
Can Current AI Algorithms Scale to Scientific Breakthroughs?

Odysseus Kaloidas
UC Berkeley

Abstract

Can today’s AI methods, including improvements they help discover, scale to major scientific breakthroughs? A system might find a relationship in existing measurements yet struggle to discover what else needs measuring, or how to explain it. These difficulties are not captured by dataset size alone, and they need not form a universal ladder. The question is how to describe them well enough to measure which ones AI is overcoming. Recursive self-improvement belongs inside this inquiry: AI can help improve the methods used by later AI. Does that feedback expand the kinds of discovery within reach, or mainly accelerate work that was already possible?

1 When Discovery Improves the Discoverer

Can today’s AI methods scale to major scientific breakthroughs, including by helping invent their own successors? An AI used for research might produce more than an answer: it might find a better way to obtain later answers. Lion searches for an optimizer, the rule that adjusts a model during training. DiscoRL learns rules for improving behavior through reward and reports transfer beyond the settings used to discover them [3; 8]. These are reasons to take the possibility seriously. They also sharpen the question: which parts of scientific discovery become easier when the method itself improves?

Scaling expands resources such as training data and computation, but it need not freeze the methods using them. Better search, tools, and learning procedures belong inside this account of progress, including when AI helps discover them.

2 Different Advances, Different Demands

Deep Blue’s chess success depended on search, a carefully engineered representation, and evaluation [2]. It did not establish general scientific invention. Yet DENDRAL already applied AI to inferring molecular structures [6]. The historical question is what makes search productive in each setting.

Consider a hypothetical laboratory problem: explaining why a reaction speeds up. In one version, the relevant measurements and possible forms of equation are supplied; the task is to find the relationship among them. In another, the recorded quantities cannot distinguish the explanations, and the investigator has to discover that an additional factor should be measured. A third version asks for a different account of the mechanism, one that predicts what will happen under conditions the existing explanation does not cover.

Finding a relationship, choosing a useful formulation, and revising an explanation place different demands on a learner. A dataset with the same number of rows could present any of them. Applying a known method to a familiar case provides a baseline, but counting examples alone would not tell us how far beyond that baseline the task goes.

These distinctions concern what the investigator has to discover. They are not mathematical complexity classes, which classify formal problems by the resources needed to solve them. A proof within familiar mathematics can be harder and more important than a modest representation change. Difficulty, novelty to the learner, and scientific importance differ: rediscovering a hidden known law tests a capability without extending knowledge.

Difficulty can depend on interacting variables, search depth, noise, and the cost of distinguishing explanations. Bellman’s grid example illustrates one burden: m values on each of d coordinates produce m^d combinations [1]. Learning need not enumerate them all. Language can supply abstractions that perception must recover, yet a short statement can still express a formidable problem.

3 More Data, or More Distinguishing Evidence?

Data size combines prior training, observations supplied for a problem, and evidence acquired during research. Chollet emphasizes prior knowledge and experience when evaluating learning efficiency [4]. Apparently learning from a few examples can conceal extensive inherited knowledge.

Suppose two explanations make identical predictions under all the conditions we have measured. Repeating those measurements more precisely cannot choose between the explanations. We need conditions under which their predictions differ, or additional assumptions that justify preferring one. The same distinction matters in causal inference: accurately predicting an observation need not tell us what would happen if we intervened [9].

Compute can expand search; an algorithm can expose reusable structure; a new observation can eliminate competing explanations. These contributions interact. None provides a universal exchange rate between tokens, experiments, and conceptual progress. Their relative importance changes across these discovery demands.

4 Measuring How Far the Frontier Moves

A success rate tells us more when we know which discovery demands were tested. Did the system receive the relevant variables, or find them? How much noise, search, or missing evidence stood in its way?

The comparison also depends on the evidence, computation, prior knowledge, and tools available. A specified task set and validation criterion let us compare progress without pretending to possess a universal scale of difficulty.

A random held-out split sets aside some examples from the available dataset for testing. It principally tests unfamiliar examples from the sampled setting. Extrapolation means extending success beyond the conditions already tested. New operating conditions, more interacting variables, or formulations the system must discover require corresponding evaluation changes. If we do not know what appeared in the model’s training, we may also mistake a familiar problem for a new one. Comparing model generations also changes more than compute alone.

Existing benchmarks begin to separate these demands. DiscoveryBench varies such features as the number of steps in an investigation and dependencies among variables in constructed tasks [7]. NewtonBench varies the complexity of both the target law and the surrounding system in interactive simulated discovery [11]. Such controls let us ask what a success rate reflects. They do not settle how far that success reaches: a long workflow can be conceptually routine, and recovering a hidden law still differs from inventing a new framework.

