FRONTIER TALK · 深度学习讲义

不要只朝目标走

开放式演化、质量多样性与 AI 生成算法

从 MAP-Elites、POET、OMNI 到自我改进 Agent 与 AI Scientist

讲者	Jeff Clune
视频来源	Bilibili「至高机器智能」· BV1Hzjg65EZd
内容长度	30:01 · 720 条带时间戳中文字幕
当前版本	不带视频画面版：零封面、零关键帧、零截图；全部机制图原创绘制
整理	cnfjlhj & Codex · 2026-08-12

这是一份基于字幕证据的教学重构，不是逐字稿，也不是视频幻灯片复刻。

怎样使用这份讲义

阅读说明与证据边界

本讲义的内容层级被刻意分成三种：

1. **内容复原**：按 30 分钟、720 条带时间戳中文字幕，恢复讲者的论证、例子、转折和结尾问题；
2. **机制解释**：为学习目的增加公式、伪代码式拆解与原创图。这些是讲义重构，不冒充视频原始内容；
3. **边界校准**：对字幕噪声、愿景性判断、缺少实验预算/误差条的强主张明确降格。

依照本次的默认模式，讲义**没有使用视频封面、关键帧、截图、storyboard 或画面 OCR**。所以它能精确复原口头主线，却不能声称复刻讲者的每一页幻灯片。视频链接仅用于来源导航：https://www.bilibili.com/video/BV1Hzjg65EZd/。

先抓住全片的一句话

真正困难的创新，不是把一个固定分数一路爬到最高，而是维护一个**多样且可复用的发现档案**，让系统不断产生新解、新任务和新环境，并允许后来才显现价值的**垫脚石（stepping stones）**继续参与搜索。

阅读路线

1. 第 1–2 章回答：为什么单一目标会欺骗搜索？archive 为什么比一个冠军更重要？
2. 第 3–4 章回答：MAP-Elites 与 POET 怎样把“多样解”升级为“共同演化的任务生态”？
3. 第 5–7 章回答：foundation model 如何成为 interestingness 判别器、奖励代码生成器与环境生成器？
4. 第 8–9 章回答：怎样把同一循环扩展到 agent scaffold、记忆系统和科学研究流程？
5. 第 10 章专门做能力边界审计，防止把研究愿景误读为已实现的 AGI/ASI。

目录

怎样使用这份讲义 ii

第一章 为什么“朝目标走”反而会失败 1

1.1 迷宫：距离不是可达性 . . . 1

1.2 微波炉、计算机与延迟显现的价值 . . . 1

1.2.1 本章小结 . . . 2

第二章 创新引擎：从一个冠军到一座档案馆 3

2.1 自然演化与人类文化的共同配方 . . . 3

2.2 三种算法到底分别扩张什么 . . . 3

2.2.1 本章小结 . . . 4

第三章 MAP-Elites：每一种类型都保留最优解 5

3.1 把“属于哪一类”与“这一类里多好”分开 . . . 5

3.2 为什么它不是“质量与多样性各打一个分” . . . 5

3.3 损伤适应与 Go-Explore：archive 的两种用法 . . . 6

3.3.1 本章小结 . . . 6

第四章 从固定地图到开放生态：POET 7

4.1 QD 的天花板：环境还是固定的 . . . 7

4.2 POET：环境与 agent 成对共同演化 . . . 7

4.2.1 本章小结 . . . 8

第五章 AI 生成算法：让“设计 AI”本身被学习 9

5.1 从手工模块到元搜索 . . . 9

5.2 三个互相卡住的层级 . . . 9

5.2.1 本章小结 . . . 10

第六章 OMNI：怎样找到“值得学习”的任务 11

6.1 巨大任务空间为何反而更难 . . . 11

6.2 Foundation model 作为 interestingness 代理 . . . 11

6.3 自然语言任务必须落到可执行奖励 . . . 12

6.3.1 本章小结 . . . 12

第七章 从 OMNI 到 OMNI-EPIC：让代码生成环境 13

7.1 Darwinian-complete 是表达能力，不是完成度 . . . 13

7.2 现实系统仍被算力和样本效率卡住 . . . 13

7.2.1 本章小结 14

第八章 自动设计 Agent：从档案搜索到自我修改 15

8.1 搜索对象不再只是模型权重 15

8.2 Darwin–Gödel Machine：开放档案 + 自修改 15

8.3 HyperAgents：迁移“如何改进自己”的知识 16

8.3.1 本章小结 16

第九章 记忆与科学自动化：ALMA 与 AI Scientist 17

9.1 ALMA：把记忆架构也变成搜索对象 17

9.2 AI Scientist：把循环扩展到科研生产线 17

9.2.1 本章小结 18

第十章 统一框架：生成、选择、档案与目标切换 19

10.1 这些项目为什么其实在做同一件事 19

10.2 从工程视角重写 Clune 的结尾问题 19

10.2.1 本章小结 20

第十一章 边界审计：这场演讲不能证明什么 21

11.1 六个必须拆开的命题 21

11.2 怎样验证一个真正的 stepping stone 21

11.2.1 本章小结 22

第十二章 总结、术语表与复习路线 23

12.1 一页压缩 23

12.2 核心术语表 23

12.3 30 分钟时间轴 24

12.4 复习题 24

12.5 最后带走的五句话 25

来源说明 26

第一章　为什么“朝目标走”反而会失败

1.1　迷宫：距离不是可达性

视频字幕 00:00–00:52

讲者开场给出一个故意反直觉的命题：对极难问题，过度执着于最终目标可能让搜索失败；暂时忽略它、转而探索新区域，反而更容易抵达目标。迷宫是最小反例：agent 位于左下角，奖励函数只鼓励它缩短与终点的几何距离。最陡的方向指向北方，但北方是一堵墙，于是优化器一次次做“按照指标完全正确、按照真实任务毫无用处”的事。

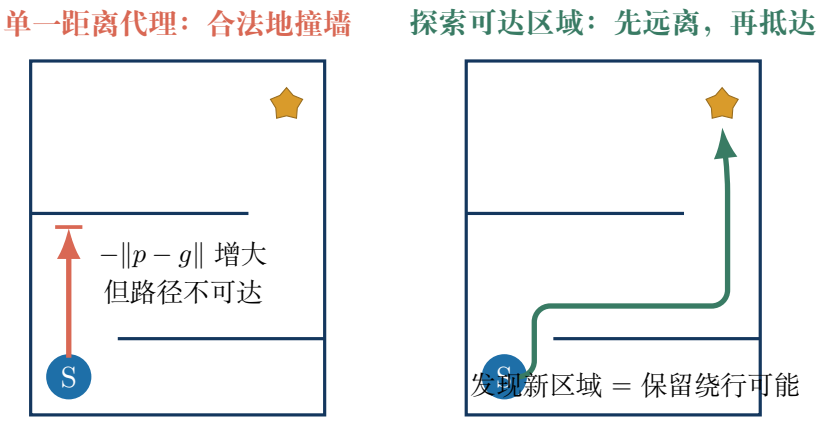


图 1.1：距离最短与路径可达是两种不同的信息。原创教学重绘；非视频截图

若位置为 p、目标为 g，一个常见代理奖励可以写成

$$r_{\text{proxy}}(p) = -\|p - g\|_2.$$

它只衡量**几何接近**，没有表达墙、门、钥匙或通道等**拓扑可达性**。因此失败不是优化器“没优化好”，而是代理目标忠实地把它导向了错误山峰。

别把它听成“目标没有用”

