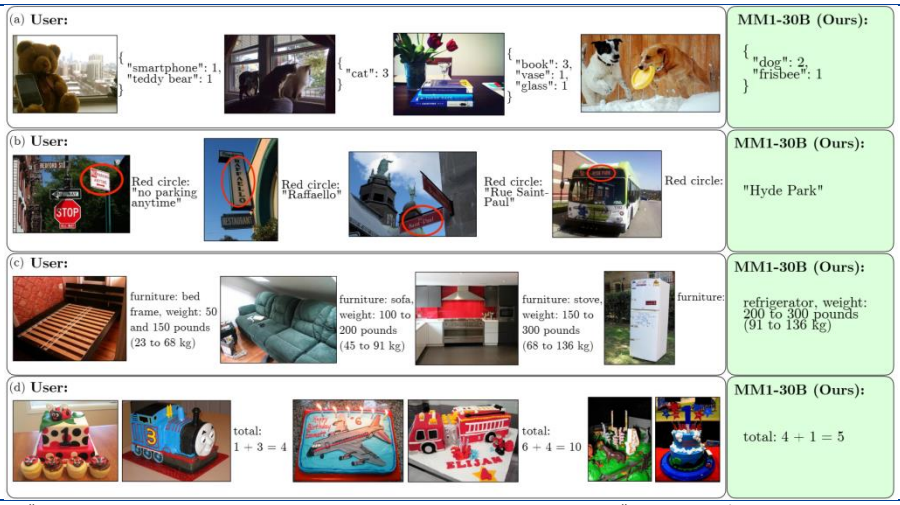
图表 1： MM1 大模型效果演示 3
图表 2： A 系列芯片 NPU 算力（Tops） 4
图表 3： M3 系列芯片信息 4
图表 4： M1 芯片 CPU 能效曲线 4
图表 5： M1 芯片 GPU 能效曲线 4
图表 6： 原有的 Mac 上芯片布局 5
图表 7： M1 芯片架构 5
图表 8： AMD 的 hUMA 架构 6
图表 9： 不同模型上 1Token 的推理延迟 6
图表 10： 苹果统一内存架构下的带宽 7
图表 11： 内存的吞吐量随块的大小和线程数增加而增加 7
图表 12： 预测变量不会改变零样本任务的准确率 7
图表 13： Sliding Window 示意图 7
图表 14： 行列捆绑 8
图表 15： 使用不同技术后的 I/O 延迟 8
图表 16： 日常场景中的模糊引用 9
图表 17： 苹果把指代对象分为三类 10
图表 18： 训练和测试数据集数量为千级 10
图表 19： 不同数据集下模型预测准确率 10
图表 20： UI 界面中的任务及其引用对象复杂 12
图表 21： Ferret-UI 训练及测试数据量 12
图表 22： Ferret-UI-anyres 技术架构 12
图表 23： Ferret-UI-anyres 表现优异 13
图表 24： 高级任务实验结果 13
图表 25： 不同模型对同一问题的回答对比 14
图表 26： 不同模型效果比较 14
图表 27： 不同评价框架下 OpenELM 表现 15
图表 28： 微调在不同参数水平上都能大幅提升模型表现 15
图表 29： 行业重点公司 15

1 苹果 AI：硬件/框架/模型全面布局

苹果在 AI 领域全方位/多层次布局。虽然苹果的 AI 布局相对低调，但其在 AI 领域的研究进展和投资表明了其对这一领域的重视和决心，并且已经目标明确得在做很具体的应用导向型研究。同时苹果的布局极具体系性，涉及 AI 生态的多个环节。从硬件的芯片，到开发者框架和工具，各类针对端侧 AI 的优化部署策略，上层的大模型，以及 AI 能力的入口 Siri 等，苹果均进行了相关的布局。

- 2023 年 12 月：1、苹果面向研究开发人员推出了专为苹果芯片设计的机器学习框架 MLX，增加了对统一内存的支持，可以同时调用内存、显存，并延续了苹果一贯的低学习成本和优秀生态及交互优势；2、发布 LLM in a flash 论文，利用闪存解决端侧模型运行时内存不足的问题。
- 2024 年 3 月：1、相关报道显示苹果收购加拿大初创公司 DarwinAI，这家初创公司以其深度神经网络技术的高效小型化而闻名。2、苹果发布 300 亿参数的多模态大模型 MM1，由于进行了大规模的多模态预训练，MM1 有不错的图像识别和推理能力，非常擅长在用户输入的图像和文本中寻找“规则”，并能够结合日常知识和数学推理能力给出答案。3、多次宣传 Macbook 强大的 AI 能力；4、发布 ReALM 模型，提出屏幕人机交互的 AI 方案。
- 2024 年 4 月：1、发布 Ferret-UI 模型，让模型更加理解屏幕；2、发布开源端侧模型 OpenELM，提出“分层缩放”策略提高准确率，降低训练需求；3、购买图片数据进行 AI 训练。
- 2024 年 5 月：1、库克宣称在 6 月 WWDC 大会上公布新的 AI 功能。

图表1：MM1 大模型效果演示



来源：《MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training》，华福证券研究所

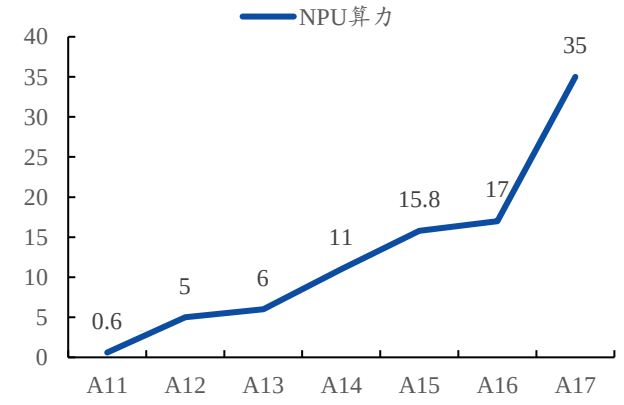
2 硬件端优势明显，在 AI 浪潮下进一步扩大

2.1 芯片 AI 相关能力持续进化

A 系列芯片 NPU 算力快速进化，A17 Pro 算力达到 35Tops。由于手机与外界交互更多，如人脸解锁，物体识别，音视频质量优化等，苹果近年来在 A 系列芯片 NPU 上发力较多。2017 年首次搭载，芯片具有 2 核 NPU，0.6Tops 算力，A12/ A13 就分别具有了 4/8 核 NPU，A14 往后均为 16 核 NPU。NPU 算力快速增长，A17 相比 A16 提升超过 100%。除了制程进步带来的算力增加，核心数量的增加是更重要的原因，反应了苹果在芯片上对 NPU 的倾斜。

M 系列芯片内存、带宽优势明显。苹果 M 系列芯片大内存、高带宽优势明显，在 AI 负载中优势将进一步扩大。M3/M3 Pro/M3 max 的内存分别为 24/36/128GB，作为对比，目前英伟达最高端的笔记本 GPU 产品 RTX 4090 的显存容量为 16GB。M3/M3 Pro/M3 max 的带宽分别为 100/200/400GB/s，我们选取英特尔 2023 年发布的 Meteor Lake 芯片作为对比，其最高带宽为 120GB/s。

