Crashing Waves vs. Rising Tides: Preliminary Findings on AI Automation from Thousands of Worker Evaluations of Labor Market Tasks*

Matthias Mertens
MIT FutureTech

Adam Kuzee
MIT FutureTech

Brittany S. Harris
MIT FutureTech

Harry Lyu
MIT FutureTech

Wensu Li
MIT FutureTech

Jonathan Rosenfeld
MIT FutureTech

Meiri Anto
MIT FutureTech

Martin Fleming
MIT FutureTech

Neil Thompson
MIT FutureTech

March 2026

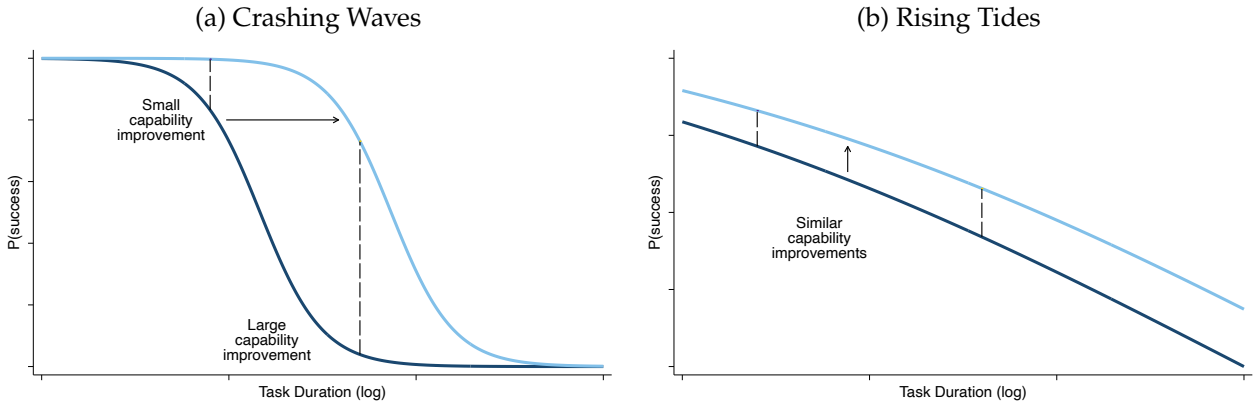
Abstract

We propose that AI automation is a continuum between: (i) **crashing waves** where AI capabilities surge abruptly over small sets of tasks, and (ii) **rising tides** where the increase in AI capabilities is more continuous and broad-based. We test for these effects in preliminary evidence from an ongoing evaluation of AI capabilities across over 3,000 broad-based tasks derived from the U.S. Department of Labor O*NET categorization that are text-based and thus LLM-addressable. Based on more than 17,000 evaluations by workers from these jobs, we find little evidence of crashing waves (in contrast to recent work by METR), but substantial evidence that rising tides are the primary form of AI automation. AI performance is high and improving rapidly across a wide range of tasks. We estimate that, in 2024-Q2, AI models successfully complete tasks that take humans approximately 3–4 hours with a 50% success rate, increasing to about 65% by 2025-Q3. If recent trends in AI capability growth persist, this pace of AI improvement implies that LLMs will be able to complete most text-related tasks with success rates of, on average, 80%–95% by 2029 at a minimally sufficient quality level. Achieving near-perfect success rates at this quality level or comparable success rates at superior quality would require several additional years. These AI capability improvements would impact the economy and labor market as organizations adopt AI, which could have a substantially longer timeline.

*Corresponding authors: Matthias Mertens, mmertens@mit.edu; Neil Thompson, neil_t@mit.edu. We thank David Autor for his insightful comments. We are grateful to Annie Lin, Amelia Michael, Peter Olkhovets, and Tiffany Wang for their excellent work as research assistants. We also thank Justin Viola for excellent software engineering work. Funding for this research was provided by Open Philanthropy and a technology company.

1 Introduction

Figure 1: Crashing Waves vs Rising Tides in AI Automation



Notes: Diagram of the distinction between AI automation that comes as "Crashing Waves" (Panel (a)) and "Rising Tides" (Panel (b)).

Recent evidence by Kwa et al. (2025) suggests that, as models improve, AI capabilities surge abruptly for tasks that previously appeared out of reach, as if a "crashing wave" suddenly reaches them (our characterization), as shown in Figure 1(a).[1]

In this paper, we contrast this crashing wave phenomenon with a "rising tide" (Figure 1(b)), in which performance is lifted more broadly across the task space. The central difference between the two phenomena is the slope of the relationship between AI success on tasks and (log) task duration. For crashing waves, this relationship can be well described by a steep logistic curve.[2] AI progress is then a rightward shift of the curve, which translates into large, concentrated automation for tasks near the tipping point due to sudden improvements in capabilities of what systems can do. In practice, this would lead to harsh surprises for human workers. Over just a short period of time, they would observe AI models going from nearly always failing to nearly always succeeding.

By contrast, rising-tide automation has a flatter success–duration relationship, with AI performance being more similar across tasks of different lengths. This would still be represented by a logistic curve, but a much flatter one. *The same amount* of AI progress would then translate into more gradual automation under the rising-tide view, such that individual workers are less likely to be blindsided by AI. A rising tide could, however, still be quite disruptive if it happens quickly.

The main insight of this paper is that, across a large set of realistic and representative labor-market tasks addressable by LLMs, the downward slope between task success and task duration is, on average, surprisingly flat — i.e., more consistent with a rising tide rather than a crashing wave.

Our analysis draws upon a broader ongoing research effort that collects novel evaluations of LLM outputs by domain experts across more than 40 models and covering over 20,000 unique task

[1] Kwa et al. (2025) study 170 research and software-engineering tasks and document rapid growth in the maximum task duration (measured as human completion time) that LLMs can solve at a 50% success rate. Subsequent work by METR (2025) extends this analysis to additional stylized benchmarks. By contrast, we focus on non-deterministic, realistic, and representative labor-market tasks.

[2] Task duration can plausibly relate to the serial dependence of tasks: longer tasks may require completing more coupled sequential sub-steps. We formalize this interpretation in Section 4.2. For related discussions, see, for instance, Dziri et al. (2023).

examples ("instances") based on more than 10,000 O*NET tasks that are at least partially text-based (O*NET (2024)).[3] Outputs are scored by human evaluators with relevant on-the-job experience. Given the paper's primary focus on automation, we center our analysis on success measures defined by expert evaluations indicating that the LLM output required no human intervention to be at least minimally-successful. **The survey is ongoing; while we have already collected a substantial share of the data, the results reported here should be interpreted as preliminary.**

Leveraging thousands of task instances that range from minutes to multiple weeks of human work, we find:

1. **The success-duration curve is relatively flat — consistent with a rising tide of AI automation.** This pattern holds across models of different sizes, as well as different model vintages across time. Automation within particular "job families" (e.g., management or community and social service) also follows the same rising-tide pattern in most cases. There are, however, meaningful differences in the slope of the success-duration curves across job families, as one would expect given the differences in task structure (see Section 4.2).

2. **AI performance is high.** Across all LLMs in our survey, we find strong capabilities. Models can do a minimally-sufficient job without human edits on roughly half to three-quarters of potential tasks presented to them. And, for example, by 2024-Q3, frontier models were already able to achieve 50% success rates on (LLM-addressable) tasks that take humans about a day.[4]

3. **AI performance is improving rapidly across a wide range of task durations.** Our estimates imply that between 2024-Q2 and 2025-Q3, frontier models went from achieving a 50% success rate on 3- to 4-hour tasks to 1-week tasks, and achieving a 70% success rate on 1-minute tasks to 1-hour tasks. The implied "doubling time," or the calendar time between model releases needed for newer models to achieve the same success rate on tasks which are twice as long, equals 3.8 months and is estimated with relatively high precision. Relative to prior benchmark-based evidence (Kwa et al., 2025; METR, 2025), these improvement rates are rapid, placing them at the upper end of existing estimates. Similarly, success rates at given task lengths increase quickly. The average failure rate (1 minus the success rate) halves every 2.4-3.2 years across tasks which take 5 minutes to 24 hours (for humans to complete). Over our observation period (2024-Q2 to 2025-Q3), this corresponds to annual success rate increases between 15-16 percentage points.

4. **The performance gains from increasing model size are different than those from newer model vintages.** As new models are created, they outperform similarly sized models from the past and the shift in the success-duration curve is approximately parallel. So, success on both short and long-duration tasks is improved. By contrast, larger models released at the same time as smaller models do outperform on short-duration tasks, but have only modest improvements on longer-duration tasks. We plan to quantify the relative contribution of these two channels in ongoing work.

[3] GDPval from OpenAI has done something related, but for an order of magnitude fewer occupations (Patwardhan (2025)).

[4] As discussed later, these findings will not translate directly to shares of job automation, because of sampling issues, "last mile" costs (Fleming et al., 2024), and other reasons.

Taken together, our findings imply that for realistic and representative real-world labor-market tasks that are text-based — or partially text-based — AI capabilities are already substantial and poised to expand broadly. But, rather than arriving in crashing waves that transform a certain set of tasks at a time, progress typically resembles a rising tide, with widespread gains across many tasks simultaneously. The pace of improvement we document is still rapid for attaining high, though not exceptional, success rates (e.g., ~80%). Extrapolating our estimates into the future suggests that most of the tasks that we study could reach AI success rates of 80%-95% by 2029 (at a minimally sufficient quality level), suggesting potentially substantial labor-market impacts as this tide continues to rise. At the same time, the flat (rising-tide) logistic relationship between model performance and task duration implies that achieving near-perfect performance will take considerably longer. This provides a window for worker adjustment, particularly in tasks with low tolerance for errors. We note that our estimates assume AI progress continues at the pace observed over the past two years and should therefore be interpreted as an upper-bound (i.e., particularly fast) scenario. There are several reasons to expect a potential slowdown in AI capability growth, including limits to scaling compute, as well as possible slowdowns in hardware progress and algorithmic innovation.

2 Results

2.1 Approach and Baseline Results

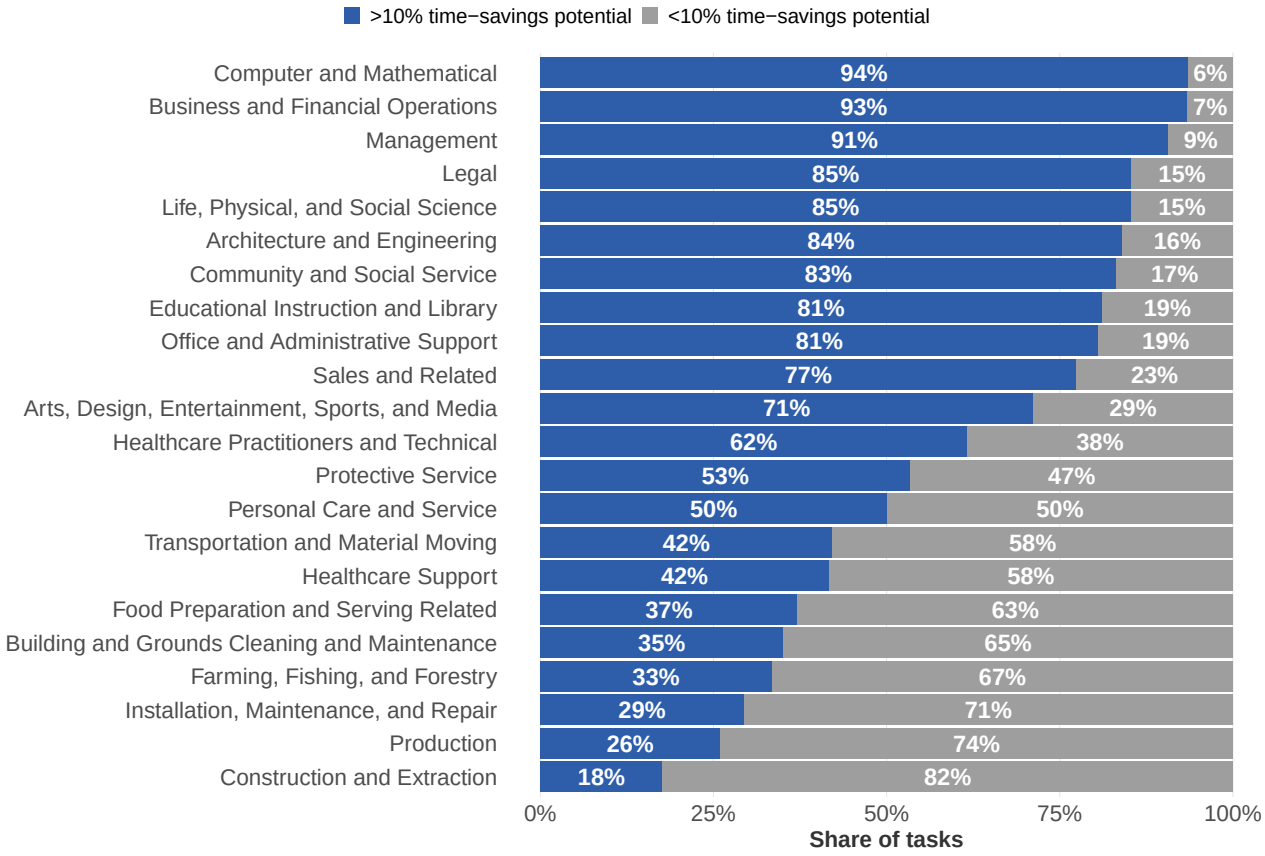
Survey Data. We collected novel survey data evaluating LLM performance on more than 11,000 labor market tasks drawn from the O*NET task classification. Here, we provide a high-level overview of the data collection process and refer to Section 4.1 for a detailed description. First, we used GPT-4 to screen O*NET tasks for automation potential. Specifically, we retained tasks for which LLM assistance was estimated to generate at least a 10% time savings. This filtering ensures that our sample focuses on tasks with meaningful cognitive or informational components, rather than, for example, purely physical activities with no plausible language-model relevance. Figure 2 reports by the O*NET job family, the share of tasks that meet this criterion. For each selected task, we constructed two task instances, which were then completed by more than 40 LLMs (5 models per instance). Model outputs were evaluated by human evaluators with relevant on-the-job experience. We only used task instances that were judged as realistic and representative for the underlying task by human evaluators. Evaluators provided contextual information on the task (most importantly, time requirements) and rated each model response on a 1–9 scale related to how a hypothetical manager would view the quality of the response. A score of 1 indicates that the output would need to be redone from scratch, whereas a score of 9 indicates above-average performance relative to a human worker. In our main analysis, we use a binary indicator that equals one when a manager would judge the response to be "minimally sufficient" without edits (i.e., a rating of ≥ 7). We treat this outcome as our primary task-level measure of automation potential throughout the paper.[5] In some analyses, we also apply stricter

