Can AI Developers' Profits Survive Open-Weight Competition?

Thomas J. May *

California State University, Fullerton

November 8, 2026

ABSTRACT

Large language models (LLMs) are an easily imitable technology. In a task-assignment model with imitation, I estimate that the capabilities of open-weight models, which can be used for free in place of proprietary models, lag the proprietary frontier by only about five months. If open-weight and proprietary models are equally costly to serve, each automated task must repay its development cost within the roughly five months before open-weight models imitate it. This raises a puzzle: what must we assume for the observed level of investment in AI models to be profitable?

*Author's Email: tmay@fullerton.edu. Author's Website: MayEcon.com.

1 Introduction

The capabilities of large language models (LLMs), the heart of the current generation of AI technologies, have been rapidly growing. At the same time, open-weight models, models whose weights are freely available to use and build on, have been keeping up with the proprietary frontier, lagging behind the best proprietary model by approximately five months on average.[1] All LLM developers have access to the same building blocks. Knowledge distillation allows developers to train their models on the outputs of a superior model. Additionally, formal intellectual property offers proprietary developers little protection against imitation. This poses a challenge to the profitability of proprietary model developers. Customers could avoid paying high prices by employing "good enough" open-weight models.[2]

This paper makes three contributions. First, it adds imitation (Aghion, Harris, Howitt and Vickers, 2001) to task-based models of automation (Acemoglu and Restrepo, 2018). In existing task models, the developer of an automation technology captures the gains from a task indefinitely. This can overstate the reward from automation if tasks can be imitated by a competitive fringe. Second, it shows how to estimate the imitation rate δ from AI capability data. I map capability indices into the model's law of motion for imitated tasks and estimate the imitation rate structurally, which turns a descriptive lag into the parameter that governs the return to automation. I estimate that open-weight models imitate a proprietary advance at a rate of about 2.3 per year, a mean lag of a little over five months. Third, the estimate yields a break-even condition for the AI investment boom (Van Nieuwerburgh, 2026): absent a lasting cost advantage, a proprietary developer must recover a task's development cost before open-weight models imitate it, about five months.

2 Model

Time is continuous and firms discount the future at rate r. Output combines a continuum of tasks $i \in [0,1]$ through a Cobb-Douglas aggregator $\log Y = \int_0^1 \log y_i \, \mathrm{d}i$. Each task can be performed by labor or by AI. Within the AI sector, there is a representative proprietary developer and a representative open-weight developer. The proprietary developer attempts to automate tasks. Tasks are ordered so that the developer automates them in sequence: at time t, tasks $i \leq I_t$ are automated and the rest are performed by labor. The proprietary developer chooses the rate x_t at which it automates new tasks, $\dot{I}_t = x_t$, at flow cost $C(x_t, I_t)Y$. Cost rises with I_t because the remaining tasks are harder

[1] The sample period here starts with the July 2024 release of Llama 3.1, and the lag is measured on a variety of capability indices. More details on capability indices are covered in Section 3 and the data on the open-weight lags can be found in Appendix A, Table A.1.

[2] While the average consumer might have a higher willingness to pay for convenience, larger, cost-minimizing businesses could rent servers and host models for a share of their tasks and pay for proprietary developers' models only for frontier tasks. The result would be a significant reduction in revenue for the proprietary developers.

to automate.[3] The open-weight developer simply imitates the proprietary developer[4]: each task the proprietary developer holds alone is imitated at an exogenous Poisson rate δ. Let P_t denote the measure of automated tasks not yet imitated. The law of motion of P_t is $\dot{P}_t = x_t - \delta P_t$.