图表2：A 系列芯片 NPU 算力（Tops）



来源：cpu-monkey，华福证券研究所
注：数据为 Pro 款机型搭载的 A 系列芯片数据

图表3：M3 系列芯片信息

	M3	M3 pro	M3 max
制程，晶体管数量	3nm，250亿	3nm，370亿	3nm，920亿
CPU	8核，4*高性能核心+4*高效能核心	12核，6*高性能核心+6*高效能核心	16核，12*高性能核心+4*高效能核心
GPU	10核，硬件加速光线追踪	18核，硬件加速光线追踪	40核，硬件加速光线追踪
NPU	16核神经网络引擎，18TOPS	16核神经网络引擎，18TOPS	16核神经网络引擎，18TOPS
支持内存	24GB	36GB	128GB
内存带宽	100GB/s	200GB/s	400GB/s

来源：Apple 官网，cpu-monkey，华福证券研究所
注：Pro 和 Max 选取可选最高规格

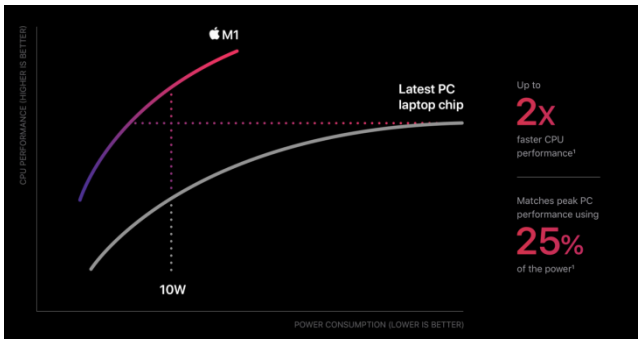
2.2 M 系列芯片将 ARM 和统一内存架构推向大众消费市场

2020 年 11 月，苹果在 Mac 上告别英特尔芯片，推出首款自研 M 系列 SoC 芯片，综合性能大幅提升。首次采用 5nm 制程封装了 160 亿晶体管，在处理器和内存架构上都大幅革新。M1 芯片将中央处理器速度提升至最高 3.5 倍，图形处理器速度提升至最高 6 倍，机器学习速度提升至最高 15 倍。相比性能提升，功耗降低更为惊人。对比当时市面最新的 PC 处理器，CPU 在同样性能表现下功耗仅为 25%，GPU 在同样性能表现下功耗仅为 1/3。反映到整机设备上，搭载 M1 的 Macbook Air 续航最长达 18 小时，比之前多出 6 小时。

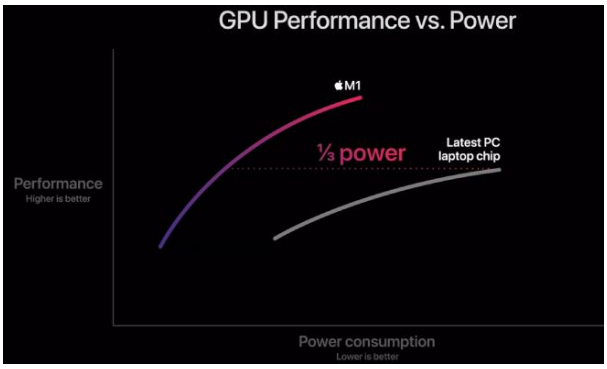
除了 ARM 架构的低功耗，高效率之外。UMA 也成为性能提升、功耗降低、节省内部空间的一大核心技术。

图表4：M1 芯片 CPU 能效曲线

图表5：M1 芯片 GPU 能效曲线



来源：Apple，华福证券研究所

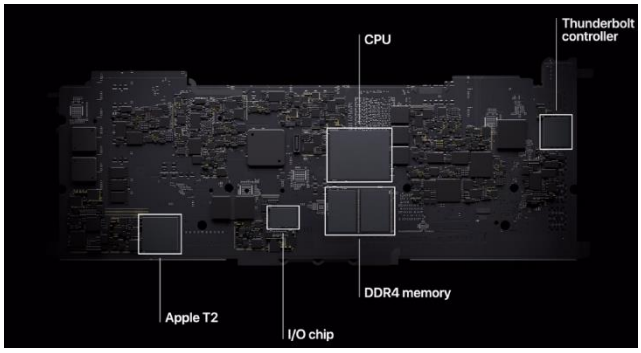


来源：Apple，华福证券研究所

M 系列芯片采用统一内存架构（Unified Memory Architecture，UMA）。不同于以往电脑端的 CPU，M 系列芯片是 CPU，还有 GPU、内存、PCIE 控制器、雷电接口控制器、神经网络引擎等组件组合的芯片，大大节省了机身内部的空间和功耗。

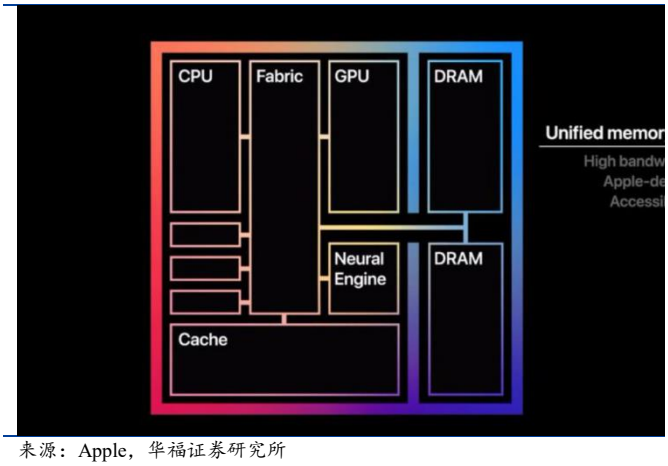
由于无需在多内存池之间复制数据就能访问数据，统一内存进一步提升了综合性能，实现高带宽、低延时。传统方案的 CPU 和 GPU 即便放在同一颗 SoC 上，但是由于 CPU 和 GPU 不同的访问习惯及数据结构，CPU 和 GPU 针对内存的存取空间是分开的，需要在内存的不同空间之间来回复制数据。M1 平台之上的 RAM 内存，面向 CPU 和 GPU 等不同的处理器时，采用统一可访问的内存池，可以在相同的内存地址访问相同的数据，因而在传输带宽、降低延时、降低功耗上表现优异。