[5]Scores 1–9 refer to, respectively: “Not useful: Requires complete rework (needs to be started over from scratch)”; “Not useful: Requires extensive rework (almost everything needs to be changed)”; “Not useful: Requires substantial rework (most elements need to be changed)”; “Useful with edits: Requires major edits (needs substantial work)”; “Useful with edits: Requires moderate edits (needs some work)”; “Useful with edits: Requires minor edits (needs some refinement)”;

quality thresholds—ratings of ≥ 8 or 9 —capturing responses that are of "average" or "superior" quality without edits, respectively.

Figure 3 displays the distribution of reported task durations. Observed tasks range from approximately 10 minutes to several days, with most tasks in our survey taking between 20 minutes and 10 hours for humans to complete.

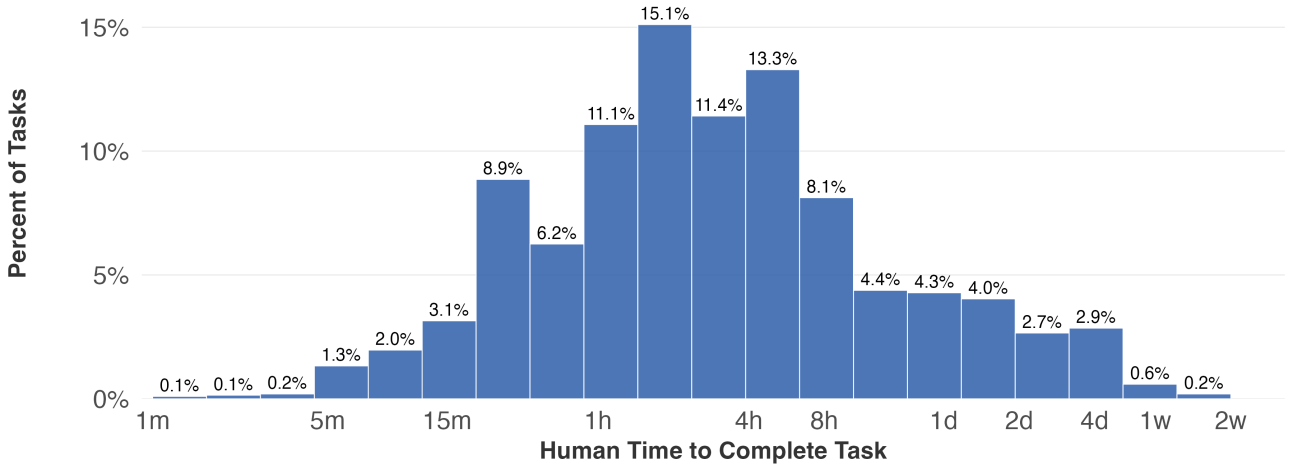
Figure 2: Share of O*NET tasks with 10% Time Saving Potential Included in the Survey by Job Family



Notes: The figure shows the share of tasks for each O*NET job family that our method classifies, using GPT-4, as offering at least 10% potential time savings from LLM use.

"Useful as is: Requires no edits to be minimally sufficient"; "Useful as is: Requires no edits to be of average quality"; and "Useful as is: Requires no edits to be of superior quality".

Figure 3: Task Duration Histogram



Notes: The figure shows the distribution of task durations collected in the survey so far across 20 equally sized log-spaced bins.

Regression framework. The key relationship we study is how LLM performance varies with task duration. Our main specification estimates the following logistic model:

$$\Pr(Y_{ijsm} = 1) = \Lambda\big(\alpha + \beta \log_{10} T_{js}\big) = \frac{\exp\big(\alpha + \beta \log_{10} T_{js}\big)}{1 + \exp\big(\alpha + \beta \log_{10} T_{js}\big)}. \tag{1}$$

Here, $\Lambda(\cdot)$ denotes the logistic CDF and α is a constant. Y_{ijsm} is an indicator equal to one if the evaluator reports that a hypothetical manager would accept the response without edits—i.e., the survey rating is ≥ 7 (we also consider thresholds of ≥ 8 and $= 9$). The main regressor, T_{js}, measures task duration in time units. Indices i, j, s, and m denote the evaluator, O*NET task, task instance, and model, respectively. We estimate Eq. (1) by maximum likelihood. The logistic specification follows prior work (Ge et al., 2026; Kwa et al., 2025). In Section 4.2, we provide one possible micro-foundation for Eq. (1) under which the slope coefficient β admits a structural interpretation: it can be mapped to the number of sequentially dependent steps required to complete a task.[6]

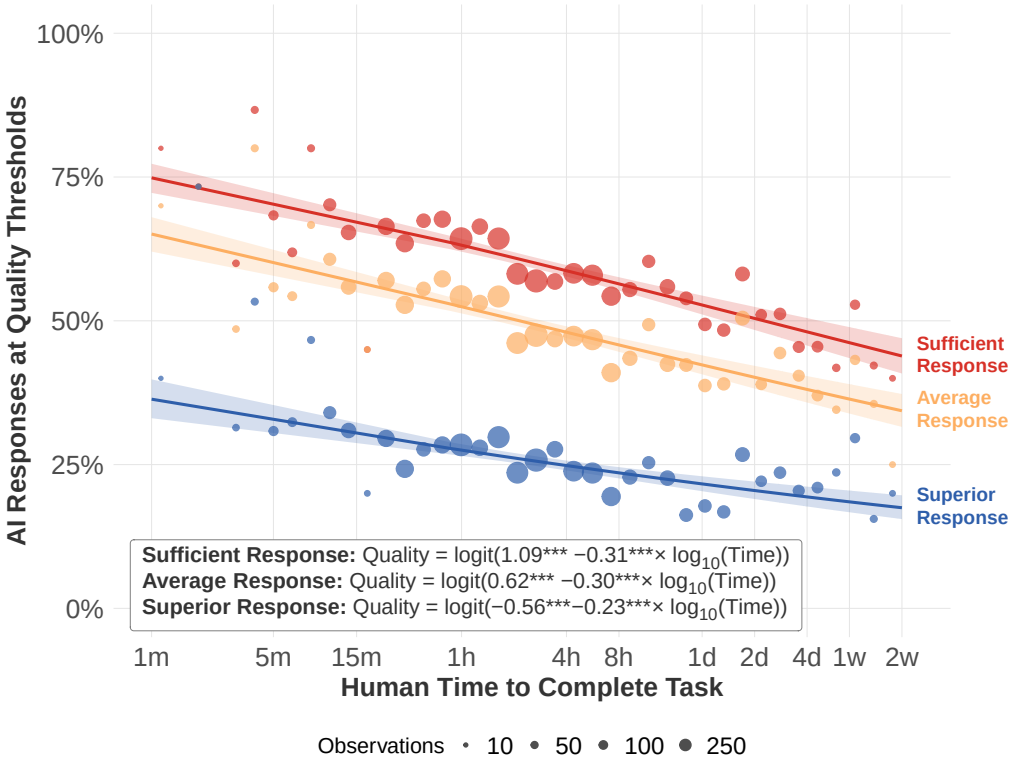
Pooled results. Figure 4 plots estimates of Eq. (1) for three "useful without edits" thresholds: minimally sufficient (score ≥ 7), at least average quality (score ≥ 8), and superior quality (score $= 9$). Binned scatter points summarize the raw data.

At the ≥ 7 threshold, a tenfold increase in task duration is associated with a 0.31 decline in the log-odds of success.[7] Evaluated at the sample mean acceptance rate of 60%, this implies a drop in

[6] We emphasize that this is only one possible (plausible) micro-foundation, offered to aid interpretation of the empirical relationship we estimate (and that prior work has studied). Our results *do not require* this particular interpretation: the relationship is identified empirically, and we do not rule out alternative micro-foundations or interpretations.

[7] We replicate the ≥ 7 analysis in Appendix Figure A.2 using specifications that control for occupational and model-specific differences. Estimated slopes are similar, ranging from -0.22 to -0.32.

Figure 4: Task Instance Automation by Required Task Completion Time



Notes: Each line plots the estimated *logistic* relationship between AI response quality and the time required to complete a task instance, based on Equation (1) estimated without controls (with 95% confidence bands). Coefficients are shown as log-odds on the figure. Standard errors are clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%. The red line corresponds to responses that are minimally sufficient or better, (score ≥ 7), the yellow line to responses which are average-quality or better (score ≥ 8), and the blue line to superior-quality responses (score $= 9$). Dots represent binned raw data: we partition task instances into 40 equally sized, log-spaced time bins and compute success rates and sample sizes within each bin. For each quality threshold, two of the 40 bins contain no observations.

predicted acceptance of about 7.6 percentage points.[8] The slope is slightly flatter at higher levels of response quality, with slope estimates of -0.30 for ≥ 8 and -0.23 for $= 9$. As expected, the curves also shift downward for higher quality levels, reflecting the greater difficulty of meeting stricter quality standards at any task length. The slope differences imply that success probabilities decline more sharply at shorter durations, with the gap between levels of quality narrowing as tasks become longer (yet, differences are still notable).

2.2 Extended Analyses

Job Families. We estimate Eq. (1) separately by O*NET job family at the ≥ 7 threshold and report the results in Table 1. Three patterns stand out. First, success rates are substantial across the board, pointing to broad potential for LLMs to handle real-world labor-market tasks. Second, average success varies meaningfully by domain, ranging from 47% in "Legal" to 73% in "Installation, Maintenance, and Repair" (recall that we restrict attention to text-based and partially text-based tasks and exclude purely physical activities). Third, and most importantly, the success–duration slopes differ sharply across job families. The estimated slope coefficients span a wide range, implying that the relationship between LLM performance and task duration is not portable across labor-market domains. In roughly

[8] Because $\text{logit}(0.60) \approx 0.405$ and $\frac{1}{1+\exp(-(0.405-0.31))} \approx 0.524$.

a quarter of job families, the slope is statistically significantly negative, with estimates between -0.25 and -0.93—equivalent to a 6.1 to 22.8 percentage-point decline in predicted success for a tenfold increase in task duration, evaluated at a 60% baseline success rate. In the remaining job families, the slope is statistically indistinguishable from zero. Table 1 also visualizes the implied logistic curves for each domain; gray curves indicate statistically insignificant estimates.

Larger vs. smaller and newer vs. older models. A key question is how ongoing progress in LLMs affects these patterns. A strength of our data is that it spans many models, allowing us to separate two distinct channels of improvement: increases in model size versus newer model releases. In Figure 5, Panel (a) compares large ($>$ 100B parameters) and small ($\leq$ 100B parameters) models using the success threshold score ≥ 7.[9] The estimated curve for large models is more strongly downward sloped than for small models ($\beta = -0.36$ vs. -0.26), implying an outward rotation: large models' performance advantage is greatest for short tasks and attenuates as task duration increases.

Panel (b) instead splits frontier models by publication date (pre- vs. post-2025). In contrast to Panel (a), the curves exhibit an almost purely parallel shift, with nearly identical slope coefficients ($\beta = -0.31$ and -0.32). This pattern indicates that newer models exhibit improved performance by roughly the same amount across the task-duration distribution, rather than disproportionately benefiting short tasks.

