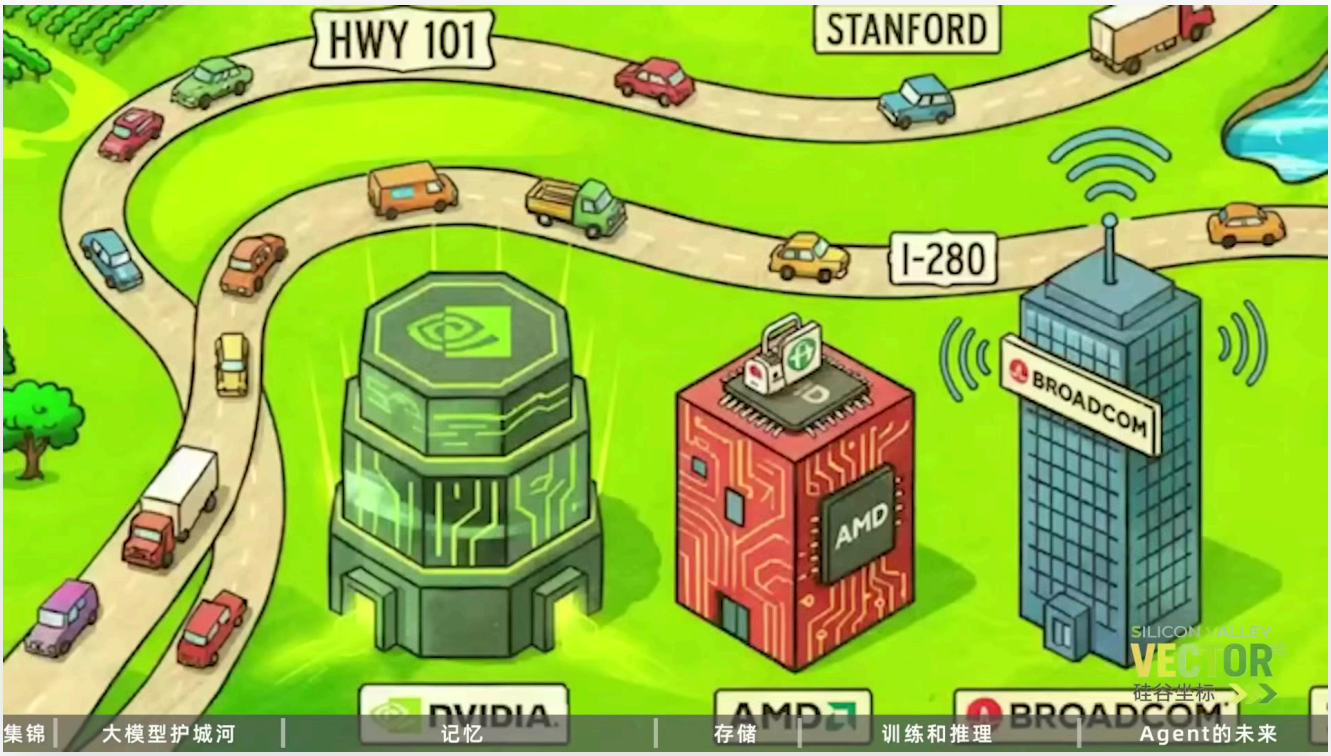
硅谷坐标 x 前Meta AI总监田渊栋: 解析大模型护城河、记忆存储瓶颈与Agent对社会冲击 – 图文解说

> 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

本期田渊栋从大模型护城河讲到持续学习、记忆和 Agent 的社会影响，重点解释为什么模型能力会扩散，而真正难复制的部分可能来自数据、系统、推理效率和工程验证。对谈进一步拆解 Knowledge Distillation、**开放与闭源**、**长期记忆**、Titans/RLM、Test-Time Scaling、Prompt Injection，以及个人助理和 AI 对工作的影响。**核心判断**是：AI 会替代大量重复劳动，但在高容错成本、需要验证和**现实反馈**的环节，人类仍会长期参与。

01. 访谈开场与研究方向（00:04–01:37）



场景 1

本节主旨 访谈开场与研究方向

完整内容

田渊栋：硅谷里面很难有一个秘密能保留很久。**可能一个新的方案弄出来之后，过了一两三个月，大家可能都知道一点了。**

曹卿云：在你的愿景里面，未来有一个能够**持续学习**的 AGI，它是一个脑容量在不断扩大的，还是一个脑容量固定，但是会持续地做记忆的升华和主动遗忘的?

田渊栋：我觉得应该是后者，当然它扩大当然更好。我觉得以后大**内存**应该是一个很大的趋势，因为有这样的一个需求。老黄也好，AMD 他们也好，当然都想要把**内存**变得越来越大，那最后就会导致**存储**会有这样的压力。

田渊栋：我觉得这个是 AI lab 竞争的需求。**推理**过程不一定是人类语言，可以是一个抽象语言，可能是用某种高维向量来表示的。

曹卿云：让这个**推理**变得更有效率，以更快的速度把这个推理找出来。最后问你的一个问题，你下一站去哪里？

田渊栋：**大家好，非常感谢今天有幸被 Silicon Valley Vector 的同学们邀请过来做一个访谈。**我是田渊栋。

田渊栋：之前在 Meta 做研究总监，主要是做强化学习、搜索优化，还有大模型的一些推理和应用。在 Meta 已经快 **11 年**了，现在已经出来自己开公司、创业。

画面说明

片头是带有 HWY 101、Stanford 和 I-280 标牌的硅谷地图，画面中有 NVIDIA、AMD、Broadcom 的芯片或建筑图示；底部栏目可见“**大模型护城河**”“记忆”“**存储**”“训练和**推理**”“**Agent** 的未来”。

02. 模型竞争与护城河排序（01:43–08:58）

场景 2

本节主旨 模型竞争与护城河排序

完整内容

曹卿云：田董，今天欢迎你来到《硅谷坐标》，跟大家分享一下你在 AI 前沿看到的一些最新动向。我们看到 2026 年才过两个月，其实在模型这个赛道里竞争非常激烈。不管是开源模型还是闭源模型，都发布了最新的版本。你是怎么看待现在各个大模型之间你追我赶的局面？似乎这样的领先优势很快就会被追上来。

田渊栋：我觉得这是一个非常普遍的现象。应该说从大模型爆发以来，也就是 2022 年年底以来，这样的趋势正在慢慢加剧。而且在技术上，其实有蒸馏这样的技术。**所以一个不是特别好的模型，可以通过蒸馏更强模型的输出，很快达到更强模型的水平。**这种趋势应该说是不可避免的。随着以后更多的人掌握这些技术、掌握这个流程，应该会有更快的迭代速度。现在确实已经非常快了，可能已经快接近人类生理极限了。

田渊栋：**至于这些 AI lab 除了保持领先优势，完全要取决于每个 AI lab 的定位和方向。**有些 AI lab 背后是大公司，那么对他们来说，现金流是不愁的。不愁现金流的话，对这些 AI lab 来说，他们的目的是向大家展示这个公司本身的技术实力和非常强的人才储备。不停发布各种各样更新的模型和更新的结果，能够让大家知道这个公司仍然处在人工智能的顶尖，或者说第一梯队。

