硅谷坐标 x Fireworks 联创Benny Chen:开源模型、token增速、推理优化和模型定制 – 图文解说

> 一张关键画面对应一段完整解说，按原视频时间推进。

文章摘要

本期围绕《硅谷坐标 x Fireworks 联创Benny Chen:开源模型、token增速、**推理**优化和模型定制》展开，按时间顺序讨论开源模型、垂直 **Agent** 与下一波协作形态、开源追赶与蒸馏的可持续性、ARR 话语权与垂直模型的机会、Token 增长与 coding / co-worker 需求。**以下内容保留完整问答、例子、数字、画面信息和限定条件，适合先读这段摘要建立框架，再结合每个关键帧逐段阅读。**

01. 开源模型、垂直 Agent 与下一波协作形态（00:02–01:54）

场景 1

本节主旨 开源模型、垂直 Agent 与下一波协作形态

完整内容

Benny Chen（陈宇飞）： 把一个 vertical-specific agent 做好，和解决**黎曼猜想**，可能是完全两种工作。但是 Frontier Lab 现在可能更在意能不能解决黎曼猜想。我们看到越来越多 vertical SaaS agent，它们的医疗（原声词形可能为“衣藻”，这里按上下文理解）做得越来越细，洗数据的流程也越来越细。这类工作对开源模型的 customization 来说很有意义。长期来说，靠**蒸馏**并不是特别可靠的路径。

曹卿云： 它可能短期内确实可以帮助大家达到一个比较好的水平。有没有哪个方向现在的 token 消耗量没有那么多，但你们看到它的 token 增长其实非常快?

Benny Chen（陈宇飞）： Co-worker 肯定是一个大的方向。大家现在绝大部分时间都在照顾理科生，但这其实是一个非常反直觉的东西：从现在看到的例子来看，照顾文科生的需求会越来越大，比如各种 **slide**、**Word**，以及最近兴起的 **video generation**。很多 engineer 还担心自己会失业，但我觉得能做 **eval** 的人真的少之又少。大家如果有兴趣，可以多转一下“怎么做 eval”，教你老板到底能不能采购开源还是闭源；如果把 eval 做好，把开源切到闭源，可能就把你工资的**三五倍**省出来了。

曹卿云： 哈喽，Benny，欢迎来到《硅谷坐标》。**今天我们先聊你对整个大局势的判断，再聊 Fireworks 的两门生意：一个是推理优化，一个是定制模型，同时也聊 Fireworks 的核心竞争力。**先看最大的一个问题：**开源模型 versus 闭源模型**。过去一年最重要的产业变量，就是开源模型以非常快的速度逼近闭源模型的能力，而价格只有闭源模型的**几十分之一**。

画面说明

开头先出现带地图元素的《硅谷坐标》片头，随后切入访谈现场；第一个选中的画面保留了 Benny Chen 的姓名、Fireworks AI 联合创始人、Meta 经历与 UCLA 等下三分之一字幕。后续画面是演播室双人机位，没有额外技术图表。

02. 开源追赶与蒸馏的可持续性（01:55–06:52）

场景 2

本节主旨 开源追赶与蒸馏的可持续性

完整内容

曹卿云： 展望未来，你觉得开源模型和闭源模型之间的版图和格局会是什么样？

Benny Chen（陈宇飞）： 我们团队很多人都是做开源软件出身的，绝大部分人是 **PyTorch** 以及使用 PyTorch 的人，之前在 **Meta**，后来出来创业。我们非常相信，在非常有价值的软件这一层，开源的东西最终都会 **catch up**。不管是 **operating system**，最后开源也 catch up 了，还是各种各样的 **database**，最后开源也 catch up 了。**所以开源能不能 catch up 这件事，我们还是很相信；只是今年的发展比我想象中快。**首先，闭源模型的 **ARR** 涨得非常快；其次，开源模型的 **capability** 也追得非常快，比如我们最近看到的 K3、DSP4 Flash，还有最新的 **MUSE**。Meta 也说要开源 Open Source Spark。开源这边竞争如此激烈、模型的性价比又这么高，这让我非常意外。我们的 team 坚信它最终会 catch up，只是没想到 catch up 得这么快。

曹卿云： 我们知道一部分开源模型是通过**蒸馏**保持领先。靠蒸馏保持领先的开源模型可持续吗？闭源模型有什么办法避免这件事？

Benny Chen（陈宇飞）： 闭源模型那几个 **API** 基本都会修好。之前确实有各种问题：一个大模型出来几个结果之后，重新喂给一个小模型，小模型再处理一下，就可以把它的 **thinking token** 弄出来；这说明闭源模型厂商之前在 **security** 上有 gap。但这个东西修起来很快，没有理由说闭源模型厂商做不好。所以长期来看，蒸馏不是特别可靠的路径，短期内却能帮助大家达到一个比较好的水平。

以我的了解，美国很多厂商，比如 Meta，可能真的没有那么多；之前（另一家厂商的名称 ASR 不清）可能也不是很多。所以开源模型的 success 是否 contingent on 蒸馏？我觉得不一定。只要开源相关的**算力/数据量足够大且清晰**，并且数据价格慢慢下来，不一定需要蒸馏，很多人也能用到非常好的开源模型。开源厂商有很多事情可以做来保持竞争优势，闭源厂商也有很多手段；但开源厂商不一定需要永远做到比闭源有优势。

