Efficiency vs Capability in the Intelligence Explosion

Karthik Tadepalli*

July 21, 2026

Preliminary draft, circulated for feedback.

Abstract

I model two channels by which AI progress can feed back into the automation of AI research and lead to a self-sustaining acceleration. Capability improvements allow AIs to perform a larger share of AI research tasks, while efficiency improvements raise AI research output per unit of compute on tasks already automated. I separate these margins in a task-based R&D model of AI progress. A software intelligence explosion (SIE)—a self-sustaining acceleration in AI progress with other inputs held fixed—requires both margins to improve sufficiently fast. Efficiency must improve fast enough to overcome the rivalry of compute, while capability must improve fast enough to prevent human researchers from bottlenecking AI progress. Neither margin can substitute for a shortfall in the other. I argue that this distinction is essential for empirical studies of an SIE, because residual measures of algorithmic efficiency include capability improvements and thus are biased toward overestimating the likelihood of an SIE. I conclude with some extensions that this model is uniquely suitable for investigating.

1 Introduction

Frontier AI systems are increasingly being used to accelerate R&D into future AI systems, in a positive feedback loop known as recursive self-improvement [Favaro and Clark, 2026]. If this feedback loop becomes self-sustaining, recursive self-improvement could lead to a *software intelligence explosion* (SIE), which would lead to much more dramatic AI progress than in the past. The likelihood of an SIE is one of the most important questions in AI policy today. This paper formalizes a model of AI development in which AI progress feeds back into itself through two margins. *Capability* is the share of research tasks that AI can perform, while

*Center for the Governance of AI. karthik.tadepalli@governance.ai

efficiency is useful AI research output per unit of compute on automated tasks. Capability improvements and efficiency improvements are, respectively, increases in these two quantities. I use this model to establish the conditions for an SIE and draw lessons for how to measure it empirically.

The benchmark model of idea production [Jones, 1995] summarizes research with a scalar stock of efficiency units. The pace of technological progress is entirely a function of this stock, with no distinction between types of researchers. In this representation, sufficiently many low-productivity researchers substitute for any high-productivity researcher. The same logic makes sufficiently many copies of a less capable AI a substitute for a more capable model. Put differently, there is some N such that N copies of me are better at AI research than Alec Radford. This is an implausible assumption in the context of AI research, yet it is inherited from the semi-endogenous model by most models of AI development.

The model in this paper replaces scalar research progress with two channels through which AI progress feeds back into itself. Capability changes where AI can be deployed; efficiency changes how much AI research input is supplied by a given compute stock.

I show that recursive self-improvement in this model is governed by the tighter of two constraints. The first constraint is the amount of AI research that the fixed compute stock can supply at the current level of efficiency. The second is the amount of complementary human research available beyond AI's capability. Improving AI research therefore requires both higher efficiency and a sufficiently rapid reduction in the set of tasks that still require humans. Whichever margin improves more slowly eventually determines the speed of AI progress.

I show that an SIE occurs if and only if two conditions are met: efficiency must improve fast enough to overcome the rivalry of physical compute, *and* capability must improve fast enough to relax the human bottleneck on AI research. Efficiency improvements alone eventually run into Baumol effects, where the remaining human-only tasks prevent self-sustaining acceleration. Capability improvements alone fail for a more surprising reason: the rivalry of physical compute means that automating a larger share of tasks only spreads the compute stock thinly across all tasks. Even an exponentially growing compute stock cannot prevent this depletion. Thus, both capability improvements and efficiency improvements must be fast enough to generate an SIE.

Next, I show that standard approaches to measuring capability and efficiency in the literature are biased toward overestimating the likelihood of an SIE. In particular, estimating algorithmic efficiency as the residual between an AI performance index and compute combines capability and efficiency improvements, and thus overestimates true efficiency improvements. Similarly, the Epoch Capabilities Index conflates efficiency and capability improvements. I

discuss the requirements empirical measures of the two margins must satisfy before they can be used to estimate the likelihood of an SIE.

Finally, I conclude with some possible extensions: important questions in the economics of AI that this two-margin model is uniquely suited to answering. The current draft does not develop any of these extensions in detail; doing so will be the next stage of this project.

Section 2 presents the task environment. Section 3 derives the research-capacity bounds and the two-margin condition for self-sustaining acceleration, including the case of growing compute. Section 4 relates the result to existing models. Section 5 maps its parameters to empirical measures and derives the residual-measurement bias. Section 6 develops extensions of the two-margin framework.