田渊栋：我觉得谷歌其实是一个很好的例子。他们通过发布各种各样不同的模型和不同的结果，也可以让大家知道，确实谷歌应该是在这方面非常领先。**像他们最近用 Gemini 做很多比较难的数学问题，发现用最新版的 Gemini 3.1 Pro，能够找到一些很好的数学问题的解。**这些问题可能以前是未解问题，但后来发现，通过大模型的搜索和探索，才能找到一些很好的证明。这样的话，大家会觉得谷歌仍然领先，仍然是在大模型竞赛中占据第一梯队。我觉得这是一个很好的策略。

田渊栋：**还有一些公司，像初创公司，他们这种你追我赶，一方面是需要证明自己很强；另外一方面，通过这个方式可以获得更多融资、获得更多人的认可，然后会有下一轮资金的注入，让这个 AI lab、让这个初创公司能继续活下去。**我觉得这是两种不同的策略。但是也许以后总有一天，初创公司如果钱烧完的话，需要在钱烧完之前找到一个商业模式，让它能够活下来。像 OpenAI 现在其实也在考虑如何在 ChatGPT 中做广告，对吧？把广告插入 ChatGPT 的对话中，或者放在旁边的栏里面，让大家看到相关的广告信息、一些知识。这样的话，它也找到自己的现金流输入，可以让自己长久地活下去。我觉得这对于大厂和初创公司来说都是必不可少的，一定要证明自己的模型非常强。所以我相信在不久的将来，这个趋势还会继续发展下去。

曹卿云：在你看来，AI lab 真正可持续的护城河是什么？是算力和 infra，还是数据、算法，还是人才？在未来的 3 到 **5 年**，如果让你给这四个维度排一个序，你会怎么样去排？

田渊栋：我觉得最重要的应该是数据。倒不是，可能数据和 infra 是比较重要的，但是 infra 应该是慢慢也会有一些变化。因为现在用 AI 写代码的趋势越来越厉害。以我自己来说，我会觉得比如跟三个月前相比，我的效率应该说提高了至少 **10 倍**，大概是这样的一个逻辑。

田渊栋：**所以我觉得以后可能会有越来越多的人开始用 AI 写代码的方式来构建自己的系统，来做自己的一些查错，或者做一些能让运行变得非常顺畅的日常工作，可能会让 AI 来代替。**如果是这样的话，infra 这个护城河可能会有一些下降，这是我的一些想法。但是数据本身可能还是比较难，数据应该是很重要的，特别是对一些比较难的垂直领域，或者说这个领域上的数据非常少的话，那么你没办法用很少的数据训练出很好的模型。所以数据还是一个比较重要的因素。

田渊栋：至于算法，目前看起来改动不是特别大。有很多方法其实就在原来的算法上做一些小修小补，就能够做出来。一些比较 fancy 的修改可能不一定有用。像 DeepSeek 可能在一个月前发布了一篇文章，就是 MRS，做一个对残差连接的魔改，他觉得非常有意思。**但是后来大家可能发现，有些算法可以改得非常简单，跟原来差不多，但效果还是可以的。**所以在算法上的一些修改，不一定会导致完全不一样的结果。目前算法处于一个比较稳定的状态，这是一个很大的瓶颈。我相信现在可能处于这样的状态：要不大家改来改去，改不出东西来；要不就是完全不一样的新方案，能够把原来的东西颠覆掉。我们现在处于这样的一个状态，但这个跳变什么时候发生，现在还不太清楚。我也希望这样的跳变能够发生，这样我们就可以得到下一代模型。

田渊栋：但如果假设跳变不发生的话，现在算法可能没有数据和 infra 重要，还是这样子。另外一方面，人员会流动。**你会发现各大 research lab 之间，人员会有很多变化。**两个月前这个大佬在这个地方，过两个月他跑到另外一个地方去了。通过人员流动，很多新的想法和新的思路会从一个地方流到另外一个地方。

田渊栋：所以硅谷里面很难有一个秘密能保留很久。可能一个新的方案弄出来之后，过了一两个月、两三个月，大家可能都知道一点，可能就传开了，知道怎么做了。**所以算法和人才应该没有那么重要，重要的是数据和 infra，大概是这样的逻辑。**

田渊栋：至于算力本身，也是一个很大的瓶颈。**算力主要是大厂和初创公司之间的区别，但大厂之间算力可能相对来说都不是差特别多，都有很大的算力配额给 AI lab。**

曹卿云：**在你想象的终局里，你觉得会是一家或者几家大模型独大，再加上几个垂直领域数据比较独特的小公司吗？**你想象未来是这样吗？

田渊栋：**我觉得有可能是这样子的。**

画面说明

画面一帧展示“知识蒸馏 **Knowledge Distillation**”，图中大模型 Teacher Model 的知识输出指向小模型 Student Model；另一帧的下方字幕写着“护城河排序：数据 > Infra > 算法 > 人才”。

03. 开源模型与日常使用（09:04–12:39）

场景 3

本节主旨 开源模型与日常使用

完整内容

田渊栋：还有一个问题就是，今年开源模型在网上有很多讨论，包括一些争议。你是怎么看待开源模型的发展呢？**我觉得开源模型是一个非常重要的方向。**地球上不可能只有闭源模型，如果只有闭源模型，会导致一个非常糟糕的将来。

田渊栋：我可能在 2023 年的时候就有这个想法：我们的模型一定要是开源的，至少要有开源的一席之地，这是非常重要的。为什么呢？我觉得对于一个指数增长的技术来说，最大的、可能的最坏结果，是少数人掌握了这个技术，而大多数人不知道。少数人用这个技术去做一些不太好的事情。

田渊栋：如果是这样，首先地球上大部分人可能获得不了这个技术带来的便利，然后还会产生很大的等级区分。这是开源模型要避免的事情。有开源模型之后，大家就变得平权了，大部分人可能获得大致一样的计算能力和模型能力。**有了这个之后，大家能够同步地往前走。**

田渊栋：能不能往前走，换一个比较有意思的说法就是：如果大家都有核武器，产生了威慑，相对来说就会有一个比较好的平衡点。**如果只有某一些人，或者某一类人有这样的工具，可能也会产生一些不必要的问题。**

田渊栋：我以前的公司，比如 Meta，在一年前还是比较适合、比较想要开源的。**但是现在可能更偏向闭源的策略。**我觉得这个策略本身没有什么问题，因为完全取决于公司本身的战略。对公司来说，如果觉得开源有利于竞争、有益于发展，那就开源；如果觉得闭源有利于发展和竞争，就用闭源的方式。公司没有必要非常清楚地坚持一定要做一件事情，特别是这件事情如果跟它的主营业务没有关系，其实可以灵活。

田渊栋：我现在用得最多的模型，其实现在应该都在用吧。比如 OpenAI 的模型在用，Transformer 的模型也在用，然后开源模型我也在用。像 GLM 5、MiniMax 2.5，这也还是不错的。**MiniMax 2.5 可能比较快一点，还是挺不错的。**

田渊栋：你会发现有很多事情让我觉得吃惊。可能半年前你去用这些模型，它们还没办法做一个完整的任务，可能有各种各样的问题。**但是现在再去用，你去问 Claude Code，它当然做得很好；你去问 MiniMax，其实也可以，做得还不错。**它有一些问题，比如什么地方会忘记什么东西，但是大概的流程和逻辑基本上还是正确的。