视频的迷宫只证明：当局部代理指标与真实任务结构错位时，强力优化会放大错位。它不证明所有目标导向搜索都失败，也不证明 novelty search 在任何任务上都更优。正确结论是：**不要让一个过早、局部、单一的分数，消灭所有暂时不涨分的可达路径。**

1.2　微波炉、计算机与延迟显现的价值

视频字幕 00:52–02:34

讲者随后把迷宫从玩具问题扩展到创新史。他用两组故事说明：

- 若唯一目标一直是“更快烹饪、烟更少”，雷达研究与巧克力融化的偶然观察不会被纳入路线，微波炉就不在当前目标的直线上；
- 若算盘时代只奖励“更多算术运算”，真空管与电力研究也不会因为当下算盘分数而得到保存，但

它们后来成为现代计算的关键技术条件。

双足机器人例子更贴近算法：我们本想提高前进速度，却观察到某个候选会爬行，另一个会单脚平衡。单看即时速度，它们可能都应被淘汰；从能力图谱看，它们却可能成为稳定行走、受损恢复或复杂运动的中间技能。

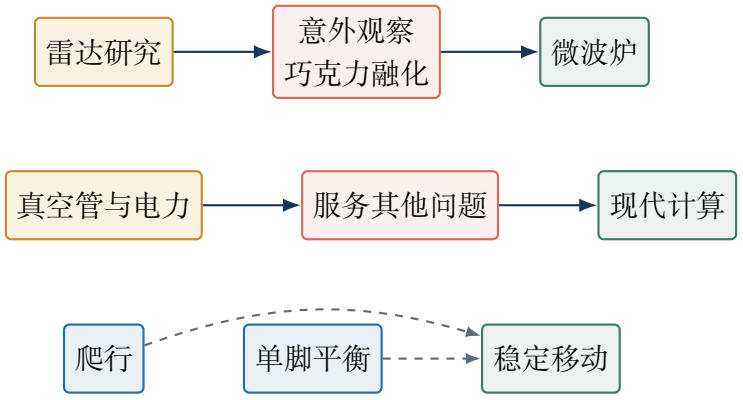


图 1.2: stepping stone 往往来自另一个问题，而不是最终目标的直线缩小版。原创教学重绘；非视频截图

教学解释：目标切换不是“见异思迁”

Goal switching 的精确含义是：当搜索在任务 A 上不涨分、却在任务 B 上形成了可验证的新能力时，允许 B 暂时成为新的局部目标，并把结果归档。它是机会主义的课程生成，不是把任何噪声都当成果。

1.2.1 本章小结

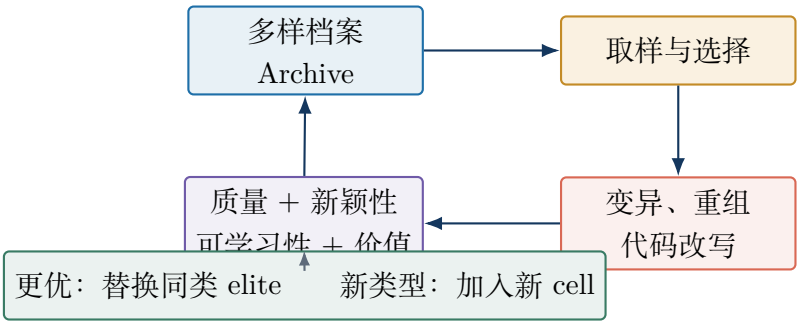
- 代理指标可以在数学上被完美优化，却在真实任务上失败；
- 困难创新依赖价值延迟显现的中间结构；
- 需要改变的不是“有没有目标”，而是**目标是否唯一、是否过早、能否切换、是否保存旁支能力**。

第二章　创新引擎：从一个冠军到一座档案馆

2.1　自然演化与人类文化的共同配方

视频字幕 02:34–03:35

讲者把自然演化与人类文化都抽象为**创新引擎（innovation engine）**。它们的状态不是“当前唯一最好对象”，而是一个装着物种、理论、证据、技术与设计的集合。系统从集合取样，修改、变异、重组，再问两个不同的问题：它是否在某一类里更好？它是否创造了此前没有的类型？



循环的产物不是单点最优，而是一组可再次组合的 stepping stones

图 2.1：创新引擎的最小循环。原创教学重绘；非视频截图

Archive 的核心作用

Archive 不只是备份。它改变了未来能从哪里出发：一个今天不是冠军的对象，只要足够不同、可复用或在某一类中表现好，就可能成为明天进入新区域的父代。

2.2　三种算法到底分别扩张什么

视频字幕 03:35–04:15

讲者把路线分为三层。它们共享“生成–评估–档案”的骨架，但扩张对象不同：

表 2.1: 三层路线的搜索对象与输出

路线	主要扩张对象	希望得到的结果
Quality-Diversity	固定任务中的行为类型与高质量解	每一种行为类型中的 elite，而非一个全局冠军
Open-Ended Algorithms	环境、任务、生态位、课程及解	新解不断创造新挑战，任务空间随运行而增长
AI-Generating Algorithms	架构、训练算法、目标、环境、agentic system	让“制造更好 AI 的流程”本身由 AI 学习和改写

注意：这三层不是互斥产品名，而是递进的设计问题。QD 先解决“不要只留一个赢家”；开放式算法进一步问“为什么任务本身不能增长”；AI-GA 再问“为什么搜索者、训练方法和科研流程不能一起被学习”。

2.2.1 本章小结

- 创新引擎必须同时有 generation、selection 与 archive；
- 新颖性不是装饰，它给延迟显现的价值留下存活空间；
- 三类算法的本质差异，是被允许演化的对象范围越来越大。

第三章　MAP-Elites：每一种类型都保留最优解

3.1　把“属于哪一类”与“这一类里多好”分开

视频字幕 04:15–06:22

MAP-Elites 先定义一个**行为描述符** $b(x)$，例如机器人的身高、体重、步态频率或接触模式。描述符决定候选 x 属于哪个 cell；另一个函数 $Q(x)$ 衡量它在该 cell 内的质量。二者承担完全不同的角色：

$$c = b(x), \qquad e_c \leftarrow \begin{cases} x, & c \text{ 为空}, \\ x, & Q(x) > Q(e_c), \\ e_c, & \text{其他情况}. \end{cases}$$