2 Model

AI development requires a continuum of research tasks $i \in [0, 1]$, to be performed by a fixed stock $L > 0$ of human researchers or a fixed stock $Z > 0$ of physical compute (on which copies of the frontier AI model run).[1]

Technology. Frontier AI technology is indexed by $A_t \in [1, \infty)$. A model at level A can perform research tasks $i \leq \alpha(A)$, where capability $\alpha(A)$ satisfies

$$\alpha(A) = 1 - A^{-\mu}, \qquad \mu \geq 0, \tag{1}$$

and its efficiency (useful research output per unit of compute on automated tasks) is

$$\nu(A) = \nu\, A^{\eta}, \qquad \nu > 0,\ \eta \geq 0 \tag{2}$$

The parameter μ is the capability elasticity, while η is the efficiency elasticity. Define *effective compute* as the AI research input supplied by physical compute at the current level of efficiency,

$$X_t \ \equiv \ \nu(A_t)\, Z,$$

whose physical component Z is a rival compute stock, and whose software component $\nu(A_t)$ is produced by the loop itself.

[1]The fixed stock of physical compute is assumed to capture the fact that on the short timescales for which people worry about an SIE, compute cannot be part of the feedback loop. Section 3.1 considers how the results would change with a growing compute path.

Block	Statement
Technology: capability	$1-\alpha(A)=A^{-\mu}$
Technology: efficiency	$\nu(A_t)=\nu A_t^{\eta}$; effective compute $X_t=\nu(A_t)Z$
Research tasks	$r_t(i)=\ell_t(i)+\nu(A_t)z_t(i)\,\mathbf{1}\{i\le\alpha(A_t)\}$
Research aggregation	$R_t=\big(\int_0^1 r_t(i)^{\rho}\,\mathrm{d}i\big)^{1/\rho},\ \sigma\in(0,1)$
Innovation: law of motion	$\dot{A}_t=\delta R_t A_t^{1-\beta}$
Resources: researchers	$\int_0^1 \ell_t(i)\,\mathrm{d}i=L$
Resources: physical compute	$\int_0^{\alpha(A_t)} z_t(i)\,\mathrm{d}i=Z$

Table 1: The economic environment.

Research Production. Research task i's output is produced from human researcher input $\ell_t(i)$ and effective AI input, which are perfect substitutes within a task, AI being usable only on automated tasks:

$$r_t(i) \;=\; \begin{cases} \ell_t(i)+\nu(A_t)\,z_t(i), & i\le\alpha(A_t),\\ \ell_t(i), & i>\alpha(A_t), \end{cases} \tag{3}$$

Tasks are complements in aggregate research output, being combined with a constant elasticity of substitution $0<\sigma<1$:

$$R_t \;=\; \left(\int_0^1 r_t(i)^{\rho}\,\mathrm{d}i\right)^{1/\rho}, \qquad \rho\equiv\frac{\sigma-1}{\sigma}\;<\;0, \tag{4}$$

Researchers and compute are fully employed,

$$\int_0^1 \ell_t(i)\,\mathrm{d}i \;=\; L, \tag{5}$$

$$\int_0^{\alpha(A_t)} z_t(i)\,\mathrm{d}i \;=\; Z \tag{6}$$

Research output drives the growth of the technology level,

$$\dot{A}_t \;=\; \delta\,R_t\,A_t^{1-\beta}, \qquad \delta>0,\ \beta>0 \tag{7}$$

Finally, we need to specify an allocation rule for labor and compute. I assume that the allocation is chosen to maximize R_t, consistent with labor and compute being allocated internally to a profit-maximizing AI developer. Write $\alpha^*\equiv X/(L+X)$ and define *research capacity* as aggregate research output R under the task assignment that maximizes it.

Proposition 1 (Task Allocation)**.** *Fix $\alpha \in (0,1)$, $L > 0$, and $X \geq 0$. At the optimal task assignment, research capacity is*

$$R(\alpha; L, X) \;=\; \begin{cases} \left[\, \alpha^{1/\sigma} X^{\rho} + (1-\alpha)^{1/\sigma} L^{\rho} \,\right]^{1/\rho}, & \alpha \leq \alpha^{*}, \\ L + X, & \alpha \geq \alpha^{*}. \end{cases} \tag{8}$$

Proof. See Appendix A.1.

3 Explosion Conditions

To analyze an SIE, start by fixing capability α and allowing effective compute X to grow arbitrarily large, since it is accumulable. This thought experiment gives the most research the economy could produce at that level of capability: automated tasks cease to constrain output, while the remaining human-only tasks still must be supplied by the fixed researcher stock. Define the *human-bottleneck bound* by

$$\overline{R}(\alpha) \equiv \lim_{X \to \infty} R(\alpha; L, X) \tag{9}$$

As a result of this abundance, human-only tasks become the main bottleneck to research inputs.