Panels (c)–(f) replicate Panels (a)–(b) for stricter quality thresholds (≥ 8 and $= 9$). We observe the same qualitative patterns at lower overall success rates: (i) comparing larger versus smaller models again yields an outward rotation, and (ii) comparing newer versus older models again produces an approximately parallel shift. Appendix Tables A.1, A.2, and A.3 confirm that these differences are statistically significant; the slight apparent rotation between newer and older models in Panel (f) is not.[10]

A natural interpretation is that improving longer-duration tasks is more demanding than improving short-duration tasks — and in particular that long-duration tasks, even if they are ultimately sequences of coupled short-duration ones (see Section 4.2), could require additional training / reinforcement learning over how to combine them. Another potential explanation for this pattern could be differences in the model release and performance patterns between small and large models.

[9] In Figure 5, **Larger models** include: Claude Opus 3, Claude Opus 4.1, Claude Sonnet 3.7, Claude Sonnet 4, DeepSeek R1, DeepSeek V3, Gemini 1.5 Pro, Gemini 2.5 Pro, GPT-4, GPT-4o, GPT-5, GPT-OSS 120B, Llama 3.1 405B Instruct, Llama 4 Maverick 400B, Llama 4 Scout 109B, Mistral Medium, o3, o4 mini, and Qwen 3 235B. **Smaller models** include: Claude Haiku 3, Claude Haiku 3.5, Gemini 2.5 Flash Lite, Gemma 3, Gemini 2.5 Flash, GPT-3.5 Turbo, GPT-4o mini, GPT-5 mini, GPT-5 nano, GPT-OSS 20B, Granite 3.3 2B, Granite 3.3 8B, Llama 2 7B, Llama 2 70B, Llama 3.1 8B, Llama 3.1 70B, Qwen 2 7B, Qwen 2 72B, Qwen 3 14B, Qwen 3 32B, and QwQ-32B. In Panel (b), we estimate Equation (1) separately for newer and older models. **Newer models** include: Claude Sonnet 3.7, Claude Sonnet 4, Claude Opus 4.1, DeepSeek R1, Gemini 2.5 Pro, Gemini 2.5 Flash, Gemini 2.5 Flash Lite, Gemma 3 1B, GPT-5, GPT-5 mini, GPT-5 nano, GPT-OSS 120B, GPT-OSS 20B, Granite 3.3 2B, Granite 3.3 8B, Llama 4 Maverick 400B, Llama 4 Scout 109B, o3, o4 mini, Qwen 3 14B, Qwen 3 32B, Qwen 3 235B, and QwQ-32B. **Older models** include: Claude Haiku 3, Claude Haiku 3.5, Claude Opus 3, DeepSeek V3, GPT-3.5 Turbo, GPT-4, GPT-4o, GPT-4o mini, Gemini 1.5 Pro, Llama 2 7B, Llama 2 70B, Llama 3.1 8B, Llama 3.1 70B, Llama 3.1 405B Instruct, Mistral Medium, Qwen 2 7B, and Qwen 2 72B.

[10] The appendix tables further show that these results remain statistically significant (and quantitatively similar) when jointly including indicators for model size and model vintage, thereby purging publication-time effects from the size comparison and, conversely, size effects from the vintage comparison.

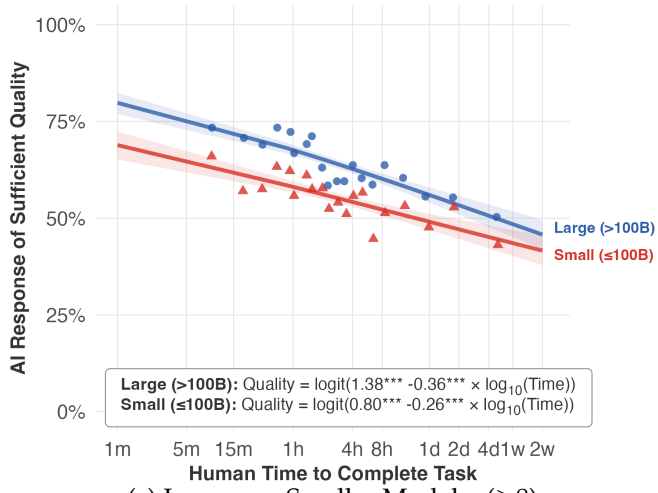
Table 1: Task Automatability And Task duration: by Job Family

Job Family	N	Success Rate	Coef. β	SE	Fit
Personal Care and Service	485	68.9%	-0.93***	0.27	
Architecture and Engineering	580	52.8%	-0.45*	0.18	
Arts, Design, Entertainment, Sports, and Media	1,035	55.2%	-0.41***	0.12	
Management	1,920	52.7%	-0.31**	0.10	
Educational Instruction and Library	1,615	60.4%	-0.28*	0.14	
Community and Social Service	720	62.2%	-0.25	0.16	
Computer and Mathematical	1,955	55.9%	-0.25**	0.09	
Sales and Related	855	62.6%	-0.25	0.15	
Business and Financial Operations	1,475	56.7%	-0.20	0.12	
Healthcare Practitioners and Technical	1,395	65.9%	-0.20	0.12	
Life, Physical, and Social Science	595	51.8%	-0.18	0.16	
Office and Administrative Support	1,865	63.2%	-0.17	0.10	
Installation, Maintenance, and Repair	120	72.5%	-0.13	0.58	
Healthcare Support	475	63.8%	-0.12	0.23	
Production	185	68.6%	-0.11	0.27	
Legal	250	46.8%	-0.04	0.32	
Transportation and Material Moving	245	70.6%	-0.02	0.25	
Food Preparation and Serving Related	835	65.5%	-0.01	0.17	
Building and Grounds Cleaning and Maintenance	140	65.0%	0.04	0.31	
Protective Service	210	61.4%	0.16	0.22	
Construction and Extraction	210	71.0%	0.23	0.32	
Farming, Fishing, and Forestry	40	62.5%	0.72	0.79	

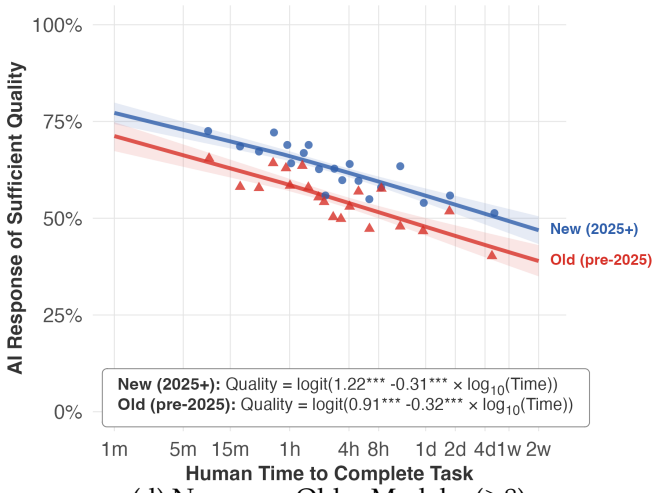
Notes: The table reports job-family-specific logit slopes from regressing manager acceptance on $\log_{10}$(time to complete). "N" denotes the number of observations. The "success rate" is the share of task instances with a score ≥ 7. "Coef. β" reports the slope estimate from Eq. (1) (estimated without controls), and "SE" reports standard errors clustered at the participant level. "Fit" provides a compact visualization of the estimated relationship, color-coded by slope magnitude, with statistically insignificant estimates shown in gray. Standard errors are clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Figure 5: Task Automation and Task Length: Model Size and Vintages

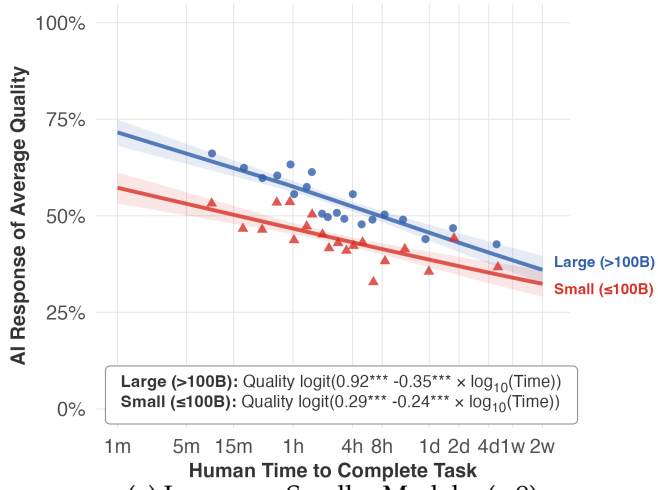
(a) Larger vs. Smaller Models: (≥7)



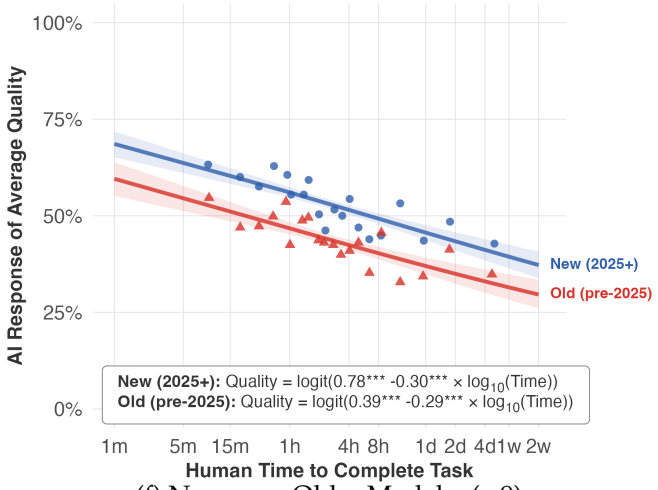
(b) Newer vs. Older Models: (≥7)



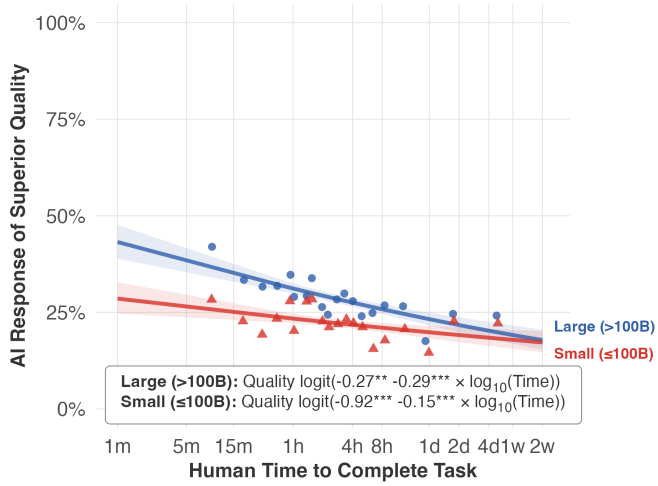
(c) Larger vs. Smaller Models: (≥8)



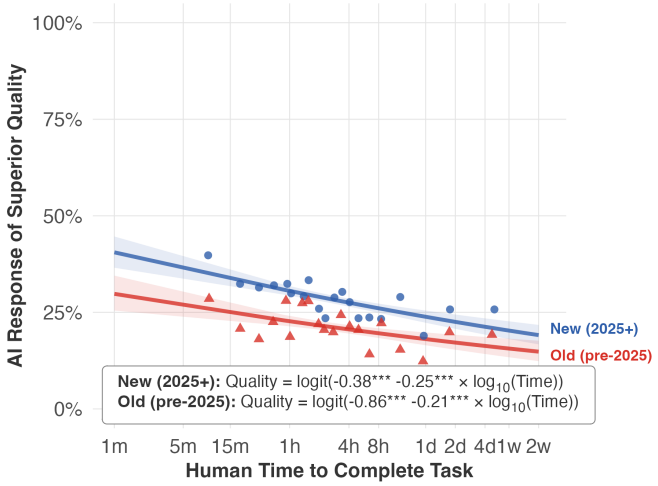
(d) Newer vs. Older Models: (≥8)



(e) Larger vs. Smaller Models: (=9)



(f) Newer vs. Older Models: (=9)



Notes: Panels (a)–(b) estimate Eq. (1) separately for larger vs. smaller models and newer vs. older models using "sufficient quality" (score $\geq$ 7) as the outcome. Panels (c)–(d) repeat this for "average quality" (score $\geq$ 8), and panels (e)–(f) for "superior quality" (score = 9). Coefficients are reported as log-odds. Model group definitions (large/small; new/old) are listed in Footnote 9. Standard errors are clustered at the participant level. Significance levels: *** 1%, ** 5%, * 10%.