其中：

- x 是新候选；$b(x)$ 把它映射到行为单元 c；
- e_c 是该 cell 当前保存的 elite；
- Q 只比较同一行为类型内谁更好；
- 空 cell 被填充，已有 cell 只有遇到更优同类才替换。

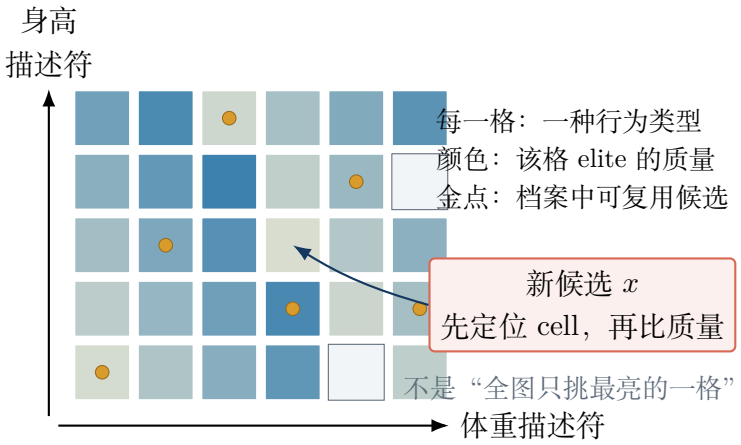


图 3.1： MAP-Elites 的二维行为档案示意。原创教学重绘；非视频截图

3.2　为什么它不是“质量与多样性各打一个分”

若把质量与新颖性压成一个标量，例如 $Q(x) + \lambda N(x)$，超参数 λ 决定了系统愿意为多样性牺牲多少质量。MAP-Elites 采取另一种结构：**先分类型，再在每类里争冠军**。于是系统不需要用同一条折中线比较一只矮而灵巧的机器人与一只高而稳定的机器人。

这带来三种学习价值：

1. **覆盖**：看见哪些行为区域已经有解，哪些仍为空白；
2. **复用**：环境变化或机器人受损时，从不同 cell 取替代方案；
3. **谱系**：回溯最终 elite 的祖先，观察它经过哪些看似绕路的区域。

Descriptor 是 MAP-Elites 的命门

如果描述符不稳定、与真实行为无关，或者偷偷把最终性能本身当作多样性维度，档案就只是漂亮的网格。好的 descriptor 应表达可区分、可复现、对未来复用有意义的行为差异，而且不能泄漏未来答案。

3.3 损伤适应与 Go-Explore：archive 的两种用法

视频字幕 06:22–08:18

视频报告了两个应用方向。其一是在模拟中预先生成大量不同步态；机器人受损后，不必从零重新学习全部控制，而是从行为档案里试出仍可工作的步态。其二是 Go-Explore：保存已经到达的状态与路径，回到有潜力的状态后再继续探索，避免每次都从起点穿越漫长的无奖励区。

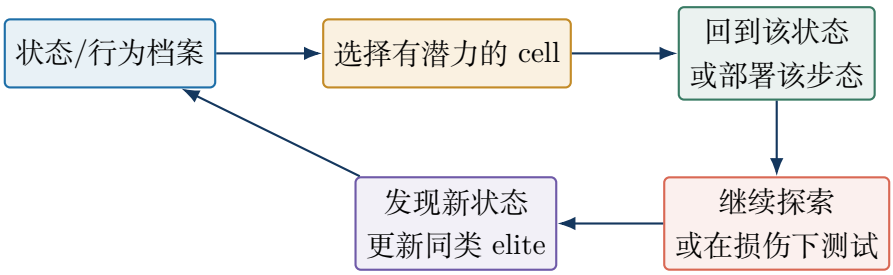


图 3.2: 同一 archive 原理可支持“从中间状态继续探索”或“从备选行为中恢复”。原创教学重绘；非视频截图

讲者的强口径要降格阅读

字幕中出现“彻底解决 Atari 探索难题”“打破人类世界纪录”等强表述。本讲义只保留为“讲者在所述基准上报告突破”，不把它扩张成“所有强化学习探索问题均已解决”。具体游戏集合、协议、预算与结果需要回到原论文核验。

3.3.1 本章小结

- behavior descriptor 回答“属于哪类”，quality 回答“同类中谁更好”；
- MAP-Elites 的目标不是一条 Pareto 折中线，而是整张地图的逐格精英；
- archive 使系统能覆盖、复用、恢复，并保留迂回谱系。

第四章　从固定地图到开放生态：POET

4.1　QD 的天花板：环境还是固定的

视频字幕 08:18–09:41

无论 MAP-Elites 运行多久，只要任务表示被锁死，它最多得到更多种、更好的机器人，不会突然创造一种新的数学问题或开始改进 AI 本身。讲者由此把开放性定义为更强的要求：系统产生的新解还要改变后续问题空间，像叶片创造食草生态位、食草动物又创造捕食、合作与寄生关系。

开放性不是“随机生成得没有尽头”

一个真正有意义的 open-ended process 必须满足：新产物不仅不同，还会创造新的可学习挑战、可复用能力或评价维度。否则“无限”只是无界噪声，而不是持续创新。

4.2　POET：环境与 agent 成对共同演化

视频字幕 09:41–11:53

POET（*Paired Open-Ended Trailblazer*）把环境也放进档案。简化后的循环是：

1. 从环境库选择一个环境并轻微变异；
2. 用现有 agent 集评估新环境；
3. 只有当环境**不是太简单**、**不是完全不可解**，并且与已有环境足够不同，才加入档案；
4. 为环境训练或迁移 agent；
5. 允许某个环境中学到的策略跳到另一个环境，形成非线性课程。

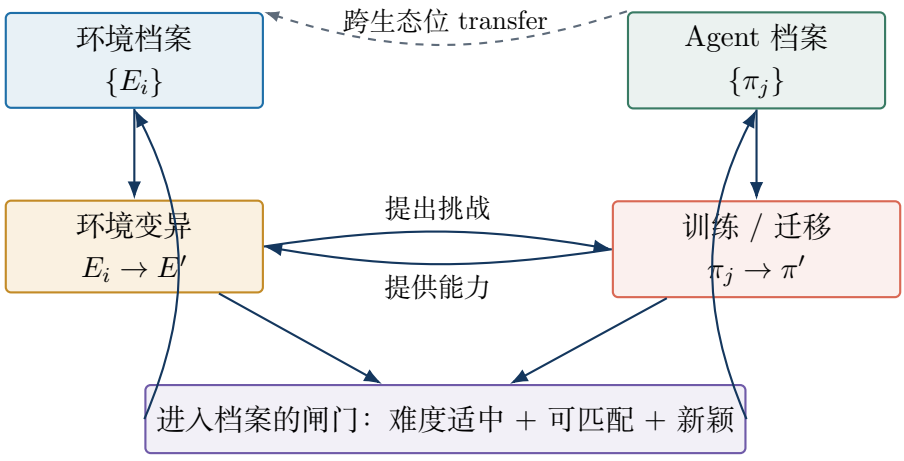


图 4.1: POET 的关键是“任务库 × agent 库”的耦合，而不是单任务难度递增。原创教学重绘；非视频截图

视频强调，一些困难障碍课程无法通过直接训练解决，人类手工安排的单线课程也会失败；最终成功路径依赖 agent 在多个环境间迁移，经过事先无法规划的 stepping stones。

