FRONTIER SCALING LAWS FOR AGENTIC COST: MEASURING DIMINISHING RETURNS IN AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

The last few years have seen a dramatic acceleration in AI capabilities, most notably in the time horizon metric, which relates the success rate of an AI agent to the number of hours it would take a human to complete the same task. When measured in human-hours, the tasks an AI agent can complete 50% of the time have been doubling approximately every 4 months for the last three years. At the same time the direct inference-time cost of running an AI agent has also experienced rapid growth. Thus, the question remains: which of these two is growing more quickly? That is, for every doubling of time horizon, by what factor does inference-time cost grow? Building on methods from item response theory, we show that at the price-performance frontier there are significant diminishing returns to paying more for AI inference. In particular, we find that cost scales as a superquadratic power law in time horizon.

1 INTRODUCTION

Scaling the size of neural networks, and the amount of compute they use at inference time has led in the last few years to a dramatic acceleration of AI capabilities. AI systems have gone from struggling with grade school mathematics (Cobbe et al., 2021), to winning gold at the IMO (Luong et al., 2025), and just recently solving a Clay Mathematics institute millennium problem with a natural-language proof of over 100 pages along with a formally verified lean certificate (OpenAI, 2026). This rate of progress is both remarkable and concerning, and thus projecting forward the growth rate of AI capabilities is a critical problem, with the pressing need to inform both researchers and society as a whole on how this technology will progress.

Two principled methods for this type of forward projection were introduced by Kinniment et al. (2025) and Ho et al. (2025). The former measures the capabilities of an AI system by the maximum length of task–measured in hours it would take a human to complete–that the AI can solve 50% of the time. The latter provides a method grounded Item Response Theory (IRT) for producing a single, aggregate latent capabilities score, that can be tracked over time. Both methods point to exponential growth in AI capabilities as time progresses.

However, in the real world all exponential growth must at some point end, and so the question remains how to determine where AI progress is going in the future. In particular, the exponential growth in AI capabilities has been accompanied by exponential growth in both model size and in the amount of thinking done by the model at inference time. A model's size times the number of inference time tokens used is proportional to the computational cost in FLOPs. Model size is also growing exponentially over time, and inference compute scaling has been a key driver of recent performance growth. The picture is further complicated by the fact that algorithmic improvements are also exponentially reducing the model size required for a given capability level, while hardware improvements have reduced the cost for a given number of FLOPs. With all these increasing and decreasing exponentials working against each other, it becomes unclear how we should think about the future trajectory of AI capabilities and their economic and societal impact.

In this paper, we propose to address this problem by determining the scaling relationship between *the inference cost in dollars for a given level of AI performance, as measured by 50% time horizon*. We argue that this is a particularly meaningful metric, because it can be compared directly to a human baseline via hourly wages. For example, if this cost scales super-linearly in time horizon, it would be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

difficult for an AI system to completely replace a human software engineer on a task that would take weeks to complete, even if the AI system is much cheaper on individual 8 hour tasks.

To determine the cost versus time horizon scaling, we utilize hierarchical item response theory to model dollar costs as a function of latent capabilities and benchmark difficulties (Fox, 2010; Lord, 1980). We then use the tight relationship between latent capabilities and 50% time horizon to fit a power law to cost versus time horizon data, with our overall results displayed in Figure 1. We find that costs C scale super-quadratically as a function of time horizon, with a mean estimate of $C \propto T^{\gamma}$ for $\gamma = 2.81$. Such scaling demonstrates significant diminishing returns to increased inference spend.

By definition, human cost to solve a task scales linearly in human time. This suggests that there is some trade-off, where shorter-horizon tasks can be very cheaply automated with AI agents, while long-horizon autonomy remains prohibitively expensive compared to human labor. This scaling gives quantitative evidence supporting the current paradigm in agentic coding, where AI agents are used to cheaply and rapidly accomplish tasks that would take humans several hours, while at the same time requiring frequent oversight and intervention from humans to accomplish high level goals over longer time horizons.

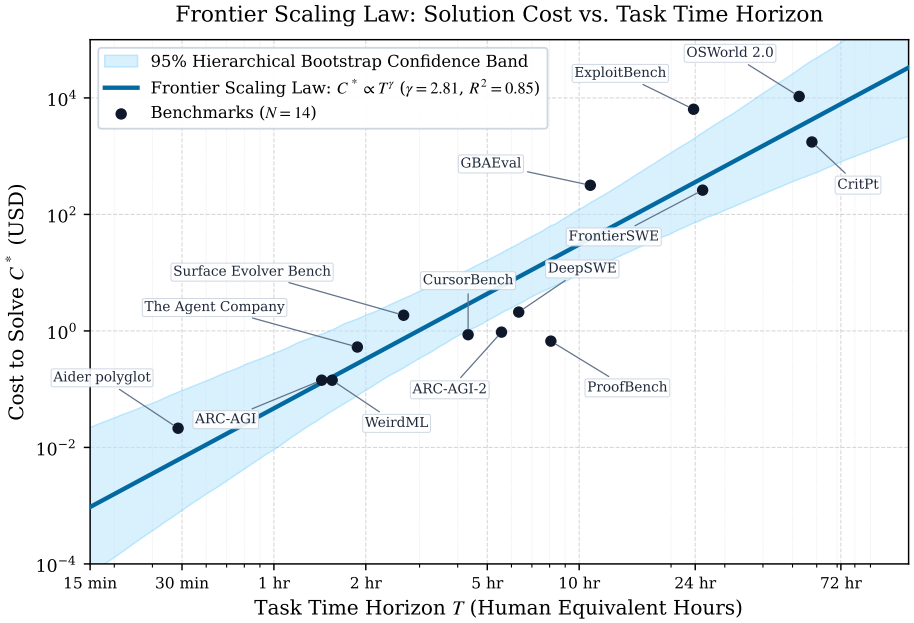


Figure 1: **Frontier Scaling Law: Solution Cost vs. Task Time Horizon.** Minimum cost C^* (in USD) required to achieve $\geq 50\%$ task completion probability as a function of human task completion time T (hours) across 14 benchmarks. Grey circles represent the individual benchmark price-vs-50%-time-horizon. The solid blue line is ordinary least-squares regression in log-log space ($\log_{10} C^* \propto 2.81 \log_{10} T$, $R^2 = 0.85$), with 95% hierarchical bootstrap confidence band $[1.96, 3.58]$. Cost scales as a superquadratic power law $C^* \propto T^{\gamma}$ in time horizon, showing significant diminishing returns to additional inference-time cost.