田渊栋：**所以这个其实让我非常吃惊。**我觉得模型进步已经这么快了，这是一个很有意思的现象。以后可能也会有更多新的好模型冒出来，然后给我们的日常生活、工作效率提高很大的助力。

曹卿云：接下来今天想重点跟你聊一聊大模型的记忆，这也是你研究的重点方向之一。**你先跟大家讲一讲，大模型到底是怎么记东西的?**

田渊栋：**大模型的记忆一直是一个很大的问题，我们在——**

画面说明

画面左侧标题为“开源 vs 闭源模型”，列出 GPT-4o、Claude 3.5、Gemini，以及 LLaMA 3、DeepSeek、Mistral；另一帧字幕写着“Meta 为什么从开源走向闭源？”。

04. 长上下文与两类记忆（12:40–20:23）

场景 4

本节主旨 长上下文与两类记忆

完整内容

田渊栋：**2023 年**上半年的时候，其实我们已经开始做一些这样的工作了。当时我们做了一些长文本、长上下文的大模型拓展。**2023 年 6 月份**的时候，我们有一篇文章叫 Position Interpolation，当时研究的是如何把大模型的 context window，也就是上下文窗口长度变长。

田渊栋：本来这个长度可能只有 2K 或 4K token 这样的量级。那么当时我们在想，怎么样把这个东西拉长？在我们的方法出来之前，大家一直以为，要把窗口拉长，就要用更长的数据去训练它。**但是这个训练过程非常耗时。**比如一个模型训练完了，窗口是 2K，也就是 2048 个 token，训练完之后窗口就这么大，定死了。如果想扩展，就得再拿一大堆很长的上下文数据，让它再训练一遍。这个过程非常慢、非常痛苦，而且要花很多很多卡，模型质量还不一定好。

田渊栋：当时我们发现一个很有意思的现象：只要把长上下文窗口映射到短上下文窗口，把每个 token 输入时的位置信息简单地除以二，就可以映射过来。除以二之后，再去做微调、再去训练，所需的训练代价就小很多，质量还不错。后来这个现象被大家广泛应用。从这开始，这篇文章应该说是这个方向的开山之作之一。大家发现可以这么做之后，窗口突然之间就变得很长。应该说从 **2023 年**下半年开始，有很多工作开始做长上下文的预测。我记得包括 Gemini、Kimi，其实都有一些这样的工作，研究怎么样把窗口变得非常长。

田渊栋：这之后我们也有一些其他工作，比如 Attention Sink，中文翻译可能叫“注意力陷阱”或者“注意力汇聚”，我也不确定哪个是官方的。它的逻辑是，如果我只保留整个句子前面的几个 token，把中间那些东西全部去掉，这个模型还是能够输出比较正确的话。虽然中间那部分被去掉之后没有记忆，但是它说话至少还是连贯的。**你真的问它事实，它可能会开始出现幻觉，但至少不会出现爆炸性的结果。**

田渊栋：我们之后还做了一些 extension，比如把中间去掉的东西再拿回来。我们有一篇文章叫 H2O，也就是 Heavy Hitter Oracle。**通过这个方式，首先记忆的大小是固定的；另外，把重要的记忆拿回来之后，它还能保证一些比较关键的问题回答正确。**

田渊栋：这些文章都是关于记忆的，都是在研究怎么让两种目标达到平衡。一种是希望把过去所有上下文全部记住，这需要花很多代价，但主要效果比较好；另外一种是有选择性地去掉一些过去的记忆，这样可以保证**存储**大小没有那么大，同时模型还能输出比较好的结果。最近一些记忆文章，其实都遵循这个逻辑：一头是用大量**内存**把记忆都存下来，但速度慢、**存储**大；另一头是去掉一些记忆，速度变快、**内存**变小，但有可能会忘记一些东西。

田渊栋：像之前一直很火的吸引注意力模型，它的逻辑是把过去的上下文压缩成一段固定长度的向量，压缩完之后把这段向量作为过去的记忆。**好处是记忆量非常少，但如果真的想找到过去历史中的所有细节，吸引注意力可能就不太行，因为有限的空间容纳不下无限的过去历史，所以这里面有一个 trade-off。**

田渊栋：我们先讲的这些，都是用短期的上下文记忆，这部分很重要。还有一部分记忆是在模型的权重里面，这就是更长期的记忆。这种模型权重的记忆，从预训练开始就建立起来了。预训练就是把很多很多数据、海量的整个 Internet 放进训练里面，让它进行大规模训练。这样，它的记忆权重会从初始化开始，慢慢演化到一个比较好的状态。这个记忆演化的状态，就是**长期记忆**。

田渊栋：**长期记忆**其实规范了模型本身对这个世界的整体理解。这部分记忆很难被改变，而且对模型的能力有很强的影响。如果模型在预训练时训练得不太好，最后在做后训练时，就会像一个比较笨的孩子：什么事情都必须给你讲得很清楚，他才能把它记下来；他无法举一反三，新的事情、新的任务来了之后又不会了，你得再给他说一遍，把过程一步说清楚，他才能做出来。

田渊栋：但是如果训练得比较好，他对这个世界的理解非常深刻、非常一般化，有泛化能力，就像很聪明的孩子，一点就通。**在后训练过程中，他很快就能适应后训练的任务，并且能够举一反三。**大概是这样的两类记忆。

田渊栋：**现在记忆中间，你觉得研究者碰到的最难的问题是什么？**

田渊栋：我觉得最难的问题，是记忆如何从背诵、从死记硬背到顿悟，这样的一个过程。**你去看任何一个小孩子怎么学说话，就会发现这是一个很难的过程。**我有时候会觉得我家女儿的学习过程很有意思。我们有自己的孩子，看看她怎么学习，就可以看看 AI 的学习能力跟她有什么区别。

田渊栋：小孩子在三岁或者四岁之前，你跟他讲很多东西没有用，因为他记不住。你跟他说再多，他也会觉得第一是记不住，第二是开始哭闹，不想跟你学。**但是过了一段时间，突然在某一天，这些事情他都会了。**我一直在想，这个小孩子的脑子是怎么长的？他会在某些情况下、一定时间之后，让内部的记忆发生一些重组。

画面说明

一帧图示“线性注意力 Mamba / SSM”，对比 Transformer“全文回看”和 Mamba“随手摘录”，并标出 $O(n^2)$ 与 $O(n)$；另一帧是田渊栋访谈近景，姓名条写着“田渊栋，**前 Meta AI 研究总监**”。

05. 从死记硬背到顿悟（20:24–26:20）

场景 5

本节主旨 从死记硬背到顿悟

完整内容

田渊栋：重组之后，记忆的表示发生了变化。**变化之后，他突然之间理解了之前无法理解的一些逻辑，然后用这个逻辑举一反三，可以做很多很多事情。**

田渊栋：这其实很有意思。比如数数，你可能在两三岁的时候教孩子数数，教他一二三、怎么样放，他可能只记住非常机械的东西：这个东西加这个东西，但不是特别清楚。**但是过了一段时间，比如四岁以后，他会突然对数字的大小开始有感觉，大概猜出来这两个数字相加是什么，突然猜出来一些两位数以及它们之间的关系。**很多时候不用大家教，他就自动会了。