Predicted task duration and success rates over time. Next, we examine how success rates have evolved over time, extending the earlier comparison of newer versus older models. To that end, we estimate a modified version of Eq. (1) that includes a linear trend for model release dates (R_m), which models a shift in the logistic curve under a constant slope (Section 4.2 shows how this specification theoretically relates to Eq. (1)).[11] Formally:

$$\Pr(Y_{ijsm} = 1) = \Lambda(\alpha + \delta R_m + \beta \log_{10} T_{js}) \,, \tag{2}$$

We estimate Eq. (2) using only *frontier* models (see figure notes).[12] We focus on the ≥ 7 threshold ("no edits required"). Based on the estimating Eq. (2), the reported lines in Figure 6, Panel (a), display the evolution of success rates for different task durations. To validate the model fit across the data, we additionally report point estimates from quarter-specific regressions of Eq. (1), which provide a more flexible and demanding (i.e., less precise) specification than our baseline model in Eq. (2). Reassuringly, both approaches yield consistent patterns (and we stick to our baseline model for interpretation).

We find that success probabilities increase rapidly and at a similar pace across all task-durations (or, equivalently, initial success rates), consistent with a broad-based capability improvement across the task-duration distribution (a fast rising-tide pattern). Based on these curves, we approximate failure-rate halving times (the failure rate is 1 minus the success rate), which equal 2.4–3.2 years over this period.[13] These halving times are high and correspond to an improvement of success rates by 15-16 percentage points across the depicted task durations over our observation period (we extrapolate the curves beyond our observation period in Figure 7). Note that tasks spanning varying durations (from five minutes to 24 hours) do not exhibit strikingly different success rates in terms of levels. For example, in 2025-Q3 predicted success rates range from about 79% to 58%, reinforcing the flat success–duration profile documented above.

Figure 6, Panel (b), presents the mirror image of Panel (a): estimating Eq. (1) for each quarter, we find the implied task duration achievable at a given probability of success. The curves are all parallel shifts because of the logarithmic vertical axis. Already by 2024-Q2, our econometric model predicts that frontier systems reached a 50% success rate (score ≥ 7) on tasks that take humans three hours to complete.

At higher success rate thresholds, implied feasible task duration is much shorter: for example, at an 80% success rate threshold, predicted task duration never exceeds about five minutes during our period of analysis. Nonetheless, improvements are rapid at every success rate threshold. We illustrate this pace by computing doubling times that we report alongside the figure (doubling times are directly inferred from the linear relationship between log-duration and time in Panel (b)). The implied feasible task duration roughly doubles every 3.8 months, with comparatively tight confidence bands.[14]

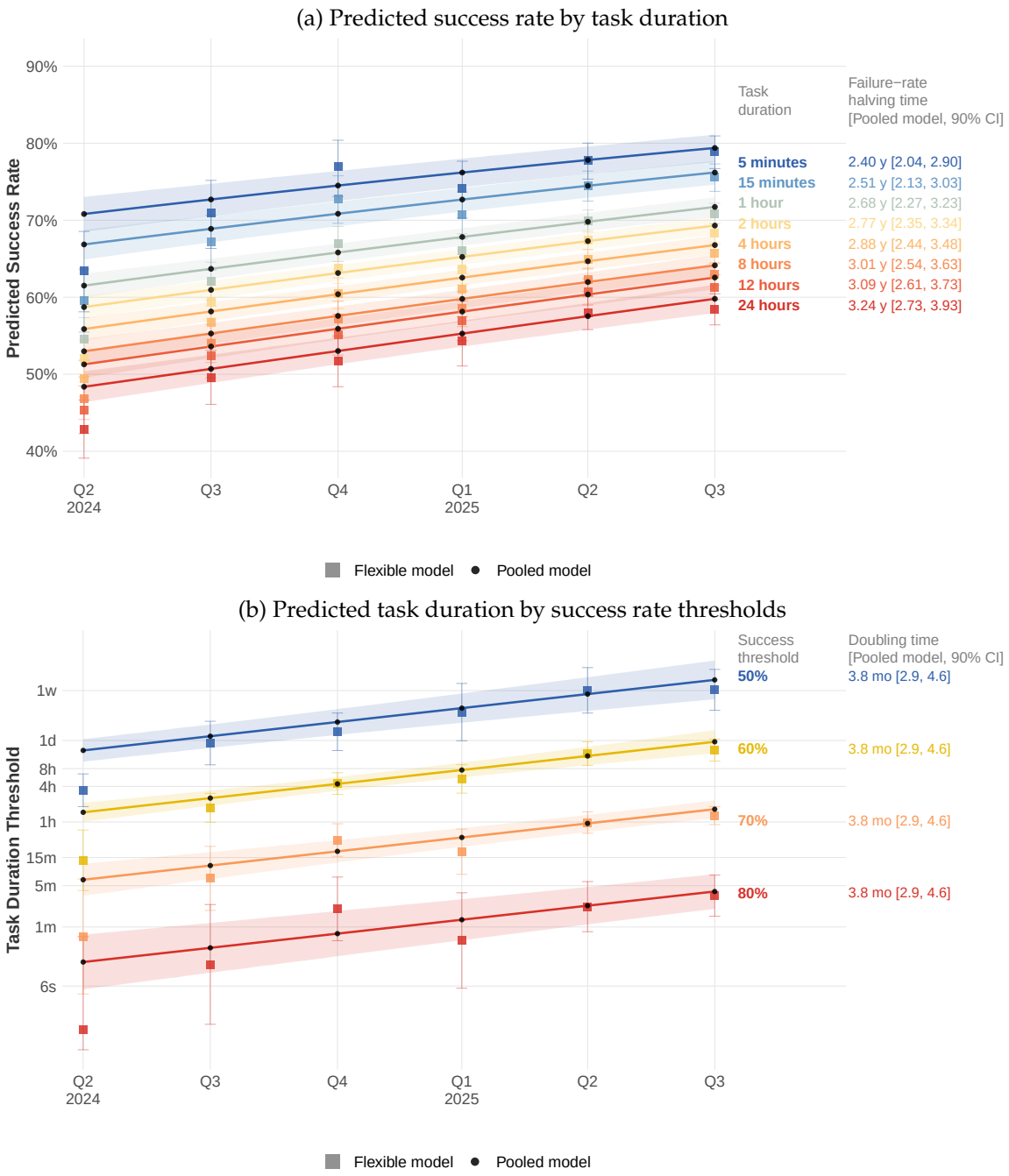
[11] In Appendix Table A.4 we additionally allow for changes in the slope over time and do not find evidence for statistically significant changes in slope coefficients over time (in line with Figure 5). We therefore rely on a model without time-depending slope coefficients.

[12] Results are qualitatively robust to alternative definitions of frontier models; see Appendix Figure A.3.

[13] Halving times are calculated locally and are not stable over time. We estimate these rates at the midpoint of our sample (December 2024) and compare the failure rate at this point to the rate one year later to derive the implied halving time.

[14] Note: the doubling time is also affected by the slope of the logistic curve. We illustrate this in Appendix Figure A.4. At

Figure 6: Task Duration and Success Rate Thresholds over Time



Notes: The lines in both panels are derived from estimating Eq. (2) on all task-level observations for frontier models across the full observation period (i.e., our "pooled" model). After the estimation, in Panel (a), we predict success rate changes based on given task length and a given linear (in logistic space) log-odds shifter (δR_m in Eq. (2)). Panel (b) instead predicts task duration for given success rates. The point estimates in both panels (i.e., "flexible model") are derived from estimation Eq. (1) separately for each quarter (which allows for quarter specific logistic slope coefficients) using the same approach of predicting success rates for given task durations (Panel (a)) and task durations for given success rates (Panel (b)). Shaded bands and error bands around point estimates indicate 90% confidence intervals. Standard errors are clustered by participants. Failure-rates halving times are locally approximated at the midpoint of the curves. Frontier models are Claude Opus 3, GPT-4, and Qwen 2 72B Instruct for 2024-Q2, Claude Opus 3, Gemini 1.5 Pro, GPT-4o mini, Llama 3.1 405B Instruct, and Qwen 2 72B Instruct for 2024-Q3, DeepSeek V3, Gemini 1.5 Pro, GPT-4o, GPT-4o mini, and Llama 3.1 405B Instruct for 2024-Q4, Claude Sonnet 3.7, DeepSeek R1, DeepSeek V3, Gemini 1.5 Pro, and GPT-4o for 2025-Q1, Claude Sonnet 3.7, Claude Sonnet 4, DeepSeek R1, DeepSeek V3, Gemini 2.5 Flash, Gemini 2.5 Pro, o3, o4 mini, and Qwen 3 235B for 2025-Q2, and Claude Opus 4.1, Claude Sonnet 3.7, Claude Sonnet 4, Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-5, GPT-5 mini, o3, and o4 mini for 2025-Q3.

2.3 Future Impacts

Can our findings inform expectations about future AI capabilities? Using estimates from Eq. (2), we project how success rates evolve as newer models are released (i.e., extending Figure 6, Panel (a) into the future and to unobserved task durations/initial success rates). Starting from a given success rate in 2024-Q2 (or, equivalently, a given initial feasible task duration level), we extrapolate the effect of subsequent model releases, captured by R_m, using the estimated coefficient, δ.

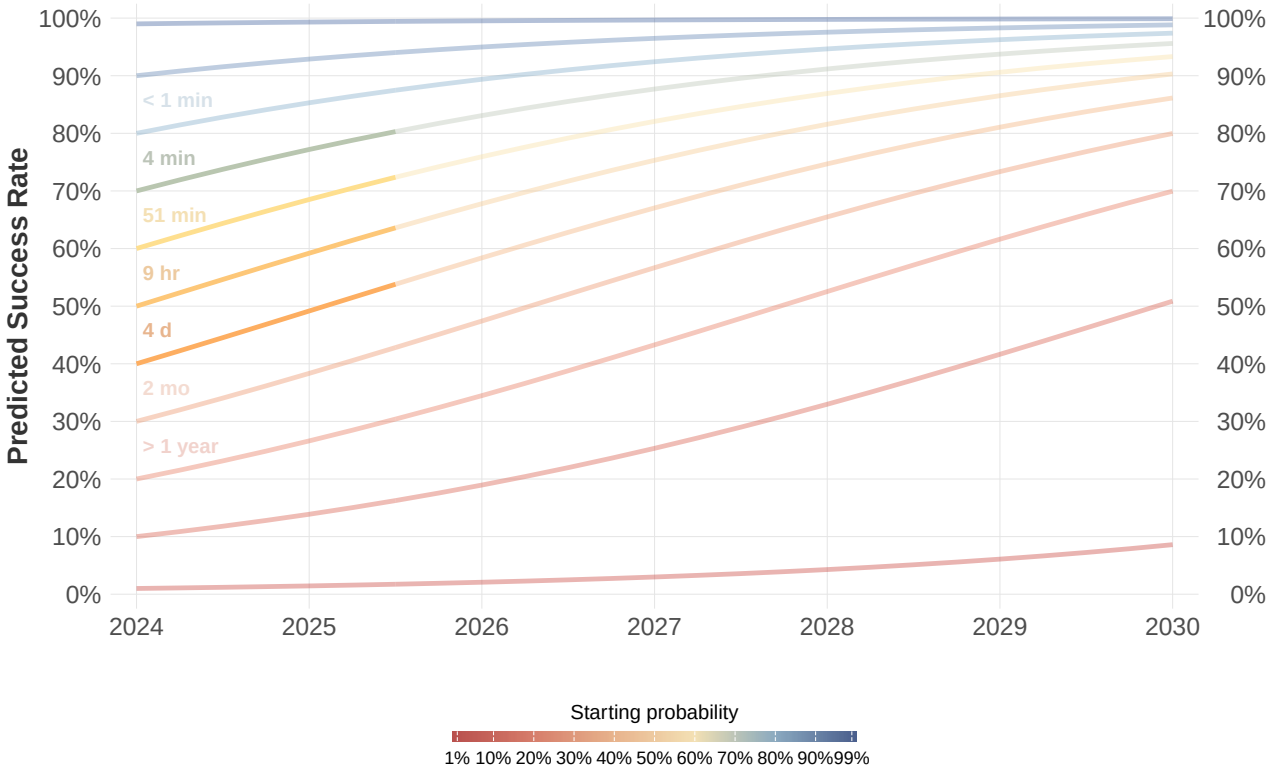
Figure 7 shows the resulting projections. The faded portions of each curve indicate extrapolations beyond the observed period or into regions with limited data support (curves are shown in full saturation only where the implied task duration falls within the 5th to 95th percentile range of the observed distribution). As shown by the saturated portions of the curves, the bulk of observed tasks lies within a relatively narrow success-rate range of 40% to 70%. Because the release-date term, δR_m , enters additively in the logit specification of Eq. (2), it implies a sigmoidal path in probability space. As a result, projected trajectories differ depending on the initial success rate.

Consistent with our earlier results, projected gains are gradual rather than abrupt. Nevertheless, the pace of improvement remains substantial for reaching high success rates across most text-based labor market tasks; most tasks are projected to attain AI success rates of 80%–95% by 2029 at a minimally sufficient quality level (with the majority of tasks in our survey being a few hours long, corresponding to a success rate of close to 90% in 2029). At the same time, due to the logistic relationship (and the flat estimates logistic slopes), our findings suggest that achieving consistently near-perfect performance (i.e., success rates close to 100%) across most text-based tasks may still take years, especially for tasks where current performance remains low. Therefore, while progress is significant, widespread automation, particularly in domains with low tolerance for errors, may still be some distance away.