Proposition 2 (Bottleneck Bounds)**.** *For every $\alpha \in (0,1)$, the human-bottleneck bound exists and is*

$$\overline{R}(\alpha) = \frac{L}{(1-\alpha)^{1/(1-\sigma)}} \tag{10}$$

Research capacity satisfies

$$2^{-\frac{\sigma}{1-\sigma}} \cdot \min\left\{X,\ \overline{R}(\alpha)\right\} \;\leq\; R(\alpha; L, X) \;\leq\; \min\left\{L + X,\ \overline{R}(\alpha)\right\} \tag{11}$$

Proof. See Appendix A.2.

In the fixed-compute baseline, effective compute is $X(A) = \nu Z A^{\eta}$ and the closed dynamics are

$$\dot{A} = \delta R(\alpha(A); L, X(A))\, A^{1-\beta}, \qquad g_A(A) = \delta R(\alpha(A); L, X(A))\, A^{-\beta} \tag{12}$$

Away from the allocation kink, define the elasticity of research capacity with respect to effective compute as

$$\varepsilon_{R,X} \equiv \frac{\partial \ln R}{\partial \ln X}$$

Log-differentiating (12) gives the local decomposition

$$\frac{\mathrm{d}\ln g_A}{\mathrm{d}\ln A} = \underbrace{\varepsilon_{R,X}\eta}_{\text{efficiency}} + \underbrace{\frac{\partial \ln R}{\partial \alpha}\frac{\mathrm{d}\alpha}{\mathrm{d}\ln A}}_{\text{capability}} - \underbrace{\beta}_{\text{idea difficulty}} \tag{13}$$

Efficiency increases the effective input available on tasks AI already performs. Capability makes that input usable on a broader task set. Idea difficulty subtracts from the return to both margins.

Definition 1 (Software Intelligence Explosion)**.** Holding L and Z fixed, the model exhibits a software intelligence explosion if AI progress eventually raises its own growth rate at a sustained proportional rate:

$$\liminf_{A\to\infty} \frac{\mathrm{d}\ln g_A}{\mathrm{d}\ln A} > 0 \tag{14}$$

Locally, acceleration occurs when

$$\frac{\mathrm{d}\ln g_A}{\mathrm{d}\ln A} > 0 \quad \Longleftrightarrow \quad \frac{\mathrm{d}\ln \dot{A}}{\mathrm{d}\ln A} > 1$$

The limit condition in (14) distinguishes a self-sustaining acceleration from a temporary rise in g_A or convergence to a constant growth rate.

The other possible definition of an SIE corresponds to a true technological singularity, where we reach infinite progress in finite time. In this model, there is no distinction; the following proposition establishes the conditions for an SIE that hold for both definitions.

Proposition 3 (SIE Conditions)**.** *Under* (1)–(7)*, the model exhibits a software intelligence explosion if and only if*

$$\boxed{\underbrace{\eta > \beta}_{\textit{efficiency leg}} \quad \textit{and} \quad \underbrace{\mu > (1-\sigma)\beta}_{\textit{capability leg}}} \tag{15}$$

Proof. See Appendix A.3.

Each inequality rules out a different failure. When both hold strictly, AI progress raises its own growth rate persistently. When one condition holds at equality and the other holds weakly, the growth rate approaches a positive constant. When either inequality fails strictly and is the tighter constraint, that margin eventually pulls the growth rate down.

Efficiency alone. Suppose efficiency improves fast enough to overcome idea difficulty, but capability does not: $\eta > \beta$ while $\mu \leq (1-\sigma)\beta$. Productivity rises rapidly on automated tasks, but the complementary human-only task share contracts too slowly. Those tasks become an increasingly important share of the effective cost of research and eventually govern

aggregate progress. This is the familiar Baumol effect emphasized by Aghion et al. [2019]: rapid progress in some tasks cannot sustain aggregate growth when essential laggard tasks remain. Formally, for any efficiency path and any physical-compute path,

$$g_A(A) \leq \delta L A^{\mu/(1-\sigma)-\beta} \tag{16}$$

Efficiency improvements therefore pile up behind the human bottleneck when capability improves too slowly.

Capability alone. The failure of capability improvements without efficiency improvements is more unusual. In a standard growth model, the machine input is capital, and capital's most important dynamic feature is that it is accumulable: more output supports more investment, which expands the capital stock and creates a positive feedback loop. Compute plays the role of capital in an AI R&D model, but compute has two economically distinct components. *Physical compute* is the rival stock of chips used to train and run AI models. It is fixed in this model, to represent the short time horizon over which an SIE could take place. *Effective compute* is the AI research input those chips supply, $X = \nu(A)Z$, and it grows only when efficiency $\nu(A)$ improves. Higher capability makes the fixed physical stock usable on more tasks, but it does not add chips or raise their effective output. Without sufficient efficiency improvements, capability improvements therefore spread a fixed rival input across more tasks rather than accumulating more of that input. This is the key economic difference between compute in this model and capital in a standard growth model.