图表6：原有的 Mac 上芯片布局



来源：Apple，华福证券研究所

图表7：M1 芯片架构

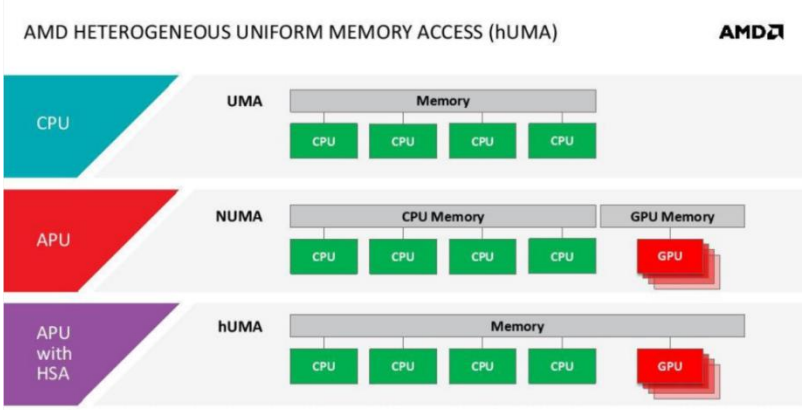


来源：Apple，华福证券研究所

苹果不是第一家尝试使用统一内存的公司，但苹果却是首家将其推向大众主流市场的厂商。AMD 在 13 年就提出了 hUMA 方案，与当前苹果的统一内存方案理论类似。但是在 PC 领域，要实现 CPU、GPU 协作的高效性，需要开发生态的配合，因此 AMD 在该领域的进展并不如人意。苹果作为芯片设计商、操作系统开发商、

PC 设备 OEM 厂商的多重角色，凭借对生态的强力把控，以 M1 为硬件基础推行自己的 UMA 架构，让 UMA 走向大众消费市场。

图表8：AMD 的 hUMA 架构



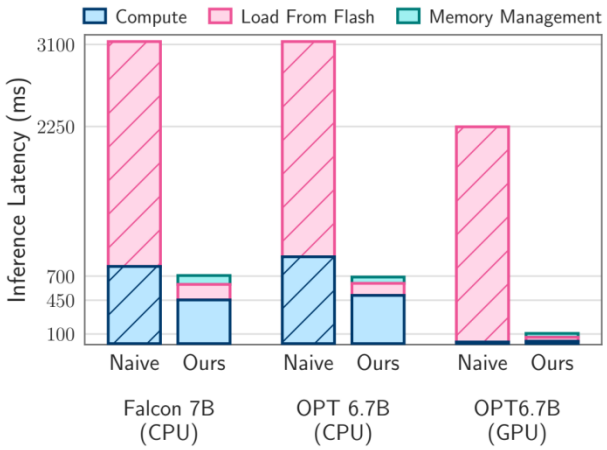
来源：AMD，COOL3C，华福证券研究所

2.3 Flash-LLM：利用闪存解决运行大模型遇到的内存瓶颈

2023 年 12 月，苹果发布论文《LLM in a flash:Efficient Large Language Model Inference with Limited Memory》，提出利用闪存解决端侧模型运行时 DRAM 不足的限制，同时针对闪存带宽不够的问题，苹果提出了两个技术方案：窗口化（windowing）和行列捆绑（row-column bundling）。

效果优异，可支持模型参数量加倍，GPU 上推理速度提升 20-25 倍。通常在推理阶段，LLM 都是直接加载到 DRAM 中的，一个 70 亿参数的模型需要超过 14GB 内存才能以半精度的方式加载参数，大大限制了边缘设备搭载 LLM。苹果通过软硬件协同优化，使设备能够支持运行的模型大小达到了原来的两倍。同时在这项技术的加持之下，LLM 的推理速度在 Apple M1 Max CPU 上提高了 4-5 倍，在 GPU 上提高了 20-25 倍。

图表9：不同模型上 1Token 的推理延迟

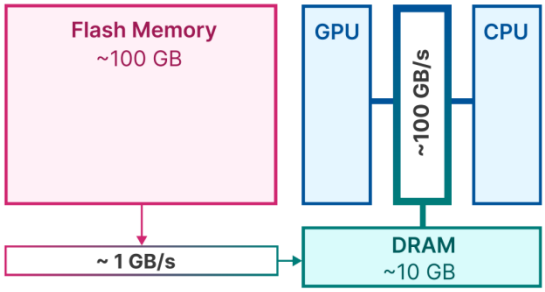


来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

从硬件出发，苹果的技术创新基于存储自身的硬件特点提出两大关键点。1、

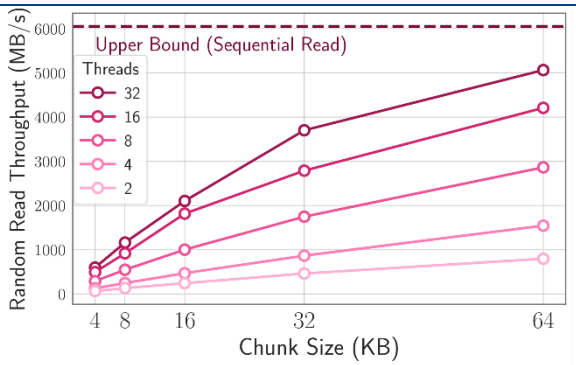
DRAM 的特点是小存储容量，传输速率高，Flash 的特点是大存储容量、传输速率低。针对这一特点，**减少需要从闪存传输的数据量成为第一个关键点。**2、读取大块、连续的数据会比读取小块、非连续的数据更高效，可以在读取量不变的情况下提升整体吞吐量，**通过优化数据块的大小提升传输速率成为第二个关键点。**

图表10：苹果统一内存架构下的带宽



来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

图表11：内存的吞吐量随块的大小和线程数增加而增加



来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

苹果提出三种方法减少数据传输量。（1）**把部分参数有选择性的常驻在 DRAM 中**，而无需进行全模型的加载，提高了计算效率和访问速度，从而提高推理性能。对于常见的 LLM 而言，它的模型参数主要由 Attention 参数和 MLP 参数两部分构成，其中 Attention 参数占比约为 1/3，MLP 参数占比约为 2/3。除此之外，还有参数量级很小的 Embedding 层的参数。因为 Attention 参数量相对较少，所以**苹果的方案是将 Attention 参数和 Embedding 层的参数直接加载到 DRAM 中。**（2）**仅迭代传输闪存中必要的、非稀疏数据到 DRAM 进行处理。**由于 MLP 层的输出只有不到 10%的值是激活状态（不为 0），而激活 MLP 层的哪些神经元与当前的输入相关。苹果利用前馈网络（FFN）模型中固有的稀疏性进行预测，每次在推理时动态加载预测为激活神经元对应的参数。（3）**使用滑动窗口技术进行神经元数据管理**，在内存中保留最近一部分输入标记的神经元数据，只对多余的参数进行删除，缺少的参数进行加载。

图表12：预测变量不会改变零样本任务的准确率

Zero-Shot Task	OPT 6.7B	with Predictor
Arc Easy	66.1	66.2
Arc Challenge	30.6	30.6
HellaSwag	50.3	49.8

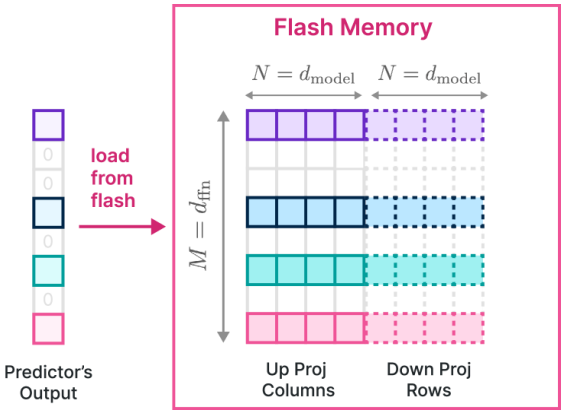
来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