The marginal cost of a task with labor is w. The proprietary developer has marginal cost c_P, and the open-weight developer has marginal cost c_O. I assume that $c_P \leq c_O < w$.[5] When a task is first automated, the proprietary developer limit prices against labor, setting $p_i = w$ and earning a per-unit profit of $w - c_P$. Because open weights are free, once a task is imitated a competitive fringe of open-weight providers stands ready to serve it at cost c_O, so the proprietary developer limit prices at $p_i = c_O$. Define the proprietary developer's profit margin before imitation as $m_P = \frac{w-c_P}{w}$ and its margin after imitation as $m_O = \frac{c_O-c_P}{c_O}$. The proprietary developer's flow profits are therefore

$$\Pi_t = m_P P_t Y + m_O(I_t - P_t)Y. \tag{1}$$

Writing the value of the proprietary developer as $V(I,P) = v(I,P)Y$,

$$rv(I,P) = \max_x \; m_P P + m_O(I-P) - C(x,I) + x\big[v_I(I,P) + v_P(I,P)\big] - \delta P\, v_P(I,P). \tag{2}$$

The reward for automating a task is

$$a = \frac{1}{r+\delta}\left(m_P + \frac{\delta}{r} m_O\right). \tag{3}$$

Consider a counterfactual economy with $\delta = 0$. The automation reward in that counterfactual economy is $\bar{a} = \frac{m_P}{r}$. It follows that

$$\frac{\bar{a}}{a} = \frac{r+\delta}{r+\theta\delta}, \qquad \theta = \frac{m_O}{m_P}. \tag{4}$$

If costs are equal between the two developers ($m_O = 0$), then a model that omits imitation overstates the reward from automation by a factor of $1 + \frac{\delta}{r}$.

3 Measuring the Catch-Up Rate

The model gives us a natural measurement framework to estimate δ. Let $O_t = I_t - P_t$ denote the measure of tasks imitated by the open-weight developer. It has the law of motion

$$\dot{O}_t = \delta(I_t - O_t). \tag{5}$$

[3]Cottier, Rahman, Fattorini, Maslej, Besiroglu and Owen (2024) estimate that the amortized cost of training frontier models has risen by a factor of 2.4 per year since 2016. I interpret this as rising automation difficulty.

[4]This excludes the possibility that the open-weight developer automates a task not already automated by the proprietary developer. Including that possibility would strengthen this paper's conclusion.

[5]I hold w fixed. In general equilibrium, automation lowers the wage, which lowers the limit price and would further reduce proprietary profits.

Table 1: Estimated imitation rate

	Imitation rate δ (per year)		Mean lag $12/\delta$ (months)	
Index	Estimate	95% CI	Estimate	95% CI
ECI	2.27	[1.91, 2.76]	5.3	[4.4, 6.3]
Math-ECI	2.25	[1.91, 2.75]	5.3	[4.4, 6.3]
SWE-ECI	2.09	[1.67, 2.69]	5.7	[4.5, 7.2]

I briefly summarize how I estimate δ here; details are in Appendix B.

We cannot measure I_t or O_t directly. Instead, AI models are evaluated on a series of benchmarks. I interpret each benchmark as a collection of tasks. Instead of focusing on performance on a single benchmark, I employ the Epoch Capabilities Index (ECI) developed by Ho, Denain, Atanasov, Albanie and Shah (2025), which combines models' scores on several dozen benchmarks into one capability score on a common scale. Epoch also documents a method for building domain subindices, which I use to construct math and software engineering subindices. Let ECI_t^I and ECI_t^O denote the ECI scores for the frontier proprietary model and the frontier open-weight model, respectively. Scores change only when new models are released, so the proprietary frontier is a step function. I assume there is some common, increasing, invertible function f linking each side's task set to its ECI score. For my main specification, I assume f is linear. Let $g = f^{-1}$.

The proprietary frontier is treated as observed, $I_t = g(\text{ECI}_t^I)$, which is a step function. I then propagate O_t from the start of the sample using (5). Each open-weight record i, released at time t_i, is a noisy reading of the open-weight frontier: $y_i = f(O_{t_i}) + \varepsilon_i$, where y_i is the record's ECI score. I estimate δ and the parameters of the measurement equation jointly by maximum likelihood, with 95% confidence intervals from the profile likelihood, and report the results in Table 1. I repeat the estimation for the math and software engineering subindices.