Critically, our estimation method identifies the minimum cost presently required to solve a task with 50% probability, and thus automatically takes into account the rapidly falling cost required to achieve a given level of performance. Despite this falling cost, we find that our estimate of the scaling exponent γ has remained remarkably stable from January 2025 to the present, with consistently tightening confidence intervals. That is, the exact superquadratic scaling exponent we observe is not an artifact of one particular set of models and benchmarks, but a consistent feature of modern AI systems built on LLMs.

We further perform stability analysis on the inclusion of the selected set of benchmarks, and compute bootstrap confidence estimates across our entire estimation pipeline, which show that superquadratic scaling holds under more than 96% of bootstrap samples. Overall, our results demonstrate the

diminishing returns to dollars spent on inference time compute, and can be an important component in forecasting if and when AIs will be able to complete tasks currently done by humans.

2 Related Work

Our methodology builds on work by Epoch AI (Ho et al., 2025) on the use of Item Response Theory to aggregate AI benchmarks, and METR (Kinniment et al., 2025) on the notion of an AI system's 50% time-horizon. These works focus on the task of predicting future AI system performance by identifying key indicators of capabilities progress and measuring how they evolve over time. We provide more details about these methods in Section 3, as they form key components of our overall methodology.

The goal of predicting modern AI system performance as compute scales up goes back to the original work on pretraining scaling laws of Kaplan et al. (2020); Hoffmann et al. (2022). At a similar time, the work of Srivastava et al. (2022) designed a large collection of over 200 diverse benchmark tasks, and utilized an average of normalized scores to report overall performance.

More recently, there has also been extensive work on benchmarking of frontier models, and aggregating data from many benchmarks in order to predict downstream task performance. The work of Ruan et al. (2024) on observational scaling laws leverages low-rank decomposition of a model vs benchmark evaluation matrix in order to identify a low dimensional subspace that captures much of the variation in model capabilities. This allows for predicting performance on new downstream tasks, simply by taking training FLOPs and observed performance on some other subset of benchmarks as input. Zhang et al. (2026b) take a different approach, and use quantile regression to predict the attainable frontier performance of an effectively post-trained model as a function of pretraining FLOPs. This allows model developers to diagnose if their post-training is sufficiently good at converting pre-training FLOPs into downstream task performance.

The work of Dominguez-Olmedo et al. (2025) demonstrates that variability in the amount of similar data that a model has encountered during training can lead to unstable model rankings across benchmarks. Building on this Zhang et al. (2026a) demonstrates that light fine-tuning on target benchmark tasks can stabilize the ranking of LLMs across benchmarks, indicating that varying levels of post-training effectiveness have significant impact on benchmark rankings.

3 Methodology

There are a priori an immense variety of ways to measure AI capabilities, via hundreds of possible benchmarks and aggregation schemes. This makes direct modelling and forecasting difficult, without some specifically grounded metrics and aggregation schemes. For the first issue, the METR 50% time horizons provide a particularly well-grounded evaluation metric (Kinniment et al., 2025). These values are determined in a two-step process. First a large dataset of software engineering tasks are collected, and human engineers are timed to determine how long each task takes. Second, an AI system attempts all the tasks, to determine the longest human-time cost task subset that it can solve 50% of the time. This value is known as the 50% time horizon, and METR documents that it has been increasing exponentially over the past several years (Kinniment et al., 2025). However, the most recent models have saturated the original METR time-horizon dataset, and so meaningful measurements of the current frontier progress are no longer possible.

For the second piece of the puzzle, the Epoch Capabilities Index (ECI) (Ho et al., 2025) provides a method grounded in Item Response Theory (IRT) for producing a single, aggregate latent capabilities score θ_i for each model i from the evaluations of many AI models across different benchmarks. Most importantly, every model need not be evaluated on every benchmark, because IRT provides a natural interpretable way to stitch together sparse evaluations of models across benchmarks into a single aggregate score for each model. There is also a strong correlation between this latent capabilities score and log time horizon. Hence, the continued linear growth of the Epoch capabilities index over time implies continued exponential growth in 50% time horizons.

Using data on average cost per task across several benchmarks, we formulate a hierarchical item response theory model in order to estimate the USD cost C of achieving a given level of latent capabilities θ. The IRT model also produces estimates of success probability on each benchmark,

even for models not tested on that benchmark. Hence, by regressing θ versus log time horizon, we are able to estimate a minimum cost C^* to achieve 50% success on each benchmark. Finally, we fit a power law to the minimum cost versus time horizon data, to achieve the results shown in Figure 1.

In the rest of this section, we introduce our hierarchical item response theory model,(Section 3.1), and describe the data we used and how it was selected (Section 3.2). We provide details of the cluster bootstrap uncertainty estimation procedure we used to estimate confidence intervals in Section B.

3.1 Hierarchical Item Response Theory Cost Model

Building on the work of Ho et al. (2025), we utilize a variant of hierarchical item response theory (Van der Linden, 2007) to model both success rate and inference time cost as a function of latent capability and benchmark difficulty.

Latent Capability and Difficulty. Let $\mathcal{M} = \{1, \ldots, M\}$ be a set of AI models, and $\mathcal{B} = \{1, \ldots, B\}$ a set of benchmarks. We view each benchmark $j \in \mathcal{B}$ as a collection of tasks across some spectrum of difficulty. Next, we define $Y_{ij} \in \{0, 1\}$ to be the indicator random variable determined by sampling a uniformly random task from benchmark j and testing if model i succeeds ($Y_{ij} = 1$) or fails ($Y_{ij} = 0$). Following standard item response theory, we model the probability distribution of Y_{ij} as a sigmoid, depending on a latent model capability $\theta_i \in \mathbb{R}$, a latent benchmark difficulty $b_j \in \mathbb{R}$, and a benchmark-specific discriminability parameter $a_j > 0$ which measures how well benchmark j is able to distinguish between different levels of model capability.

$$P(Y_{ij} = 1 \mid \theta_i, b_j, a_j) = \sigma\big(a_j(\theta_i - b_j)\big) = \frac{1}{1 + \exp\big(-a_j(\theta_i - b_j)\big)}, \tag{1}$$

Thus far, this model is identical to the standard IRT model used by Ho et al. (2025), where the fit value of θ_i is called the Epoch Capabilities Index (ECI) for each model, and the fit value of b_j is called the Epoch Difficulty Index (EDI) for each model. The model assumes that benchmark scaling and saturation follows a logistic curve, determined by how capable a model is, with significant empirical support for this modelling choice given in the original work of Ho et al. (2025). Note also that when $\theta_i = b_j$ for a given model i and benchmark j, then $P(Y_{ij} = 1 \mid \theta_i, b_j, a_j) = 0.5$ i.e. latent capability equals latent difficulty at precisely the 50% success rate threshold.