Capability improvements can still create a temporary burst. If $\eta < \beta$ and $\mu > (1-\sigma)\beta$, the human bottleneck recedes, but the effective-compute ceiling eventually governs capacity. The case $\eta = 0$ makes the temporary nature transparent: for $\mu > 0$, capacity reaches $L + \nu Z$ once

$$A \geq \widehat{A} \equiv \left(\frac{L + \nu Z}{L}\right)^{1/\mu},$$

and thereafter $g_A = \delta(L+\nu Z)A^{-\beta}$. Relative to its value at $A = 1$, the growth rate is amplified by less than $1 + \nu Z/L$ before it declines.

3.1 Growing Compute

It may seem like the core result is an artifact of our decision to model compute as fixed; if compute were growing exponentially, we might expect that it could substitute for efficiency improvements, such that compute growth and capability improvements alone could carry an SIE. As it turns out, this is false.

Proposition 4 (Growing Compute)**.** *Let Z_t grow exponentially at rate g_Z, and classify an acceleration as self-sustaining only if it persists when the exogenous compute path is held fixed, as in Definition 1. The conditions for a software intelligence explosion remain identical to the conditions in Proposition 3.*

Proof. See Appendix A.4.

Intuitively, exponential growth in physical compute adds a constant to the growth rate of effective compute. It cannot make effective compute grow superexponentially, because it is not part of a feedback loop—so its growth is constrained to be exponential. Thus, a rising g_X must come from a rising growth rate of efficiency. Compute also does not enter $\alpha(A)$, so it cannot relax the capability condition for an SIE.

Growing compute can still change the observed path. If $\eta < \beta$ and capability improves sufficiently quickly, it can sustain ordinary exponential growth in A; the growth rate approaches $g_Z/(\beta - \eta)$. If $\eta = \beta$, the observed growth rate can rise while Z_t is growing, but freezing the compute stock removes that rise. If $\mu/(1-\sigma) \leq \beta$, the human bottleneck continues to bind regardless of the compute path. Thus exponential compute growth can change the pace of progress without changing whether the acceleration is self-sustaining.

4 Related Models

The task structure nests or approaches several familiar models:

- **Semi-endogenous growth.** Setting $\mu = 0$ gives $\alpha(A) = 0$, so AI input is unavailable on every task and $R = L$. Equation (7) becomes the standard semi-endogenous idea-production equation [Jones, 1995]. Eth and Davidson [2025] instead model a rising stock of effective researchers. Their model corresponds to full automation of AI R&D in this model, $\alpha = 1$, where $R = L + X$.

- **Task automation.** The production side uses the task-automation mechanics of Aghion et al. [2019] and Jones and Liu [2024]: capital and labor are perfect substitutes within a task, tasks are complements in aggregate, and technological progress has both an extensive automation margin and an intensive capital-productivity margin. The correspondence is nearly exact after relabeling AI as capital. Human researchers play the role of labor, physical compute supplies the capital stock, capability α is the extensive margin, and efficiency ν is the intensive margin. More generally, the feedback loop over effective compute echoes the positive feedback from capital in a standard growth model: a productive machine input raises output, which can support more of the machine input and further raise output. Many familiar capital-accumulation mechanisms therefore

carry over to AI R&D. Physical compute is fixed here, so effective compute accumulates endogenously only through efficiency improvements. The replacement of capital with effective compute—the product of a fixed physical input and endogenous efficiency—is responsible for most of the surprising conclusions in this model.

- **Automation-based AI R&D.** Cunningham and Shetty [2026] model AIs that pick every apple within their height, with the apples picked increasing the height of future robots. This is the only other SIE model that focuses on the range of automated tasks as a feature affecting the SIE. However, the main difference is that they assume AIs are abundant, which abstracts from the rivalry of compute that is likely to bind in the short term.

5 Empirical Mapping

The primary reason to model feedback loops in AI development is quantitative; we want to know whether current AI progress is in an intelligence explosion regime. This paper is a theoretical framework, not a quantitative calibration; however, it has important implications for how we should interpret the data commonly used to calibrate SIE models. Eth and Davidson [2025] treat both acquiring new tasks and becoming more competent at tasks a system can already attempt as capability improvements, and treat achieving a given capability level with less compute as an efficiency improvement. Those definitions are intuitive, and useful for a broad accounting of software progress. But for the purpose of SIE accounting, a different boundary is necessary.

Efficiency. Efficiency is useful task output per unit of compute on automated tasks. By definition, efficiency improvements must be measured on a fixed task set. Most algorithmic progress measures are contaminated because they use changing task sets; a notable exception is the fixed-task approach of Ho et al. [2024].