表示空间决定开放性的上限

视频随即承认：POET 演示仍局限于各种障碍赛道。环境数量变多，不等于领域变多。如果环境生成器只能表达赛道，再长时间的开放循环也不会生成逻辑题、实验设计或记忆系统。这一瓶颈直接引出后面的 OMNI 与代码生成环境。

4.2.1 本章小结

- QD 扩张固定任务内的解，POET 同时扩张环境与解；
- 难度适中、新颖性和跨环境迁移共同构成自然课程；
- 真正的开放性仍受环境表示语言限制。

第五章　AI 生成算法：让“设计 AI”本身被学习

5.1　从手工模块到元搜索

视频字幕 11:53–13:57

讲者观察到一个长期趋势：随着数据与计算增加，手工设计的环节常被学习型环节替代。他据此提出更激进的研究纲领：既然 AI 的架构、训练算法、学习目标、环境与 benchmark 都是人设计的，为什么不让 AI 学习如何产生和改进这些对象？

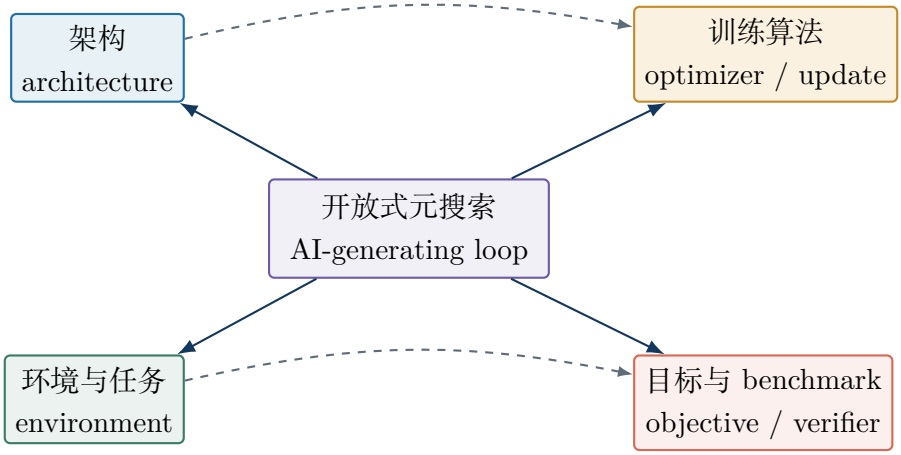


图 5.1: AI-GA 把“AI 系统的哪些部分可被生成”从单一架构扩展到整个训练闭环。原创教学重绘；非视频截图

讲者用自然进化的长期过程作可行性类比：在巨大时间和计算规模下，演化最终产生了样本效率很高的人类智能。他希望复制这类开放生成过程，并通过人工计算平台加速。

类比不是 AGI 证据

“自然演化产生人类”只能说明开放生成过程**可能**产生复杂智能，不能证明某个当前算法在可承受预算内会得到 AGI/ASI，也不能推出“计算越多就必然持续变强”。这里是 Clune 的路线判断，而不是实验结论。

5.2　三个互相卡住的层级

AI-GA 的难点不是“让 LLM 多写几段代码”，而是三层必须闭合：

1. **任务层**：生成什么任务，怎样判定它可学、有趣、不重复？
2. **agent 层**：修改 prompt、工具、计划、记忆或代码后，怎样评价是否真正改进？
3. **元层**：谁来改进任务生成器、evaluator 和自我改进算子本身？

如果只有第一层，系统可能批量生成无意义任务；只有第二层，agent 会过拟合固定 benchmark；元层没有独立约束，生成器与 evaluator 还可能共同漂移。

5.2.1 本章小结

- AI-GA 把“手工设计 AI”改写为“搜索制造 AI 的过程”；
- 可生成对象包括架构、训练算法、环境、目标与 evaluator；
- 这是研究纲领，成败取决于任务、评价和元改进闭环，而不是一句“让 AI 自我改进”。

第六章　OMNI：怎样找到“值得学习”的任务

6.1　巨大任务空间为何反而更难

视频字幕 13:57–15:13

当搜索空间扩展到自然语言或任意环境时，绝大多数候选不是好课程：有的太简单，agent 已经会；有的太难，长期没有学习信号；有的虽可执行却没有意义；还有的只是旧任务换皮。均匀采样等于在任务海洋里捞针。

一个常见信号是**学习进度（learning progress）**。讲义可用下式表达其直觉：

$$LP_t(\tau) = \left|\overline{R}_t(\tau) - \overline{R}_{t-\Delta}(\tau)\right|,$$

其中 τ 是任务，$\overline{R}_t$ 是当前窗口的平均表现，Δ 是比较窗口。若表现正在变化，任务可能处在“可学但尚未学完”的甜区。

Learning progress 只回答一半问题

高学习进度说明 agent 正在发生变化，不说明任务具有长期价值。一个无聊、重复甚至奖励可被钻空子的任务，也可能短期进步很快；反之，一个重要但反馈稀疏的任务可能暂时没有进度。

6.2　Foundation model 作为 interestingness 代理

视频字幕 15:13–16:08

OMNI 的转折是：与其由研究者手写“什么叫有趣”，不如查询在互联网文本上预训练的 foundation model，让它近似判断候选任务是否新颖、是否与已掌握任务显著不同、是否符合人类常识中的“值得注意”。然后把这个判断与 learning progress 合并，持续生成课程。