We use empirically measured benchmark scores normalized to $[0, 1]$ as our data for $P(Y_{ij} = 1 \mid \theta_i, b_j, a_j)$, and fit the model in (1) using least squares regression. The model is invariant to subtracting a constant k from all the θ_i and β_j, as well to rescaling θ_i and β_j up by a factor λ, while scaling a_j down by the same factor λ. Hence, to fix a characteristic offset and scale, we use the same standard as the Epoch Capabilities Index, and normalize our results so that Claude Sonnet 3.5 has $\theta_i = 130$ and GPT-5 has $\theta_i = 150$.

Extension to Inference Time Cost. Next we extend the standard IRT model above to the hierarchical IRT setting where we also take into account inference time costs. In particular, we define $C_{ij} > 0$ to be the average inference time cost in USD per task attempt by model i and benchmark j. We model the log cost as scaling linearly in model capability θ_i with a fixed benchmark-specific offset c_j and joint cost scaling slope η across all benchmarks and models.

$$\log_{10}(C_{ij}) = c_j + \eta \cdot \theta_i \tag{2}$$

To understand this modelling choice better, observe that due to the sigmoid in the original model (1), the units of the latent capability θ_i are naturally thought of as being log probabilities $\theta_i \propto \log p_i$. These numbers p_i can be thought of as an intrinsic measure of benchmark-independent success probability. Hence, exponentiating both sides of (2) yields a power law scaling of cost as a function the p_i.

$$C_{ij} \propto 10^{c_j} \cdot p_i^{\eta}. \tag{3}$$

Understood this way, our model assumes that there is some intrinsic likelihood to succeed at any given task for model i, and the cost on benchmark j scales as a power law p_i^{γ} in this likelihood, with a benchmark-specific coefficient 10^{c_j}. We view this as a natural modelling choice, because cost is largely determined by model size and inference-time FLOPs, which in turn are directly linked to intrinsic likelihood for a model to succeed at a task (i.e. p_i), scaled by how many inference time flops

are typically required on the benchmark (i.e 10^{c_j}). We further empirically validate this modelling choice in Section 4.2 by comparing its predictions to individual empirical estimates of costs.

We fit the vector $\mathbf{c} = (c_1, \ldots, c_m)$ of benchmark cost intercepts and the joint cost slope η via ordinary least squares

$$\hat{\mathbf{c}}, \hat{\eta} = \arg\min_{\mathbf{c},\eta} \sum_{i,j} \Big(\log_{10}(C_{ij}) - c_j - \eta \cdot \theta_i \Big)^2, \tag{4}$$

In order to avoid outliers arising from very weak models that fail on most tasks, but are extremely cheap, we include in the regression only models i that succeed with at least 50% probability. If no such models are available for a given benchmark j, then we include only the best-performing model for that benchmark.

Predicting the Cost for 50% Success. The item response theory equation (1) predicts that for a benchmark with latent difficulty b_j, a model with latent capability $\theta_i = b_j$ will have exactly 50% success rate on tasks in the benchmark. Hence we can directly plug the value $\theta_i = b_j$ into (2) to obtain an estimate C_j^* for the cost required to achieve 50% success on benchmark j.

$$\log_{10}(C_j^*) = c_j + \eta \cdot b_j \implies C_j^* = 10^{c_j + \eta \cdot b_j}. \tag{5}$$

In other words, once the parameters of our hierarchical IRT model are estimated, we can predict the 50% success cost threshold for each benchmark with an analytical closed-form. We empirically validate this conversion between cost and 50% success in Section 4.2.

Converting Latent Difficulty to 50% Time Horizon. We use METR estimated 50% time horizons T_i for each model i, and least squares regression to fit conversion offset κ_0 and slope κ parameters for the equation

$$\theta_i = \kappa_0 + \kappa \cdot \log_{10}(T_i). \tag{6}$$

This gives a linear conversion rate between latent capabilities/difficulty and time 50% time horizon. Once again using the fact that a model with $\theta_i = b_j$ would succeed with probability exactly 50% on benchmark j, we rearrange (6) and plug in $b_j = \theta_i$ to predict the time horizon T_j for each benchmark j:

$$\log_{10}(T_j) = \frac{b_j - \kappa_0}{\kappa}. \tag{7}$$

Thus we obtain a linear relationship between log time horizon and benchmark difficulty scores.

The Cost vs. Time Horizon Scaling Law. Finally, having estimated difficulty b_j and 50% success rate cost C_j^* for each benchmark j, we fit an ordinary least squares regression to estimate parameters α and β relating cost to benchmark difficulty via the formula:

$$\log_{10}(C_j^*) = \alpha + \beta \cdot b_j \tag{8}$$

We then use our linear conversion parameters κ and κ_0 to convert benchmark difficulty b_j to time horizon T_j, giving us the final power law scaling for cost versus time horizon $\log_{10}(C_j^*) = (\alpha + \beta\kappa_0) + \gamma \log_{10}(T_j)$, where $\gamma = \kappa\beta$. Here, γ is the scaling exponent, with $\gamma = 1$ denoting linear scaling of cost with time horizon $\gamma = 2$ quadratic scaling, and so on.

3.2 Empirical Data Sources

To estimate the initial IRT model in equation (1) we use the full set of benchmark evaluations published by Epoch AI[1]. This data contains evaluations of 270 models across 62 benchmarks, with a total of 2,932 model-benchmark evaluations. Notably, the number of total possible model-benchmark pairs is $270 \times 60 = 16,200$, demonstrating the necessity of using the IRT model to fit together latent capabilities and difficulties on a consistent scale. For the cost estimation, we used 14 benchmarks which report cost-per-task, yielding a total of 471 model-benchmark evaluations across 146 models. Finally, we used the METR dataset of 50% time horizons which contains evaluations of time horizon for 37 different models. See Section A for additional details on the included benchmarks.