我也是买苹果手机的人。以前用安卓时，我觉得 **Google** 那些手机好用多了，但还是有各种小问题；如果价格没有那么敏感，我可能还是买苹果。最后利润大头可能还是被一直领先的人拿走，但这和开源做得很大完全不冲突：run rate 最后可能大头在开源，闭源可能永远 claim 一些优势，价格不那么敏感的人就会选闭源。

所以我觉得它长期的优势可能更在于**用户心智**。企业采购时我们以前开玩笑说，**“no one gets fired for buying IBM”**。IBM 是否一直能维持这个优势我不知道，但很长一段时间大家偏保守、都想买 IBM。这个变化可能不会那么快；如果人们相信继续采购 cloud 或 open 能找到更好的东西，变化还会慢一点，这可能是闭源模型接下来**两三年**的优势。至于**五年、十年**之后，我确实看不清。

曹卿云： 所以，如果靠蒸馏保持领先，长期不能靠这个方式缩小和闭源的差距；这件事最终会通过内部团队加强 security 来避免。但像 Meta，或者像其他非蒸馏路线的模型，它们会追得很快吗？gap 会保持吗？

Benny Chen（陈宇飞）： Meta 的 gap 相对没那么大了：它能继续 **scale up**，有足够的 talent 和 GPU，所以 gap 不会特别大。其他模型厂商（部分名称 ASR 不清），包括 reflection 或 **Thinking Machine**，也有各种东西可以 catch up。我没理由相信 gap 会一直存在，我觉得 gap 会越来越小。

就像刚才说苹果和安卓有 gap，可能是各种小 gap，但你真给我一个安卓，我不能用吗？我觉得也可以用，没有本质差别。大家最后还是看效果，最后效果好不好可能远大于价格高低。效果方面，**frontier 大模型**做得更好，因为模型毕竟更大、做得更早、更细，所以很多事情确实需要开源模型花更多精力去做。

画面说明

以 Benny 的单人中近景为主，画面下方持续显示节目字幕；中间短暂切换到主持人提问和双人远景。没有出现实际的模型架构或价格图表，画面主要用于记录对谈主体。

03. ARR 话语权与垂直模型的机会（06:53–10:23）

场景 3

本节主旨 ARR 话语权与垂直模型的机会

完整内容

Benny Chen（陈宇飞）： 长期来看，**OpenAI** 和 **Anthropic** 这些闭源 SOTA 模型的 **ARR** 肯定会继续涨。但有一个东西我不是很确定：ARR 的 accounting，以及和下游的分成怎么处理。可能很多 revenue 是百分之百算进去的，但其实有一部分要分给云厂商；如果分成比例越来越大，ARR 可能往上涨，但它的 gap revenue 不一定会一直涨。它上市之后大家会怎么分析这个故事，我也不清楚。**未来分成比例可能意味着 SOTA 模型拿到更小的分成，话语权会弱一点；我们平时看 AI，确实没有特别看他们的分成比例，上市后 visibility 可能会好一点。**

还有一个问题：**Harvey** 是你们非常重要的标杆客户。未来像 Harvey 这样的**垂直应用模型**会跑出来，还是 SOTA 模型做法律模型会更好？你站哪一边？

曹卿云： 到底是像 Harvey 这样的垂直应用模型会跑出来，还是 SOTA 模型的法律模型会更好?

Benny Chen（陈宇飞）： 我们当然站 **vertical-specific** 这一边。把一个 vertical 做好需要很大精力。我们是 infrastructure provider，大家经常开玩笑说，明年大家可能一起完蛋，AGI 直接把我们都替代掉；但这只是玩笑。把一个 vertical 做好，可能需要大量测试，要知道用户需求，以及这些需求怎么表达在产品上，这是一大部分工作。

我们很多 customer 也是 vertical-specific agent。比如 **Harvey** 是很出名的 vertical-specific law agent。各种 vertical 都要通过 **evaluation** 表达出来，用来评判工作好坏，这本身就是非常大的工作量。还有 **Doximity**，他们在平台上训练了一个 medical agent，等于在做 ChatGPT 的竞品；它更偏 **deep research**，确保医生看到正确的 reference，给病人下判断时不会出现 reference 有问题。北美以外我们最大的合作客户叫 **Heidi**，是北美以外最大的 medical scribe / medical search company 之一，他们也在平台上做了很多 fine-tuning 来取得更好的效果。

每个 vertical 的 **evaluation criteria** 确实不一样。我们希望 **a thousand flowers bloom**，希望各行业的 vertical SaaS agent 都成功，因为每个 vertical 都有很多特别细的工作。用一个特别大的通用模型把这些细致工作都做好很困难；coding、legal、medical 每个 vertical 都有各种细节，这就是这些 SaaS company 的 alpha。我们认为一个 **frontier lab** 如果把所有这些东西都做了，成本会特别高，ROI 可能特别低。

画面说明

画面在主持人提问、Benny 回答之间切换，部分候选帧保留姓名下三分之一字幕；没有展示 Harvey、Doximity 或 Heidi 的界面，相关信息均来自对谈文字。

04. Token 增长与 coding / co-worker 需求（10:23–13:53）

场景 4

本节主旨 Token 增长与 coding / co-worker 需求

完整内容

曹卿云： 今年有这么多优秀、质量很好的开源模型，你们看到的 token 增长量是什么样?