4 A Profit Puzzle

Returning to (4) with the estimated δ and $r = 0.05$, Panel (a) of Figure 1 shows the overstatement factor $\frac{\bar{a}}{a}$ as a function of the share of margin kept after imitation. If the two developers have the same marginal cost ($c_P = c_O$), a standard task-assignment model would overstate the reward from automation by a factor of 46.4. If the proprietary developer keeps some of its margin after imitation, the factor falls, because the developer retains a permanent source of profit. Panel (b) of Figure 1 shows the flow profit, in terms of development costs K, the proprietary developer must earn while it is ahead to break even: it falls from 2.32 when the developer keeps nothing after imitation ($\theta = 0$) to 0.42 when it keeps a tenth of its margin, approaching the no-imitation benchmark of $r = 5\%$ as $\theta \to 1$. Under the assumption that $c_P \ll w$, $\theta \approx 1 - \frac{c_P}{c_O}$. Figure 2 shows the break-even profit as a function of the cost ratio. If open-weight providers pay 10% more than

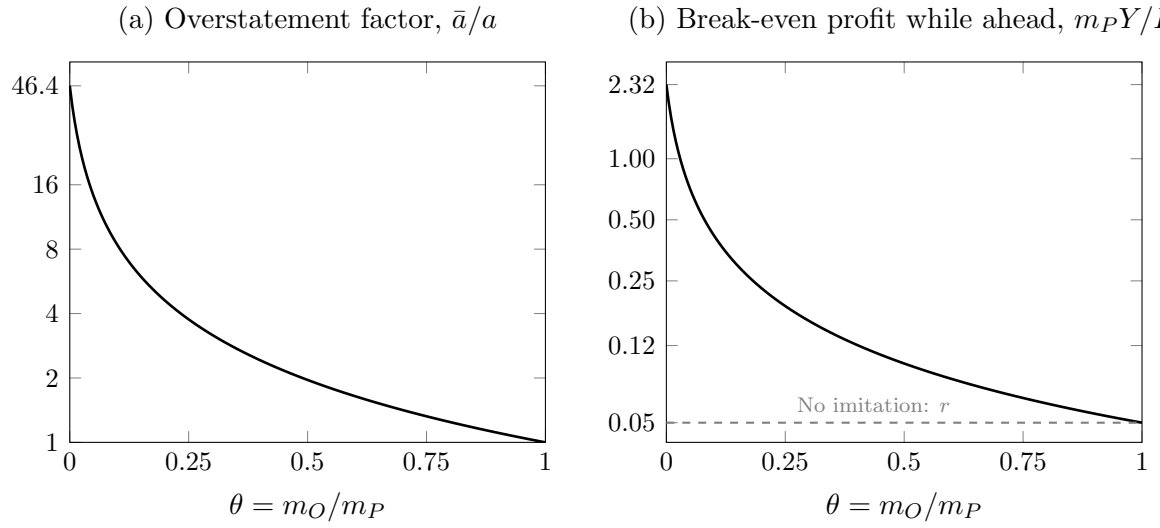


Figure 1: Overstatement and break-even profit by the margin kept after imitation. Both panels plot the y-axis on a log scale.

the proprietary developer to serve a task ($c_O = 1.1\, c_P$), the proprietary developer breaks even only if, while it is ahead, it earns a flow profit of at least 45% of its development cost per year. An open question is whether any persistent advantage can survive competition from unaffiliated hyperscalers renting the same hardware that proprietary developers use. Additionally, neither figure accounts for the investment costs of lowering the proprietary developer's marginal costs. These costs could offset the reduction in break-even profits, meaning the results should be treated as a lower bound on required profits.