[1]The full EpochAI model evaluation dataset is available for download at `https://epoch.ai/data/benchmark_data.zip`

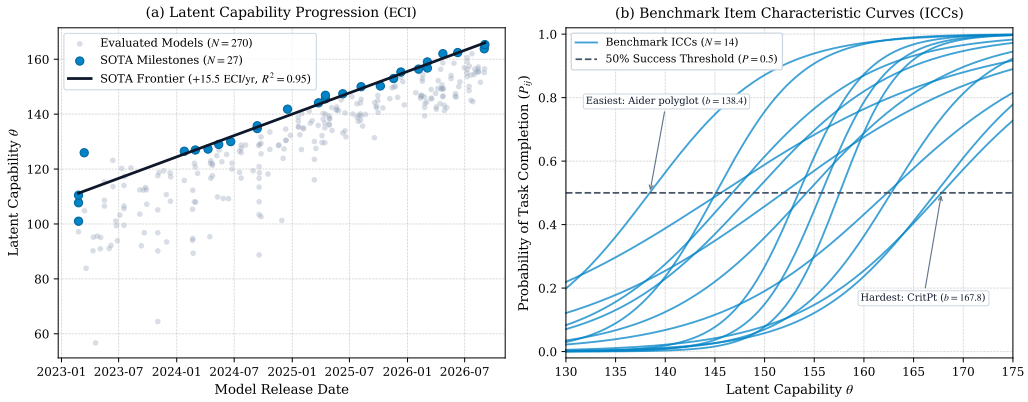


Figure 2: **Latent Capability Progression and Benchmark Difficulty Item Response Curves.** (a) Latent capability scores θ_i versus release date from the least squares regression fit to the IRT equation (1) for the full set of 270 models evaluated on 62 benchmarks. Solid blue points denote SOTA latent capability milestones. The black line is a linear regression fit to the SOTA models, showing a slope of 15.5 latent capability points per year, with $R^2 = 0.95$. The early 2023 outlier is GPT-4, and the highest latent capability point is GPT-6 Astra. (b) Item Characteristic Curves (ICCs) plot the predicted success probability as a function of latent capabilities given by (1) for each of the 14 benchmarks we use for cost data.

4 Results

The main result from our methodology is reported in Figure 1, which shows that the cost C^* to achieve 50% success rate scales as a power law in the 50% time horizon T. The linear regression fit between $\log(C^*)$ and $\log(T)$ has $R^2 = 0.85$, indicating that the power law fits the data well. Formally we have,

$$\log_{10}(C^*) = -1.33 + 2.81 \cdot \log_{10}(T \text{ in hours}). \tag{9}$$

Exponentiating both sides of the equation yields a superquadratic power law relationship between cost and time horizon.

$$C^*(T) \propto T^{\gamma}, \quad \text{with } \gamma = 2.81 \quad \text{Bootstrap 95\% CI: } [1.96, 3.58]). \tag{10}$$

Furthermore, over the $B = 1000$ bootstrap samples, we observe $P(\gamma > 2.0) = 96.8\%$, demonstrating that superquadratic scaling holds robustly.

To aid with intuition, we put concrete numbers into the model to see what sort of cost growth it predicts. plugging 8 hours into our model yields a predicted cost of $C^* = \$16.13$ for an hourly rate of just over \$2 per hour, while 80 hours yields a predicted cost of $C^* = \$10,415.56$ and hence an hourly rate of \$130.19. This demonstrates the rapid cost growth of agentic AI: 50% success for one human day's worth of work is much cheaper than minimum wage, while the cost for 50% success on two human week's worth of work approaches the top end of human wages. Going much further out in the power law becomes prohibitively expensive compared to human labor.

In the rest of this section we go through each step in our estimation pipeline and empirically evaluate the validity of the assumptions made. In Section 4.1 we investigate the IRT model and its predictions of latent capabilities and difficulties, in Section 4.2 we compare our model's estimated 50% success rate costs C_j^* to the empirical costs of the cheapest model to exceed 50% on each benchmark, and in Section 4.3 we evaluate the conversion from latent difficulties to 50% time horizons. Finally, in Section 4.5 we test the stability of our predictions over time. We further include additional results on leave-on-out benchmark cross-validation in Appendix C.

4.1 Model Capability (ECI) and Benchmark Difficulty (EDI)

We begin by reporting the results of fitting the standard IRT model defined in (1). Figure 2(a) shows how latent capabilities θ_i have evolved over time, with a linear regression on the SOTA-on-release

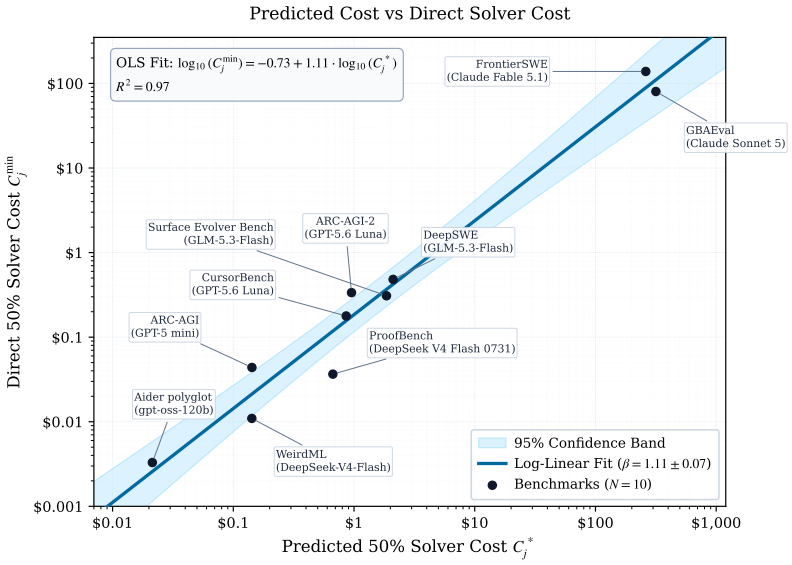


Figure 3: **Empirical Validation: Predicted Cost vs. Direct Solver Cost.** Predicted 50% success rate cost estimates C_j^* versus the cheapest empirical cost $C_j^{\min}$ for a model that scored at least 50% on benchmark j. Data is report for the 10 benchmarks where at least one model was able to score over 50%. The regression line is the ordinary least squares fit $\log_{10} C_j^{\min} = -0.73 + 1.11 \log_{10}(C_j^*)$ with $R^2 = 0.97$.

models showing a consistent gain of 15.5 points per year from early 2023 to late 2026 with $R^2 = 0.95$. If fit only to points starting in September 2024, when the first OpenAI o1 reasoning model was released, the frontier linear regression fit becomes 14.7 points per year with $R^2 = 0.98$.

In Figure 2(b) we plot the Item Characteristic Curves (ICCs) for each of our cost-reporting benchmarks. The ICC curve of a benchmark represents the predicted success probability P_{ij} for a model with latent capability θ_i on benchmark j. These predicted curves are determined by plugging the fit values of capability θ_i, difficulty b_j, and discriminability a_j into (1).