图表13：Sliding Window 示意图

来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

基于神经元共同激活的行列捆绑，增加块大小提高传输吞吐量。通过对列和行，每次加载将加载 2x 块，而不是单独读取列或行，使每次读取的块大小增加 1 倍。

图表14：行列捆绑



来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所

多种技术合计可将 1Token 的 I/O 延迟从 2130ms 降低至 87ms。苹果在 M1 Max 芯片上配合 1T 的 SSD 进行实验，并且设定只有 50%的内存可用于大模型计算。相比于标准方式，通过参数常驻 DRAM 的方式将模型加载时需要从 Flash 加载到 DRAM 的数据量从 13.4GB 降低到 6.7GB，依次增加参数预测/滑动窗口/行列绑定技术，IO 延迟可降低至 738ms/164ms/87ms。

图表15：使用不同技术后的 I/O 延迟

Configuration				Performance Metrics			
Hybrid	Predictor	Windowing	Bundling	DRAM (GB)	Flash→ DRAM(GB)	Throughput (GB/s)	I/O Latency (ms)
✗	✗	✗	✗	0	13.4 GB	6.10 GB/s	2130 ms
✓	✗	✗	✗	6.7	6.7 GB	6.10 GB/s	1090 ms
✓	✓	✗	✗	4.8	0.9 GB	1.25 GB/s	738 ms
✓	✓	✓	✗	6.5	0.2 GB	1.25 GB/s	164 ms
✓	✓	✓	✓	6.5	0.2 GB	2.25 GB/s	87 ms

来源：《LLM in a flash: Efficient Large Language Model Inference with Limited Memory》，华福证券研究所
备注：实验设定只有 50%内存可用，模型为 OPT 6.7B 16 位模型

3 苹果在模型端针对端侧场景进行深度布局

3.1 ReALM：屏幕交互新思路-端侧场景化小模型

3.1.1 ReALM 为端侧 AI 交互打开新思路

ReALM 是一种可在端侧运行的实用高效实体识别系统。2024 年 3 月，苹果发布论文《ReALM:Reference Resolution As Language Modeling》，主题在于解决非对话实体（non-conversational entities）中的指代消解（Reference resolution）问题。当前 LLM 在非实体对话中的指代消解问题会严重影响端侧 AI 使用体验，ReALM 提出了一个利用大语言模型建立和解析各类指代对象（尤其是非对话实体）的系统。最小的 8000 万参数模型在屏幕实体识别的准确率上也比原有系统提升了 5%以上。在与 GPT-4.0 的对比中，8000 万参数模型与 GPT-4.0 性能相当，而更大的 ReALM 模型则明显优于它，且在屏幕领域表现更加出色。

3.1.2 指代消解问题的解决将大幅提升端侧 AI 的可用性

模糊的引用在日常场景中很常用。人类语言通常包含模棱两可的引用，例如“他们”或“那个”，其指代的含义在上下文中是显而易见的。因此使用和理解上下文的能力对于 AI 应用来说是至关重要的，否则将无法理解用户的意图以完成任务。

图表16：日常场景中的模糊引用

Speaker	对话
用户	显示我附近的药店
智能体	这是我找到的清单
智能体	...（已列出）
用户（例1）	打电话给彩虹路的那个人
用户（例2）	呼叫底部的那个
用户（例3）	拨打此号码（显示在屏幕上）

来源：《ReALM: Reference Resolution As Language Modeling》，华福证券研究所

非对话框形式下的实际场景中，指代的实体种类更多，可能不被上下文包含，指代消解问题更突出。指代消解问题可以概括为：由于无法理解“它”指的是什么，导致错误的理解和回复。相对云端 AI 对话框，实际应用场景中实体类型更多。ReALM 模型主要针对屏幕 UI 场景，苹果把该场景下的对象分为三类：多轮对话实体、屏幕实体和后台运行的实体。在这种场景下，针对非文本数据的指代消解问题更突出。

图表17：苹果把指代对象分为三类

类型	解释和举例
对话实体	多轮对话中的各类。可能来用户（例如，当用户说“呼叫妈妈”时，妈妈的联系人将是相关实体），也可能来自虚拟助理（例如，智能体向用户提供可供选择的列表）。
屏幕实体	当前显示在用户屏幕上的各类实体
背景实体	来自后台进程的相关实体，这些实体不一定是用户在屏幕上看到的内容或他们与智能体交互的直接部分;例如，闹钟开始响起或背景中播放的音乐

来源：《ReALM: Reference Resolution As Language Modeling》，华福证券研究所

3.1.3 Apple 新路线致力于解决 AI 模型在端侧的应用瓶颈

现有模型在端侧核心不足在于指代消解、硬件计算能力有限。现有的模型在端侧运行有诸多不足，其中核心问题主要为以下几点。

1、模型完全运行在端侧的时候功耗要求高；同时硬件计算能力有限，运行大模型较为困难；

2、在屏幕场景下，指代消解问题突出；无法把屏幕上的但在历史对话轮次中未提及的实体纳入对话。

3、大模型必须集成应用程序接口 API，泛化能力较弱。

针对现有大模型在端侧应用存在的不足，ReALM 通过以下创新得以大幅改进：

ReALM 创新点 1：传统的端到端特征工程需要大量的人工特征和规则，适应新领域的成本很高，同时对上下文理解弱。而现有大模型硬件要求高。**苹果提出的新范式：用 LLM 进行端到端的建模。**

ReALM 创新点 2：将历史会话、屏幕内容等不同内容的实体统一编码为文本输入给 LLM，实现同时处理对话和非对话实体。**这样的好处是 ReALM 并不直接处理图像数据，将指代消解问题变为了建模问题，并且使用单个语言模型解决多种任务。同时仅为文本数据降低了模型大小，能够在小内存的 iPhone 上本地运行，并使 Siri 拥有了视觉能力。**

3.1.4 ReALM 效果突出，小参数实现更高效果

效果优异，尤其是在屏幕实体领域。苹果基于谷歌在 2022 年提出的 Flan-T5 微调模型，使用了包括千数量级的对话、屏幕、背景实体进行训练和测试。最终效果突出。与以前的特征工程方式 MARRS 相比提升巨大，最小的 80M 模型在对屏幕实体的识别上获得了 5%的准确率提升，与大参数的 GPT-4 相比也不遑多让。在代表了泛化能力的未知数据集的准确率上，ReALM 大幅超过 MARRS 和 GPT-3.5，与 GPT-4 接近。

图表18：训练和测试数据集数量为千级

图表19：不同数据集下模型预测准确率

Dataset	Train	Test
Conversational	2.3k	1.2k
Synthetic	3.9k	1.1k
On-screen	10.1k	1.9k