田渊栋：所以这个过程很重要。**但是在理解上，我们现在还是有很大的困难：这件事情怎么样发生，什么时候发生，怎么样让它发生得更快？**能不能让模型的学习变得更有效率，更像小孩子那样学得更快？这是一个问题。

田渊栋：现在有很多记忆的方式，一些新的范式怎么做记忆呢？它们的过程还是比较机械化的，没有那么像小孩子学习时那么灵动。**比如最近的一些文章，像 Google 那篇叫 Nested Learning 的文章，这是 Google 去年一篇比较火的文章。**

田渊栋：他们的想法非常简单：在设计网络架构时，打破优化器和神经网络架构的界限，希望所有东西都是一类东西，叫 associative memory。**什么叫 associative memory？**在生物学、脑科学里面有这样的概念：我输入一个东西进去，它就出来一个东西，是一一映射，像一张表格一样。比如我输入“今天”，输出“今天天气是什么”，是非常清楚的映射。他希望把神经网络里面的每一部分学习过程都映射成 associative memory。

田渊栋：这样相当于把神经网络里面每一部分都作为记忆模块的一个特例。这个逻辑本身挺有意思，因为这样可以把事情统一化。但另一方面，我不是很认同这个方向。**对我来说，所谓 memory 还是效率不高，因为它只是把一个点记住了，然后把那个点弹出来。**

田渊栋：但是对人来说，学到一定程度之后，你会发现他对这个世界有一个整体的理解。这个理解可能就像以前诸葛亮说的，看一件事情要“观其大概”：我不看细节，但是对大概理解得非常深入之后，很快就能够对这个问题有一个很好的答案。**所以我觉得 associative memory 还比较简单，对于人类记忆的形成过程，模型的建模还不够。**这个问题本身很大，可以慢慢再去解决。

曹卿云：在你的愿景里面，未来有一个能够**持续学习**的 AGI，它是一个脑容量不断扩大的，还是脑容量固定，但是会持续做记忆的升华和主动遗忘的？

田渊栋：**我觉得应该是后者。**

田渊栋：**当然它扩大当然更好，但是我觉得后者是让一个人变得聪明，或者让一个 AI 有飞跃性进展的一个很重要的因素。**我觉得后者更重要。

田渊栋：前者其实可以认为更像 Internet。全世界的 Internet 越来越大，可以把 terabytes、petabytes 这样的数据放进**内存**、放进硬盘里。那么多数据堆积在一起，让检索变得非常有效率，但它并没有升华成一个有自主意识的人，或者一个对问题有更深理解的人。过去的搜索时代并没有做到这一点，还是需要人把数据整合起来，然后获得一些新的知识。

田渊栋：**但是大模型来了之后，一个很重要的贡献就是，它通过训练把数据和知识整合到了权重里面。**这个整合让模型对数据和知识的理解上升了一个层次。你再去查询这个模型，它给你的回答就非常灵动，很像人了。

田渊栋：这就是这一代技术跟上一代技术的区别。**上一代技术比较机械，给你单独精准匹配，匹配完之后得到结果；现在这一代技术已经开始比较灵动，更像人。**它的思维更像人，能对问题有一些分析和理解，这是很大的区别。

田渊栋：这两代之间最大的区别，就是因为我们用了神经网络、用了大模型。大模型做了训练这件事，让知识的表示和**存储**发生了质的变化。如果从大模型出发再往前走一步，让知识的表示和**存储**发生更加质的变化，或者让训练和存储的代价变低、学习能力变强，能够很快适应这个时代、适应变化的世界，可能就是下一代新的模型。

曹卿云：**现在我们看到 context window 的长度越来越长，你认为之后还会继续这样持续变长吗？**天花板在哪里？

田渊栋：我觉得会。**很难有天花板，因为现在有这个需求。**

画面说明

画面展示“统一记忆架构 **Titans**”，左侧是 Attention 层、FFN 层、Optimizer，右侧通过“统一化”指向多个“记忆单元 Memory Unit”；另一帧为双人访谈画面，字幕写着“未来 AGI 的记忆形态”。

06. 上下文长度与存储压力（26:20–37:12）

场景 6

本节主旨 上下文长度与存储压力

完整内容

田渊栋：以前我们用大模型，主要还是因为跟它聊天。聊天的频次不会特别高，一个人在网上泡一天、十个小时，能聊多少话？可能聊十万字，已经很可怕了。**这个数字基本上是一天一个人能够消耗的 token 上限。**

田渊栋：但是现在大模型的用处不是聊天了，很多时候是用来写代码，或者做一些问题的分析。这种情况下，动不动就要把整个代码库放进去，或者让模型做多轮工具调用，做多轮分析问题、解决问题的迭代。**这样很快就会发现，上下文超过一万，或者超过十万，经常会出现这种情况。**你很快就会发现上下文用完了。

田渊栋：另外，大家非常热衷于希望模型能够长期工作，不需要人干预。这是最近半年各大厂都想做的事情。本来工具可能每隔三分钟就要跟你汇报一次，说“我做完了，下一步是什么不知道”。这对人的注意力要求很高。**之后我们当然希望模型能够工作一个礼拜，或者几天几夜，然后不要任何人干预。**

田渊栋：这样的话，就需要大量 token、大量上下文，让模型能够工作。**所以总的上下文应该越来越长，而且上下文越长，它对这个世界的理解就越深入，决策应该也会越准确。**我相信这是一个很大的趋势，很难改变。

田渊栋：最近有一些把上下文 memory 做得比较好的机制，比如最近比较火的 CloudBot，大家都知道这个 CloudBot。**它的 memory 机制比较有意思，是把 memory 组织成各种各样的 Markdown 文件，有短期的，也有长期的。**

田渊栋：这些文件首先是可读的，human readable。我们真的去看这些文件，把其中一行去掉，就让它真的不记得这件事了。虽然这个做法很 awkward、很难看，但如果我不想让它记住这件事，就可以这样做。**文件本身也有层次感，有些是近期任务，有些是远期但非常重要的记忆。**通过这个方式，可以让 bot 对这个世界理解得更加深刻。我觉得这个设计挺有意思，会拿去看一下。

田渊栋：另一方面，虽然这个设计本身很有意思，但我们的最终目的当然是希望 AI 能够自动发现这个设计。如果让 AI 自动发现这个设计，就要给 AI 很长的上下文，让它自己去挑。**这意味着你做任何探索，都需要从很长的上下文开始，然后希望 AI 慢慢在里面挑，挑到一些好的，塞到当前的对话里面去，把这件事情做好。**

田渊栋：**这方面的代表工作就是最近 MIT 的一篇文章，叫 Recursive Language Model。**那篇文章是在研究，怎么样把上下文作为一个数据库，然后动态地从里面调东西出来，用它做预测、做 decoding。我觉得这个趋势以后还会持续，会有更多文章出来。

田渊栋：**但是不管怎么样，最终也许上下文比较短，但在研究过程中还是需要很长上下文。**从这开始，我觉得这个趋势很难扭转。大家第一反应都是先把东西都塞进去，看有没有效果，然后慢慢再往下掉。

曹卿云：我接下来想跟你聊的话题，也跟刚刚聊的非常相关。我想从物理世界再来看模型这一端的发展。从去年开始，我们看到**内存**和**存储**的产业链都出现供不应求的状态。AI 的需求非常大，产能跟不上，所有产品都开始进入提价周期。甚至我们看到 Google、Microsoft、NVIDIA 的采购高管，很