Benny Chen（陈宇飞）： 我们的量增长得还不错。最近融资时有公开消息，我记得好像是每天**四五十个 T token**；这个数字应该比 Gemini 和 OpenAI 公开的数字都大。可以认为，我们平台上走的 **open-source model** 量已经超过 Gemini 和 OpenAI 的 **B2B API** 量，这对开源模型来说是非常好的事情。

头部应用确实是 **coding**，其次是 medical、co-worker。很多工作在我们的观察中被重新 frame 成 coding 问题。假如某个行业有自己的 **vertical-specific SaaS application**，这个工具解决该 vertical 的特定问题，那么这些工具可以被转换成 **RL 的靶子**，模型就能训练成熟悉地使用这些工具

来解决 vertical 问题。

原先我们确实没想到，**PowerPoint 和 Excel 也是 coding 问题**。只要把怎么用 PowerPoint、怎么用 Excel 转换成 coding 问题，已有的 coding 工具就可以立刻用上，也可以立刻 tune 出比较好的 vertical-specific application。

所以绝大部分问题都被重新定义成 coding；如果不是 coding，可能就是 deep research。Deep research 和 coding 可能是两个很大的类别，很多小类别会尝试判断自己到底属于 deep research、coding，还是两者兼有，再把 vertical-specific SaaS agent 和 tune 的 model 编排到这个分类里。

曹卿云： 有没有哪个方向现在 token 消耗量不多，但 token 增长很快?

Benny Chen（陈宇飞）： **Co-worker** 肯定是大方向。大家现在绝大部分时间在照顾理科生，但这是反直觉的；我反而觉得照顾文科生需求会越来越大，比如 slide、Word，以及 video generation。绝大部分工作最后还是服务于人。很多人喜欢把数学榜刷得很高，但我不确定这个意义在哪里；之后服务文科生的 co-worker，我们会做很多，Fireworks 也做了 **Fireworks Nexus**，希望服务好各个公司的 co-worker。

一年半以前 **computer-use agent** 出来时，我觉得惊为天人：模型不需要各种乱七八糟的 API 调用，直接用电脑，电脑里的活都能干。当时这个词叫 **unhobbling**。但一年半之后我反过来觉得，Computer Use **Agent** 可能是一个偏小的 vertical，绝大部分工作从文字那边走。

很多成功的 Chinese open-source model 是纯文本的；**MUSE** 确实是 multimodal，但中文开源模型也能做到很高的 ARR，而它其实是纯文本模型。很多工作回到纯文本，是因为文本模型便宜，大家为了 **cut cost** 把工作流往文本靠。随着 VM 的价格越来越便宜、VM 做得越来越好，我相信 computer-use agent 仍然很有前途，只是如何把这条路径优化好还在探索。

画面说明

画面以 Benny 单人访谈为主，间或切到双人远景和主持人提问；候选字幕出现“token 趋势”“coding”等节目包装字样，但没有真实 token 仪表盘或软件界面。

05. 北美开源模型支出：Token 与 Revenue 的差别（13:53–15:25）

场景 5

本节主旨 北美开源模型支出：Token 与 Revenue 的差别

完整内容

曹卿云： 能否帮我们估算整个北美客户在开源模型上的支出?

Benny Chen（陈宇飞）： 这个有点难，你问 analyst 可能比问我好。但 **token 比例**可能容易一点。现在公开的 router 榜会影响判断，因为榜单很大一部分受 **free traffic** 影响；某些公司会做 incentive，让大家免费用模型，在 OpenRouter 上就会影响看出来的比例。

我经常和公司沟通：我们确实做得还可以，去年大概从 **100 million 做到 1 billion**，但有些公司的增长比我们快得多，可能已经到 **70 billion**（具体单位是原声说法，未展开）。**所以看 token 消耗量没有那么 honest。**因为某个模型的消耗量和另一个模型的消耗量，即使 token 数一样，也不能说明价

值一样。

我觉得看 **revenue** 最真实。大家的付费意愿还是往最好的方向移动，大家可能没有那么 price-sensitive，就会永远付费给最好的 model。我们和 customer 合作做 **Specialized Intelligence**，绝大部分时候看的不一定是性价比，而是绝对的最好：这个 task 能不能超过 **Opus**、超过（原声中另一个模型名听写为 Fabo/不确定），这是最重要的工作。比如我们和 **DeepForce** 合作，也是看最后效果能不能超过 Opus，以 Frontier Model 为 benchmark、以客户 revenue 为 benchmark，这样比较 honest。

画面说明

画面以主持人提问后切到 Benny 回答，约在 14 分钟处出现带 Benny 身份信息的下三分之一字幕；视频没有展示 OpenRouter 榜单或支出图表。

06. 未来六个月的开源流量与模型偏好（15:26–17:47）

场景 6

本节主旨 未来六个月的开源流量与模型偏好

完整内容

曹卿云： 光看 token 消耗量有一点 deceptive。你觉得未来六个月，整个北美客户在开源模型上的支出或 token 消耗量会有什么趋势?