The second important and unintuitive element of this definition is that improved quality at performing the same task is an efficiency improvement, rather than a capability improvement. If a model's code quality improves for the same compute cost, this may seem like a capability improvement, but for SIE accounting it raises efficiency. Treating within-task performance gains as capability would assign progress to α even though the automated task share has not changed.

The critique of treating quality and quantity as interchangeable still applies in this model, but only at the task level. Within a task, AI quality and AI cost are interchangeable: an AI

that can do a task outstandingly well is isomorphic to an AI that can do that task adequately at low cost. This assumption is more palatable within a task than at the level of AI research as a whole.

Capability. In this model, capability is the automated task share α. Each task's automation status is binary: either it is automated or it is not. Improvements in model ability that do not change α are efficiency improvements rather than capability improvements.

Unfortunately, capability improvements as constructed here are very difficult to measure empirically. The Epoch Capabilities Index [Epoch AI, 2026] is not a measure of capability in the model's sense. It stitches together different benchmarks to assemble a continuous latent index, meaning that it combines capability improvements with efficiency improvements.

In contrast, METR's time-horizon measure [Kwa et al., 2025] has a closer correspondence to $\alpha(A)$. Because it measures the boundary at which a model's task-specific success probability is 50%, its construction reflects the potential capability of AI models when applied to R&D. Under (1), if task difficulty q has tail $\Pr(q > Q) \propto Q^{-k}$ and the model adequately completes tasks up to $q(A)$, then

$$1 - \alpha(A) = \Pr(q > q(A)), \qquad \mu = k \, \frac{\mathrm{d} \ln q}{\mathrm{d} \ln A}$$

Time-horizon progress can inform the second term; given a calibration of the difficulty-tail parameter k, time-horizon progress gives us μ. Estimating k is difficult because it requires the tail of the difficulty distribution for the AI R&D tasks that matter economically, rather than only the tasks represented in an evaluation suite.

Calibrating the capability condition $\mu > (1 - \sigma)\beta$ also requires σ and β. The substitution elasticity σ is a distinctive requirement of the capability leg. Note from (8) that σ is also the elasticity of substitution between human labor and compute, which means it is amenable to empirical approaches like Whitfill and Wu [2025]. The idea-difficulty elasticity β is also challenging to estimate within AI R&D, but this challenge is shared by both legs: the efficiency condition compares η with the same β. A quantitative SIE assessment must therefore combine a capability trend, a research-task difficulty distribution, a substitution elasticity, and an estimate of diminishing returns to ideas.

5.1 Residual Bias

Conventional algorithmic-efficiency estimates often infer software progress from changes in a performance measure after accounting for compute. In this model, the relevant performance

object is the technology level A, observed through its growth rate $g_A = \dot{A}/A$. Define

$$\varepsilon_{R,X} \equiv \frac{\partial \ln R}{\partial \ln X}, \qquad s_{R,\alpha} \equiv \frac{\partial \ln R}{\partial \alpha}.$$

Holding R&D productivity δ and human research labor L fixed, the law of motion and $X = \nu Z$ imply

$$\mathrm{d}\ln g_A + \beta\,\mathrm{d}\ln A = \varepsilon_{R,X}\big(\mathrm{d}\ln\nu + \mathrm{d}\ln Z\big) + s_{R,\alpha}\,\mathrm{d}\alpha. \tag{17}$$

Suppose an observer measures A and physical compute Z, adjusts for idea difficulty, but does not hold capability fixed. If the observer attributes all adjusted progress unexplained by physical compute to efficiency, the resulting residual estimator is

$$\mathrm{d}\ln\widehat{\nu} \equiv \frac{\mathrm{d}\ln g_A + \beta\,\mathrm{d}\ln A}{\varepsilon_{R,X}} - \mathrm{d}\ln Z. \tag{18}$$

Proposition 5 (Residual Efficiency Bias)**.** *Wherever $\varepsilon_{R,X} > 0$, the residual estimator satisfies*

$$\mathrm{d}\ln\widehat{\nu} = \mathrm{d}\ln\nu + \frac{s_{R,\alpha}}{\varepsilon_{R,X}}\,\mathrm{d}\alpha. \tag{19}$$

Below the saturation threshold, $s_{R,\alpha} > 0$, so a capability improvement biases estimated efficiency improvements upward. The bias vanishes on a fixed task set and after allocation saturates.

Proof. See Appendix A.5.

The bias has a direct interpretation: a broad progress measure credits the research contribution of newly usable tasks to $\widehat{\nu}$, even though the model assigns that contribution to capability improvements. An evaluation on a fixed task set imposes $\mathrm{d}\alpha = 0$ and isolates efficiency improvements.

6 Extensions

The two-margin framework makes several additional questions expressible.