4.2 Empirical Validation against Direct Single-Attempt Solvers

In order to validate our method for predicting the cost for 50% success via (5), we next directly compare the predictions of our hierarchical IRT method with the actual empirical cost needed to reach a score of 50% on each benchmark. In particular, in Figure 3 we plot our predicted cost C_j^* versus the cheapest cost $C_j^{\min}$ of any model that reached a score of at least 50% on benchmark j. We fit a linear regression in log space for $\log(C_j^*)$ versus $\log(C_j^{\min})$, which reveals a slope of 1.11 ± 0.07 with $R^2 = 0.97$. A slope of 1 would indicate perfect cost estimation if we actually had models that empirically reached a score of exactly 50%. However, since we are simply selecting the cheapest model that was *above* 50% on each benchmark, it is expected that the slope is slightly larger than 1 as for more difficult benchmarks we have fewer models that achieve 50% success and thus less tight estimates of the minimum cost.

Note that this validation test contains 10 of our 14 cost-reporting benchmarks, because the hardest of the four–*The Agent Company*, *ExploitBench*, *OSWorld 2.0*, and *CritPt*–have no model that succeeds with probability greater than 50%. This demonstrates the importance of having a method to estimate the value C_j^*, even for benchmarks where no model has yet achieved 50% success.

To provide further validation, we also directly compared the raw dollar costs of C_j^* versus $C_j^{\min}$, which we found achieved Pearson correlation $r = 0.94$. That is, there is a strong linear relationship even when measured in raw dollars between our predictions and the empirical minimum cost to achieve score larger than 50%.

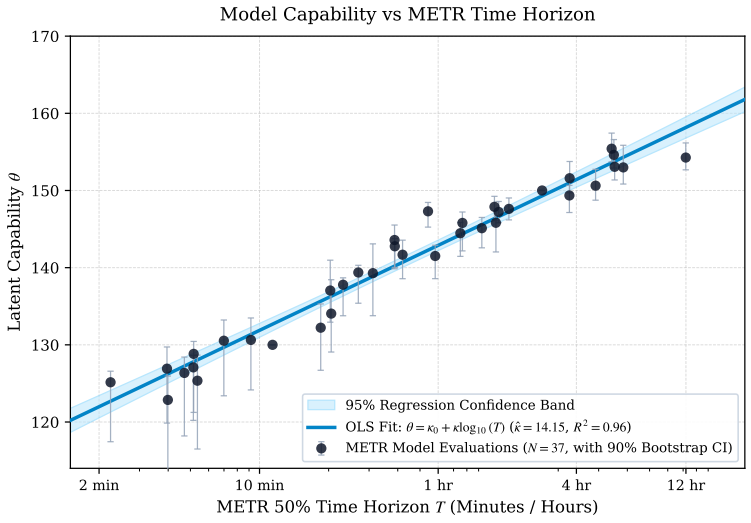


Figure 4: **Model Capability vs. METR Time Horizon.** Latent capability θ_i versus METR Time Horizon for 37 models evaluated on METR tasks (Kinniment et al., 2025). The solid blue line is least squares regression yielding $\theta = 117.72 + 14.15 \log_{10}(T)$, $R^2 = 0.96$, with analytical 95% confidence shaded region. The error bars on individual model evaluation points denote 90% bootstrap confidence intervals computed by refitting latent capabilities on $B = 1000$ bootstrap samples.

4.3 Model Capability and METR Time Horizon Calibration

In Figure 4 we report a linear regression between latent capabilities θ and 50% task completion time horizon in minutes, using data from the 37 models evaluated on the METR task suite (Kinniment et al., 2025). The linear regression fit is given by

$$\theta = 117.72 + 14.15 \cdot \log_{10}(T \text{ in minutes}) \quad R^2 = 0.96. \tag{11}$$

Recalling our IRT methodology, a model with latent capability $\theta_i = b_j$ achieves exactly 50% success probability on a task with difficulty b_j. Thus we can use the θ vs T slope estimated in(11) to convert task difficulty b_j directly into time horizon units T_j for each benchmark j. The conversion ratio is given by:

$$\log_{10}(T_j \text{ in minutes}) = \frac{b_j - 117.72}{\kappa}, \quad \log_{10}(T_j \text{ in hours}) = \frac{b_j - 142.82}{\kappa}. \tag{12}$$

Applying this conversion we get a range of predicted 50% time horizons from $T \approx 29$ min with $b_j = 138.44$ for Aider polyglot, to 57.9 hr with $b_j = 167.82$ for CritPt.

4.4 The Final Cost vs Time Horizon Scaling Law

To complete our estimation model we fit the linear regression of (9) to obtain the slope β and intercept α of log cost versus difficulty, and then convert difficulty into time horizon.

$$\log_{10}(C_j^*) = \alpha + \beta \cdot b_j \implies \log_{10}(C_j^*) = -1.33 + 2.81 \cdot \log_{10}(T_j \text{ in hours}), \tag{13}$$

Here we have used our fit value of κ to convert difficulties b_j into time horizons T_j via (12). The power law exponent $\gamma = 2.81$ is similarly obtained above by multiplying the fit slope $\beta = 0.1989$ by the fit value of $\kappa = 14.15$, giving us our final scaling relationship presented in Figure 1.

4.5 Stability and Robustness Analysis

In this section, we analyze the stability of our predicted power law scaling over time, as well as the sensitivity of our results to the inclusion or exclusion of individual benchmarks. Our overall results

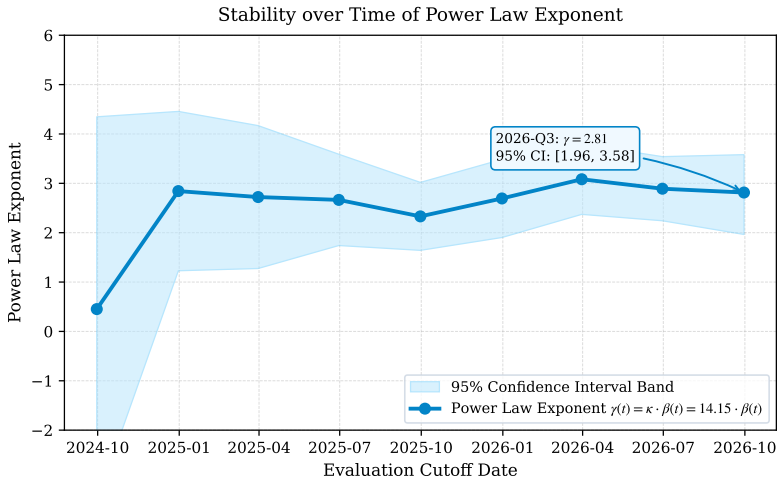


Figure 5: **Stability over Time of Power Law Exponent.** We re-run our full analysis pipeline on each subset of model-benchmark evaluations that were first available in a given quarter ranging from Q3 2024, to Q3 2026. For each quarter t, we plot $\gamma(t)$, the power law exponent estimated from data up until quarter t, along with the associated 95% confidence interval. After initial volatility due to the small number of evaluations available early on, by January 2025 $\gamma(t)$ stabilizes at a level quite close to its final value of $\gamma = 2.81$.

demonstrate that the superquadratic power law scaling exponent estimate has remained steady for nearly the last two years, and is highly stable to the removal of individual benchmarks containing cost data.