Finally, we note an important cautionary point. The predictions in Figure 7 rely on extrapolation, under the assumption of a stable logistic relationship and continued improvement at the rate observed over recent years. Particularly in the tails of the logistic function (most relevant for our forward projections) alternative functional forms could yield materially different estimates of when AI reaches specific success thresholds. There are substantial reasons to question whether such progress can continue unchanged. For frontier models, compute has already scaled by several orders of magnitude (Mertens et al., 2026), and there are good reasons why continued investment in scaling may become prohibitively high (Thompson et al., 2020, Brown et al., 2026). In addition, algorithmic progress may be showing signs of slowing (Gundlach et al., 2025), while physical limits may impose increasingly binding constraints on further hardware advances (Shalf, 2020). These considerations suggest that future gains in AI capabilities may proceed more slowly than our extrapolations imply. Accordingly, our estimates are best interpreted as an upper bound under continued trend growth.

the extreme, a nearly flat line shifted just slightly up (and thus comprising almost no additional automation) would exhibit an enormous fast doubling rate. As such, our preferred measure of change is the change in success rates over time (Panel (a)), rather than the shift in task-duration at given success rates (Panel (b)).

Figure 7: Predicted AI Success Rates Over Time



Notes: The figure reports predicted AI success rates over time based on estimates of Eq. (2) using all task-level observations for frontier models across the full sample period. In Panel (a), we project changes in success rates as a function of task duration and a linear log-odds shift in logistic space (captured by δR_m in Eq. (2)). Relative to Figure 6, Panel (a), we extend these predictions into future periods and into regions of the data with sparse or no observations (specifically, beyond the 5th to 95th percentile range of the task duration distribution). The saturated segments of the curves reflect predictions grounded in observed data, whereas the faded segments indicate extrapolated regions with limited or no empirical support. Frontier models are Claude Opus 3, GPT-4, and Qwen 2 72B Instruct for 2024-Q2, Claude Opus 3, Gemini 1.5 Pro, GPT-4o mini, Llama 3.1 405B Instruct, and Qwen 2 72B Instruct for 2024-Q3, DeepSeek V3, Gemini 1.5 Pro, GPT-4o, GPT-4o mini, and Llama 3.1 405B Instruct for 2024-Q4, Claude Sonnet 3.7, DeepSeek R1, DeepSeek V3, Gemini 1.5 Pro, and GPT-4o for 2025-Q1, Claude Sonnet 3.7, Claude Sonnet 4, DeepSeek R1, DeepSeek V3, Gemini 2.5 Flash, Gemini 2.5 Pro, o3, o4 mini, and Qwen 3 235B for 2025-Q2, and Claude Opus 4.1, Claude Sonnet 3.7, Claude Sonnet 4, Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-5, GPT-5 mini, o3, and o4 mini for 2025-Q3.

3 Discussion

AI capabilities. Our preliminary results inform the trajectory of AI capabilities and their implications for labor markets and the broader economy. Across realistic and representative tasks, the association between LLM performance and task duration is well approximated by a relatively flat, near-linear relationship rather than a steep, wave-like pattern. Put differently, models perform, on average, not dramatically differently on short versus long tasks across most job domains (one notable exception: personal care and service tasks). With successive model releases, this relationship shifts upward in an approximately parallel fashion. We refer to this pattern as *rising-tide* automation. Under this view, capability improvement typically does not arrive in isolated "bursts" confined to a narrow set of tasks. Instead, LLMs improve performance broadly across both short- and long-duration tasks.

Importantly, the flat success–duration gradient does not imply that LLM capabilities are growing slowly, nor that workers will be insulated from AI automation effects. We observe a significant AI success rate improvements across the vast majority of partially text based tasks in our survey. Our findings do suggest, however, that workers are likely to have some visibility into these changes, rather than facing discontinuous jumps in AI-driven automation. Even under an arguably *upper-bound* (i.e., aggressive) projection, we estimate that by 2029 AI systems will complete most text-based tasks with success rates of 80%–95% (at a minimally sufficient quality level), while near-perfect automation remains several years further out. While such gradualism is not inherently protective, it may provide workers with more time to adjust, particularly compared to a "crashing wave" scenario, in which automation risks appear limited until shortly before widespread disruption occurs.

Comparison to recent work by METR. Across specifications, the estimated relationship between success and task duration is consistently shallow: slope coefficients are small, and most observations lie in the approximately linear region of the logistic curve. This contrasts with Kwa et al. (2025) and METR (2025), where the curve gradients are steeper and, as a result, shorter/longer tasks are located closer to the logistic tails. Put differently, through the lens of Figure 1, prior benchmark-based evidence aligns more closely with a crashing wave-like pattern, whereas our results support a rising-tide view across most job domains.

One potential explanation for the contrasting results could be the coverage of very short tasks. Our data do not include extremely short (and simple) items with near-certain success, such as producing a single programming command, whereas task durations in Kwa et al. (2025) range from seconds to several hours. However, this cannot fully account for the difference: we continue to observe a near-linear success–duration relationship even at longer durations. Moreover, in an unreported robustness check, we re-estimated the Kwa et al. (2025) curves while excluding the shortest tasks and consistently obtained steep slopes ($\beta = -1.08$ in the full sample versus $\beta = -0.99$ after excluding short tasks), suggesting that task coverage alone is unlikely to explain the discrepancy.

A potentially more important explanation is that the underlying task environments differ. Kwa et al. (2025) focus on relatively deterministic research and software-engineering tasks (e.g., identifying a subtle bug or implementing a known algorithm), whereas we study a broad set of real-world, text-based labor-market tasks that often involve non-deterministic instances and a wider mix of domain knowledge and skills.[15]

Another potential explanation is that pooling across models (and across domains) may attenuate the estimated slopes. In particular, larger models may be disproportionately better at completing shorter tasks, as suggested by Figure 5. While our sample is still limited, in an unreported analysis, we do not find that model-specific curves are systematically steeper than our pooled estimates, which is reassuring. We will investigate this issue further as the number of survey responses grows.

As in Kwa et al. (2025) and METR (2025), we translate the estimated curves into implied doubling times for feasible task duration at fixed success rate thresholds. Kwa et al. (2025) report doubling times of roughly 4–7 months at the 50% success rate threshold, while METR (2025) find broadly similar but somewhat faster rates of 2–6 months (with some exceptions). Our estimates fall toward the faster end

[15]METR (2025) analyze multiple benchmark datasets but do not report logistic shapes across benchmarks.

of this range, with a doubling time of about 3.8 months at the 50% threshold.[16] We also find that doubling times *shorten* at higher success rate thresholds, reflecting faster AI improvement at higher reliability levels.

Finally, under our baseline ≥ 7 threshold ("no edits required, minimally sufficient"), overall performance levels differ markedly. On our real-world labor-market tasks, frontier models in 2024-Q2 attain an implied feasible task duration of roughly three hours at a 50% success rate, substantially above the 8–15 minutes reported in Kwa et al. (2025). However, this comparison should be interpreted with caution, as Kwa et al. (2025) examines deterministic tasks, whereas our tasks are non-deterministic and drawn from real-world labor-market settings. In addition, the implied feasible task duration decreases sharply as we tighten the scoring threshold, falling to just 9 minutes at the ≥ 8 (average quality) threshold.[17]

Labor Market Impacts. Importantly, the success rates achieved by LLMs in this analysis should *not* be interpreted as implying that a corresponding share of tasks can (or should) be automated today, for three main reasons. First, our data collection is incomplete and likely over-represents occupations that are easier to survey (e.g., potential candidates are more numerous or more willing to participate). It is plausible that harder-to-survey occupations are also harder to automate, which would bias our estimates upward and imply lower overall automation success rates. Second, our survey setup provides the information required for LLMs to perform each task. In practice, integrating such information may be difficult, costly, or subject to regulatory constraints, rendering some tasks infeasible to automate in real-world settings.[18] Third, our analysis does not account for the economic attractiveness of deployment, which prior work identifies as a key determinant of automation feasibility Svanberg et al. (2024). In particular, "last-mile" implementation costs (Fleming et al., 2024) may be substantial and could limit adoption, especially among smaller firms.

The ultimate labor market impacts are further complicated by the distinction between task automation, which we analyze, and worker impacts that happen at the occupation level. As shown in Autor and Thompson (2025), the loss of individual tasks does not necessary hurt the employees. Indeed, based on the expertise of task and how that relates to the occupation's bundle of tasks, automation could increase or decrease wages. It could also increase or decrease the employment in that occupation. The labor impacts will thus be a combination of task automation that happens and occupational-level responses that follow.

Further Limitations. Our findings are preliminary and are based on an incomplete subset of an ongoing survey (as discussed above). Moreover, our analysis is restricted to tasks we classify as offering at least 10% potential time savings from LLM use. Accordingly, we do not claim rapid automation potential across all human work. Rather, our evidence points to a broad and fast-rising expansion of AI capabilities within text-centric real-world labor-market tasks where LLMs plausibly deliver meaningful time savings.

[16] Again, owing in part to a much shallower logistic curve. See Appendix Figure A.4.

[17] At the ≥ 9 (superior quality) threshold, no task duration reached a success probability above 50%.

[18] We plan to address these issues in follow-on work.

Another limitation of our approach is that, despite careful task and survey design, we may fail to capture important dimensions of real-world work and its evaluation. In particular, we require each task instance to be self-contained, with all relevant information provided in the prompt. This constraint limits our ability to represent tasks that depend on interaction with external artifacts (e.g., opening and editing files, navigating software environments) or on repeated, multi-turn interactions with other people that cannot be condensed into a single vignette. We partially mitigate this concern by restricting our analysis to instances that evaluators deem realistic and representative of the underlying task. Nonetheless, some dimensions of work are excluded by construction.

A related concern is that respondents without relevant task experience may attempt to complete the survey. To address this, we implement extensive validation procedures for both respondents and their answers before inclusion in the analysis. For a subset of occupations, we can directly verify prior work experience for some participants; we find no meaningful differences in responses between these individuals and those who pass our implicit validation checks.

Finally, our survey design evaluates success based on the final output rather than the process used to produce it. While this aligns with how many tasks are assessed in practice, it may introduce measurement noise if outputs are imperfect proxies for true task performance. Such noise is unlikely to bias our estimates materially, but it may reduce explanatory power.

Future work. In future work, we will provide a more detailed account of our survey and extend the analysis in several directions. This includes (i) updating our findings using the full sample, (ii) offering a more comprehensive view of AI capabilities across tasks and occupations, and (iii) examining the implications for labor automation, including which task-specific constraints are likely to hinder real-world adoption. In particular, it will be important to reconcile the high levels of performance we document across many task domains with the still limited adoption of AI at the firm level—especially outside of high-tech industries (McElheran et al., 2024).

4 Methods

4.1 Survey Collection Details

Figure 8 provides an overview on the survey design and data collection that we detail in the following. The prompts we used during the survey design are detailed in Appendix E.

Task selection. We base our definition of labor market tasks on the O*NET 29.2 database (O*NET (2024)). Because not all tasks have meaningful LLM automation potential (e.g., predominantly physical tasks), we used GPT-4—the most advanced OpenAI model at the time—as a classifier to identify tasks where LLMs could lead to at least 10% time savings.[19] This procedure yielded 11,768 out of 18,786 tasks (62.6%) with at least 10% estimated automation potential .

[19]This is similar to Eloundou et al. (2024). The difference is that while Eloundou et al. (2024) use a 50% time savings threshold to classify automation exposure, our 10% threshold identifies a wider array of tasks with economically meaningful automation potential.

Task instance generation. For each of the qualified O*NET tasks, we generated six task instances using GPT-5 with a structured prompt that incorporated the occupation title and the corresponding O*NET task description. We constrained each instance to a single coherent scenario, including technical details only when necessary for the role. Task difficulty was calibrated to reflect the expected performance of an experienced worker and aligned with the occupation's typical education and training requirements.

Each instance was limited to approximately 150 words to reduce evaluator cognitive load, incorporated at least one professional perspective (e.g., technical, procedural, interpersonal, or strategic), and followed a standardized format to ensure consistency.

Task instance filtering. We implemented a filtering process to ensure that generated task instances aligned with our research objectives. Specifically, we used GPT-5 to verify that each instance (i) represented a meaningful portion of the corresponding O*NET task, (ii) could be completed by an LLM (i.e., required text output only), and (iii) contained all necessary information to solve the task without external inputs. All task instances that met these criteria were included in the survey. Instances that did not meet these criteria were regenerated and re-evaluated, with up to 10 iterations per task. We excluded 232 O*NET tasks for which we could not generate a sufficient number of valid instances in this step. Ultimately, we included 11,536 tasks with 69,216 task instances in the survey. Appendix E provides details on the filtering prompts.