1. **Joint difficulty.** The baseline holds efficiency constant across automated tasks. A natural extension lets the tasks reached later by capability improvements also require more compute per unit of output. This creates a direct dependence between the two legs: capability improvements expose increasingly expensive tasks, so the efficiency threshold depends on the joint tail of task difficulty and compute cost rather than on η alone.

2. **Directed innovation.** Research effort can be allocated between improving capability and improving efficiency. Because self-sustaining acceleration requires both margins, private incentives to improve the currently scarcer input need not coincide with the allocation that most rapidly relaxes the binding bottleneck. Adding this choice would turn (μ, η) into endogenous variables and permit analysis of when labs favor capability, efficiency, or a balanced portfolio.

3. **Inference scaling.** There is some evidence that models unlock higher capabilities when given larger inference budgets [AISI, 2026]. This can be modelled by allowing α to depend on a task-specific compute budget, with the total compute stock still fixed. Inference scaling converts efficiency into capability because efficiency improvements free compute that can be used to reach harder tasks.

4. **Distillation.** Distillation transfers the capability of a more capable model to a cheaper model and therefore converts capability into efficiency—the inverse of inference scaling. This gives distillation a special power to relax the efficiency constraint on an SIE. In scenarios where capability improvements are fast but efficiency improvements are slow, distillation can unlock an SIE that would otherwise be bottlenecked by the fixed compute stock.

These are all questions that cannot be expressed within a benchmark AI R&D model, but can be expressed once capability and efficiency are modelled separately. Answering them is the next step for this project.

References

Philippe Aghion, Benjamin F. Jones, and Charles I. Jones. Artificial intelligence and economic growth. In Ajay Agrawal, Joshua Gans, and Avi Goldfarb, editors, *The Economics of Artificial Intelligence: An Agenda*, pages 237–282. University of Chicago Press, 2019.

AISI. More compute, more capability: Why AI agent evaluations need to account for test-time compute. AISI blog, July 2026. Accessed: 2026-07-20.

Tom Cunningham and Manish Shetty. An apple-picking model of AI R&D. Blog post, https://tecunningham.github.io/posts/2026-03-13-apple-picking-ai.html, 2026.

Epoch AI. Epoch capabilities index: Methodology. https://epoch.ai/data/eci-documentation/methodology, 2026.

Daniel Eth and Tom Davidson. Will AI R&D automation cause a software intelligence explosion? Technical report, Forethought Research, 2025.

Marina Favaro and Jack Clark. When ai builds itself. *Anthropic Institute, June*, 4:2026, 2026.

Anson Ho, Tamay Besiroglu, Ege Erdil, David Owen, Robi Rahman, Zifan Carl Guo, David Atkinson, Neil Thompson, and Jaime Sevilla. Algorithmic progress in language models. Technical report, arXiv preprint arXiv:2403.05812, 2024.

Benjamin F. Jones and Xiaojie Liu. A framework for economic growth with capital-embodied technical change. *American Economic Review*, 114(5):1448–1487, 2024.

Charles I. Jones. R&d-based models of economic growth. *Journal of Political Economy*, 103 (4):759–784, 1995.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. Technical report, arXiv preprint arXiv:2503.14499, 2025.

Parker Whitfill and Cheryl Wu. Will compute bottlenecks prevent an intelligence explosion? Technical report, arXiv preprint arXiv:2507.23181, 2025.

A Proofs

A.1 Task allocation

Proof. Since $\rho < 0$, maximizing R is equivalent to minimizing $\Phi \equiv \int_0^1 r(i)^\rho \, \mathrm{d}i$, and all tasks are staffed at an optimum.

Step 1 (Within-Block Equalization). Let $\ell_A \equiv \int_0^\alpha \ell$ be labor assigned to automated tasks, so the automated block has total input $X + \ell_A$ and the human-only block has $L - \ell_A$. The map $s \mapsto s^\rho$ is strictly convex on $(0, \infty)$ (its second derivative is $\rho(\rho - 1)s^{\rho-2} > 0$ because $\rho < 0$), so by Jensen's inequality, for a block of measure m with total input T, $\int_{\text{block}} r^\rho \geq m \, (T/m)^\rho = m^{1-\rho} T^\rho$, with equality iff r is constant on the block; the constant profile is feasible because inputs are perfect substitutes within tasks. Hence

$$\Phi(\ell_A) \;=\; \alpha^{1-\rho}(X + \ell_A)^\rho + (1-\alpha)^{1-\rho}(L - \ell_A)^\rho, \qquad 1 - \rho = \tfrac{1}{\sigma}$$