Benny Chen（陈宇飞）： 我们坚信开源消耗量会继续上涨。我对 Fireworks 很有信心，肯定能把事情做好；与此同时，大家对于最好模型的付费意愿可能不会有太多改变。**会有越来越多人想做最好的模型，所以我们会越来越努力，和更多人把 open-source customized model 的效果做得非常好，进而 drive 更多付费意愿。**

具体比例更取决于有多少人有 conviction 去 customize open-source model。很多人担心 customize 一个 model 成本高、成功率不高；但如果越来越多人在 quality 上和 OpenAI、（原声中另一个模型名听写不清）竞争，开源模型的 traffic 就会越来越大；反之，闭源模型的 traffic 或 ARR 会越来越大。

曹卿云： 北美客户比较不同的开源模型时，更喜欢哪些?

Benny Chen（陈宇飞）： 国内几家差不多：**Kimi**、**GLM**、**MiniMax**、**DeepSeek**，它们轮流上来，哪个刚 launch，哪个 traffic 就可能大一点。前面忘了说 **Qwen（千问）**，它其实也很大，所以国内可能是五家：Kimi、GLM、Qwen、MiniMax、DeepSeek。国外可能有 **Nimbletron**、**Gemma**，以前是 Llama，现在是 MUSE（其中 Nimbletron 的词形按 ASR 保留，原声不够清楚）。

很难排名，因为谁更新、谁刚 launch，谁的 traffic 就会上去；新的 launch 后，旧 model 的 traffic 可能就下来。比如今天，也就是 **August 2026**，肯定是 Kimi 的 traffic 最大。

曹卿云： 从你们角度看，coding 现在的渗透率，特别是开源模型的渗透率怎么样?

Benny Chen（陈宇飞）： 最近还是偏低。Twitter 上大家会开玩笑说“open code 用 DS4 永远用不完”之类，但我们的观察是，**Claude 的 ARR 或 Codex 的 ARR** 还是远远高过开源模型 ARR 的。Fireworks 最近推 Fireworks Nexus，也希望把开源模型带到更多公司；这个工作很多时候本质是 **go-to-market**，不是 technical 工作。

画面说明

主持人提问与 Benny 回答交替出现，场景中有“双人访谈”“模型上线”等节目字幕包装；没有展示模型排名或流量数据的实际图表。

07. Coding 的 TAM、降价与被保留的 Fine-tune（17:48–19:57）

场景 7

本节主旨 Coding 的 TAM、降价与被保留的 Fine-tune

完整内容

Benny Chen（陈宇飞）： 我们观察到 Claude 或 Codex 的 ARR 还是远远高过开源模型 ARR。Fireworks 最近推 **Fireworks Nexus**，希望把开源模型带到更多公司里；这件事很多时候本质是 **go-to-market**，不是 technical 工作。

曹卿云： 你怎么看 coding 市场的 **total addressable market（TAM）**？它的上限在哪里？我们现在还处在比较早期的阶段吗？

Benny Chen（陈宇飞）： 讲 TAM 可能要分 **钱**和 **token 消耗量**。Token 数字肯定会一直上涨，这一点我非常坚信。但大家说 TAM 时，讲的往往是 revenue。假如每天使用的模型都在降价，比如某些模型降价，就会 drive 更多 consumption；但 revenue 往哪个方向走，我不确定。以前可能最值钱的公司是 **General Electric**，但大家都用上了电，最后最挣钱的公司也不一定是 General Electric，这很难判断。

曹卿云： 每次一个比较强大的语言模型上线，平台上客户自己做的 **fine-tune 模型**有没有被洗出去？有的留下、有的洗出，两类 customer 最大区别是什么？

Benny Chen（陈宇飞）： 去年确实洗出去很多，今年我觉得洗出去的很少，这是一个非常 **positive signal**。如果这些模型一直被洗出去，我们也不会增长这么快。现在我们发现，很多 vertical 前进的方向可能和 Frontier Lab 不一样：一个 vertical-specific agent 做好，和解决黎曼猜想，可能是完全两种工作；Frontier Lab 可能更在意能不能解决黎曼猜想、能不能制造新的药，而 vertical SaaS agent 把医疗等领域的 eval 和洗数据流程做得越来越细，这对开源模型 customization 很有意义。

画面说明

画面在双人远景、主持人提问和 Benny 单人近景之间切换；候选帧出现“coding 当前渗透率”“新模型上线”等节目字幕，没有实际 TAM 曲线或 fine-tune 面板。

08. 垂直 SaaS 的 Jagged Intelligence（19:58–22:25）

场景 8

本节主旨 垂直 SaaS 的 Jagged Intelligence

完整内容

曹卿云： 如果像 **Harvey** 或 **EvenUp** 这样的公司，今天让两家闭源模型厂商把资源全部投入法律这个 vertical，有没有可能打败 Harvey 和 EvenUp?

Benny Chen（陈宇飞）： 肯定可以，但这样支撑不了它们一个 **trillion-dollar 估值**，对它们来说不是聪明行为。我坚信这些 vertical SaaS agent 的 **evaluation** 做得很细，数据洗得很好，能够确保沿着它们认为正确的路径前进。

我们经常开玩笑说这叫 **jagged intelligence**：它们在某个方向上做得很好，但它们 tune 出来的 model 能不能解决黎曼猜想？我觉得绝对不可能。对于 AGI 公司来说，如果做出来的是 jagged intelligence，它可能只能服务这个客户，不能让一个模型服务很多很多人。