长时间待在韩国首尔，有人说他们在首尔的时间可能比在 Mountain View 还长，就是为了保证三星、海力士的产能。

曹卿云：感觉 AI 的发展正在把全栈**存储**，包括热数据、温数据和冷数据的产能全部占据。你觉得最大的增量来自哪里？可能是刚刚说的 context window 增量？这个增量可持续吗？未来有什么好的解决办法？

田渊栋：**首先，context window 变长肯定是一个重要增量。**另外一个增量还是训练模型的需求。现在模型还是很大，比如 Kimi K2，是一个 trillion parameter 的模型；DeepSeek 的大小也是六百多个 billion parameter。现在模型大小已经变成标配了。以前可能觉得模型好大，现在我们认为五百、六百 billion 到一个 trillion，这个大小成了模型的标配。如果达不到这个大小，可能就会觉得模型能力不太行。

田渊栋：这个数字跟以前大概有十倍的差距。以前可能会觉得一个开源模型七十几个 B，七十个 B 的模型大概还行，OK。**但是现在大家胃口变大了，希望模型有更大的参数集，训练效果更好。**

田渊栋：这会导致训练模型时，我希望用同样的算力搭配更多**内存**，这样效率会更高。如果同样的算力内存不够，就要讨论很多问题：同样一个模型，单卡放不进去怎么办？你得把内存切片，把模型切片，可能用 tensor parallelism、data parallel，或者 expert parallel。

田渊栋：比如一个很大的矩阵，一张卡存不下，可以切成几部分，横着切，也可以竖着切，用各种方式切，让每部分权重放在不同的卡上。这样可以减少单卡上的内存消耗，但代价是卡和卡之间要通信。**通信需要时间，会增加整个 pipeline 的延迟，也就是 latency。**

田渊栋：相当于一张卡上内存不够，就要通过增加延迟的方式让系统跑起来，这是一个不太好的 trade-off。比较好的方法是，在单卡的 CPU 上加很多内存。加大内存之后，可以把整个模型放进一张卡，或者放进一个八卡的大机器上。**这样通信代价降低，训练过程变快，出结果的速度也变快。**

田渊栋：大家如果想在 AI 竞争中胜出，当然希望速度更快、更好，让设计变得更简单，减少出 bug 的概率。**所以我觉得以后大内存应该是一个很大的趋势，因为有这样的需求。**自然，老黄也好，AMD 也好，都想要把内存变得越来越大，最后就会导致存储有这样的压力。我觉得这是 AI lab 竞争的需求。另外，上下文变大也有这个需求。这两方面的需求都有，有了需求之后，所有卡的内存都在往更大的方向走，这是非常正常的。

田渊栋：比如现在去外面租卡，你会租 H100 还是 H200？我觉得 H200 肯定更好，因为 H200 内存更大。同样的算力，有更大的内存，就有更好的办法做训练。同样一个模型，用 H200，内存大了之后，就可以用更少的卡获得同样的性能。**这样算下来其实还是合算的，所以大内存的卡会那么受欢迎，还是这个原因。**

曹卿云：可以说，随着未来 **Agent** 应用的数量变多、任务复杂度增加，后面还有多模态、世界模型，这个需求是不是就无解了，会是指数型增长？

田渊栋：确实很大，也很难。我刚才只说 language model，还有 image model。**图片进去之后，需要高清地进去，比如 4K 照片进去，那内存要很大。**主要是 activation，因为单图片模型主要不是参数量很大，而是一张图或者一个批次的图进去之后，需要大量内存存中间结果，这部分会花很多内存。

田渊栋：内存大了之后，本来批次训练可能用 128，现在可以放 256 张图片进去，训练速度是不是更快了？所以这个东西卡住了训练速度和效率，也包括最终 serving 时 **Agent** 的效率、速度和容量。这是一个很大的问题。大家都想要大内存，内存厂商就会有一个瓶颈，而且这个瓶颈很难被克服。

曹卿云：**所以这是短期还是长期的问题？**你现在还没有看到一个好的解决办法？

田渊栋：我现在挺难看到很好的解决方法，还挺困难的。当然最近也有一些方案。比如最近有一个公司，把整个大模型的权重都刻到 ASIC 电路里面去，通过这种方式提高速度。**原本要用内存去存权重，现在把权重移到 ASIC 电路里面，这样就可以用多出来的内存存别的东西。**

田渊栋：这是一种方案，但可能不够灵活。如果模型改了一点点，那些电路就没用了。**所以如果现在想做研究、做比较灵活的探索，还是需要原来的架构。**

曹卿云：我们再回来讲预训练。**训练这块，反正你一直对 Scaling Law 是比较悲观的态度，觉得 Scaling Law 只不过是拿更多权重、更多数据、更多算力把模型变大，但这个过程其实非常不有效率，效率很低。**

画面说明

画面一帧展示“递归语言模型 **RLM**”，对比传统方式把全部内容输入模型与 **RLM** 动态检索相关章节；另一帧为双人访谈，字幕写着“**存储**危机：AI 正在吃掉全世界的全栈**内存**和**存储**”。

07. Scaling Law 的资源约束（37:13–37:49）

场景 7

本节主旨 Scaling Law 的资源约束

完整内容

曹卿云：**但好像现在的大厂，包括 Google、OpenAI，他们还是继续朝着 Scaling Law 的方式往下发展。**你是怎么样评论这件事情的?

田渊栋：首先，因为我刚才说过，我一直的论点是 Scaling Law 是 work 的，只是它需要大量资源，需要指数级资源去支持它。刚才我们说**存储**是一个很大的问题：如果现在都卡在**存储**上，大家还能不能把这个 scheme 做起来？这是一个问题。

田渊栋：另外电力也是一个问题。**电力能不能保证这个很大的集群跑起来，而且要保证电力供应稳定，这些都是很大的问题。**

画面说明

画面是田渊栋访谈近景，字幕写着“预训练 Scaling Law 的路径依赖”；底部栏目仍可见“**大模型护城河**”“记忆”“**存储**”“训练和**推理**”“**Agent** 的未来”。

08. 路径依赖与持续学习（37:50–41:21）

场景 8

本节主旨 路径依赖与持续学习

完整内容

田渊栋：对于公司来说，我觉得对大厂来说，他们有自己的路径依赖。因为大厂已经把所有的 team 都建好了，每个 team 各司其职，把事情做成。**所以很难让大厂转方向去做一个不太可能、或者很难看到希望的新方向，这是很困难的事情。**

田渊栋：所以大厂一定会做路径依赖，把原来的那条路径走到底。一定会这样。为什么 OpenAI、Google 会往这条路上继续走？**一个是别无选择；另外，这条路不需要花太多脑筋，见效快，只要花时间和精力把原来的事情做得更好，就能把整个问题做得更好，是比较安全的一条路径。**

田渊栋：所以你会发现，人的一些 insight 加上大量数据的生成和训练，确实能让模型变得更强，这个问题不大。只是它最后的收益会有多大的 diminishing return，这其实是一个问题。比如，如果模型再加 **10 倍**算力、**10 倍**数据、**10 倍**人力，能把模型往前提一点点，直到它能变强，他们就会去做。到最后发现 return 越来越少了，大厂才会去想一些别的方法。

曹卿云：**你有没有看到一些新的范式，觉得是比较有希望的?**

田渊栋：现在其实也很难说。最近有一些 learning 的文章，想探索新的范式。还有一些文章在研究，做 reinforcement learning 的时候能不能不用 model weights updates，这些也挺有意思，但都还在探索中，还没有成气候。**可以尝试一下，但是要让它能够 skill up，现在还比较困难。**

曹卿云：刚刚你提到 continuous learning，为什么模型训练完之后，记忆就停留在当时，而没有办法**持续学习**?