Another possibility is that there is some set of tasks that proprietary model developers will be able to automate but that are not easily imitable. In the model, this corresponds to an imitation rate that varies across tasks, $\delta(i)$. If $\delta(i)$ falls sharply for some range of tasks, the proprietary developer's profits would come almost entirely from that range, and the case for investing in automation would rest on reaching it rather than on the tasks automated so far. If such tasks were already being automated, we would expect the gap between the two frontiers to widen on the benchmarks that measure them. The estimates do not show this: the imitation rate is nearly identical across the overall, math, and software engineering indices (Table 1), and the time lag has stayed within a few months throughout the sample (Table A.1).

A world where technologies like large language models end up being primarily open-weight is not out of the question. Consider the servers these models are trained and run on. Those servers most likely run Linux, an open-source operating system that beat out proprietary competition in the server operating system market. Linux did not end profits in computing. It moved them from the operating system to the hardware and cloud services that run it, and the profits from AI may follow the same path, away from the developers of the models.

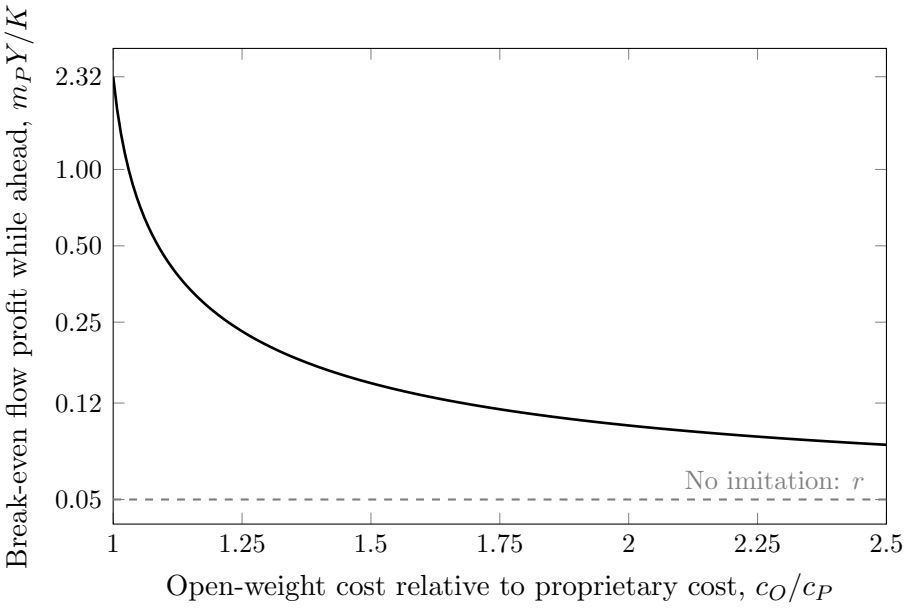


Figure 2: Cost advantage needed to break even. The y-axis is on a log scale.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the author used Claude in order to review code and proofread the text. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

Acemoglu, Daron and Pascual Restrepo, "The Race between Man and Machine: Implications of Technology for Growth, Factor Shares, and Employment," *American Economic Review*, 2018, *108* (6), 1488–1542.

Aghion, Philippe, Christopher Harris, Peter Howitt, and John Vickers, "Competition, Imitation and Growth with Step-by-Step Innovation," *The Review of Economic Studies*, 07 2001, *68* (3), 467–492.

Cottier, Ben, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen, "The Rising Costs of Training Frontier AI Models," 2024.

Ho, Anson, Jean-Stanislas Denain, David Atanasov, Samuel Albanie, and Rohin Shah, "A Rosetta Stone for AI Benchmarks," 2025.

Van Nieuwerburgh, Stijn, "Financing the AI Buildout," *Brookings Papers on Economic Activity*, 2026. Fall 2026, forthcoming.