We evaluate the stability of our power law exponent over time in Figure 5. From 2024 to early 2025, there are fewer than 6 benchmarks and 100 total model evaluations reporting costs, resulting in large initial uncertainty. However, throughout all of 2025 and the first 3 quarters of 2026, as more benchmarks were added the mean estimate of γ has stayed in a very narrow window between 2.81 and 3.08. At the same time, the 95% confidence estimated has been narrowing, so that we can confidently reject subquadratic power law scaling by 2026. We provide additional details on the stability over time analysis, as well as a leave-one-out cross-validation analysis on the included benchmarks in Appendix C.

5 Conclusion

Overall, we have shown that our hierarchical item response theory model fits the empirical cost and performance data well, and predicts superquadratic scaling of costs as a function of the 50% time horizon. Starting from early 2025, when the first reasoning models OpenAI o1 pro and o3 mini had just been released, until the third quarter of 2026 which included the release of Claude Fable 5.1 and GPT-6 Astra, the estimated power law scaling exponent has remained steady with mean value close to $\gamma = 2.81$. Thus, we believe this power-law scaling represents a fundamental property of the current reasoning/agentic LLM paradigm.

Hence, until a new approach to long-horizon AI systems is introduced, we believe these results can be informative in predicting the progression of AI capabilities, and their impact on the world. In particular, they suggest that there will be a certain type of complementarity, where AI agents rapidly and cheaply accomplish tasks that would take humans a few hours or days, but require human-in-the-loop guidance to continue to produce useful work over longer horizons.

There is also the possibility that a future change in paradigm will now longer satisfy superquadratic cost scaling. This would plausibly mean a much higher level of direct substitution of AI for human labor, rather than the current complementarity. We believe our scaling model can be used to identify if this has come to pass, allowing for society to make more informed decisions about how we should use AI.

AI USE STATEMENT

This work does not use generative AI tools in text generation and its core research contributions.

REFERENCES

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Evan Chu, Rajan Agarwal, Abishek Thangamuthu, Brendan Graham, Justus Mattern, et al. FrontierSWE: An ultra-long-horizon benchmark for autonomous software engineering. https://www.frontierswe.com, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ricardo Dominguez-Olmedo, Florian Eddie Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *International Conference on Learning Representations*, volume 2025, pp. 68564–68594, 2025.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Jean-Paul Fox. *Bayesian Item Response Modeling: Theory and Applications*. Springer, New York, NY, 2010.

Paul Gauthier. Aider: AI pair programming in your terminal — polyglot benchmark. https://aider.chat/docs/leaderboards/, 2024.

Yann Hénon. Surface evolver bench: An agentic benchmark for 3D geometric physics modeling. https://yann-henon.github.io/surface-evolver-bench/, 2024.

Anson Ho, Jean-Stanislas Denain, David Atanasov, Samuel Albanie, and Rohin Shah. A rosetta stone for AI benchmarks. *arXiv preprint arXiv:2512.00193*, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Wenqi Huang, Charley Lee, Leonard Tng, and Serena Ge. DeepSWE: Measuring frontier coding agents on original, long-horizon engineering tasks. *arXiv preprint arXiv:2607.07946*, 2026.

Håvard Tveit Ihle. WeirdML: Evaluating LLMs on novel machine learning engineering tasks. https://htihle.github.io/weirdml/, 2024.

Naman Jain, Arvid Lunnemark, Aman Sanger, Sualeh Asif, Michael Truell, and Alexander M. Rush. Composer 2 technical report. *arXiv preprint arXiv:2603.24477*, 2026.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Megan Kinniment, Lawrence Chan, Thomas Kwa, Ben West, Joel Becker, et al. Measuring AI ability to complete long software tasks. *arXiv preprint arXiv:2503.14499*, 2025.

Seunghyun Lee and David Brumley. ExploitBench: A capability ladder benchmark for LLM cybersecurity agents. *arXiv preprint arXiv:2605.14153*, 2026.

Frederic M. Lord. *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.

Thang Luong, Edward Lockhart, et al. Towards robust mathematical reasoning. *arXiv preprint arXiv:2511.01846*, 2025.

OpenAI. On the navier–stokes millennium prize problem. https://openai.com/index/navier-stokes-solution/, September 2026.

Nikil Ravi, Kexing Ying, Vasilii Nesterov, Rayan Krishnan, Elif Uskuplu, Bingyu Xia, Janitha Aswedige, and Langston Nashold. FormalProofBench: Can models write graduate level math proofs that are formally verified? In *ICLR 2026 Workshop on AI Verification in the Wild (VerifAI-2)*, 2026. URL https://arxiv.org/abs/2603.26996.

Yangjun Ruan, Chris J Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of langauge model performance. *Advances in Neural Information Processing Systems*, 37:15841–15892, 2024.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Wim J Van der Linden. A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika*, 72(3):287–308, 2007.

Tianbao Xie, Deyi Zhang, Jmin Chen, Xiang Yue Fan, Zekun Shen, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Frank F. Xu, Yufan Song, Boxuan Shi, Bowen Li, Graham Neubig, et al. The agent company: Benchmarking LLM agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Stephen Yang, Ege Erdil, and Tamay Besiroglu. GBA Eval: Evaluating AI agents on building a full-scale emulator. https://gbaeval.com, 2026.

Guanhua Zhang, Ricardo Dominguez-Olmedo, and Moritz Hardt. Train-before-test harmonizes language model rankings. In *International Conference on Learning Representations*, volume 2026, pp. 74776–74808, 2026a.