它们需要一个合理的分布。我不能天天纠结是不是和 EvenUp、Harvey 竞争，而是要把盘子里的所有 customer 都照顾好。不能只做好川菜，或者只做好粤菜；很多师傅只做好一道菜，才敢收很多钱，但 **AGI 公司**要做的是把“麦当劳”做好，把所有人的口味调好。最多是进入中国时照顾中国人的口味，进入美国时照顾美国人的口味；不可能把加州和 **Kentucky** 的口味都改掉，否则就和 **In-N-Out** 没有本质差别了。这个例子可能走得有点远，但 In-N-Out 每天和 McDonald 竞争，同时把加州和 Nevada 照顾好、把牛肉做好。

这个东西和 language model 不一定完全 comparable，但把所有人的口味照顾好，和把一个 vertical 的口味照顾好，可能是两个不同的工作。**如果真有 AGI 公司能把所有人的口味都照顾得很好，我相信它肯定能挣很多钱，只是现在还没有观察到这一点。**

曹卿云： 企业考虑闭源还是开源模型时，会权衡哪些因素?

Benny Chen（陈宇飞）： 很大的一个点是**信任**。**Fireworks 既做偏 coding-plan 的 Fireworks Max co-worker，也做 customized model，所以很大一部分是信任问题。**

画面说明

主持人和嘉宾的问答镜头中，主持人提问时保留节目下三分之一字幕；选中画面以 Benny 单人机位为主，没有显示 Harvey、EvenUp、In-N-Out 或模型输出界面。

09. 企业信任、Go-to-market 与云端部署（22:25–24:42）

场景 9

本节主旨 企业信任、Go-to-market 与云端部署

完整内容

Benny Chen（陈宇飞）： 企业信任的是这个 **ecosystem** 能不能继续 support 接下来一两年要做的事情，因为合同要签一两年。他们和 Fireworks 签合同，不会只在意我们是不是二道贩子，而会在意我们平台上 serve 的 model 接下来一两年能不能 fulfill requirement。

信任建立是长期过程。我们要和更多 customer 沟通 success case，沟通哪里做得好、哪里做得不好、开源的好处和不好，并且非常 transparent 地聊这些事情。**闭源模型的信任可能建立得更好，因为它们每天可以在 Twitter 上发“我们又 hack 了某个公司的 system”或“我们又把黎曼猜想解出来了”。**这是它们更擅长的地方。

所以最大的差别其实是 go-to-market capability：开源模型以及开源模型公司的 go-to-market capability 远弱于闭源模型，大家的 mindshare 大部分还在闭源模型上。

曹卿云： 企业会考虑本地部署还是云端部署？以前看到很多本地部署，现在是不是越来越多云端部署？

Benny Chen（陈宇飞）： 这也是信任问题：能不能做好 **security**，能不能在各种 **VPC** 上把这一轮做好。但云端部署最好的一点是 **multi-tenant**。很多企业未必能 saturate 自己想租的那么多 GPU；如果我们聚合足够的 demand，saturate 这些 workload 就很容易。如果这个东西在企业运行成本中的占比越来越大，multi-tenancy 是更好的行为，大家会希望在 cloud 上做 multi-tenant setup，和别人一起 share cost，双方 **ROI** 都会更好。

画面说明

画面在 Benny 单人近景、主持人近景与双人远景间切换，节目字幕出现“go-to-market capability”等访谈包装；没有实际 VPC、云架构或 GPU 画面。

10. 从 PoC 到生产：Eval 是缺口（24:43–27:10）

场景 10

本节主旨 从 PoC 到生产：Eval 是缺口

完整内容

曹卿云： 企业端 AI 从现在的 PoC 到真正生产，adoption 节奏和关键变量会是什么？

Benny Chen（陈宇飞）： 可以多看这些 **data company 的 ARR 和 adoption**。我相信越来越多公司会采购 open-source model，但它们需要 evaluation，需要说服自己这个采购合理。很多人没有能力做 **in-house evaluation**，就需要 third-party data company 进来帮忙把 eval 做好，也可以拿来做 fine-tuning。所以 data company 的 revenue diversification 和 run rate，是很有意思的指标。

如果一直是几个大的 lab 作为大客户、占这些公司的大头，反过来 customer 可选择的事情就少了。

曹卿云： 企业从 PoC 到生产的过程中，失败主要有哪些原因?

Benny Chen（陈宇飞）： 就是 **eval 不清晰**，特别容易是 PoC 做得不好。我们和很多人聊，会问他们内部有什么样的 eval；虽然已经是 **2026 年**，但还是有不少人只是 vibe，试一下效果好不好就 OK。即使这件事已经占到公司运营 cost center 很大比例，仍然可能没有 rigorous 的 eval。

我经常开玩笑说，很多 engineer 还担心自己会失业，但能做 eval 的人真的少之又少。大家如果有兴趣，可以多转一下“怎么做 eval”，教你老板到底能不能采购开源闭源；把 eval 写好，把开源切到闭源，可能就把你工资的**三五倍**省出来了。这是很多公司缺少的 muscle，也是影响 PoC 最大的点。

曹卿云： 所以 eval 还有很多 job 没被 fulfill?

Benny Chen（陈宇飞）： 我觉得新兴 SaaS 公司和旧 SaaS 公司差别甚至不大，这可能是个暴论。以前 SaaS 公司内部有一套 **unit test**，比如 SQL engine 跑起来需要满足哪些测试；现在 vertical SaaS company 有一堆 eval，agent deliver 出去之前要满足哪些 eval。这些都需要积累、沉淀，需要大量时间和钱把分布做好，所以差别没那么大。

曹卿云： 大家的工作可能慢慢从写测试转到写 eval，只是看每个人多快适应。

Fireworks 的两门生意，一个是**推理**优化，一个是定制模型。先看**推理**优化，未来空间有多大?