A Additional Tables and Figures

Table A.1: Measuring open-weight model lag[6]

Panel A: New open-weight bests on ECI since Llama 3.1-405B

Open-weight model			Proprietary model matched		Lag
Release	Model	Score	Release	Model	(months)
Jul 2024	Llama 3.1-405B	128.8	May 2024	GPT-4o	2.3
Sep 2024	Qwen2.5-72B	129.0	May 2024	GPT-4o	4.2
Dec 2024	Phi-4	130.5	Sep 2024	o1-mini	3.0
Dec 2024	DeepSeek-V3	132.3	Sep 2024	o1-mini	3.4
Jan 2025	DeepSeek-R1	138.8	Dec 2024	o1	1.1
Apr 2025	Qwen3-235B-A22B	139.2	Dec 2024	o1	4.3
May 2025	DeepSeek-R1 (May 2025)	141.1	Dec 2024	o1	5.3
Jul 2025	Qwen3-235B-A22B-Thinking	143.8	Mar 2025	Gemini 2.5 Pro	3.8
Sep 2025	DeepSeek-V3.2-Exp	144.9	Apr 2025	o3	5.5
Nov 2025	Kimi K2 Thinking	145.8	Apr 2025	o3	6.7
Dec 2025	DeepSeek-V3.2	146.1	Apr 2025	o3	7.5
Jan 2026	Kimi K2.5	148.1	Aug 2025	GPT-5	5.7
Apr 2026	GLM-5.1	149.8	Aug 2025	GPT-5	8.0
Apr 2026	Kimi K2.6	150.9	Nov 2025	Gemini 3 Pro	5.0
Jun 2026	GLM-5.2	151.9	Nov 2025	Gemini 3 Pro	6.9
Jul 2026	Kimi K3	157.9	Mar 2026	GPT-5.4 Pro	4.4

Panel B: Lag in months by measure

Measure	Bests	Mean (median)	SD	Range; latest
ECI	16	4.8 (4.7)	1.9	1.1 to 8.0; 4.4
Math-ECI	14	3.9 (3.6)	2.2	0.2 to 8.0; 4.4
SWE-ECI	14	4.8 (5.2)	2.6	1.1 to 8.1; 1.2

B Estimation Details

This appendix describes how the imitation rate δ reported in Table 1 is estimated.

B.1 Data

Capability scores come from Epoch AI's benchmark database (accessed October 30, 2026). The ECI is the published Epoch Capabilities Index (Ho et al., 2025), scaled

[6]The lag at a new open-weight best is the time since a proprietary model first reached the same score. Panel A lists every open-weight model that set a new open-weight best on ECI from the release of Llama 3.1-405B onward, the first proprietary model to reach that score, and the lag between their release dates. Panel B summarizes these lags for each measure: the number of new open-weight bests, the mean and median lag, the standard deviation of the lag, the range, and the latest lag. Release dates are from Epoch AI. ECI scores carry a standard error of about two points, roughly two months of frontier progress, so individual lags are uncertain by a month or two.

so that Claude 3.5 Sonnet scores 130 and GPT-5 scores 150. I rebuild the math and software engineering subindices with Epoch's documented domain-specific method: each benchmark's difficulty and slope are held at their values from the general ECI fit, and each model's capability is refit by least squares on the benchmarks Epoch assigns to the domain, for models with at least two such benchmarks. Every score comes with a 90% interval: Epoch's bootstrap interval for the ECI, and for the subindices an interval from Epoch's 500 bootstrap draws of the benchmark parameters combined with 10 resamples of the model's domain benchmarks. I convert each interval to a standard error, $s_i = (\text{upper}_i - \text{lower}_i)/(2 \times 1.645)$. The median standard error is about 2.0 points on the ECI and 1.7 points on the subindices.

Models are dated at their release date and classified using Epoch's accessibility field. A model is open-weight if its weights can be downloaded, including under restricted-use or non-commercial licenses, and proprietary if it is available only through an API or hosted service or is unreleased. Twenty-nine of the 538 model scores have missing or ambiguous labels and are classified by hand.