Step 2 (Labor Across Blocks). Write $r_A = (X + \ell_A)/\alpha$ and $r_N = (L - \ell_A)/(1-\alpha)$ for per-task inputs. Differentiating,

$$\Phi'(\ell_A) \;=\; \rho\left[\alpha^{1-\rho}(X+\ell_A)^{\rho-1} - (1-\alpha)^{1-\rho}(L-\ell_A)^{\rho-1}\right] \;=\; \rho\left[r_A^{\rho-1} - r_N^{\rho-1}\right]$$

Since $\rho < 0$ and $s \mapsto s^{\rho-1}$ is strictly decreasing, $\Phi' < 0$ exactly when $r_A < r_N$: labor flows toward the block with lower per-task input. The interior solution equalizes $r_A = r_N$, i.e. $\ell_A^* = \alpha L - (1-\alpha)X$, which is nonnegative iff $\alpha \geq \alpha^*$; then $r(i) \equiv L + X$ and $R = L + X$. If $\alpha < \alpha^*$ then $r_A > r_N$ already at $\ell_A = 0$ (as $X/\alpha > L/(1-\alpha)$ iff $\alpha < \alpha^*$), so $\Phi'(0) > 0$ and the corner $\ell_A = 0$ is optimal, giving the first branch of (8) with $1 - \rho = 1/\sigma$. □

A.2 Bottleneck bounds

Proof. Compute-Abundant Limit. Fix $\alpha \in (0,1)$. Because $\alpha^* = X/(L+X)$ converges to one, the first branch of (8) applies for all sufficiently large X. Since $\rho < 0$, X^ρ converges to zero, and therefore

$$\lim_{X\to\infty} R(\alpha; L, X) = \left[(1-\alpha)^{1/\sigma} L^\rho\right]^{1/\rho} = \frac{L}{(1-\alpha)^{1/(1-\sigma)}}$$

Upper Bounds. $R \leq \int_0^1 r = L + X$ is the power-mean inequality ($M_\rho \leq M_1$ for $\rho \leq 1$). For the second: for any feasible allocation, the human-only block receives only labor, total at most L, so by Jensen (as in Proposition 1) $\Phi \geq \int_\alpha^1 r^\rho \geq (1-\alpha)^{1-\rho}\left(\int_\alpha^1 r\right)^\rho \geq (1-\alpha)^{1-\rho} L^\rho$, the last step because $s \mapsto s^\rho$ is decreasing and $\int_\alpha^1 r \leq L$. Raising to $1/\rho < 0$ flips the inequality: $R \leq L(1-\alpha)^{(1-\rho)/\rho} = \overline{R}(\alpha)$, using $(1-\rho)/\rho = 1/(\sigma-1)$.

Lower Bound. On the corner branch ($\alpha \leq \alpha^*$), $\Phi = T_A + T_N$ with $T_A = \alpha^{1/\sigma} X^\rho$, $T_N = (1-\alpha)^{1/\sigma} L^\rho$. Then $\Phi \leq 2\max\{T_A, T_N\}$, so

$$\begin{aligned} R &\geq 2^{1/\rho} \min\left\{T_A^{1/\rho}, T_N^{1/\rho}\right\} \\ &= 2^{-\frac{\sigma}{1-\sigma}} \min\left\{\alpha^{-\frac{1}{1-\sigma}} X, \overline{R}(\alpha)\right\} \\ &\geq 2^{-\frac{\sigma}{1-\sigma}} \min\left\{X, \overline{R}(\alpha)\right\}, \end{aligned}$$

using $1/\rho = -\sigma/(1-\sigma)$ and $\alpha \leq 1$. On the flat branch, $R = L + X \geq X \geq \min\{X, \overline{R}(\alpha)\} \geq$ the left side. □

A.3 SIE conditions

Proof. If $\mu = 0$, then $\alpha(A) = 0$ and $R = L$, so the elasticity of R with respect to A is zero. Moreover, $\dot{A} = \delta L A^{1-\beta}$ cannot diverge in finite time because $\beta > 0$. Suppose $\mu > 0$.

On the corner branch, let

$$T_A = \alpha^{1/\sigma} X^{\rho}, \qquad T_N = (1-\alpha)^{1/\sigma} L^{\rho}, \qquad \omega_A = \frac{T_A}{T_A + T_N}, \qquad \omega_N = 1 - \omega_A$$

Differentiating the exact allocation formula gives

$$\frac{\mathrm{d}\ln R}{\mathrm{d}\ln A} = \frac{1}{\rho}\left[\omega_A\left(\rho\eta + \frac{1}{\sigma}\frac{\mathrm{d}\ln\alpha}{\mathrm{d}\ln A}\right) + \omega_N \rho \frac{\mu}{1-\sigma}\right] \tag{20}$$

The extra term from α converges to zero. If $\eta < \mu/(1-\sigma)$, then ω_A converges to one; if $\eta > \mu/(1-\sigma)$, then ω_N converges to one; and if the two elasticities are equal, both terms in brackets converge to $\rho\eta$. Thus the elasticity of R converges to $\min\{\eta, \mu/(1-\sigma)\}$.