田渊栋：主要问题还是因为预训练阶段学习的是大量数据，所以得到的表征，或者说内部的一些知识，它们的学习结构比较特殊。**这个结构能够举一反三，给后面的后训练提供对问题比较好的理解。**

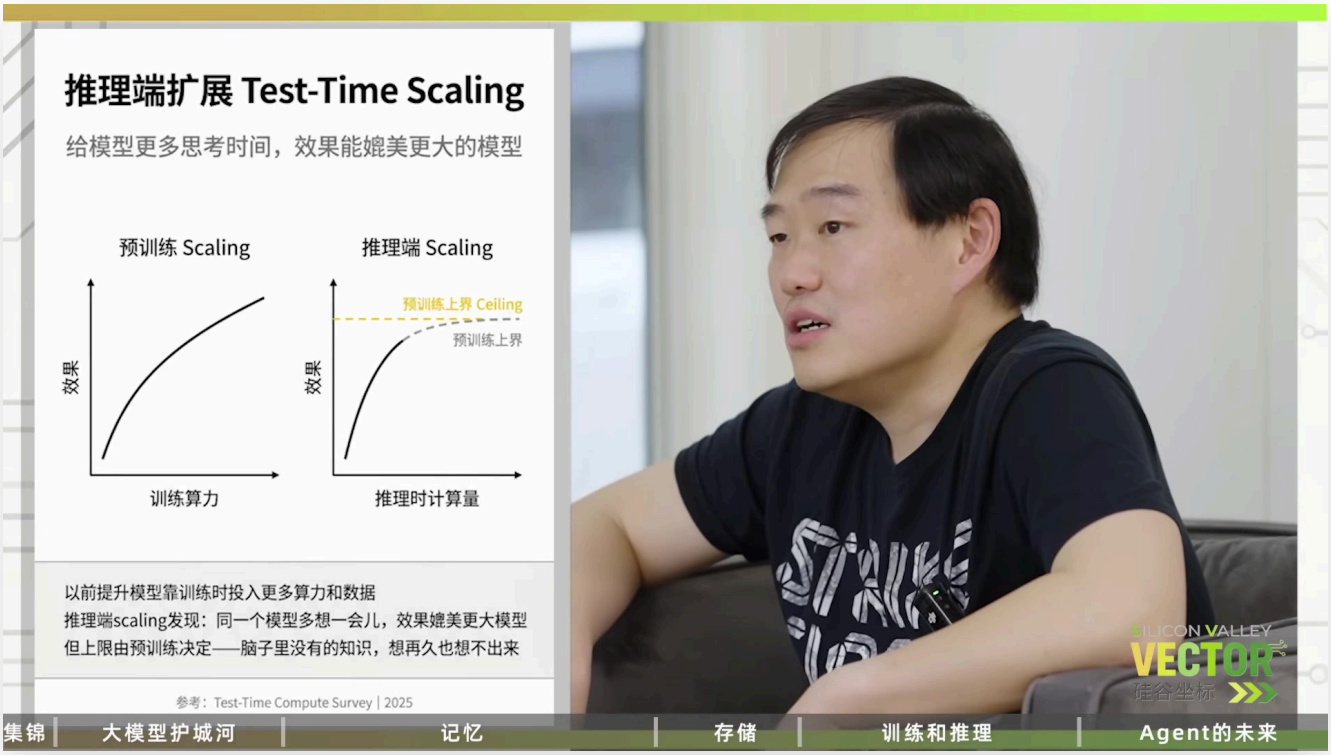
田渊栋：但是后训练，或者说 continuous learning 这条路径的问题在于，它能学到的东西比较有限。**它可能只能学到过去某个单独领域的一些知识，所以泛化能力没有那么强。**可以让它在这个领域里效果还不错，有一些效果，但是要让它泛化到其他地方，可能需要在 at some point 把这些数据再放回预训练里面去，让它回炉重造，这样它的能力才会变得更强。

田渊栋：大概现在就是这样的想法和状态。现在还没有办法做到用很小的样本、很少的数据，一点点积攒资源，然后让模型突然之间产生飞跃性的变化。现在还比较难。**持续学习**肯定是一个愿景，或者说是一个重要的方向，希望能够把这件事情做成。

画面说明

两张访谈近景字幕分别写着“预训练 Scaling Law 的路径依赖”和“新训练范式的探索”；画面没有额外可见图表或产品界面。

09. 推理扩展与隐空间（41:26–50:13）



场景 9

本节主旨 推理扩展与隐空间

完整内容

曹卿云：我们再来讲讲**推理**侧。**推理**侧好像现在也有一个 Scaling Law，是说一个小一点的模型，给它更多思考时间，增加 test-time compute 的时间，也能取得跟大一点的模型差不多的效果。这是以后发展的一个趋势吗?

田渊栋：这应该说是 **2024 年**下半年的时候，o1 刚刚出来。那个时候大家确实很兴奋，觉得 test-time scaling 这条路线不错。横轴变成运行时、推理时的计算资源消耗，纵轴变成推理时的效果。看到这个 scaling law，大家会很兴奋。

田渊栋：我觉得这也是最近一年强化学习能力发展的一个原因。突然之间所有人都去做强化学习推理，做强化学习，做 verifiable reward 这样的方向。**这个方向其实已经做了挺多，但是后来也有一些不同意见：强化学习最终的上界可能是被预训练锁死的，这是一种可能性。**

田渊栋：为什么？因为预训练提供了大量思维素材和思维方式，只是这些思维素材和思维方式在推理过程中被放大了。比如预训练时告诉 AI，这道数学题可能有 100 种方法去想，100 种方法里有些对、有些不对，大部分不对，可能有一个对就可以了。**强化学习通过不停迭代、不停搜索，找到了这 100 种可能性中的一种可行性，并加以放大。**这样经过强化学习之后，模型就能以很快、很高的概率把题解决了。

田渊栋：如果前面的预训练对世界理解不够深刻，对问题了解得不够透彻，它可能永远无法发现后面的解题思路。按照这个逻辑，就很容易解释为什么有人观察到，强化学习在做后训练时，上界是存在的。**有些问题如果前面做不出来，医学里面根本没有这个知识，或者医学知识不够丰富，提不出问题的解决方案，那么用强化学习也做不出来。**

田渊栋：所以强化学习的 Scaling Law 会受制于前面的预训练，受制于预训练对世界表示的上界。比如 **2024 年**下半年看到的 test-time scaling，这个 law 可能相当于前面画一点，之后这样走；还是会这样走，不好说。我有点倾向于认为，再往后 scale 的时候也可能这样走，慢慢出现上界，把模型的能力限制住。

田渊栋：所以现在大家开始做 continuous learning。大家意识到，单纯用后训练的一些推理、不改变模型权重是不够的，希望在后训练推理过程中同时改变模型权重。**这样模型才会对世界有更好的表示，也许这条线能让它走得更快，这就是大家关注方向发生变化的原因。**