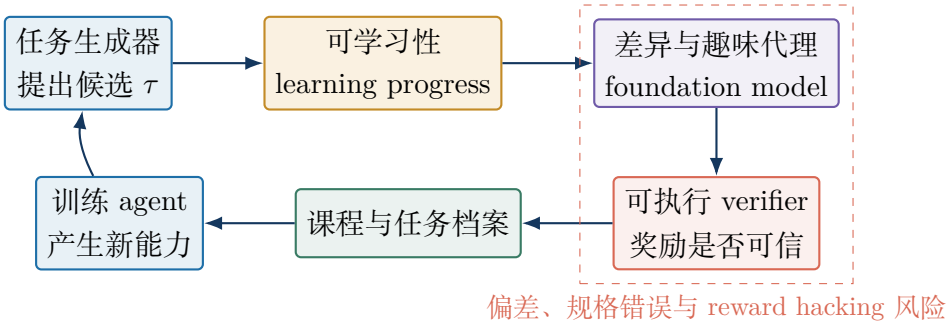


图 6.1： OMNI 不是只让模型“想点子”，而是让多种筛选信号形成闭环。原创教学重绘；非视频截图

“有趣”在这里是模型化的人类观念

Foundation model 只是一个被训练数据和提示词塑造的判别器。它可能偏好语言上新奇的任务、复制互联网偏见，或把“不同表述”误判为“不同能力”。因此 interestingness 必须被当作待检验代理，而不是客观价值函数。

6.3 自然语言任务必须落到可执行奖励

视频字幕 16:08–17:46

视频用模拟厨房举例：系统可以提出“把烤面包机倒置放在水壶上”。但一句自然语言不是训练信号；系统还必须自动判断任务是否完成。当时视频模型效果不佳，项目让 GPT 编写奖励函数，并读取模拟器状态来验证物体关系。

由此形成一条不可缺的链：

$$\text{自然语言任务} \longrightarrow \text{可执行 verifier} \longrightarrow \text{状态查询} \longrightarrow \text{奖励} \longrightarrow \text{策略更新}.$$

OMNI 由此能构造类似“找到苹果 → 识别刀 → 拿起刀 → 切苹果”的课程。关键不是顺序看起来合理，而是每一步都处在 agent 的可学习区间，并为后续任务提供能力。

自动写奖励不等于自动写对规格

代码能运行，只说明 verifier 可执行；不说明它准确表达自然语言意图。状态遗漏、边界条件、代理钻空子和 evaluator 偏差都可能让系统在错误奖励上高速进步。

6.3.1 本章小结

- 巨大任务空间的瓶颈从“能否生成”转为“哪些值得训练”；
- learning progress、interestingness 与可执行 verifier 分别回答不同问题；
- foundation model 给出可用的人类趣味代理，但不拥有客观价值判断。

第七章　从 OMNI 到 OMNI-EPIC：让代码生成环境

7.1　Darwinian-complete 是表达能力，不是完成度

视频字幕 17:46–18:51

只要 OMNI 仍依赖固定厨房模拟器，它的任务都被厨房的对象、动作和物理规则限制。视频进一步提出 **Darwinian-complete search space**：让系统直接生成环境代码与奖励代码。由于通用编程语言可以表达任意可计算过程，任务族原则上可以跨越物理控制、逻辑、编程、问答乃至新的模拟器。

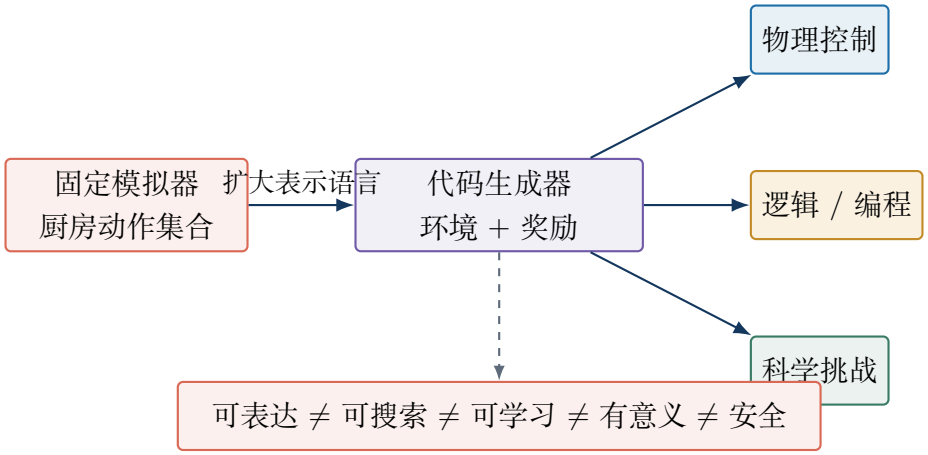


图 7.1: 代码扩大的是潜在环境表示空间，不会自动解决筛选与训练难题。原创教学重绘；非视频截图

7.2　现实系统仍被算力和样本效率卡住

视频字幕 18:51–20:40

视频明确承认现实约束：系统没有无限算力，强化学习仍然样本低效，所以演示只在受限平台内生成并学习部分任务。移动平台、多米诺、复杂建筑、球门与清洁任务展示的是生成器的多样性；“运行一亿年会怎样”是思想实验，不是已经完成的长期实验。

这一区分至关重要：

- **表达层**：代码理论上能描述什么；
- **提议层**：foundation model 实际会提出什么；
- **求解层**：RL/agent 在预算内能学会什么；
- **选择层**：系统如何判断任务值得保存；
- **安全层**：哪些环境或奖励即使有趣也不应被执行。

最常见的概念偷换

“图灵完备”描述语言的表达能力；“Darwinian-complete”在视频中是开放环境空间的研究设想。二者都不能推出系统已经探索完所有环境，更不能推出它能学会、可靠评价或安全执行所有可计算任务。

7.2.1 本章小结

- OMNI-EPIC 试图把环境表示从固定模拟器扩展到代码；
- 表达能力扩大后，任务提议、训练预算、verifier 与安全反而更关键；
- 视频展示和“一亿年算法”属于研究方向与思想实验，不能写成长期能力证明。

第八章　自动设计 Agent：从档案搜索到自我修改

8.1　搜索对象不再只是模型权重

视频字幕 20:40–22:34

视频的第二支柱是提升解决环境的 agent。今天的 agentic system 通常由 prompt、工具调用、计划循环、反思、记忆、代码与外围 scaffold 组成，人类手工连接这些部件。AI-GA 的问题是：能否把整个系统表示成可修改对象，维护多样档案，从中选择一个版本，让模型改写、部署、评估，再按质量或差异加入档案？

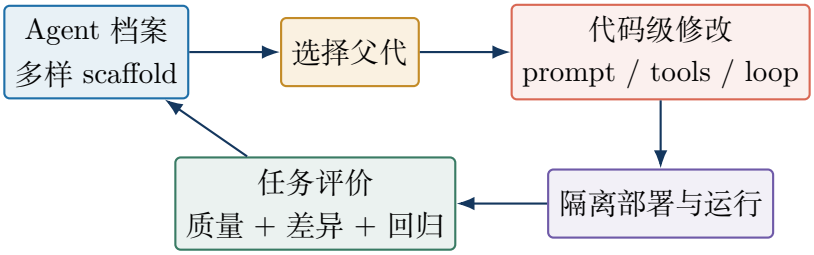


图 8.1: 自动设计 agentic system 的开放档案循环。原创教学重绘；非视频截图

视频描述系统随时间超过同期手工方案，但字幕没有给出任务数、模型、预算、统计不确定性。讲义因此把它理解为项目机制与讲者报告，而不是“自动设计普遍优于人类”的定论。

8.2　Darwin–Gödel Machine：开放档案 + 自修改

视频字幕 22:34–23:56

Darwin–Gödel Machine（DGM）把两种思想组合：Darwin 式的多样演化档案，以及 Gödel machine 式的自我修改。视频的消融叙事强调：只做贪婪 hill climbing 结果较差；保留多样 agent archive 会改善；禁止 self-improvement 也会变差；两者结合表现最好。