Hanlin Zhang, Jikai Jin, Vasilis Syrgkanis, and Sham Kakade. Prescriptive scaling reveals the evolution of language model capabilities. *International Conference on Machine Learning*, 2026b.

Minhui Zhu, Minyang Tian, Xiaocheng Yang, Tianci Zhou, Penghao Zhu, Eli Chertkov, et al. Probing the critical point (CritPt) of AI reasoning: A frontier physics research benchmark. *arXiv preprint arXiv:2509.26574*, 2025.

A Additional Details on Benchmark Dataset

In this section we give additional details on the benchmarks included in our analysis.

A.1 Hierarchical Item Response Theory Datasets

Raw Source Archive. We use the full dataset of benchmark evaluations published by Epoch AI at `https://epoch.ai/data/benchmark_data.zip`. This contains 62 Benchmarks, 58 of which are included in the official Epoch Capabilities Index (ECI). We use all 62 benchmarks, in order to include as many as possible that span the full difficulty range, are tested on recent models, and report costs per task. The four that we include in this way are GBAEval, CursorBench, CritPt, and FrontierSwe.

A.2 Benchmarks with Cost Reporting

For fitting the cost portion of our hierarchical IRT model, we utilize the following 14 benchmarks from the Epoch AI dataset.

1. **Aider Polyglot** (Gauthier, 2024): Measures code repair and code completion for single files in the languages Python, JavaScript, Go, and C++.
2. **ARC-AGI** (Chollet, 2019): Two abstract reasoning benchmarks based on few-shot understanding of transformations between visual grid patterns.
3. **ARC-AGI-2** The sequel to ARC-AGI, created after the original had been saturated.
4. **WeirdML** (Ihle, 2024): A collection of out-of-the-ordinary Machine Learning coding tasks.
5. **Surface Evolver Bench** (Hénon, 2024): An agentic coding benchmark requiring writing code for the Surface Evolver physics simulator to demonstrate mathematical reasoning and coding for a rare programming language.
6. **ProofBench** (Ravi et al., 2026): A benchmark of formalizing mathematical theorems with the Lean 4 interactive theorem prover.
7. **CursorBench** (Jain et al., 2026): A benchmark of large-scale agentic coding workflows involving multi-file coding.
8. **CritPt** (Zhu et al., 2025): A very difficult set of olympiad-level physics problems.
9. **DeepSWE** (Huang et al., 2026): A long-horizon agentic coding benchmark involving editing whole repositories.
10. **FrontierSWE** (Chu et al., 2026): An extremely long-horizon agentic software engineering benchmark, allowing up to 20hrs of wall-clock time for agentic execution.
11. **The Agent Company** (Xu et al., 2024): A large-scale agentic workplace simulator, requiring software engineering, project management, and administrative tasks.
12. **GBAEval** (Yang et al., 2026): A benchmark where the agent is given 24 hours of execution time to build a Game Boy Advance emulator from scratch.
13. **ExploitBench** (Lee & Brumley, 2026): Agentic cybersecurity benchmark involving capture-the-flag style cyber exploits.
14. **OSWorld 2.0** (Xie et al., 2024): A benchmark of real-world computer use for Desktop GUI tasks.

For GBAEval, models AI agents are run for 24 hours, and their total score and cost are reported at frequent checkpoints. We report the dollar cost for the first checkpoint where the model reached its final score. This is necessary, because weaker models will continue to run for the full 24hrs while not making any progress, artificially inflating their cost to achieve a given score level.

B Non-Parametric Bootstrap

Our method for estimating scaling laws has several stages with multiple least-squares regressions as well as cost estimates produced in analytical closed form. In order to capture the uncertainty in

these estimates we use a cluster joint non-parametric bootstrap (Cameron et al., 2008; Efron, 1979) to correctly account for the correlations in model evaluations across benchmarks.

Cluster Resampling Protocol. Let $\mathcal{M}$ be the set of $M = 270$ models. We re-run our entire analysis pipeline while resampling models with replacement. When a model is resampled, all of the associated benchmark evaluation scores and costs for that model are paired with it, to preserve correlation. For each bootstrap sampling iteration $s \in \{1, \ldots, B\}$ for $B = 1000$ we do the following:

1. **Model-Level Cluster Resampling:** We sample M models with replacement from $\mathcal{M}$: $\mathcal{M}^{(s)} = \{m_1^{(s)}, \ldots, m_M^{(s)}\}$. Every benchmark score and reported cost-per-task for the model $m_i^{(s)}$ is included in the bootstrap sample $\mathcal{D}^{(s)}$. If model i is sampled k times, all of this data appears k times in $\mathcal{D}^{(s)}$.
2. **IRT Model Refitting:** We re-fit the IRT model from (1) on $\mathcal{D}^{(s)}$,
$$P(Y_{ij} = 1 \mid \theta_i, b_j, a_j) = \sigma\big(a_j(\theta_i - b_j)\big)$$
which yields bootstrap estimates for latent capabilities $\hat{\theta}_i^{(s)}$, benchmark difficulties $\hat{b}_j^{(s)}$, and benchmark discriminabilities $\hat{a}_j^{(s)}$.
3. **Cost Estimation:** We re-fit the cost versus latent capabilities model (2) on $\mathcal{D}^{(s)}$,
$$\log_{10}(C_{ij}) = c_j + \eta \cdot \theta_i$$
giving us bootstrap estimates for benchmark offsets $\hat{c}_j^{(s)}$ and cost-capability slope $\hat{\eta}^{(s)}$. We then directly plug in the values of $\hat{b}_j^{(s)}$ to equation 5, to obtain 50% success rate bootstrap cost estimates
$$\log_{10}(\hat{C}_j^{*(s)}) = \hat{c}_j^{(s)} + \hat{\eta}^{(s)}\hat{b}_j^{(s)}, \tag{14}$$
4. **METR Time Horizon Conversion Uncertainty:** To estimate uncertainty from the conversion of benchmark difficulties to time horizons using the METR data, we use the Gaussian posterior estimates of the regression in (11)
$$\begin{pmatrix} \kappa_0^{(s)} \\ \kappa^{(s)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{\kappa}_0 \\ \hat{\kappa} \end{pmatrix}, \boldsymbol{\Sigma}_{\text{METR}} \right), \tag{15}$$
where $\hat{\kappa}_0 = 117.72$, $\hat{\kappa} = 14.15$, and $\boldsymbol{\Sigma}_{\text{METR}}$ is the parameter covariance matrix, all of which are estimated from the full METR time horizon dataset.
5. **Final Scaling Law Uncertainty Estimation:** We re-fit least squares regression on $\big(\log_{10} \hat{T}_j^{(s)}, \log_{10} \hat{C}_j^{*(s)}\big)$ using equation 8 and conversion parameter $\hat{\kappa}^{(s)}$ and $\hat{\kappa}_0^{(s)}$
$$\log_{10} \hat{C}_j^{*(s)} = \hat{\alpha}^{(s)} + \hat{\gamma}^{(s)} \log_{10} \hat{T}_j^{(s)}, \tag{16}$$
which gives us bootstrap estimates for $(\hat{\alpha}^{(s)}, \hat{\gamma}^{(s)})$ as well as the parameter covariance matrix $\boldsymbol{\Sigma}^{(s)}$. Because we have 14 benchmarks in the regression and are estimating 2 parameters, our model has 12 degrees of freedom. Hence, we sample from the Student's t_{12} posterior:
$$\begin{pmatrix} \tilde{\alpha}^{(s)} \\ \tilde{\gamma}^{(s)} \end{pmatrix} \sim t_{12}\left(\begin{pmatrix} \hat{\alpha}^{(s)} \\ \hat{\gamma}^{(s)} \end{pmatrix}, \boldsymbol{\Sigma}^{(s)} \right). \tag{17}$$