曹卿云：你现在看到的推理端，未来 3 到 **5 年**最可能有前景的推理演变路线是什么样？

田渊栋：**首先，我们的推理过程能不能更快、更有效率，这是重要的。**我们有不同的方案。比如能不能让推理链变得更短、更有效率。之前有文章做隐空间推理，我们的推理过程每一步不一定是人类语言，可以是抽象语言，可能用某种高维向量来表示。

田渊栋：用高维向量表示，一个高维向量可能顶一句话，或者顶更长的一段话，让推理变得更有效率，用更快的速度把推理找出来，这是一种做法。**另外我们也证明，隐空间推理的一个好处是，推理向量里面可以同时存几个不同的推理路径，所以效率可能比正常的语言推理更高。**

田渊栋：为什么？可以认为隐空间推理的向量相当于量子力学的一个叠加态。叠加的推理路径可以同时处理很多不同的探索路径，通过这个方式让推理效率更高。这是一种可能的做法。**最近也看到很多组在这个方向做探索，看看隐空间推理能提高多少效率。**

田渊栋：另外一条路是最近比较火的 parallel thinking，也就是平行推理。**这里面还是用语言做推理，但是希望在某种情况下，语言推理能够并行进行，而不是串行进行。**比如你跟大家说，接下来有 1、2、3、4、5 五点要求，领导讲话时会这么说。有五点要求之后，这五点要求可以分别自己做推理。一旦五点要求的提纲列出来，这五点要求本身的推理过程就可以自动进行。

田渊栋：这五个要求可以同时进行，而不需要按顺序进行。**这样可以利用更多计算资源做推理，不需要等前面的推理结束之后再做下一步推理，会更快。**

田渊栋：当然还有一些方法，比如怎么样让推理链变得更加扎实、更加短，同时抓住本质；或者做推理时，怎么样很快去掉不必要的、明显错误的推理链条，让推理过程变快。**我们这边有一篇文章叫 DeepComf，通过这个方式能够降低推理的 token 使用量，同时效果还更好一点。**

曹卿云：明白。**那 Coconut 它不是并行的，还是串行的?**

田渊栋：**它还是串行的，只是每一步推理的结构、里面的表示不是语言，而是 latent vector，也就是隐空间变量。**

曹卿云：**这个对 token 的消耗是不是更大了?**

田渊栋：从长度上来说，它是变短了。当然还有其他消耗。以前要存一个 token ID，现在要存一个整个 vector，这样对**存储**的消耗会变大。你会发现设计各种算法时有这种 trade-off：一方面变强，另一方面变弱，怎么样平衡是一个问题。

曹卿云：**还想问你一个 hallucination，也就是幻觉的问题。**你觉得幻觉问题最后会怎么解决?

田渊栋：幻觉问题应该说是根植于模型本身的结构。在训练过程中，模型权重除了学到一些有益的结构之外，可能还学到一些没有意义的东西。**这些没有意义的东西在训练数据集上没有作用，但是如果测试时输入的数据超过了数据集本身分布的边界，这些没有学好的权重方向就可能影响它。**

田渊栋：用数学语言来说，权重本身有两部分：一部分是有 signal 的子空间，另一部分叫 null space，也就是没有信号的子空间。但是没有信号子空间里面，权重本身也有一部分分量在那里。**这部分分量平时没有打扰正常推理，但是如果输入数据跟训练时的推理过程不一样，这部分权重就可能**

影响你。

田渊栋：如果要比较好地解决这个问题，最终还是要打开大模型的黑箱，知道里面的权重是怎么 work 的，这样可能能更好地解决这个问题。

曹卿云：今年小龙虾 bot 出来的时候，你当时还在朋友圈发了一个帖子。你用了，感觉怎么样?

田渊栋：我其实就用了两小时，也没用太多。**因为当时我觉得，装了之后我越装心越虚。**

画面说明

一帧图示“**推理**端扩展 **Test-Time Scaling**”，对比预训练 Scaling 与**推理**端 Scaling 曲线，并标出“预训练上界”；另一帧图示“隐空间推理 Coconut”，对比传统 A→B→C→D 的串行路径与同时存在的多条路径。

10. Agent 安全与协同（50:14–55:45）

场景 10

本节主旨 Agent 安全与协同

完整内容

田渊栋：因为它必须要我把所有各种 API 的 key 都交出来，所以让我觉得有点慌。我就在网上发了一个帖子：我与其用小龙虾，不如用我现在 AI coding 的方法，自己写一个，或者写一些更加专一的工具，而不是依赖小龙虾，让它去做任何事情。这样的话我心里不是很踏实，万一它做什么坏事，我就不知道，这是一个安全性的问题。

田渊栋：现在小龙虾变成这样的模式。我也跟很多人说过，它相当于我有一个小孩子。这个小孩子是 Agent，手上握有我所有的秘密，然后到外面去跟各种人聊天，帮我把事情做完。但是它智商又不够高，所以有可能被人骗。

田渊栋：比如小孩子跑到菜市场，别人跟他说：“哎呀，你这小孩子真乖呀。**你能不能，我给你五块钱，你能不能把你家地址告诉我？**”小孩子可能就把什么东西告诉他了，接下来你家可能晚上就被人敲开门，进来偷东西。现在就是这样的状态。

田渊栋：这个小孩子脑子里可能有你所有的密码、你的 OpenAI key、Anthropic 的所有 key。这个还算好，因为如果这些东西被偷了，你可以通过某种方式把原来的 key 删除掉，这样就没事了。**但还有其他东西，比如 Google 邮箱的一些 access，或者机密文件的密码，或者能够 access 到重要 folder 的 token。**

田渊栋：**这些东西非常重要，也非常危险。**甚至你可以把别人存 password 的这个密码告诉别人，一看里面什么密码都知道了。有这个问题之后，你并不能保证这个小孩子不会被人骗走。网上有各种方式来骗，而且有专门的平台把这些小孩放在一起，让他们相互讨论，找到一个骗大人的方法。你可以认为它存在这样的问题。

田渊栋：**所以你可以用它，但是要特别注意它里面的东西。**最好的办法是一边用，一边去想它里面是怎么 work 的。我非常建议看它的代码，这样你会理解得更深刻，也许你自己可以做一个。

田渊栋：我们看到 **Agent** 越来越成为大家工作流的伙伴。你觉得这些 Agent 未来会对科技组织的组织架构，包括大家的协同方式，有什么影响?