来源：《ReALM: Reference Resolution As Language Modeling》，华福证券研究所

Model	Conv	Synth	Screen	Unseen
MARRS	92.1	99.4	83.5	84.5
GPT-3.5	84.1	34.2	74.1	67.5
GPT-4	97.0	58.7	90.1	98.4
ReALM-80M	96.7	99.5	88.9	99.3
ReALM-250M	97.8	99.8	90.6	97.2
ReALM-1B	97.9	99.7	91.4	94.8
ReALM-3B	97.9	99.8	93.0	97.8

来源：《ReALM: Reference Resolution As Language Modeling》，华福证券研究所

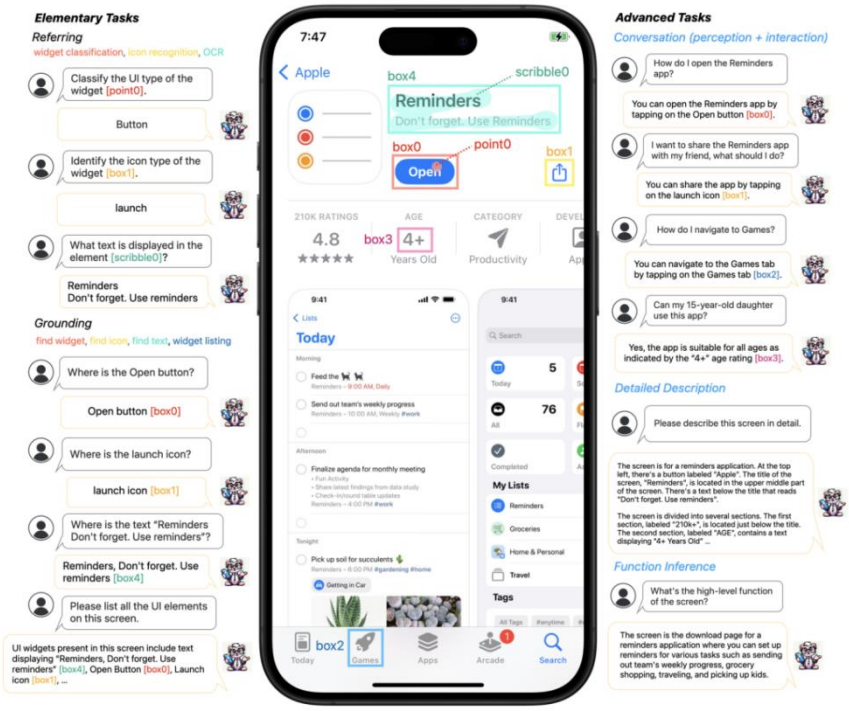
ReALM 为指代消解提出了全新的研究范式，同时有望引领端侧 AI 和云端 AI 走向不同的技术路线，端侧 AI 将不仅仅是云端 AI 在设备端的弱化版或端侧场景的特殊版，而是和云端 AI 在能力上各有所长，相互补充。

3.2 Ferret UI：让多模态 LLM 更好的理解屏幕

3.2.1 现有多模态大模型在屏幕内容理解上表现不佳

由于 UI 界面的特殊性，通用多模态大型语言模型（MLLM ）在理解和有效交互能力方面往往不足。2024 年 4 月，苹果发布论文《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，基于自身 2023 年 10 月推出的多模态大模型 Ferret 做了针对 UI 界面的改进。多模态大模型在 UI 领域表现不佳的主要原因包括：1、与现实世界的图像相比，手机屏幕的高宽比和大多数模型训练图形使用的高宽比不同；2、在与 UI 的交互中，不仅需要理解屏幕整体，还要能够集中于屏幕内的特定 UI 元素。所以 MLLM 需要识别出图表和按钮，但是他们相对来说都比较小，现有 MLLM 是把整个屏幕作为单一输入容易丢失细节。

图表20：UI 界面中的任务及其引用对象复杂



来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所

3.2.2 Ferret-UI 结合 UI 特点和需求改进模型

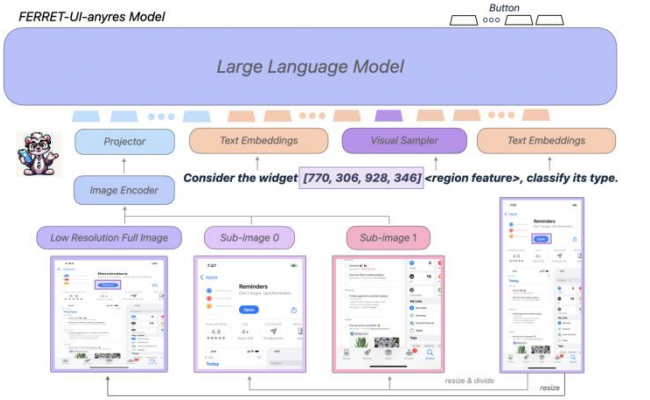
苹果从 3 方面对 UI 领域的 MLLM 进行改进。1）针对大部分模型训练数据和 UI 界面的差异，苹果在训练数据上使用 UI 界面的训练样本，该论文中的样本量仅为万级，安卓数据集包含 26527 个训练样本及 3080 个测试样本，iPhone 数据集包含 84685 个训练样本及 9410 个测试样本。2）同时为增强推理能力，苹果在基础功能之外编制了高级任务，包括详细描述、感知/交互对话和功能推理。3）加入了任意分辨率（anyres），该技术根据原始图像的高宽比获取每个子图像的高宽比，通过放大细节来解决 UI 屏幕识别的小型对象识别问题，从而提高模型对 UI 元素的理解精度。

图表21：Ferret-UI 训练及测试数据量

Platform	Resolution	Train	Test
Android	2560×1440	26,527	3,080
iPhone	1792×828	74,953	8,297
	828×1792	4,225	461
	2436×1125	5,420	635
	1125×2436	87	17

来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所

图表22：Ferret-UI-anyres 技术架构



来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所

轻量化下性能更加优秀，初级任务显著优于 GPT-4V，高级任务上 GPT-4V 表现更好。在多个基础任务和高级任务中，相比没有针对 UI 进行优化的 Ferret，Ferret-UI 提升非常显著，同时显著优于 GPT-4V。与 Spotlight 相比，Ferret-UI 在 S2W 和 WiC

两个任务中表现出了优越的性能，Ferret-UI 性能在 TaP 任务上表现稍弱，但仍然具有竞争力。Spotlight 使用了 8000 万网页截图和 2.69 亿手机截图进行预训练，而 Ferret-UI 由于数据量少，在 8 个 A100 上训练，Ferret UI 仅需要 1 天，Ferret-UI-anyres 也只需要 3 天。

图表23：Ferret-UI-anyres 表现优异

	Public Benchmark			Elementary Tasks				Advanced Tasks	
	S2W	WiC	TaP	Ref-i	Ref-A	Grd-i	Grd-A	iPhone	Android
Spotlight [30]	106.7	141.8	**88.4**	-	-	-	-	-	-
Ferret [53]	17.6	1.2	46.2	13.3	13.9	8.6	12.9	20.0	20.7
Ferret-UI-base	113.4	**142.0**	78.4	80.5	**82.4**	79.4	83.5	73.4	80.5
Ferret-UI-anyres	**115.6**	140.3	72.9	**82.4**	**82.4**	**81.4**	**83.8**	93.9	71.7
GPT-4V [1]	34.8	23.5	47.6	61.3	37.7	70.3	4.7	**114.3**	**128.2**