表 8.1: 根据字幕转述重构的 DGM 机制消融；不含伪造数值

搜索设置	开放档案	自我修改	教学解读
贪婪爬山	否	是	单一路径容易丢掉反直觉中间版本
多样档案、无自改	是	否	能保留分支，但改进算子受限
DGM 组合	是	是	既保留多样谱系，又允许代码级后代改变自身

自我修改仍必须由外部可观察结果约束

一个 agent 说“我改进了自己”不是证据。至少要比较匹配预算下的新旧任务表现、旧能力回归、工具权限与失败模式。视频没有给出这些协议细节，因此本讲义不把消融口述扩写成完整科学复现。

8.3 HyperAgents：迁移“如何改进自己”的知识

视频字幕 23:56–26:09

字幕把后续项目译作“超智能体”，本讲义保留英文候选名 **HyperAgents**。DGM 在编程任务上有一个特殊耦合：编码能力提高，修改自身代码的能力也可能一起提高；写诗、金融建模等非编程任务却没有天然耦合。视频描述 HyperAgents 把下游任务 agent 与元改进 agent 放进统一可编辑代码库，同时优化任务能力与改进能力。

更有意思的主张是元改进迁移：系统先在任务 A 学会怎样改进自己，迁移到任务 B 后即使停止继续自改，也携带一部分“如何改进”的知识。

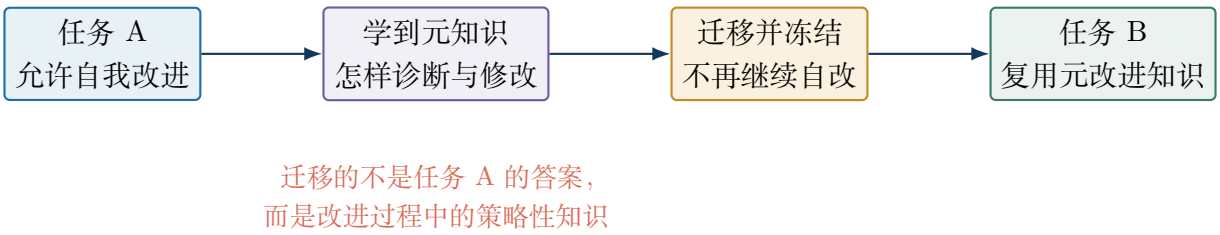


图 8.2: HyperAgents 的元改进迁移主张。原创教学重绘；非视频截图

项目名与范围仍有字幕不确定性

24:23–24:31 存在字幕缺口，作者名、论文标题及跨任务实验范围没有得到外部逐项核验。因此正文只解释视频所述机制，不声称该能力已在任意模型、任意任务上成立。

8.3.1 本章小结

- agentic system 的可搜索对象包括 prompt、工具、计划、记忆与代码；
- DGM 的核心是开放档案与自修改算子的组合；
- HyperAgents 试图把下游能力与“改进能力”解耦，从而跨任务迁移元知识。

第九章　记忆与科学自动化：ALMA 与 AI Scientist

9.1　ALMA：把记忆架构也变成搜索对象

视频字幕 26:09–27:28

持续学习 agent 需要保存过去交互、选择何时写入、怎样压缩、何时检索并放回上下文。常见做法由人手工定义这些模块。视频中的 ALMA 将同一 AI-generating philosophy 用于记忆：生成多样记忆设计，形成带分支的 archive tree，在任务上评价并继续变异。

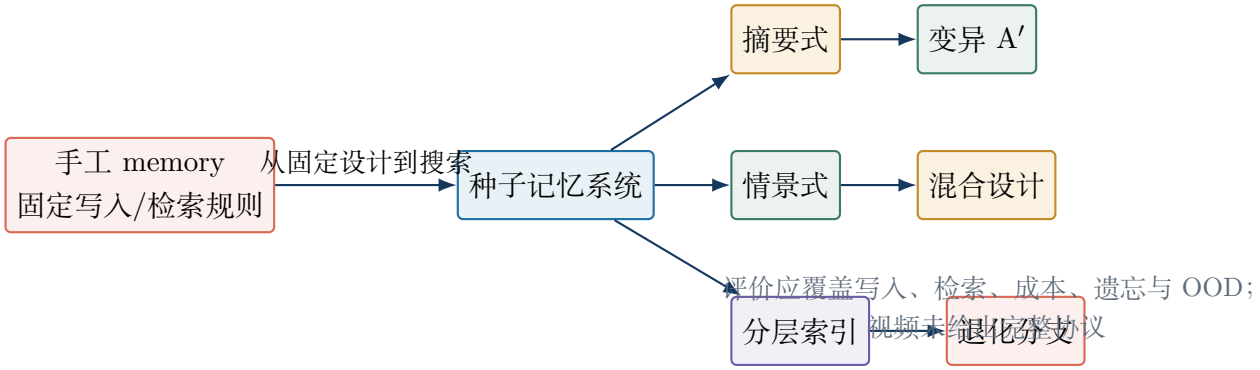


图 9.1： ALMA 的教学重构：记忆设计形成可演化的档案树。原创教学重绘；非视频截图

视频称搜索出的设计优于手工系统、计算增加时继续改善，并在分布外任务上更强。由于字幕没有提供 ALMA 全称、作者、基准、数据规模与误差，这些只能写成**讲者报告的结果**。真正的科学问题是：fitness 是否泄漏测试任务？更大上下文成本是否被公平计入？OOD 是否在设计冻结后定义？