田渊栋：以后很多基本的常识、基本的知识，或者一个人与人之间交流的东西，都可以让 Agent 来完成。**比如本来我要跟你面对面聊天，本来会跟你说我们什么时候吃个饭，或者什么时候见面、安排 meeting schedule，可能本来需要一个秘书相互对接。**现在我有 Agent，你有 Agent，你们俩对接一下，就可以把 meeting 或者其他事务性工作做完了。

田渊栋：**这是一个很大的变化。**这个趋势其实我一年多以前就看到了。当时在 Meta，我写了一个 proposal，叫 Omni Agent。我说将来可能是人和人之间的交流从 A 点来完成。当时觉得这件事可能会在五年内发生，但没想到这么快，就已经发生了。

田渊栋：以后可能会有这样的疑问：我们要不要上街购物？**要不要浏览 Amazon 的网站？**要不要去网上看各种东西？很多事务性的搜索，不是出于经验性或娱乐性的搜索，可能都可以用 AI 代替。为什么要花两个小时浏览网站找到一个东西再买？让它帮我买就行了。

田渊栋：网上已经有人跟我说，他用 Agent、用小龙虾来帮他买东西，觉得很开心、很喜欢。因为 Agent 知道他的所有 preference，当然买东西会非常好。**对小龙虾来说，它不需要花很多时间浏览网页，所有网页都是一个连接，它马上就看完了，然后找到你想要的东西。**

田渊栋：**所以这个过程应该说颠覆了整个电商逻辑，或者整个的人与人之间的交互逻辑，以后会有很大的影响力。**以后也许网站做得再花哨都没有用了。很多东西是吸引用户点击的，比如广告条，希望你点进去；或者一个很闪亮的东西、一个特价，人会看到它，然后有点进去的欲望。

田渊栋：**但是对于小龙虾、对于 Agent 来说，它没有欲望。**它的任务是把事情做成，希望找到最好的 deal，所以这些广告对它没有用。这样整个逻辑可能就不同了。具体怎么不同，现在还在演进中，但是会很有意思。

曹卿云：**所以你觉得个人助理以后会越来越强大？**比如我们以后可能就是一个 super app，包括衣食住行的需求，打车之类的都包括在里面，可能不用 Uber 这样的平台经济，而是直接用一个 super app 解决所有事情。你是这么看吗？

田渊栋：这是有可能的。**比如以前你要——**

画面说明

一帧图示“**Prompt Injection** 提示词注入攻击”，展示用户让 **Agent** 浏览网页、网页隐藏恶意指令、**Agent** 执行指令并将信息交给攻击者的流程；另一帧字幕写着“让 Agent 帮你做事前，先搞清楚它的手里有什么”。

11. Agent、就业与下一站（00:55:45–01:01:49）

场景 11

本节主旨 Agent、就业与下一站

完整内容

田渊栋：以前你要打电话约水管工，或打电话约银行，打电话约一些人，然后把事情解决了。如果这些人每个人手里都有一个 Agent，就没有必要打电话给他了。你可以让我们的 Agent 24 小时蹲在网上，只要有任何需求或者任何信息过来，它马上告诉你，马上达成协议，然后完成。所以这个效率远远高于打电话。

田渊栋：这件事一定会发生。**事务性的工作，一个是我们其实并不需要这样的经验，我们希望把这个经验交给别人，让他很快完成；我希望更多时间花在我想要的经验上面。**所以这个事情一定会发生。

田渊栋：另外，它有一个裹挟效应。如果世界上的人都用这个了，你不用就会被 kick out。**比如一个水管工说自己平时只接电话，但是其他人都开始用 bot 了。**这个 bot 可以 24 小时蹲在网上给你拉生意，还会自动把这些生意组织起来，变成一个很好的路线图让你去走。比如今天要走访五家，这五家在哪里，怎么样开车最快走一遍，这个过程都可以让 AI 自动完成。

田渊栋：如果你不用，你的效率就低于其他同行，就会被淘汰。最终通过这个方式，大家都会不得不做这件事，不管主动还是被动。有些人主动说要提高效率，有些人是被动的；如果你不做，别人做了，你就会被淘汰。**大家都会进去，所以这是一个很快的过程。**

曹卿云：**按照你的描述，是不是失业才刚刚开始，未来还会有更多?**

田渊栋：我在年终总结里面说过，我觉得现在这个变化非常大，只是洪水马上来了，很多人还没有感觉到。**因为很多人可能不是 AI 从业者，所以不知道会发生什么事情。**很多非 AI 从业者一直觉得岁月静好，突然有一天像地震一样，突然发生大事情，发现自己被裁了。

田渊栋：这个被裁不像以前那样：我跟老板有矛盾，或者我在这家公司做得不好，去另外一家公司还能找到工作。不是这样。**那个时候你会突然发现，全行业的逻辑变了，逻辑和思路跟以前不一样了，所以你的技能在任何地方都没有用了。**

田渊栋：这是很可怕的事情。我只是在洪水来之前给大家说一下，这件事会发生，只是大家不一定会听。**直到某一点，大家突然说，原来世界发生变化了，但变化已经发生了，就是这样的逻辑。**

曹卿云：**你会怎么教育下一代?**

田渊栋：对我来说，我们还是希望他能做一些他想做的事情。因为下一代，比如再过 **20 年**，会是什么样子？很难想象。现在想象力已经落后于发展的速度了，这跟以前不同。以前我写科幻小说，会想这样的想法可能五十年之内都不能发生，我就慢慢写，没关系。现在倒过来了：这个 idea 如果你再不写，就没有了，因为已经发生了，不会成为将来的科幻，而会成为过去的历史。

田渊栋：所以这个速度在不断演进，非常快。二十年之后发生什么真的不知道。我觉得能预测的可能是，人还是会在，而人最重要的是目的性，还有经验。很多事情跟目的性绑定，这部分机器不能替代。**因为机器替代之后，就意味着这东西不再是你的了。**

田渊栋：这部分应该是比较长久的东西。**比如要写一部小说，或者完成一部作品，或者一些艺术家的个人目的，达成一个结果。**这部分的缘起和整个设计，本身是艺术家通过自己的内心经历产生一个冲动，这个冲动变成一个作品。

田渊栋：这个冲动、目的和想法，是人类独有的。**所谓人类独有，不是说它不能被 AI 取代，而是说这部分被 AI 取代之后，这个作品就没有意义了。**作品的意义在哪里？在人怎么把这个意义写下来，作为自己的动机，把这件事情做成。

田渊栋：这部分是机器跟人很大的区别。**所以最终教育孩子或者教育下一代，应该是希望他有很大的动力，让他能够做自己想做的事情。**有动力的话，学习过程就会变得非常愉快，他也愿意用所有工具完成他想要做的事情。我觉得这很重要。

曹卿云：**你怎么看待现在 Agent 和大模型的发展？**大家很担心，如果 Agent 创业，想做的创业方向很快就被大模型的能力侵蚀掉。你怎么看待这些创业者？

田渊栋：一个当然是速度要快。**另外，如果有客户，客户本身会对你产生粘性；客户本身有些数据，这些数据可能变成你的护城河，这是一个重要的点。**

田渊栋：要不就做一些很快的项目，项目速度快于大模型的发展速度；要不就做一些很难的问题，这个问题现在大模型解决不了。**这两个都是有价值的，我觉得是这样子。**

田渊栋：最后，你问我下一站去哪里。我会去一个 startup 做 cofounder。**当然，我们现在的名字和方向暂时不能公开，因为现在还在融资，我们正在融 Series A，马上就要结束了，整体还是挺好的，应该说挺顺利，很多人愿意投。**

田渊栋：大概是这样。不过具体方向还有人员组成，我们暂时保密。**希望之后在一个关键的时间点可以宣布。**

曹卿云：**非常期待知道你下一步是什么样的走向。**

画面说明

画面是主持人曹卿云的近景，字幕写着“教育下一代做想做的事，工具由 AI 来完成”；底部栏目可见“**大模型护城河**”“记忆”“**存储**”“训练和**推理**”“**Agent** 的未来”。