B.2 Frontier records

On each index and for each side, I keep only records: a model's score enters only if it exceeds the best score so far among models of the same type, with models released on the same date ordered by score. The sample on each index begins with the first proprietary record, GPT-4 (March 2023) on the ECI and Math-ECI and Claude 3 Opus (February 2024) on the SWE-ECI. Open-weight records enter the likelihood only if released after that date. Table B.1 reports the number of records on each side.

B.3 Measurement model

With f linear, each index point represents the same measure of tasks, and because δ is unchanged by rescaling tasks, the law of motion can be estimated directly on index scores. The proprietary frontier is treated as observed: $I_t = \text{ECI}_t^I$, the latest proprietary record, so I_t is a step function. Between proprietary records, with $I_t = I_j$ on $[\tau_0, \tau_1]$, the law of motion $\dot{O}_t = \delta(I_t - O_t)$ has the exact solution

$$O_{\tau_1} = I_j + (O_{\tau_0} - I_j)\, e^{-\delta(\tau_1 - \tau_0)}. \tag{B.1}$$

I propagate O_t from the start of the sample through each proprietary record date and each open-weight record date using (B.1). Each open-weight record i, released at t_i with score y_i, is a noisy reading of the open-weight frontier,

$$y_i = O_{t_i} + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}\big(0, \sigma^2 + s_i^2\big)\,, \tag{B.2}$$

independently across records. The parameter σ captures model-specific deviations of an open-weight record from the open-weight frontier beyond the measurement error in its

score. The log likelihood of one index is

$$\ell(\delta, \gamma, \sigma) = -\frac{1}{2} \sum_i \left[\log\big(2\pi(\sigma^2 + s_i^2)\big) + \frac{(y_i - O_{t_i})^2}{\sigma^2 + s_i^2} \right]. \tag{B.3}$$

B.4 Initial condition

The open-weight frontier at the start of the sample, O_0, is not observed. I parameterize it as $O_0 = I_0 - e^{\gamma}$, where I_0 is the proprietary record in force at the start and γ is unrestricted. This imposes $O_0 < I_0$: the open-weight side starts behind the frontier, as the model requires ($P_0 > 0$). Because I_t is nondecreasing, O_t then stays below I_t throughout, so the fitted open-weight frontier never passes the proprietary frontier.

B.5 Estimation and inference

I maximize (B.3) over $(\log \delta, \gamma, \log \sigma)$ by L-BFGS with exact gradients from automatic differentiation, from twelve starting values ($\delta \in \{0.5, 1, 2.5, 6\}$ crossed with three starting gaps), keeping the highest likelihood. The 95% confidence interval for δ comes from the profile likelihood: it is the set of δ at which the likelihood maximized over (γ, σ) lies within $\chi^2_{1,0.95}/2 \approx 1.92$ of its maximum, with endpoints found by bisection. The mean lag is $12/\delta$ months, and its interval inverts the interval for δ. This lag is the expected time until a given task is imitated; when the proprietary frontier grows linearly in tasks, it also equals the horizontal distance between the two frontiers. It is therefore a model-based counterpart to, not the same object as, the descriptive lags in Table A.1.

B.6 A common imitation rate

The three indices measure overlapping sets of tasks, so the model implies a common δ. I estimate it by letting the starting gap and σ differ by index and maximizing the sum of the indices' profile likelihoods over δ. The common estimate is $\delta = 2.22$ (95% CI $[2.00, 2.49]$), a mean lag of 5.4 months. A likelihood-ratio test of a common δ against index-specific values gives a statistic of 0.38 on 2 degrees of freedom ($p = 0.83$). The subindices share benchmarks with the ECI, so their errors are not independent: the pooled interval is too narrow and the test is too likely to reject, which makes the failure to reject stronger evidence of a common rate.

Table B.1: Estimation samples

		Records	
Index	Sample start	Proprietary	Open-weight
ECI	Mar 2023	23	24
Math-ECI	Mar 2023	22	19
SWE-ECI	Feb 2024	16	15

Notes: Proprietary records include the record in force at the sample start; open-weight records are those released after it.