9.2　AI Scientist：把循环扩展到科研生产线

视频字幕 27:28–28:38

AI Scientist 把开放循环放到机器学习科研：产生想法、写代码、设计并运行实验、分析和可视化、撰写论文、模拟同行评审，再读取自己的结果提出下一轮想法。

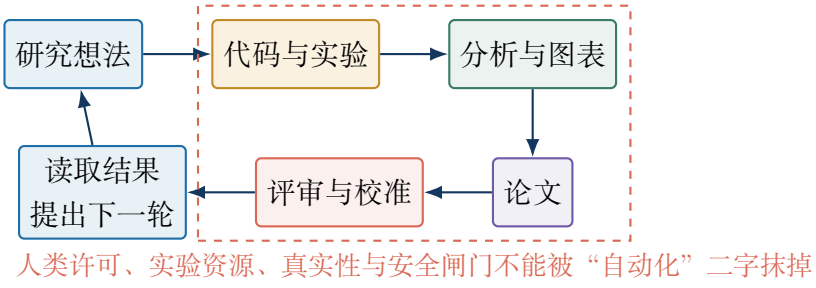


图 9.2： 科研流程自动化闭环；每一个箭头都可能引入错误，而不是自动产生真理。原创教学重绘；非视频截图

讲者认为机器学习尤其适合这条路线，因为实验平台本身就是计算机，假设可以通过代码快速测

试。他还提到在组织许可下向研讨会同行评审投稿并被接受；但字幕将会议名和百分比数字识别得含混，本讲义不写死这些细节。

流程自动化与知识可靠性是两件事

系统能完成 idea→paper 的文件流水线，只证明流程覆盖；科学可信度还需要正确问题、无泄漏实验、可靠基线、统计检验、负结果、可复现性、诚实写作和外部审查。论文被接收也不能单独证明结论为真。

9.2.1 本章小结

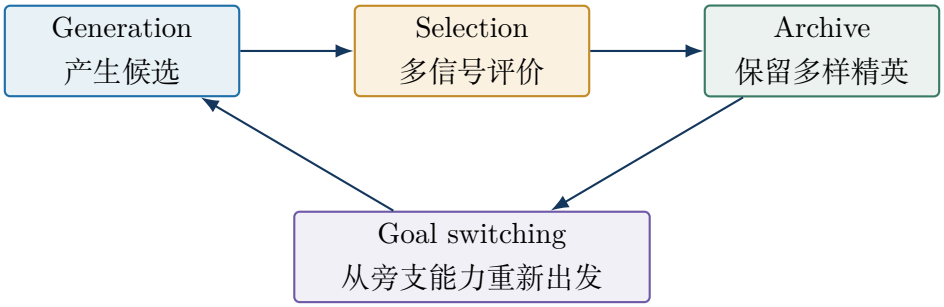
- ALMA 把 memory architecture 纳入开放式设计搜索；
- AI Scientist 把生成–评价–档案循环推进到端到端科研；
- 自动化覆盖率、科研质量与科学真理必须分层验收。

第十章　统一框架：生成、选择、档案与目标切换

10.1　这些项目为什么其实在做同一件事

视频字幕 28:38–29:57

从 MAP-Elites 到 AI Scientist，表面对象不断变化，底层骨架却高度一致：



候选可以是：行为、环境、任务、agent、记忆系统、研究想法
评价可以是：质量、新颖性、学习进度、interestingness、verifier 与风险

图 10.1: 全片的共同算法语法。原创教学重绘；非视频截图

可以用一个**讲义概念式**压缩多信号选择：

$$S(x) = \alpha Q(x) + \beta N(x) + \gamma LP(x) + \delta I_{\mathrm{FM}}(x) - \lambda R_{\mathrm{risk}}(x).$$

其中 Q 是质量，N 是新颖性，LP 是学习进度，I_{FM} 是 foundation model 给出的趣味/差异代理，R_{risk} 是风险。这个式子**不是视频给出的原始公式**，也不主张简单加权就是最佳实现；它只是提醒读者：不同项目实际上在组合不同的选择信号。

10.2　从工程视角重写 Clune 的结尾问题

讲者最后让听众思考：自己的工作中，哪些手工设计环节应该改成学习过程，并用开放式与 AI-generating principles 处理？一个可执行版本不是“把一切交给 AI”，而是依次问：

1. 当前只有一个冠军，还是保留了行为与失败模式不同的候选档案？
2. 评价是否只有一个易被钻空子的 proxy？
3. 哪些暂时不涨分的对象，可能为未来任务提供 stepping stone？
4. 任务和 benchmark 是否固定到足以让系统过拟合？
5. 生成器、evaluator 与安全闸门是否彼此独立、可检查、可回退？

迁移练习：在你自己的项目里

选择一个目前完全靠手工完成的部件，例如 prompt、任务生成、测试选择、memory、工具路由或实验设计。写出：

- 候选对象是什么；行为描述符是什么；质量函数是什么；
- 怎样保留多样 archive；何时允许 goal switching；
- 最小预算下，哪一个对照实验能区分“真实 stepping stone”与“短期 benchmark hack”；
- 哪些分支必须被硬性禁止，而不能借“开放探索”之名执行。

10.2.1 本章小结

- 全片的共同语法是 generation → selection → archive → goal switching；
- 研究对象从行为扩展到任务、环境、agent、memory 与科研流程；
- 真正困难的部分始终是评价：什么算更好、不同、可学、值得和安全。

第十一章　边界审计：这场演讲不能证明什么

11.1　六个必须拆开的命题

表 11.1: 从愿景到可验证命题的降格阅读

视频中的强表述	可以保留的含义	不能推出的结论
“忽略目标反而更容易成功”	某些 deceptive proxy 会排除绕行路线	所有目标函数都无用；随机探索普遍优于优化
“运行一亿年的算法”	追求长期产生新生态位的设计理想	已有算法经长时间运行必然持续创新
“Darwinian-complete”	环境代码具有广泛的可计算表达能力	系统已经搜索、学会或安全覆盖所有任务
“自动设计系统超过手工系统”	讲者报告特定项目中的性能趋势	任意预算、任意任务上自动设计都优于专家
“自我改进可以迁移”	所述实验中元改进知识可能跨任务复用	任意 agent 会递归自我增强，或能力必然爆炸
“AI Scientist 自动化科学”	部分机器学习科研步骤可被串成自动循环	科学结论自动可靠、无人监督，或已自动化所有科学
“通往 AGI/ASI 的最快路径”	Jeff Clune 的路线判断与研究倡议	已有比较实验证明它优于所有其他 AGI 路线

11.2　怎样验证一个真正的 stepping stone

事后看到成功谱系，很容易把每个祖先都叫作“关键垫脚石”。更严格的检验至少需要：

1. **反事实删除**：从 archive 删除候选 z 或禁止其成为父代，最终能力是否稳定下降？
2. **匹配预算**：比较开放档案与单一路径时，总训练、评价、生成和检索成本是否相同？
3. **多种子**：效果是否只来自一次幸运谱系？
4. **跨任务复用**：z 的价值是否只在发现它的任务成立，还是能打开后续任务？
5. **时间诚实**：descriptor 与选择器是否只使用当时可见信息，而非未来结果泄漏？

一个更科学的开放式算法叙事

现象：开放 archive 找到直接优化没找到的解。
归因层级：多样性覆盖、跨环境迁移、父代选择、额外计算量都可能贡献。
不能宣称：仅凭一条成功谱系，不能证明算法普遍更好。
最小区分实验：匹配预算、多种子，分别关闭 archive、goal switching、自修改或 interestingness evaluator。