Once a task instance was included in the survey, we conducted a final validation step: participants were asked whether the instance was realistic and representative of the underlying O*NET task.[20] If a participant judged an instance as unrealistic or unrepresentative, it was removed from the survey pool (including for future participants) and replaced with a new instance. This resampling procedure was feasible because, across all participants, only two task instances were evaluated per O*NET task.[21]

LLM response generation For each task instance included in the survey, we generated responses using 41 LLMs of varying scales and capabilities released between June 2023 and August 2025, including both open-weight and proprietary models (listed in Appendix F). All models were queried with an identical prompt and default generation settings to ensure that differences in expert evaluations reflect variation in model capabilities rather than prompt optimization. Responses were capped at 700 words to balance substantive depth with the cognitive burden on the expert evaluators.

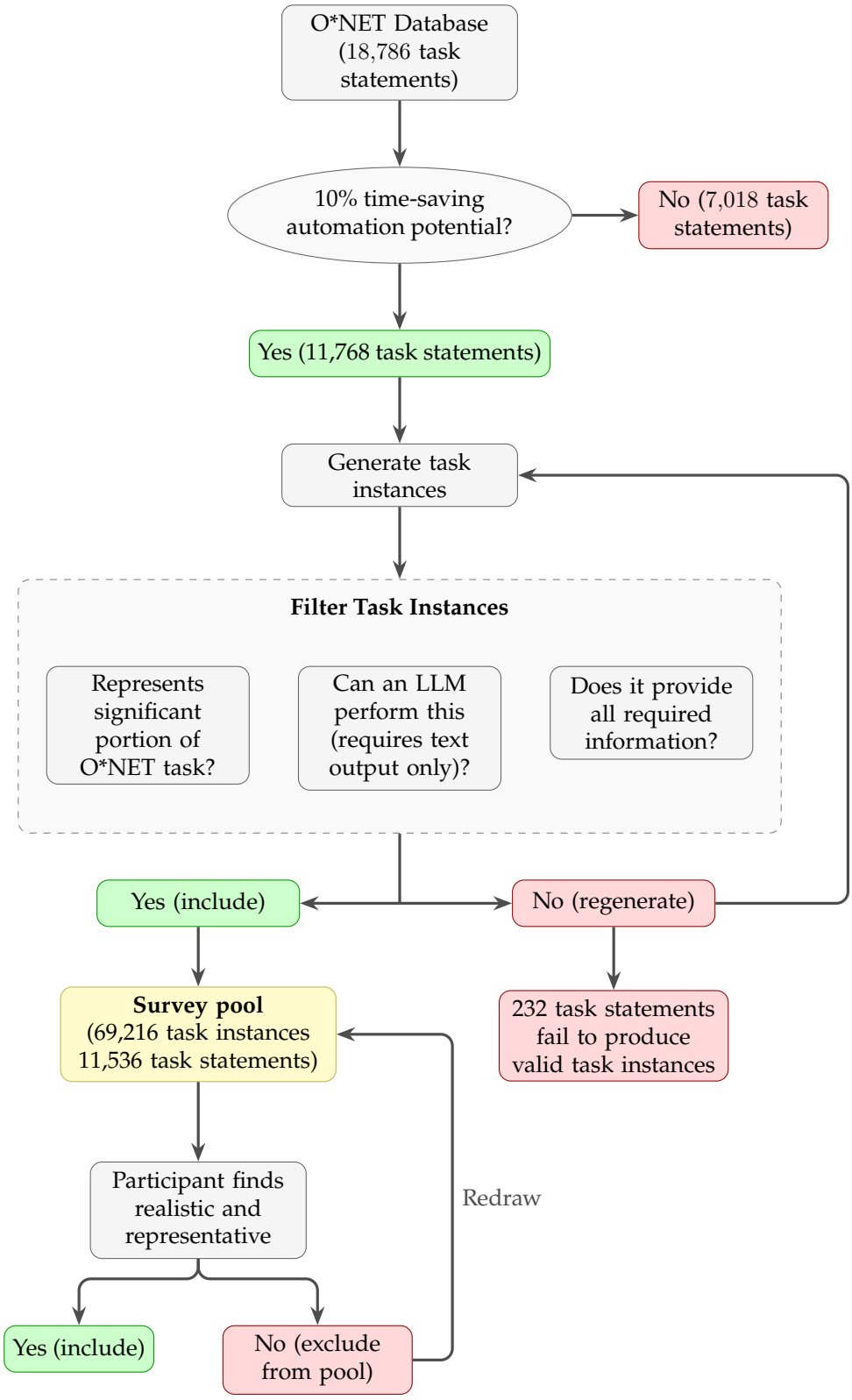
Survey data collection and participants We began collecting domain expert evaluations via the Prolific platform on September 22, 2025; data collection is ongoing. To qualify, participants had to reside in the United States, hold a platform approval rating of at least 75%, and have at least six months of work experience in the occupation under evaluation. 21% of participants who attempted to take the survey were excluded due to the work experience restriction.

At the start of the survey, participants selected a task from a list corresponding to their occupation. They were then shown a task instance and asked to confirm that they understood it and, as a secondary

[20] For a pilot subsample pertaining to 5.8% of the sample, we did not ask participants if the response was representative.

[21] In practice, no case occurred in which all six task instances were deemed unrealistic or unrepresentative; had this happened, we would have generated additional instances.

Figure 8: Task Filtering Process



validation of prior model filtering, that it was both realistic and representative of the underlying task. If any of these three conditions was not met, a new instance was provided. Once an instance passed validation, participants evaluated responses from five different LLMs.[22] Participants were not informed that the responses were generated by AI models. After reviewing all responses from the 5 different LLMs, participants were asked if they felt they had enough information and context about the task to accurately evaluate the responses.[23] We did not use data from any participant who did not agree or strongly agree.

For each of the 11,536 tasks, we collected evaluations for two task instances, with five model responses assessed per instance (all five evaluated by the same expert). Experts reported contextual task information (e.g., time required, difficulty, and frequency in the job).[24] They evaluated each model response on a 1–9 scale, where 1 indicates that the response needed to be started over from scratch and 9 indicates that the response was of above-average quality for a human worker, as discussed above.

Survey data included in these analyses underwent rigorous quality assurance. Participants who failed more than one of four attention checks were excluded (9.8%). We further excluded participants exhibiting highly repetitive response patterns (i.e., limited variation in model evaluations), providing implausible estimates of task completion time, spending only minimal time on survey pages or showing no measurable engagement, or giving internally inconsistent (contradictory) responses. In total, approximately 34.6% of collected data were excluded to ensure high data quality.

As data collection is still ongoing, the preliminary data presented here is not a perfectly representative subset of the target set of 900+ occupations the study will eventually cover. The sample presented in our analyses is composed of occupations with slightly lower wage levels and experience requirements as compared to the target distribution ($29 vs $33 median wage, 1.8 vs 2 years of work experience required). The sample also slightly over-represents occupations which require a bachelor's degree or less education, and slightly under-represents occupations which require post-graduate education. Qualitatively, our current sample represents more white-collar occupations and under-represents blue-collar work.

Summary statistics and task instance examples. Table 2 provides summary statistics on key variables used throughout the paper. We pool the data across all tasks and models and report means and distribution characteristics. Table 3 provides examples of task instances for shorter and longer tasks (according to participant evaluations) in our data. In Appendix D we list examples of LLM responses.

[22] Our data includes subsamples where this order was randomized and some where it was not. In Appendix B, we test for differences between these and show that non-randomization does not meaningfully change our results.

[23] For a pilot subsample pertaining to 2.2% of the sample, this question was not included.

[24] For a pilot subsample pertaining to 2.2% of the data, the timing of the question regarding the time required to complete the task is different: in the pilot, respondents answered this question before viewing the LLM response, whereas in the main survey, they were asked after seeing the LLM output. Appendix C demonstrates that our main findings are robust to excluding the pilot data entirely.

Table 2: Summary Statistics

	Mean	St.Dev.	P10	P50	P90
	(1)	(2)	(3)	(4)	(5)
Score	6.88	1.95	4	7	9
Manager Acceptance (0/1; 1 if score ≥ 7)	0.60	0.49	0	1	1
Time required to complete task instance (hours)	11.81	29.11	0.42	2.5	30
Observations: 17,205					

Notes: This table presents summary statistics for key survey variables. Columns (1)-(5) report means, standard deviations, and values for the 10^{th}, 50^{th}, and 90^{th} percentile.

Table 3: Task Instance Examples.

Length	O*NET Task	Task instance
5 minutes	Prepare checks that itemize and total meal costs and sales taxes.	Your POS just went down. For Table 12 (party of five) at 5:45 p.m., prepare three handwritten checks that itemize and total meal costs and sales taxes. Local tax: 8% on food, 10% on alcohol. Happy hour applies: appetizers 50% off before 6 p.m. Orders: - Appetizers: Calamari $12, Nachos $10 (manager comped 100%—do not tax). - Entrees: Burger $14, Pasta $16, Salmon $22. - Drinks: 2 Cocktails $11 each, 1 Beer $6. Split: - Check A (Guests 1–2): Burger, Pasta, 2 Cocktails, and half Calamari. - Check B (Guest 3): Salmon, Nachos (comped). Apply a $15 gift card to this check. - Check C (Guests 4–5): Beer and half Calamari. Instructions: - Show each item with price/discount, separate food vs alcohol subtotals, apply correct tax per category, then total. - Ensure the comped Nachos are $0 and not taxed. - Round to the nearest cent.
30 minutes	Assist students who need extra help with their coursework outside of class.	During evening office hours, a multilingual student who missed a key seminar needs targeted help to revise a 1,200-word literary analysis due tomorrow on how two critics interpret a single poem. The draft has a vague thesis, quotation drops, patchwriting risks, and inconsistent Modern Language Association (MLA) 9 citations. The student has an accommodation for dyslexia and prefers structured outlines and color-coded feedback. In a 30-minute Zoom, outline exactly how you will: (1) triage the draft against the rubric; (2) guide a 5-minute close reading of one stanza to generate a precise, arguable thesis; (3) build a reverse outline for paragraph coherence; (4) model integrating quotations with signal phrases and analysis; (5) correct one in-text citation and one Works Cited entry (journal article with DOI); and (6) create a clear post-session revision checklist.
4 hours	Create project status presentations for delivery to customers or project personnel.	You must prepare a 10–12 slide project status presentation for a quarterly customer steering committee (executives and engineering leads) on a 9-month SaaS integration project, now at Month 5. Include: - Baseline vs. current: SPI 0.87, CPI 0.94, 62% scope complete (baseline 68%), forecast finish slips by 3 weeks. - Milestones: Data API (done), SSO (at risk), Reporting (not started). Critical path impacted by vendor sandbox delay (14 days). - Quality: UAT defect density 0.8/Story Point (target 0.5). - Budget: $2.1M EAC vs. $2.0M BAC, 5% variance. - Change requests: CR-014 (expanded reporting) approved; CR-017 (custom SSO flows) pending. - Risks/Issues: resource contention with Security team; single point of failure on lead architect; vendor instability watch. - Stakeholder concerns: customer wants go-live unchanged. Deliver: clear RAG (red/amber/green), one-page RAID (Risks, Assumptions, Issues, Dependencies), decision requests (scope trade-offs), and a reconciled metric view (Jira says 65% complete; Finance shows 58%—explain).
1 week	Devise programs to develop executive potential among employees in lower-level positions.	You are the Training and Development Specialist at a 1,200-employee manufacturing firm with 18% annual turnover in frontline supervisor roles. The COO asks you to design a 9-month "Emerging Leaders" program for high-potential hourly and entry-level salaried employees across three shifts and two sites. Constraints: $150,000 total budget, 10% time away from job max, union environment, mixed on-site/remote access. Requirements: define unbiased selection criteria, integrate 360-degree feedback and an initial assessment center, include coaching/mentoring (executive sponsors), 3 role-rotation or stretch assignments aligned to business KPIs (safety, yield, on-time delivery), and a capstone improvement project per participant. Deliverables: cohort design (20–30 participants), curriculum outline (modalities, schedule), manager involvement plan, measurement plan (leading/lagging KPIs, promotion/readiness metrics at 6/12 months), and risk mitigation (coverage, buy-in, DEI). Describe your program design and evaluation approach.

4.2 Theoretical Foundations

Our empirical object is the probability that an LLM response would be *accepted without edits* for a given real-world task instance, as a function of the task's (human-reported) log duration, T_{js} (for instance s of task j). We estimate this relationship using logistic regression. This section motivates the specification and shows that, under one plausible micro-foundation, the estimated slope coefficient can be interpreted as reflecting the length of the underlying serial chain of sequentially dependent steps required to complete the task.