画面说明

候选画面保留了主持人关于“企业如何从 PoC 到生产”“eval 缺口”的字幕包装，随后切换到 Benny 回答；未出现真正的评测表或数据看板。

11. 推理优化、缓存与模型路由（27:11–31:05）

场景 11

本节主旨 推理优化、缓存与模型路由

完整内容

Benny Chen（陈宇飞）： **推理**优化还有很远的路。这个工作有很多细致的东西：每次新模型上线后，一开始上线的效果，和优化一年后的极致效果差别很大。我们花很多时间努力缩短这个过程，确保模型一上线就做到最好。

推理优化另一个很重要的部分是把**缓存**做好。缓存本质上不一定在 serving runtime 本身，而在 supporting infrastructure 能不能做好；这件事我们会投入越来越多精力。推理优化本质上仍然是 **labor-intensive** 的工作，有很多事可以做。

曹卿云： 相比模型提供的 API，你们的优化有什么优势？

Benny Chen（陈宇飞）： **因为我们优化的模型多，所以工作会更细。**很多 **one-point API** 能做的是把模型本身 serve 好，但我们从其他模型看到很多 workflow pattern；和开源模型厂商沟通时，我们能指出一些它们没有测到的东西，或它们没看过、而我们看过的 workflow pattern。

它们可能要做一个大而全的 API，但我们很多时候 serve 的是 **customized model**，customized model 的 workload 和 traffic pattern 都不一样，优化工作也不一样。

曹卿云： 你们做模型路由，用闭源 SOTA 模型加开源模型，不仅省成本，还能提高性能。你们和 Harvey 的联合研究提到，前沿模型平均每个任务只被请教 **0.83 次**，质量却超过单独使用最强模型。为什么多模型分工会让任务完成率更好?

Benny Chen（陈宇飞）： 工作场景会影响 **harness**，harness 会影响什么时候调用什么模型。Harvey 的例子里，我记得是 **Opus 作为 executive、GLM 作为 advisor**，这样可以把上下文特别长的工作拆成一些上下文稍短的工作。

本质上，大语言模型的 context 变长之后效果会下降。已知这个特性，又已知工作是长文本，就可以在 harness 上改进；harness 改进后，再改进 harness 具体调度模型的问题。**每个工作使用模型的方式都不一样，这些差别决定我们如何优化。**

这个工作有时像以前在 **CPU** 上做 performance optimization：代码看起来可能差不多，但为了熟悉 CPU 特性，需要把工作拆成 CPU 喜欢的 workload。我们要和 customer 一起探索，把底层模型当作文本 processor，理解 processor 的特性，根据特性拆出 harness，再根据拆完的效果把最适合的模型放在最适合的位置。

曹卿云： 从优化单 token 成本的角度，硬件哪些配置还有空间？比如 GPU 和 HBM 容量比、CPU 和 DRAM 配比、DRAM 和 NAND 配比、HBM 带宽、scale-up 带宽。

画面说明

画面主要是 Benny 单人机位，也有主持人提问和双人远景；候选字幕出现“**推理**优化”“0.83”等节目包装，但没有展示真实路由图、缓存系统或硬件结构图。

12. 硬件比例、模型架构与生态激励（31:05–33:56）

场景 12

本节主旨 硬件比例、模型架构与生态激励

完整内容

Benny Chen（陈宇飞）： 这个问题特别 specific，我想想怎么回答。对这个问题感兴趣的人，可以看两三个月前 **More Cash Potato** 的一个黑板教学视频，里面请人讲 GPU 构造。但里面有个更重要的数字，它是没有 dimension 的：本质上就是 **compute 和 memory 的比例**。Ratio 本身没有 dimension，每种硬件有不同的比例，会影响模型设计。

模型 train 之前，会先想要在什么硬件上 serve；确定硬件之后，就会改 model architecture。很多时候我们问硬件能怎么改，却忽略了绝大部分模型 train 时就是为了在现有的 **NVIDIA** 或 **AMD** 硬件上 serve。**AMD 和 NVIDIA 的比例差不多，所以 NVIDIA serve 的 model 很快可以放到 AMD 上；但放到 ASIC 上就要花很大力气，因为比例和硬件特性不一样。**

反过来，如果有人找到特别便宜、能提供类似比例的 hardware，比例不一样也可以。比如类似的 FLOPs，再 incentiveize 做模型的人去 **overfit** 到这张卡上，问题就变成：我怎么样提供足够好的 incentive，让大家 overfit 到我的硬件？首先卡的量要大，其次卡的 ecosystem 要好。所以我可能没有直接回答“硬件应该怎么改”。

还有什么空间？我觉得老黄以前讲过一个很好的点：开玩笑说，只要有电，你不买我的卡，甚至会亏得很厉害。买这张卡可能是 **cost-neutral**，因为很快能把钱挣回来；其他卡可能没有这个特性，导致 offer 其他卡时没有办法把钱挣回来。

所以问题未必是 scale-up ratio 应该是多少、要不要 HBM，甚至 NVIDIA 可能自己也不用 HBM；真正的问题是，能不能找到一张卡，把上游绝大部分价格合理的东西整合起来，通过各种 incentive program 和软件 ecosystem，激励大家在这张卡上 overfit，特别是 serving 时 overfit。