On the flat branch, $R = L + X$ and its elasticity is $\eta X/(L+X)$. This converges to η when $\eta > 0$ and equals zero when $\eta = 0$. The flat branch can persist at high A only when the effective-compute elasticity is the smaller one, so the limit is again $\min\{\eta, \mu/(1-\sigma)\}$.

Finally, the law of motion gives

$$\frac{\mathrm{d}\ln g_A}{\mathrm{d}\ln A} = \frac{\mathrm{d}\ln R}{\mathrm{d}\ln A} - \beta \longrightarrow \min\left\{\eta, \frac{\mu}{1-\sigma}\right\} - \beta$$

Let $m \equiv \min\{\eta, \mu/(1-\sigma)\}$. Inspection of the same two branches shows more strongly that R/A^m converges to a positive finite constant. Hence whenever $m = \beta$, the growth rate $g_A = \delta R A^{-\beta}$ converges to a positive constant.

Definition 1 is therefore satisfied exactly when both inequalities in (15) hold.

The same conditions characterize the finite-time definition. Because $X(A) = \nu Z A^{\eta}$ and $\overline{R}(\alpha(A)) = L A^{\mu/(1-\sigma)}$, Proposition 2 implies that, for all sufficiently large A, there are constants $0 < c < C < \infty$ such that

$$cA^m \leq R(\alpha(A); L, X(A)) \leq CA^m.$$

If $m > \beta$, then $\dot{A} \geq \delta c A^{1+(m-\beta)}$, so comparison with a superlinear power law gives divergence in finite time. If $m \leq \beta$, then $\dot{A} \leq \delta C A^{1+(m-\beta)} \leq \delta C A$ for $A \geq 1$, which rules out finite-time divergence. Thus finite-time divergence also occurs exactly when both inequalities in (15) hold. □

A.4 Growing compute

Proof. Taking the time derivative of $\ln X_t = \ln \nu + \eta \ln A_t + \ln Z_t$ gives

$$g_X = \eta g_A + g_Z = g_\nu + g_Z$$

Because g_Z is constant, $\dot{g}_X = \dot{g}_\nu$. Exponential physical compute growth therefore contributes no acceleration to effective compute; any rise in its growth rate comes from efficiency. The capability path $\alpha(A) = 1 - A^{-\mu}$ contains no compute term. Holding the exogenous compute path fixed consequently leaves both feedback elasticities in Proposition 3 unchanged, so its two conditions continue to govern self-sustaining acceleration.

For completeness, if $\mu/(1-\sigma) \leq \beta$, Proposition 2 gives $g_A \leq \delta L A^{\mu/(1-\sigma)-\beta}$ for every compute path. If $\mu/(1-\sigma) > \beta$ but $\eta < \beta$, the compute-limited dynamics balance exponential input growth against the exponent $\beta - \eta$, producing the limiting growth rate $g_Z/(\beta - \eta)$. At $\eta = \beta$, a rising observed growth rate is tied to the continuing rise in Z_t and disappears when that path is frozen. □

A.5 Residual efficiency bias

Proof. Substituting the accounting identity (17) into the residual estimator (18), dividing by $\varepsilon_{R,X} > 0$, and canceling $\mathrm{d}\ln Z$ gives

$$\mathrm{d}\ln \widehat{\nu} = \mathrm{d}\ln \nu + \frac{s_{R,\alpha}}{\varepsilon_{R,X}}\,\mathrm{d}\alpha,$$

which is (19).

It remains to establish the sign of the bias. Below the saturation threshold, Proposition 1 gives $R = \Phi^{1/\rho}$, where

$$\Phi = \alpha^{1-\rho} X^\rho + (1-\alpha)^{1-\rho} L^\rho.$$

Differentiating with respect to α yields

$$\frac{\partial \Phi}{\partial \alpha} = (1-\rho)\left[\left(\frac{X}{\alpha}\right)^\rho - \left(\frac{L}{1-\alpha}\right)^\rho\right].$$

The corner condition $\alpha < \alpha^* = X/(L+X)$ is equivalent to $X/\alpha > L/(1-\alpha)$. Since $\rho < 0$, this implies $\partial\Phi/\partial\alpha < 0$. Because $1/\rho < 0$, it follows that $\partial R/\partial\alpha > 0$, and hence $s_{R,\alpha} > 0$. Therefore $\mathrm{d}\alpha > 0$ creates an upward bias. On a fixed task set $\mathrm{d}\alpha = 0$; after saturation, $R = L + X$ is independent of α, so $s_{R,\alpha} = 0$. □