Coupled critical-path robustness as a logit model. Suppose an instance, s, of a task, j, within domain, d, requires $N_{djs}(T_{js})$ *coupled critical actions* that must be completed without a fatal error for the output to be acceptable (relative to the main analysis, we add index d to discuss domain-specificity, aligning with our analysis by job families). We express $N_{djs}(T_{djs})$ as a function of observed task duration, T_{djs}, scaled by a parameter γ_d that governs how sequentially coupled tasks are:

$$N_{djs}(T_{djs}) = N_{0d}\, T_{djs}^{\gamma_d}, \qquad N_{0d} > 0,\ \gamma_d > 0, \tag{3}$$

such that $\log_{10} N_{djs}(T_{djs}) = \log_{10} N_{0d} + \gamma_d \log_{10} T_{djs}$. A larger γ_d, implies that longer tasks in domain d become disproportionately more sequentially coupled. N_{0d} captures the baseline serial complexity within a domain.

Let B_{djsm} denote the number of serial critical actions model m can sustain before the first fatal mistake. Success occurs iff $B_{jsm} \geq N_{djs}(T_{djs})$ (and success empirically corresponds to the evaluator rating "minimal sufficient" with no edits required). If the log-horizon is logistic,

$$\log_{10} B_{djsm} = \eta_m + \nu_d + \varepsilon_{djsm}, \qquad \varepsilon_{djsm} \sim \text{Logistic}(0, \sigma), \tag{4}$$

where η_m captures a model-specific baseline robustness (how far a model can typically go in a serial chain) ν_d captures a domain-level baseline shift (how models generally perform within a domain), and ε_{djsm} denotes unobserved "failure" shocks. Given the structure of the error terms, B_{djsm}, is log-logistic, the survival probability has a closed form (Bennett, 1983; Kiefer, 1988). In particular:

$$\begin{aligned} \Pr(Y_{djsm} = 1 \mid T_{js}, m, d) &= \Pr(\log_{10} B_{djsm} \geq \log_{10} N_{djs}(T_{djs})) \\ &= \Lambda\left(\frac{\eta_m + \nu_d - \log_{10} N_{0d}}{\sigma} \ - \ \frac{\gamma_d}{\sigma} \log_{10} T_{djs} \right), \end{aligned} \tag{5}$$

where, as above, $\Lambda(z) = 1/(1 + e^{-z})$. Eq. (5) corresponds to Eq. (1), where we define: $\alpha_{md} = \frac{\eta_m + \nu_d - \log_{10} N_{0d}}{\sigma}$ and $\beta_d = -\frac{\gamma_d}{\sigma}$, and provides a micro-founded rational for our estimation approach (and the estimation approach used in other work, such as Kwa et al., 2025).[25] Through the lens of Eq.

[25]The same "first fatal error" framing yields a complementary log–log specification under a different assumption on how failures accumulate. If fatal errors arrive along serial exposure according to a Poisson process (constant hazard) or more generally a Weibull model (Weibull, 1951), then $p(T) = \Pr(Y = 1 \mid T) = \exp\{-C\, T^{\kappa}\}$ for constants $C, \kappa > 0$, implying $\log(-\log p(T)) = \log C + \kappa \log T$. This corresponds to a complementary log–log link in grouped-duration survival models (Jenkins, 1995; Prentice and Gloeckler, 1978). In Appendix Figure A.1 we re-estimate our main specification using this alternative link and obtain a qualitatively similar duration slope, suggesting a flat relationship between LLM performance and task duration.

5, differences in estimated logistic slope coefficients (as reported in Table 1), result from differences in the extent to which tasks within a domain (job family) are sequentially coupled. If longer tasks become more sequentially coupled, the slope coefficients become steeper (more negative). Shifts in the curve, on the other hand, are explained by model-specific capabilities (η_m) in solving coupled serial steps and domain-specific characteristics (ν_d).

Release-date shifts in model robustness. A parsimonious extension of this framework is to allow the location of the model-robustness distribution to drift with model release date. For the subset of frontier models used in the time-series analysis, suppose:

$$\eta_m = \eta_0 + \rho R_m, \tag{6}$$

where R_m denotes model release date and ρ captures how the distribution of $\log_{10} B_{djsm}$ shifts over time. A one-unit increase in R_m raises $\log_{10} B_{djsm}$ by ρ, implying a multiplicative increase of 10^ρ in the typical sustainable serial horizon. Substituting Eq. (6) into Eq. (4) yields:

$$\Pr(Y_{djsm} = 1 \mid T_{djs}, R_m, d) = \Lambda\left(\frac{\eta_0 + \nu_d - \log_{10} N_{0d}}{\sigma} + \frac{\rho}{\sigma} R_m - \frac{\gamma_d}{\sigma} \log_{10} T_{djs}\right). \tag{7}$$

Thus, release date generates a pure intercept shift in log-odds space: $\delta = \rho/\sigma$ captures systematic improvements in model robustness over time. Suppressing domain heterogeneity yields the empirical specification in Eq. (2): $\Pr(Y_{jsm} = 1) = \Lambda(\alpha + \delta R_m + \beta \log_{10} T_{js})$. The maintained constant-slope restriction therefore corresponds to assuming that newer models extend the length of critical paths they can sustain, but do not change how required serial depth scales with task duration (for which we find empirical support, see Table A.4).

Implications of a linear release-date trend in the logit. For a fixed task duration T, Eq. 2 implies:

$$\frac{p(T, R_m)}{1 - p(T, R_m)} = \exp(\alpha + \beta \log_{10} T)\, e^{\delta R_m}, \tag{8}$$

so a one-unit increase in release date multiplies the odds of success by e^δ. Hence the release-date effect is exactly exponential in *odds space*, not in probability space. In probability space, the implied path is logistic:

$$p(T, R_m) = \Lambda(c(T) + \delta R_m) = \frac{p_0(T) e^{\delta R_m}}{1 - p_0(T) + p_0(T) e^{\delta R_m}}, \tag{9}$$

where $c(T) = \alpha + \beta \log_{10} T$ and $p_0(T) \equiv p(T, 0) = \Lambda(c(T))$. It follows that

$$\frac{\partial p(T, R_m)}{\partial R_m} = \delta\, p(T, R_m)\big(1 - p(T, R_m)\big),$$

so absolute percentage-point gains are largest for tasks with intermediate baseline success rates and smaller near 0 or 1. In this sense, an additive linear trend in the logit produces a sigmoidal path in probability space (as shown in Figure 7).

References

Autor, D., & Thompson, N. (2025). Expertise. *Journal of the European Economic Association*, *23*, 1203–1271.

Bennett, S. (1983). Log-logistic regression models for survival data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *32*(2), 165–171.

Brown, Z. A., Martín Poza, J., & Thompson, N. (2026). The economics of ever-larger language models. *Mimeo*.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C., Le Bras, R., et al. (2023). Faith and fate: Limits of transformers on compositionality. *Advances in neural information processing systems*, *36*, 70293–70332.

Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2024). Gpts are gpts: Labor market impact potential of llms. *Science*, *384*(6702), 1306–1308.

Fleming, M., Thompson, N. C., & Li, W. (2024). The last mile problem in ai. *Washington: Brookings Institution*. http://www.brookings.edu/articles/the-last-mile-problem-in-ai/

Ge, H., Bastani, H., & Bastani, O. (2026). Are ai capabilities increasing exponentially? a competing hypothesis. *arXiv preprint arXiv:2602.04836*.

Gundlach, H., Fogelson, A., Lynch, J., Trisovic, A., Rosenfeld, J., Sandhu, A., & Thompson, N. (2025). On the origin of algorithmic progress in ai. *arXiv preprint arXiv:2511.21622*.

Jenkins, S. P. (1995). Easy estimation methods for discrete-time duration models. *Oxford Bulletin of Economics & Statistics*, *57*(1).

Kiefer, N. M. (1988). Economic duration data and hazard functions. *Journal of economic literature*, *26*(2), 646–679.

Kwa, T., West, B., Becker, J., Deng, A., Garcia, K., Hasin, M., Jawhar, S., Kinniment, M., Rush, N., Von Arx, S., et al. (2025). Measuring ai ability to complete long tasks. *arXiv preprint arXiv:2503.14499*.

McElheran, K., Li, J. F., Brynjolfsson, E., Kroff, Z., Dinlersoz, E., Foster, L., & Zolas, N. (2024). Ai adoption in america: Who, what, and where. *Journal of Economics & Management Strategy*, *33*(2), 375–415.

Mertens, M., Fischl-Lanzoni, N., & Thompson, N. (2026). Is there" secret sauce"in large language model development? *arXiv preprint arXiv:2602.07238*.

METR. (2025). How does time horizon vary across domains? https://metr.org/blog/2025-07-14-how-does-time-horizon-vary-across-domains/

O*NET. (2024). O*NET 29.2 Database. https://www.onetcenter.org/dictionary/29.2/excel/

Patwardhan, T. (2025). Gdpval: Evaluating ai model performance on real-world economically valuable tasks. *arXiv preprint arXiv:2510.04374*.

Prentice, R. L., & Gloeckler, L. A. (1978). Regression analysis of grouped survival data with application to breast cancer data. *Biometrics*, *34*(1), 57–67.

Shalf, J. (2020). The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *378*(2166).

Svanberg, M., Li, W., Fleming, M., Goehring, B., & Thompson, N. (2024). Beyond ai exposure: Which tasks are cost-effective to automate with computer vision? *Available at SSRN 4700751*.

Thompson, N. C., Greenewald, K., Lee, K., Manso, G. F., et al. (2020). The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, *10*(2).

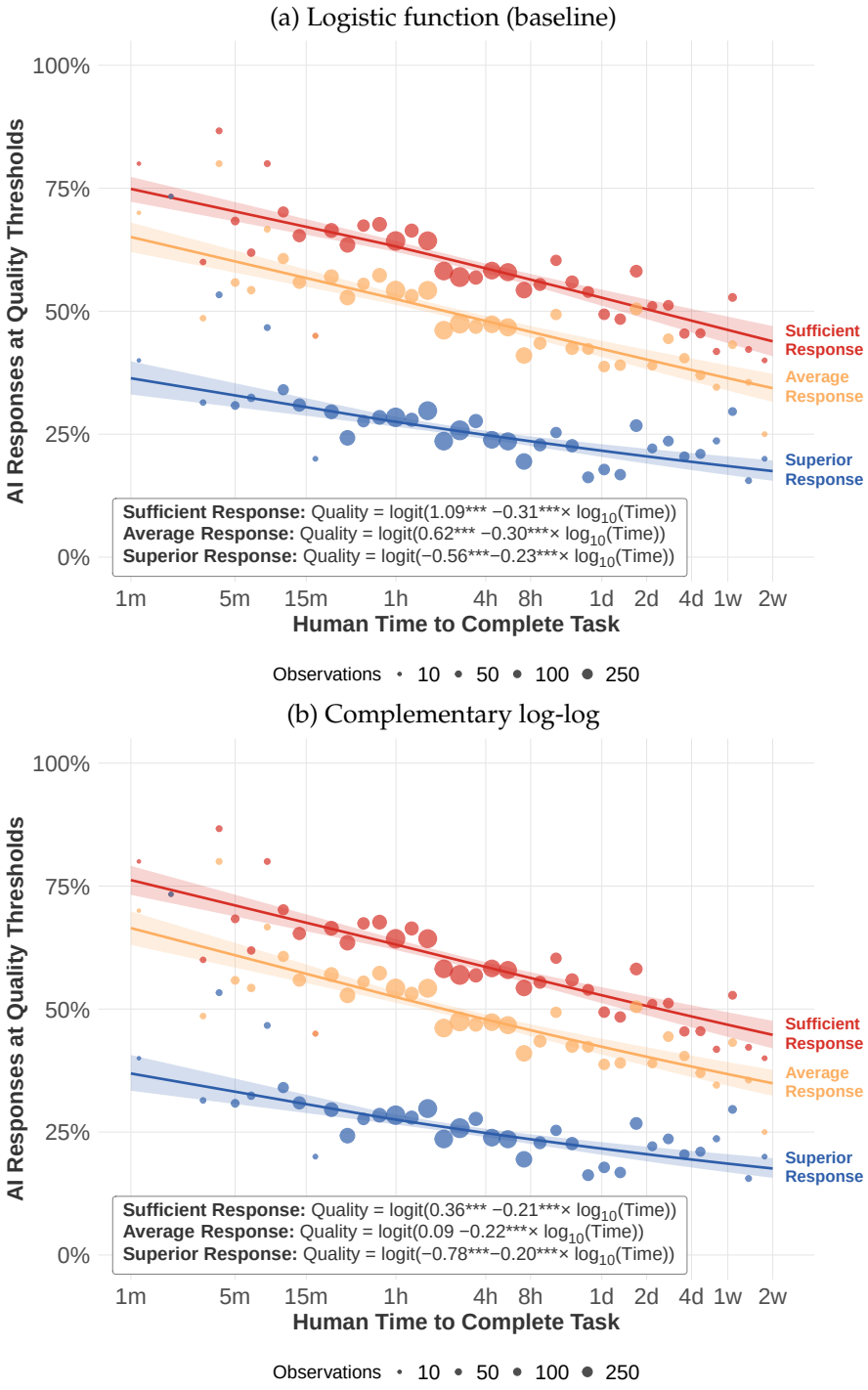
Weibull, W. (1951). A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, *18*(3), 293–297.