能做这个 incentive 的人，不管 ratio 多奇怪，最后都能成功；做不到的人，即使 ratio 和 NVIDIA 一模一样，也不能成功。

另外，我们想讲一下定制化模型。

画面说明

画面是 Benny 的单人回答与主持人提问交替，约 31–33 分钟处字幕包装出现“硬件配置优化”等字样；没有显示 GPU、HBM、DRAM 或 ASIC 的实际示意图。

13. 什么样的企业适合定制模型（33:57–34:53）

场景 13

本节主旨 什么样的企业适合定制模型

完整内容

曹卿云： 什么样的企业需要定制、什么样的不需要？这条标准线怎么决定？

Benny Chen（陈宇飞）： 我觉得 **vertical SaaS company** 定制会好一些，具体的 **end consumer** 我们不建议定制。比如前面讲的 **Doximity**、**Heidi**，它们和医院接触，知道比如十万个医院有什么需求，因此能把医疗做好，确保覆盖各种不同 case，有能力去 fine-tune。**Harvey** 可能 support 各种律所，能够覆盖各种律所，模型就会比较全，也可以 fine-tune 一个 model。

但具体到每个医院或每个 legal practice 本身，做 customized model 特别难。它们可能更适合把工作流固化到一个 **skill**，或者写一些内部工具、agent。

曹卿云： 在企业定制 **RL** 项目里，最花时间、最花钱的是哪部分？

画面说明

主持人画面出现“企业如何定制开源模型”的字幕包装，随后切到 Benny；画面没有展示医疗或法律产品界面，企业分类和例子来自对谈。

14. 定制 RL 的成本、数据量与 Rebase（34:54–36:28）

场景 14

本节主旨 定制 RL 的成本、数据量与 Rebase

完整内容

Benny Chen（陈宇飞）： GPU 肯定是消耗的大头，另外还有 data。训练时我们要和 customer 一起看 data 有没有 **reward-hacking behavior**，看 data 的 **environment stability**，以及怎样 scale up environment，这些都要和 customer 一起看。

整个定制化 RL 需要什么数据量？可能**上千个 environment** 就可以。为什么不用更多？因为 RL 过程中模型会持续探索，生成量远大于这几千个。比如每个 step 假设 **roll out 128**，那么 1000 条数据可能生成 **128K** 数据；数据量会随着探索增长，不需要很多 environment 也会有大量数据去教模型哪个对、哪个错。

曹卿云： 客户模型上线之后，未来需要“回炉”吗？

Benny Chen（陈宇飞）： 需要。新模型 launch 之后，我们都会帮他 **rebase**。只要客户的 eval 固定下来，rebase 过程就不会那么长；但 customer 会加入新的 eval、新 workflow，想 push for harder task，这时仍需要 customization work。只要 eval 固化，rebase 就会很快。

曹卿云： Rebase 是很快的一件事。

画面说明

画面在主持人关于“定制 RL 的难点”和 Benny 回答之间切换；候选帧出现“定制 RL 的难点”“模型上线之后需要回炉吗”等字幕，未展示训练曲线或 GPU 集群。

15. Fireworks 的护城河与 CSP 竞合（36:29–40:28）

场景 15

本节主旨 Fireworks 的护城河与 CSP 竞合

完整内容

曹卿云： 像 NVIDIA、CoreWeave 这样的公司，另一类是做软件优化的公司，第三类是有客户关系的 Palantir。现在硬件算力公司也在并购软件能力，比如 AI service；做**推理**优化的公司怎么保持护城河？会不会拥有重资产去买 GPU？

Benny Chen（陈宇飞）： Fireworks 现在肯定没有自己做硬件的想法，更偏“二房东”的 setup，还是想把软件这一层做好。我们相信把 **compute layer** 做好有价值。和各个公司不同的是，我们真相信 **specialized intelligence**，也真的往 **customized model** 移动。我们绝大部分 traffic 不是 vanilla base model，而是 customized model。

其他云公司可能绝大部分 traffic 还是 base model，base model 生意竞争很激烈。但我们坚信绝大部分 profit 在效果好的 model 上。效果好的 model 让 customer 有能力 charge 更多钱，因为它们能 deliver result；这样我们也能分到更多钱。**所以我们相信 customized intelligence，相信把 training 做好，客户 inference 才能赚钱。**

我们和其他 **RL service company** 的不同，是很多公司要么只做 training，要么只做 consulting；它们的 incentive 和 customer 没那么 align。我们 training 不只要效果好，还希望它影响的 traffic 一直上来：customer 通过 inference 赚钱，我们从他们赚的钱分回来。如果是 consulting company 或 pure training-infrastructure company，它们和 customer success 不够 align；我们 align 的是 influence traffic，它们可能 align 的只是 training traffic。

真正的竞争对手可能不是 New Cloud，而是 **CSP**。CSP 有能力把整条链做起来，既做 AI service，也把 inference 做好。所以我们更关注 CSP 能力以及怎么和 CSP partner，它们才像“通关的大 boss”。Nebius 或其他 New Cloud 人力有限，要把整条链做完很困难。

Fireworks 和 CSP 是竞合关系。我们和 **Azure** 合作最紧密，在 Azure 上可以直接买 Azure credit、买 Microsoft Azure credit，然后得到 Fireworks 服务；我们也和 Azure、**GCP** 合作。

曹卿云： 你们的**推理**优化会比 CSP 自己的团队做得更好吗?