来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所
注 1：S2W : screen2words, WiC : widget captions, TaP: taperception
注 2："i": iPhone, "A": Android, "Ref": Referring, "Grd": Grounding

高级任务上或将是多种模型协作发展的路线。高级任务具体看，由于采用了"任意分辨率（anyres）"技术，Ferret-UI-anyres 在详细描述和功能推理上优势非常明显，远超过开源的用户界面理解 MLLM（如 Fuyu、CogAgent），相比理解真实世界的模型 Ferret 也是全面大幅领先。在与 GPT-4V 的对比上，Ferret-UI-anyres 在与设备端屏幕更相关的任务（详细描述、功能推理）上超过 GPT-4V，但是在对话互动相关的任务中 GPT-4V 优势较大。

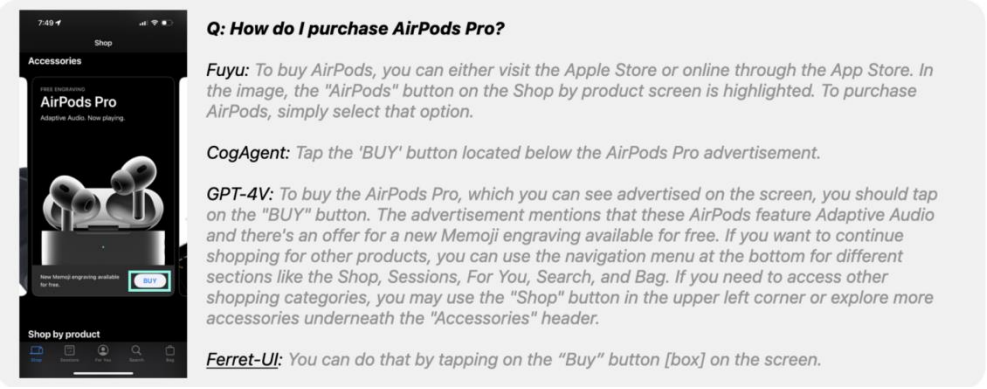
图表24：高级任务实验结果

	iPhone					Android				
	DetDes	ConvP	ConvI	FuncIn	**Avg**	DetDes	ConvP	ConvI	FuncIn	**Avg**
Ferret [53]	2.5	34.7	23.7	19.1	20.0	2.0	33.9	24.9	21.9	20.7
Fuyu [6]	5.0	24.6	18.8	35.7	21.0	2.0	20.8	44.5	36.1	25.9
CogAgent [20]	53.1	59.7	74.8	71.9	64.9	28.0	58.5	90.1	**90.5**	66.8
Ferret-UI-base	64.5	75.0	77.5	76.5	73.4	90.8	72.8	79.3	79.2	80.5
Ferret-UI-anyres	**97.4**	92.1	91.1	**95.2**	93.9	86.4	70.3	50.2	77.3	70.1
GPT-4V [1]	66.8	**105.6**	**198.5**	86.3	**114.3**	**126.6**	**109.4**	**188.6**	88.3	**128.2**

来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所
注：DetDes：详细描述、ConvP：对话感知，ConvI：对话互动，FuncIn：功能推理。

在与屏幕相关的对话交互上，GPT-4V 更倾向于提供与问题无关的信息，Ferret 更偏向简洁。高级任务具体看，为了更好的了解界面相关问题交互的准确性和相关性，苹果人工评估了 Ferret-UI 和 GPT-4V 的准确率，Ferret-UI 的准确率为 91.7%，而 GPT-4V 为 93.4%。但是 GPT-4V 往往会提供与问题无关的额外信息，在原有的评分模型中，这些详细答案比 Ferret-UI 简洁的答案得分更高。

图表25：不同模型对同一问题的回答对比



来源：《Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs》，华福证券研究所

3.3 OpenELM：多款可在端侧运行的开源小模型

2024 年 4 月，苹果发布了一款专为终端设备而设计的小模型 OpenELM，包含了 2.7 亿、4.5 亿、11 亿和 30 亿四个参数版本。**OpenELM 核心使用了“分层缩放”策略来有效分配 Transformer 模型每一层参数。模型每层的参数数量不同，从而提高准确率，并降低训练的数据需求量。**在约 10 亿参数规模下，OpenELM 与 OLMo 相比，准确率提高了 2.36%，同时需要的预训练 Token 数量减少了 50%。

图表26：不同模型效果比较

Model	Public dataset	Open-source		Model size	Pre-training tokens	Average acc. (in %)
		Code	Weights			
OPT [55]	✗	✓	✓	1.3 B	0.2 T	41.49
PyThia [5]	✓	✓	✓	1.4 B	0.3 T	41.83
MobiLlama [44]	✓	✓	✓	1.3 B	1.3 T	43.55
OLMo [17]	✓	✓	✓	1.2 B	3.0 T	43.57
OpenELM (Ours)	✓	✓	✓	1.1 B	1.5 T	**45.93**

来源：《OpenELM: An Efficient Language Model Family with Open-source Training and Inference Framework》，华福证券研究所

OpenELM 开源程度行业领先。与以往开源模型只提供权重和代码，在私有数据集上训练不同，OpenELM 基于公共数据集进行预训练和微调，并且发布了完整的框架，包括数据准备、训练、微调和评估程序，以及多个预训练的 checkpoint 和训练日志，以促进开源研究。

OpenELM 的架构在不同评价框架的实验结果中均显示有效性。由于 MobiLlama 和 OLMo 与 OpenELM 都是在类似的数据集上训练出来的，因此也是 OpenELM 的主要比较对象。与今年 2 月发布的 OLMo 开源 LLM 相比，在三种不同的评价框架下，OpenELM 较 OLMo 在 10B 左右的参数模型中准确率分别提升了 1.28%、2.36%、1.72%，并且是在 50%训练 Token 数量情况下实现的，显示了 OpenELM 架构的有效性。

模型参数数量/是否指令微调对最终效果有明显影响。参数大小对模型表现影响较大，以 zero-shot 评价框架为例，0.27B/0.45B/1.08B/3.04B 模型准确率分别为 54.37%、