11.2.1 本章小结

- 愿景、机制、演示、benchmark 结果和通用能力必须分层；
- 可表达性、可学习性、意义、可靠性与安全是五道独立门；
- 开放式算法最需要的不是更响亮的口号，而是可区分机制贡献的消融与反事实实验。

第十二章　总结、术语表与复习路线

12.1　一页压缩

从问题到路线的完整链条

单一 proxy 可能产生目标欺骗 ⇒ 保留旁支能力与 stepping stones ⇒ MAP-Elites 用行为 cell 保存每类 elite ⇒ POET 让环境与 agent 共同演化 ⇒ OMNI 用 learning progress 与 foundation-model interestingness 筛选任务 ⇒ OMNI-EPIC 用代码扩大环境表示 ⇒ DGM / HyperAgents 搜索并自改 agentic system ⇒ ALMA 搜索记忆架构 ⇒ AI Scientist 把同一循环扩展到科研流程。

Clune 的核心贡献不是声称某个单独系统已经实现超级智能，而是提供一种统一研究视角：**不要把创新压缩为一个固定 benchmark 上的单一路径；让系统维护多样谱系，并让任务、环境与改进机制本身成为可学习对象**。

12.2　核心术语表

术语	本讲义中的精确定义
Goal deception	代理目标与真实任务结构错位，优化器被合法地导向错误区域；不是算法“故意欺骗”。
Stepping stone	当前主指标上未必优秀、却能为后续搜索打开新区域的中间对象或能力。
Goal switching	发现旁支能力后，允许改变或扩展局部目标，并保留该分支继续发展。
Archive	持久保存多样候选、行为、任务或系统的搜索记忆；不等于普通 checkpoint 文件夹。
Quality-Diversity (QD)	同时追求行为覆盖和各行为区域内高质量解的一类算法。
MAP-Elites	以 behavior descriptor 划分 cell，每格保存当前同类最高质量 elite 的 QD 算法。
Behavior descriptor	决定候选“属于哪一类”的低维行为特征；与质量分数相分离。
Open-endedness	新解持续创造新问题、生态位与学习机会，使任务空间本身扩张。
POET	环境与 agent 配对共同演化，并允许跨环境迁移形成非线性课程。
Learning progress	一段时间内任务表现变化的信号；表示可能可学，不代表任务必然有价值。

术语	本讲义中的精确定义
Interestingness	对任务是否新颖、不同、值得探索的代理判断；OMNI 用 foundation model 近似人类观念。
Darwinian-complete	对广泛可计算环境的表示空间愿景；不是已搜索完、已学会或已安全覆盖。
AI-generating algorithm	让 AI 生成/改进架构、训练算法、环境、目标和 agentic system 的研究路线。
Darwin–Gödel Machine	把开放式多样 archive 与代码级自我修改结合的 agent 改进系统。
HyperAgents	视频所述、尝试把下游能力与元改进能力共同优化并迁移的项目名候选。
ALMA	视频所述、用开放档案搜索记忆架构的项目；当前字幕不足以补全论文细节。
AI Scientist	将想法、实验、分析、写作、评审与下一轮想法串成自动科研循环的系统。

12.3 30 分钟时间轴

时间	内容定位
00:00–02:34	目标欺骗；迷宫；微波炉与计算机；爬行/单脚平衡作为 stepping stones
02:34–03:35	创新引擎：收集、变异/重组、质量与新颖性评价、归档
03:35–08:18	QD、MAP-Elites、迂回谱系、损伤适应机器人、Go-Explore
08:18–11:53	固定环境的边界；open-endedness；POET 环境–agent 共演
11:53–14:57	AI-generating algorithms 的研究纲领与巨大任务空间问题
14:57–17:46	OMNI、learning progress、interestingness、自然语言任务与奖励代码
17:46–20:40	OMNI-EPIC、Darwinian-complete、现实算力与样本效率限制
20:40–23:56	行为先验；自动设计 agentic systems；Darwin–Gödel Machine
23:56–26:09	HyperAgents 与自我改进知识迁移
26:09–27:28	ALMA：开放式记忆架构搜索
27:28–28:38	AI Scientist：端到端科研循环
28:38–29:57	总结、AGI/ASI 路线判断与应用问题

12.4 复习题

1. 为什么“与终点欧氏距离更近”不能代表迷宫中的真实进展？请给出另一个工程 proxy 失真的例子。

2. MAP-Elites 中 behavior descriptor 与 quality 各自负责什么？为什么不能混成一个概念？
3. 一个候选足够新颖，是否就应进入 archive？还需要哪些条件？
4. POET 与“把同一关卡逐渐调难”有何根本不同？
5. Learning progress 为什么不能替代 interestingness？Foundation model 又为什么不能成为客观价值函数？
6. 为什么“代码是图灵完备的”不能推出“agent 能学会任意代码生成的环境”？
7. DGM 的开放 archive 与 self-modification 分别解决什么问题？怎样做匹配预算消融？
8. AI Scientist 完成一篇论文流水线后，仍缺哪些证据才能证明科学结论可靠？
9. 为你自己的一个项目定义候选、descriptor、quality、archive 与 goal-switching 规则。
10. 设计一个反事实实验，验证某个中间对象是真 stepping stone，而不是事后编故事。

12.5 最后带走的五句话

1. 困难目标的路径通常不能被最终指标提前完整描述。
2. 多样 archive 的意义，是让延迟显现的价值仍有未来。
3. 开放式不是无目标，而是多目标、可切换、可积累、能创造新问题。
4. Foundation model 让任务生成、趣味判断和代码改写更可操作，但也把偏差带进 evaluator。
5. “自动科研”最值得追问的不是能否产出文件，而是能否产生经得住独立复核的新知识。

讲者留给听众的问题

“你可以在自己的工作中应用这些理念何处？”
讲义给出的附加条件是：不要只写愿景。找出一个手工瓶颈，定义可测对象与档案，做一个匹配预算的最小对照，然后再判断开放式搜索是否真的提供了新的 stepping stones。

来源说明

- 主来源：Jeff Clune 演讲中文时间戳字幕，Bilibili BV1Hzjg65EZd，共 720 条，覆盖 00:00–29:57；视频总时长 30:01。
- 元数据：Bilibili 公共视频元数据；标题为《【Frontier】人类命运大转折！开放式演化算法 +AI 生成算法全自动科研 | Jeff Clune》，频道“至高机器智能”。
- 本讲义未进行外部论文逐项核验，因此对论文状态、作者、会议名、精确百分比和 benchmark 强结果保持字幕证据边界。
- 所有图均为本讲义原创机制图；没有使用视频封面、关键帧、截图或视频派生视觉。