Benny Chen（陈宇飞）： 至少目前是。但我不太担心具体推理优化好坏；CSP 把整条链做下来能力很强，因为它们做过这些事情。我们更在意把整个 **MLOps cycle / developer life cycle** 走完，把体验做好，这是 CSP 可能更能做到、而现有 New Cloud 未必完整的事情。

曹卿云： 所以 Fireworks 的核心竞争力是什么?

Benny Chen（陈宇飞）： 比 CSP 做得更好的地方，可能是整个 developer life cycle 做得更好；和 New Cloud 的竞争则是我们能做的链更长，而它们 serve 的更多是 base model。

画面说明

画面以 Benny 单人近景为主，穿插主持人提问、双人远景和“Fireworks 核心竞争力”字幕卡；没有显示 Azure、GCP 或云平台控制台。

16. RSI、客户需求与 Fireworks 的计算挑战（40:29–41:36）

场景 16

本节主旨 RSI、客户需求与 Fireworks 的计算挑战

完整内容

曹卿云： 现在大家都在聊 Recursive Super Intelligence（RSI）。如果 RSI 把 kernel 优化、推理优化和 training / post-training 全部自动化，你们的核心竞争力在哪里？

Benny Chen（陈宇飞）： AGI 是特别难预测的事情。如果真的有 AGI 能把上面所有活都干好，我觉得确实没什么核心竞争力。难点在于，即使 AGI 之后也需要和客户聊需求。一个 AGI 在 lab 里能不能把客户需求都想出来，这是很困难的事。如果客户需求没法通过 RSI 获得，我觉得 RSI 的成长可能也有限。它确实可能解决黎曼猜想，但我不确定它能不能解决客户需求。

曹卿云： Fireworks 从现在到未来一两年，最大的挑战是什么？

Benny Chen（陈宇飞）： 未来一两年还是 **compute**。能不能在市场上用合理的价格获得足够多的计算？现在 compute 价格非常高。公司最后 revenue 肯定还是 compute，所以如何获取足够多的 compute 是比较困难的事情。

画面说明

候选帧保留“RSI 全面自动化后，核心竞争力在哪”“未来一两年最大的挑战”等提问字幕，回答部分是 Benny 单人镜头，没有展示 RSI 或 AGI 的图示。

17. CapEx、债券压力与基础设施受益逻辑（41:36–45:04）

场景 17

本节主旨 CapEx、债券压力与基础设施受益逻辑

完整内容

曹卿云： 进入今年夏天以来，资本市场有一个很大争论：开源模型质量开始 catch up，闭源模型收入端会承压，过去趋势不一定持续，也会使 AI 未来投入承压。因此硬件股票、半导体行业出现比较大的回调。开源模型持续 catch up，你怎么看这轮资本市场趋势?

Benny Chen（陈宇飞）： ROI 高不高要反过来想。假设几个大的 player 错过这一波，即使 ROI 很高但错过了，他们可能再也回不到牌桌。**所以它们的想法是，即使 overspend，也不会有特别多 downside：最多这两年账本难看一点，折旧可能用五年、十年消化；但这不是 existential risk，它们还有二十年、三十年去消化，不是问题。**反过来，如果错过这一波、没法解决，那可能就完蛋。

所以 Google、Meta 算的不是“我 LIC 多少”（原声缩写听写不清），而是万一 LIC 特别高、我却没投入，可能会错过牌桌。New Cloud 可能要更 honest 地算清楚投入到底多高。

按我们的观察，绝大部分场景仍然是 **compute-constrained**，Fireworks 也 compute-constrained。我们希望从市场找到更多 compute，至少接下来一年都像 compute-constrained setup。一年之后很难 predict，但资本开支一旦投下去，可能要**三三年**（原声如此，按语境为约三年）才能回本；这是一个 open-ended question，回答不了。

可以看各个公司的**债券**，债券评级很 transparent。某些公司如果项目有压力，可能导致债券评级下降。假设接下来 **3–6 个月**有 player 因为债券出问题、没法 borrow 更多钱，需要 restructuring；市场上到底有几个人有钱把它捞上来？这可能是一个 **watershed moment**。在这个时刻之前很难测，因为不知道大家兜里还有多少钱。只要能捞上来，问题不大，大家可以继续；万一捞不上来，就会很困难。

开源模型追上来导致硬件价格下跌，我确实捋不清这个关系。但在我看来，开源追赶绝对 favor **infrastructure provider**：它让 model 这一层的议价能力变弱。如果钱不变，肯定 favor infrastructure 这一层的人。更接近 infrastructure 的人，股价应该涨而不是跌；比如去年 **DeepSeek** 刚出来时 NVIDIA 股价也跌，我当时也不懂大家怎么想。

所以开源肯定 favor hardware provider。你说实话，我不适合炒股，我理不清中间的逻辑；但你问我，我觉得它肯定 favor infrastructure provider。

曹卿云： 非常感谢今天 Benny 的分享，非常有启发。

Benny Chen（陈宇飞）： 谢谢，谢谢。

画面说明

后段仍为演播室访谈镜头，Benny 单人近景、主持人近景和双人远景交替，结尾回到致谢；字幕出现“资本市场”“开源模型”等提问包装，没有真实股票或债券图表。