57.56%，63.44%，67.39%。同时微调能将 OpenELM 的平均准确率提高 1-2%。因此我们认为未来端侧模型针对不同算力和场景将会百花齐放。

图表27：不同评价框架下 OpenELM 表现

Model	Model size	Pretraining tokens	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	SciQ	WinoGrande	Average	Average w/o SciQ
OpenELM (Ours)	0.27 B	1.5 T	26.45	45.08	53.98	46.71	69.75	84.70	53.91	54.37	49.31
MobiLlama [44]	0.50 B	1.3 T	26.62	46.04	55.72	51.06	71.11	83.60	53.20	55.34	50.63
OpenELM (Ours)	0.45 B	1.5 T	**27.56**	**48.06**	**55.78**	**53.97**	**72.31**	**87.20**	**58.01**	**57.56**	**52.62**
TinyLlama [54]	1.10 B	3.0 T	30.12	55.25	57.83	59.20	73.29	-	59.12	-	55.80
OpenLM [18]	1.00 B	1.6 T	31.00	56.00	**65.00**	61.00	74.00	-	60.00	-	57.83
MobiLlama [44]	0.80 B	1.3 T	28.84	49.62	60.03	52.45	73.18	85.90	55.96	58.00	53.35
MobiLlama [44]	1.26 B	1.3 T	31.91	56.65	60.34	62.18	74.81	89.10	59.27	62.04	57.53
OLMo [17]	1.18 B	3.0 T	31.06	**57.28**	61.74	62.92	75.14	87.00	59.98	62.16	58.02
OpenELM (Ours)	1.08 B	1.5 T	**32.34**	55.43	63.58	**64.81**	**75.57**	**90.60**	**61.72**	**63.44**	**58.91**
OpenELM (Ours)	3.04 B	1.5 T	35.58	59.89	67.40	72.44	78.24	92.70	65.51	67.39	63.18

(a) Results on zero-shot tasks with respect to the standard metrics defined in Tab. 3a.

Model	Model size	Pretraining tokens	ARC-c	HellaSwag	MMLU	TruthfulQA-mc2	WinoGrande	Average
Cerebras-GPT [14]	0.26 B	5.1 B	22.01	28.99	26.83	45.98	52.49	35.26
OPT [55]	0.35 B	0.2 T	23.55	36.73	26.02	40.83	52.64	35.95
OpenELM (Ours)	0.27 B	1.5 T	**27.65**	**47.15**	25.72	39.24	**53.83**	**38.72**
Pythia [5]	0.41 B	0.3 T	24.83	41.29	25.99	40.95	54.38	37.49
MobiLlama [44]	0.50 B	1.3 T	29.52	52.75	**26.09**	37.55	56.27	40.44
OpenELM (Ours)	0.45 B	1.5 T	**30.20**	**53.86**	26.01	40.18	**57.22**	**41.50**
MobiLlama [44]	0.80 B	1.3 T	30.63	54.17	25.2	38.41	56.35	40.95
Pythia [5]	1.40 B	0.3 T	32.68	54.96	25.56	**38.66**	57.30	41.83
MobiLlama [44]	1.26 B	1.3 T	34.64	63.27	23.87	35.19	60.77	43.55
OLMo [17]	1.18 B	3.0 T	34.47	63.81	26.16	32.94	60.46	43.57
OpenELM (Ours)	1.08 B	1.5 T	**36.69**	**65.71**	**27.05**	36.98	**63.22**	**45.93**
OpenELM (Ours)	3.04 B	1.5 T	42.24	73.28	26.76	34.98	67.25	48.90

(b) Results on OpenLLM Leaderboard tasks with respect to the metrics defined in Tab. 3b.

Model	Model size	Pretraining tokens	ARC-c	CrowS-Pairs	HellaSwag	MMLU	PIQA	RACE	TruthfulQA	WinoGrande	Average
OpenELM (Ours)	0.27 B	1.5 T	27.65	66.79	47.15	25.72	69.75	30.91	39.24	53.83	45.13
MobiLlama [44]	0.50 B	1.3 T	29.52	65.47	52.75	**26.09**	71.11	32.15	37.55	56.27	46.37
OpenELM (Ours)	0.45 B	1.5 T	**30.20**	**68.63**	**53.86**	26.01	**72.31**	**33.11**	**40.18**	**57.22**	**47.69**
MobiLlama [44]	0.80 B	1.3 T	30.63	66.25	54.17	25.2	73.18	33.68	**38.41**	56.35	47.23
MobiLlama [44]	1.26 B	1.3 T	34.64	70.24	63.27	23.87	74.81	35.02	35.19	60.77	49.73
OLMo [17]	1.18 B	3.0 T	34.47	69.95	63.81	26.16	75.14	**36.75**	32.94	60.46	49.96
OpenELM (Ours)	1.08 B	1.5T	**36.69**	**71.74**	**65.71**	**27.05**	**75.57**	36.46	36.98	**63.22**	**51.68**
OpenELM (Ours)	3.04 B	1.5 T	42.24	73.29	73.28	26.76	78.24	38.76	34.98	67.25	54.35

(c) Results on LLM360 tasks with respect to the metrics defined in Tab. 3c.

来源：《OpenELM: An Efficient Language Model Family with Open-source Training and Inference Framework》，华福证券研究所

图表28：微调在不同参数水平上都能大幅提升模型表现

Model Size	Instruction Tuned?	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	SciQ	WinoGrande	Average
0.27 B	✗	26.45	45.08	**53.98**	46.71	69.75	**84.70**	**53.91**	54.37
	✓	**30.55**	**46.68**	48.56	**52.07**	**70.78**	84.40	52.72	**55.11**
0.45 B	✗	27.56	48.06	55.78	53.97	72.31	87.20	58.01	57.56
	✓	**30.38**	**50.00**	**60.37**	**59.34**	**72.63**	**88.00**	**58.96**	**59.95**
1.08 B	✗	32.34	**55.43**	63.58	64.81	**75.57**	**90.60**	61.72	63.44
	✓	**37.97**	52.23	**70.00**	**71.20**	75.03	89.30	**62.75**	**65.50**
3.04 B	✗	35.58	59.89	67.40	72.44	78.24	**92.70**	65.51	67.39
	✓	**39.42**	**61.74**	**68.17**	**76.36**	**79.00**	92.50	**66.85**	**69.15**

(a) Results on zero-shot tasks with respect to the metrics defined in Tab. 3a.

Model Size	Instruction Tuned?	ARC-c	HellaSwag	MMLU	TruthfulQA	WinoGrande	Average
0.27 M	✗	27.65	47.15	25.72	**39.24**	**53.83**	38.72
	✓	**32.51**	**51.58**	**26.70**	38.72	53.20	**40.54**
0.45 M	✗	30.20	53.86	**26.01**	40.18	57.22	41.50
	✓	**33.53**	**59.31**	25.41	**40.48**	**58.33**	**43.41**
1.08 B	✗	36.69	65.71	**27.05**	36.98	63.22	45.93
	✓	**41.55**	**71.83**	25.65	**45.95**	**64.72**	**49.94**
3.04 B	✗	42.24	73.28	**26.76**	34.98	67.25	48.90
	✓	**47.70**	**76.87**	24.80	**38.76**	**67.96**	**51.22**

(b) Results on OpenLLM Leaderboard tasks with respect to the metrics defined in Tab. 3b.