When the full set of $B = 1000$ bootstrap re-samples is complete, we use empirical percentiles $[0.025, 0.975]$ for our 95% confidence intervals.

C Additional Details for Stability Analysis

In this section we provide full statistical details of our two stability analyses: the evolution of γ over time, and leave-one-out benchmark cross-validation.

Stability Over Time. In Table 1 we provide the full tabular form of the stability over time analysis from Section 4.5.

Table 1: **Stability over Time of Power Law Exponent.**

Cutoff Date	Evaluations	Models	Benchmarks	Slope β	SE(β)	Exponent γ	R^2
2024-09-30	32	21	5	0.0320	0.0865	0.45 ± 1.22	0.044
2024-12-31	49	29	6	0.2009	0.0411	2.84 ± 0.58	0.857
2025-03-31	71	35	6	0.1924	0.0368	2.72 ± 0.52	0.872
2025-06-30	146	58	6	0.1883	0.0235	2.66 ± 0.33	0.941
2025-09-30	187	74	8	0.1647	0.0199	2.33 ± 0.28	0.920
2025-12-31	221	86	9	0.1904	0.0236	2.69 ± 0.33	0.903
2026-03-31	276	100	13	0.2179	0.0228	3.08 ± 0.32	0.892
2026-06-30	372	124	14	0.2042	0.0210	2.89 ± 0.30	0.887
2026-09-30	**471**	**146**	**14**	**0.1989**	**0.0235**	$\mathbf{2.81 \pm 0.33}$	**0.857**

Leave-One-Out Benchmark Cross-Validation. Our main analysis utilized all 14 benchmarks for which we had cost data. In order to ensure that our results are robust to the exact choice of benchmarks used to compute costs, we perform leave-one-out cross validation, re-running the whole pipeline while dropping one benchmark at a time. Figure 6 reports the results of this analysis.

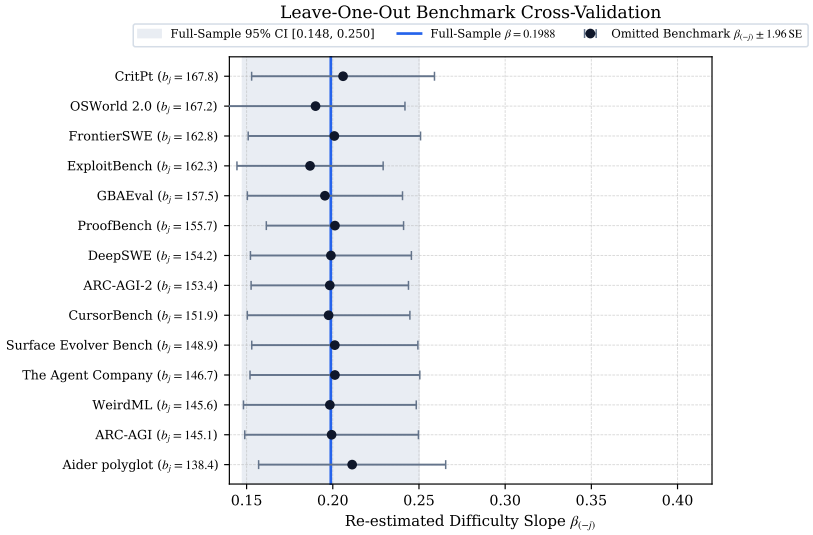


Figure 6: **Leave-One-Out Benchmark Cross-Validation.** Each individual point denotes estimated cost vs difficulty slope $\beta_{(-j)}$ when each benchmark is individually removed, along with 95% confidence interval. The blue vertical line denotes the estimated value $\beta = 0.1989$ when all 14 benchmarks are included, with 95% confidence shaded region.

Table 2: **Leave-One-Out Benchmark Cross Validation.**

Omitted Benchmark	**EDI**	**Slope $\beta_{(-j)}$**	**SE($\beta_{(-j)}$)**	**% Change**	**Exponent $\gamma_{(-j)}$**	**Cook's D_j**
Aider polyglot	138.44	0.2112	0.0276	+6.2%	2.99	0.179
ARC-AGI	145.10	0.1993	0.0256	+0.2%	2.82	0.000
WeirdML	145.58	0.1984	0.0255	-0.3%	2.81	0.001
The Agent Company	146.75	0.2012	0.0251	+1.2%	2.85	0.012
Surface Evolver Bench	148.89	0.2012	0.0245	+1.2%	2.85	0.018
CursorBench	151.88	0.1977	0.0240	-0.6%	2.80	0.021
ARC-AGI-2	153.43	0.1983	0.0232	-0.3%	2.81	0.047
DeepSWE	154.23	0.1990	0.0238	+0.0%	2.82	0.028
ProofBench	155.72	0.2013	0.0202	+1.2%	2.85	0.154
GBAEval	157.55	0.1955	0.0229	-1.7%	2.77	0.074
ExploitBench	162.33	0.1869	0.0216	-6.0%	2.64	0.269
FrontierSWE	162.75	0.2006	0.0255	+0.9%	2.84	0.005
OSWorld 2.0	167.23	0.1902	0.0264	-4.4%	2.69	0.098
CritPt	167.82	0.2062	0.0270	+3.7%	2.92	0.067
Full Sample	—	**0.1989**	**0.0235**	**0.0%**	**2.81**	—