Controlled tasks show how resources affect performance; real discoveries show whether the distinctions matter scientifically. A plateau identifies a limitation at tested budgets. A rising curve supports extrapolation only where the next problems preserve the conditions behind it.

5 Self-Improvement Inside Scientific Scaling

Better code, experiments, and search procedures can help produce the next research improvement. The Darwin Gödel Machine offers a concrete case: it improves coding-agent software around broadly pretrained foundation models while keeping their trained parameters fixed [10]. The software changes;

the underlying model is not retrained. This is AI-assisted self-improvement without autonomous development of successive foundation models.

Lion addresses transfer from small search tasks to larger targets, while DiscoRL reports transfer to unseen benchmarks [3; 8]. These signs of transfer leave open whether the methods improve the activities limiting a further breakthrough.

There is a serious optimistic possibility here. A discovery can be scientifically important even if it requires no new kind of computation. If a decisive candidate lies within reach of an existing search procedure and can be recognized reliably, better proposals or cheaper evaluation could make the difference. Conversely, a striking improvement on routine searches may leave a missing measurement or an inadequate formulation untouched. The importance of the eventual discovery and the kind of progress needed to reach it are separate questions.

This is where recursive self-improvement belongs: a system helps improve the methods used by its successors, which may in turn help produce further improvements. It need not invent a new paradigm at every cycle, or eliminate human participation before becoming consequential. Cunningham and colleagues formalize feedback while distinguishing narrow and broad capabilities and task-specific bottlenecks [5]. Faster progress in one part of research need not imply comparable gains everywhere, although a narrow improvement can unlock a major bottleneck.

Major breakthroughs may emerge from cumulative improvements to recognizable foundations. The difficult question is how to notice whether we are moving toward them. Is the next system searching a familiar space more effectively, choosing better experiments, or finding a formulation its predecessor could not express? Following those changes would give substance to both optimism and skepticism about a process that increasingly helps build its own successors.

References

- [1] Richard Bellman. *Dynamic Programming*. Princeton University Press, 2010. doi: 10.1515/9781400835386. Originally published in 1957; new introduction by Stuart Dreyfus.
- [2] Murray Campbell, A. Joseph Hoane, Jr., and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1–2):57–83, 2002. doi: 10.1016/S0004-3702(01)00129-1.
- [3] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL <https://arxiv.org/abs/2302.06675>.
- [4] François Chollet. On the measure of intelligence, 2019. URL <https://arxiv.org/abs/1911.01547>. arXiv:1911.01547.
- [5] Tom Cunningham, Lukas Althoff, Basil Halperin, Brian Jabarian, Andrew Koh, Arjun Ramani, Phil Trammell, Parker Whitfill, and Cheryl Wu. The economics of recursive self-improvement. Technical report, Elasticity Institute, 2026. URL <https://elasticity.institute/rsi-paper.pdf>. Working paper, July 13.
- [6] Edward A. Feigenbaum and Bruce G. Buchanan. DENDRAL and Meta-DENDRAL: Roots of knowledge systems and expert system applications. *Artificial Intelligence*, 59(1–2):233–240, 1993. URL <https://stacks.stanford.edu/file/druid:pj337tr4694/pj337tr4694.pdf>.
- [7] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhi-jeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models, 2024. URL <https://arxiv.org/abs/2407.01725>. arXiv:2407.01725. Preprint version consulted.
- [8] Junhyuk Oh, Gregory Farquhar, Iurii Kemaev, Dan A. Calian, Matteo Hessel, Luisa Zintgraf, Satinder Singh, Hado van Hasselt, and David Silver. Discovering state-of-the-art reinforcement learning algorithms. *Nature*, 648:312–319, 2025. doi: 10.1038/s41586-025-09761-x. URL <https://www.nature.com/articles/s41586-025-09761-x>.
- [9] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009. doi: 10.1017/CBO9780511803161.
- [10] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026. URL <https://arxiv.org/abs/2505.22954>. First released May 2025; conference version March 2026.
- [11] Tianshi Zheng, Kelvin Kiu-Wai Tam, Newt Hue-Nam K. Nguyen, Baixuan Xu, Zhaowei Wang, Jiayang Cheng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Tianqing Fang, Yangqiu Song, Ginny Y. Wong, and Simon See. NewtonBench: Benchmarking generalizable scientific law discovery in LLM agents. In *International Conference on Learning Representations*, 2026. URL <https://arxiv.org/abs/2510.07172v3>. arXiv:2510.07172v3.