Model Size	Instruction Tuned?	ARC-c	CrowS-Pairs	HellaSwag	MMLU	PIQA	RACE	TruthfulQA	WinoGrande	Average
0.27 M	✗	27.65	**66.79**	47.15	25.72	69.75	30.91	**39.24**	**53.83**	45.13
	✓	**32.51**	66.01	**51.58**	**26.70**	**70.78**	33.78	38.72	53.20	**46.66**
0.45 M	✗	30.20	**68.63**	53.86	**26.01**	72.31	33.11	40.18	57.22	47.69
	✓	**33.53**	67.44	**59.31**	25.41	**72.63**	**36.84**	**40.48**	**58.33**	**49.25**
1.08 B	✗	36.69	**71.74**	65.71	**27.05**	**75.57**	36.46	36.98	63.22	51.68
	✓	**41.55**	71.02	**71.83**	25.65	75.03	**39.43**	**45.95**	**64.72**	**54.40**
3.04 B	✗	42.24	**73.29**	73.28	**26.76**	78.24	38.76	34.98	67.25	54.35
	✓	**47.70**	72.33	**76.87**	24.80	**79.00**	38.47	**38.76**	**67.96**	**55.73**

(c) Results on LLM360 tasks with respect to the metrics defined in Tab. 3c.

来源：《OpenELM: An Efficient Language Model Family with Open-source Training and Inference Framework》，华福证券研究所

4 行业重点公司

苹果在 AI 领域的相关成果展现了其在端侧 AI 领域强大的技术储备和领先性，后续有望落地到硬件产品，建议关注：

苹果供应链：立讯精密、歌尔股份、鹏鼎控股、东山精密、长盈精密、国光电器、领益智造

端侧 IC：恒玄科技、晶晨股份、乐鑫科技、中科蓝讯、瑞芯微、矩芯科技

其它整机品牌：传音控股、漫步者

图表29：行业重点公司

公司代码	公司名称	当前市值	EPS(摊薄)				PE			
		(亿元)	2023A	2024E	2025E	2026E	2023A	2024E	2022A	2023E
002475.SZ	立讯精密	2,125	1.53	1.93	2.42	2.88	22.52	15.37	12.26	10.27
002241.SZ	歌尔股份	551	0.32	0.62	0.85	1.02	66.05	25.97	19.08	15.75
002938.SZ	鹏鼎控股	626	1.42	1.66	1.92	2.12	15.76	16.22	14.08	12.72
002384.SZ	东山精密	267	1.15	1.37	1.72	2.05	15.82	11.43	9.10	7.64
300115.SZ	长盈精密	128	0.07	0.57	0.71	0.86	186.10	18.71	14.94	12.27
002045.SZ	国光电器	72	0.63	0.56	0.68	0.88	26.22	22.60	18.61	14.34
002600.SZ	领益智造	350	0.29	0.33	0.45	0.55	23.10	14.93	11.14	9.09
688608.SH	恒玄科技	151	1.03	2.38	3.66	4.83	149.79	52.83	34.40	26.02
688099.SH	晶晨股份	235	1.20	1.77	2.56	3.29	52.36	31.78	21.94	17.06
688018.SH	乐鑫科技	86	1.69	2.34	3.31	4.52	61.07	45.68	32.37	23.67
688332.SH	中科蓝讯	67	2.10	2.82	3.81	4.73	35.96	19.84	14.69	11.84
603893.SH	瑞芯微	238	0.32	0.84	1.36	2.02	196.53	67.58	41.99	28.24
688049.SH	炬芯科技	32	0.53	0.73	0.97	1.25	71.39	36.29	27.26	21.20

688036.SH	传音控股	1,162	6.87	8.08	9.57	11.23	20.16	17.84	15.06	12.83
002351.SZ	漫步者	113	0.47	0.61	0.72	0.86	37.44	20.97	17.75	14.82

数据来源：Wind，华福证券研究所
注：
1、市值数据更新至 2024/5/8；
2、盈利预测数据来源于 Wind 一致预测

5 风险提示

技术发展不及预期：模型在设备端搭载需要软硬件技术的协同发展，若技术发展不及预期，相关产品推出时间也将延后。

场景落地不及预期：相比云端大模型，端侧更加重视和用户及场景结合，端侧模型在具体场景中的表现将影响整体产品力。

市场竞争加剧：智能手机市场若竞争加剧，相关供应链公司盈利水平或将承压。

分析师声明

本人具有中国证券业协会授予的证券投资咨询执业资格并注册为证券分析师，以勤勉的职业态度，独立、客观地出具本报告。本报告清晰准确地反映了本人的研究观点。本人不曾因，不因，也将不会因本报告中的具体推荐意见或观点而直接或间接收到任何形式的补偿。

一般声明

华福证券有限责任公司（以下简称“本公司”）具有中国证监会许可的证券投资咨询业务资格。本报告仅供本公司的客户使用。本公司不会因接收人收到本报告而视其为客户。在任何情况下，本公司不对任何人因使用本报告中的任何内容所引致的任何损失负任何责任。

本报告的信息均来源于本公司认为可信的公开资料，该等公开资料的准确性及完整性由其发布者负责，本公司及其研究人员对该等信息不作任何保证。本报告中的资料、意见及预测仅反映本公司于发布本报告当日的判断，之后可能会随情况的变化而调整。在不同时期，本公司可发出与本报告所载资料、意见及推测不一致的报告。本公司不保证本报告所含信息及资料保持在最新状态，对本报告所含信息可在不发出通知的情形下做出修改，投资者应当自行关注相应的更新或修改。

在任何情况下，本报告所载的信息或所做出的任何建议、意见及推测并不构成所述证券买卖的出价或询价，也不构成对所述金融产品、产品发行或管理人作出任何形式的保证。在任何情况下，本公司仅承诺以勤勉的职业态度，独立、客观地出具本报告以供投资者参考，但不就本报告中的任何内容对任何投资做出任何形式的承诺或担保。投资者应自行决策，自担投资风险。

特别声明

投资者应注意，在法律许可的情况下，本公司及其本公司的关联机构可能会持有本报告中涉及的公司所发行的证券并进行交易，也可能为这些公司正在提供或争取提供投资银行、财务顾问和金融产品等各种金融服务。投资者请勿将本报告视为投资或其他决定的唯一参考依据。

投资评级声明

类别	评级	评级说明
公司评级	买入	未来 6 个月内，个股相对市场基准指数指数涨幅在 20%以上
	持有	未来 6 个月内，个股相对市场基准指数指数涨幅介于 10%与 20%之间
	中性	未来 6 个月内，个股相对市场基准指数涨幅介于-10%与 10%之间
	回避	未来 6 个月内，个股相对市场基准指数涨幅介于-20%与-10%之间
	卖出	未来 6 个月内，个股相对市场基准指数涨幅在-20%以下
行业评级	强于大市	未来 6 个月内，行业整体回报高于市场基准指数 5%以上
	跟随大市	未来 6 个月内，行业整体回报介于市场基准指数-5%与 5%之间
	弱于大市	未来 6 个月内，行业整体回报低于市场基准指数-5%以下

备注：评级标准为报告发布日后的 6~12 个月内公司股价（或行业指数）相对同期基准指数的相对市场表现。其中 A 股市场以沪深 300 指数为基准；香港市场以恒生指数为基准，美股市场以标普 500 指数或纳斯达克综合指数为基准（另有说明的除外）

联系方式

华福证券研究所 上海
公司地址：上海市浦东新区浦明路 1436 号陆家嘴滨江中心 MT 座 20 层
邮编：200120
邮箱：hfyjs@hfzq.com.cn