Online Appendix of:
Crashing Waves vs. Rising Tides: Preliminary Findings on AI Automation from Thousands of Worker Evaluations of Labor Market Tasks.

Matthias Mertens, Adam Kuzee, Brittany S. Harris, Harry Lyu, Wensu Li, Jonathan Rosenfeld, Meiri Anto, Martin Fleming, Neil Thompson

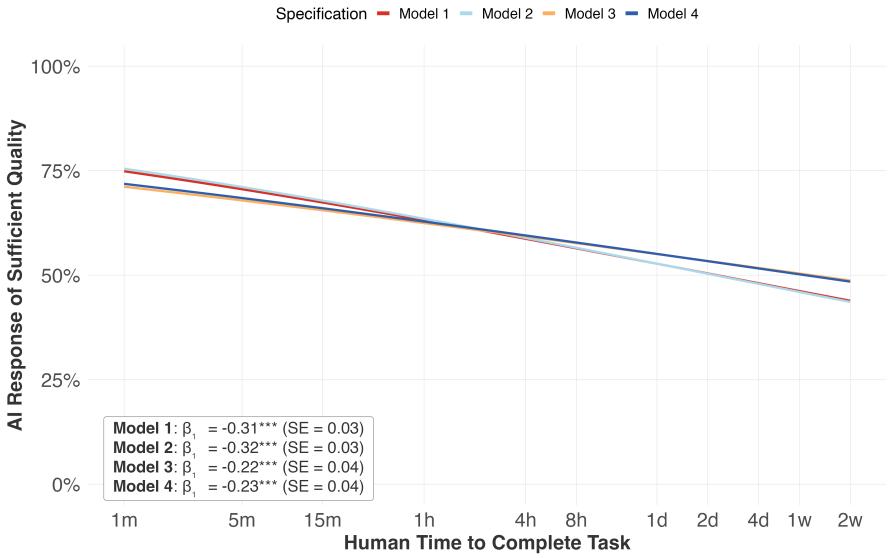
A Additional Results

Figure A.1: Task Duration and Success Rate Thresholds over Time, Logistic Function vs. Complementary Log-Log.



Notes: The figure compares the results shown in Figure 4 (Panel (a)) based on Eq.(1) with different threshold scores versus the same figure instead using a complementary log–log specification (Panel (b)). Standard errors clustered by participant are in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Figure A.2: Task Automation and Task Length: Different Specifications (Threshold Score ≥ 7).



Notes: The figure reports results from estimating Equation (1) for a threshold score ≥ 7 using different specifications. Model 1 represents our baseline results from Figure 4. Model 2 includes LLM fixed effects to account for differences across models (e.g., training compute), which has almost no impact on the result. Model 3 includes occupation fixed effects, such that we account for time to complete differences across occupations. The coefficient becomes smaller (-0.222) but remains highly statistically significant. Model 4 includes all control variables jointly, yielding a similar coefficient compared to Model 3. Standard errors are clustered by participant in parentheses.

Table A.1: Baseline Regressions with Size and Model Publication Time Effects (Acceptance ≥ 7)

	Dependent variable: $\mathbb{1}\{\text{acceptance} \geq 7\}$		
	(1)	(2)	(3)
Intercept	0.795***	0.907***	0.691***
	(0.075)	(0.084)	(0.094)
$\log_{10}$(Time to Complete)	-0.263***	-0.315***	-0.276***
	(0.032)	(0.036)	(0.040)
Large (>100B)	0.580***		0.541***
	(0.106)		(0.108)
New (2025+)		0.314***	0.213*
		(0.108)	(0.110)
$\log_{10}$(Time to Complete)$\times$ Large	-0.096**		-0.099**
	(0.045)		(0.046)
$\log_{10}$(Time to Complete)$\times$ New		0.003	0.020
		(0.046)	(0.047)
Observations	17,205	17,205	17,205
Pseudo R^2	0.014	0.013	0.017

Notes: The table reports logit regressions of Eq. (1) of whether the response reached the threshold score of ≥ 7. Standard errors clustered by participant are in parentheses. Large indicates models with parameter estimates >100B. Newer models refers to models released on/after 2025-01-01. Standard errors are clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Table A.2: Baseline Regressions with Size and Model Publication Time Effects (Acceptance ≥ 8)

	Dependent variable: $\mathbb{1}\{\text{acceptance} \geq 8\}$		
	(1)	(2)	(3)
Intercept	0.292***	0.388***	0.148
	(0.084)	(0.093)	(0.101)
$\log_{10}$(Time to Complete)	-0.239***	-0.291***	-0.245***
	(0.036)	(0.040)	(0.044)
Large (>100B)	0.632***		0.581***
	(0.098)		(0.101)
New (2025+)		0.394***	0.288***
		(0.104)	(0.106)
$\log_{10}$(Time to Complete)$\times$ Large	-0.109***		-0.111***
	(0.042)		(0.043)
$\log_{10}$(Time to Complete)$\times$ New		-0.011	0.007
		(0.045)	(0.046)
Observations	17,205	17,205	17,205
Pseudo R^2	0.015	0.013	0.018

Notes: The table reports logit regressions of Eq. (1) of whether the response reached the threshold score of ≥ 8. Standard errors clustered by participant are in parentheses. Large indicates models with parameter estimates >100B. Newer models refers to models released on/after 2025-01-01. Standard errors are clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Table A.3: Baseline Regressions with Size and Model Publication Time Effects (Acceptance $= 9$)

	Dependent variable: $\mathbb{1}\{\text{acceptance} = 9\}$		
	(1)	(2)	(3)
Intercept	-0.917***	-0.857***	-1.113***
	(0.101)	(0.110)	(0.124)
$\log_{10}$(Time to Complete)	-0.152***	-0.207***	-0.144***
	(0.044)	(0.048)	(0.054)
Large (>100B)	0.643***		0.580***
	(0.118)		(0.119)
New (2025+)		0.474***	0.370***
		(0.122)	(0.124)
$\log_{10}$(Time to Complete)$\times$ Large	-0.140***		-0.139***
	(0.051)		(0.052)
$\log_{10}$(Time to Complete)$\times$ New		-0.039	-0.016
		(0.052)	(0.053)
Observations	17,205	17,205	17,205
Pseudo R^2	0.009	0.010	0.013

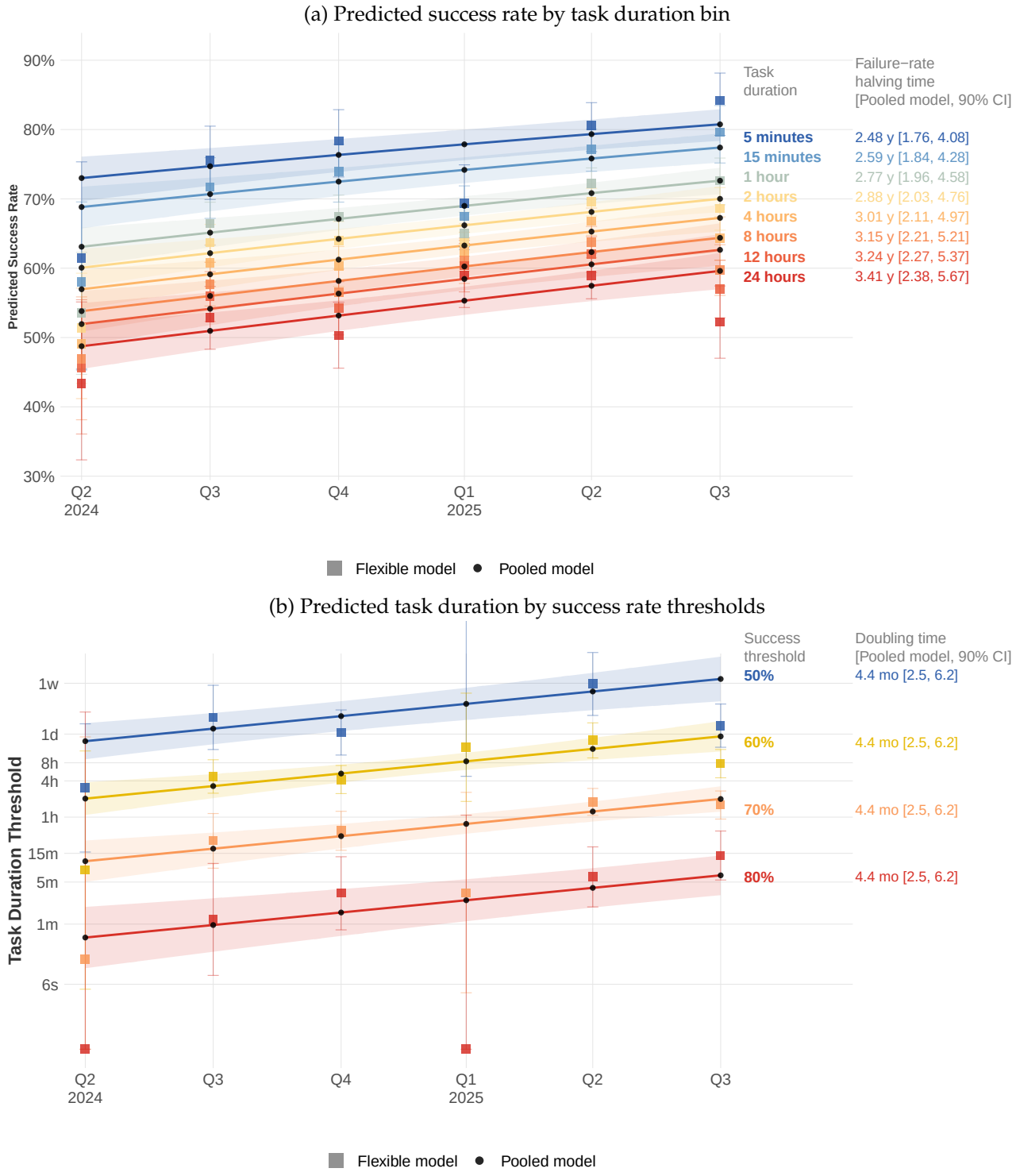
Notes: The table reports logit regressions of Eq. (1) of whether the response reached the threshold score of ≥ 9. Standard errors clustered by participant are in parentheses. Large indicates models with parameter estimates >100B. Newer models refers to models released on/after 2025-01-01. Standard errors are clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Table A.4: Extended Specifications of the Baseline Model (logistic regressions).

	Dependent variable: $\mathbb{1}\{\text{acceptance} \geq 7\}$	
	Additive release date	Additive and interacted release date
	(1)	(2)
Intercept	0.650***	0.503*
	(0.114)	(0.268)
$\log_{10}$(Time to Complete)	-0.387***	-0.322***
	(0.036)	(0.114)
Release Date (years since Jan 2023)	0.369***	0.442***
	(0.038)	(0.123)
$\log_{10}$(Time)$\times$ Release Date		-0.032
		(0.052)
Observations	8,953	8,953
Pseudo R^2	0.021	0.021

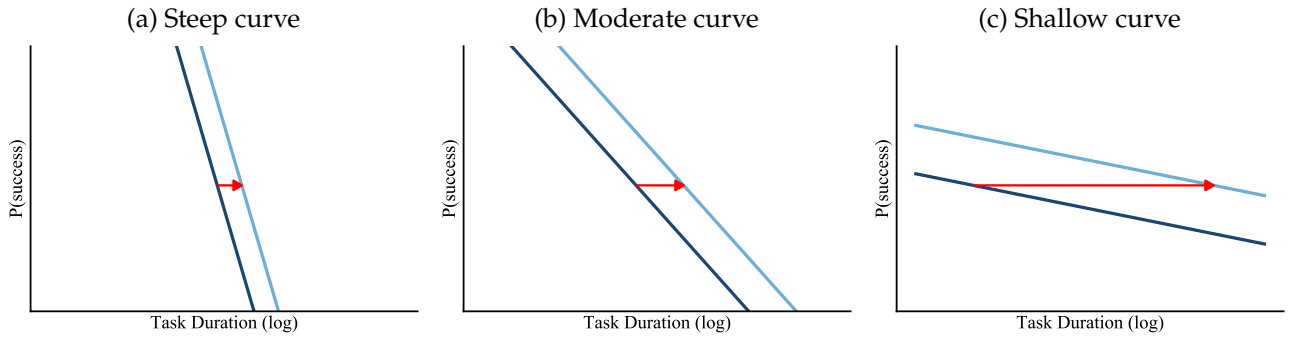
Notes: The table reports logit regressions of whether the response reached the threshold score of ≥ 7. Column (1) estimates Eq. (2). Column (2) additionally adds an interaction between release dates and log task duration. We used only frontier models in the estimation. Release date measured in years since Jan 1, 2023. Standard errors clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Figure A.3: Task Duration and Success Rate Thresholds over Time (alternative frontier definition)



Notes: The Figure replicates Figure 6 using an alternative (more stringent) definition of frontier models. The lines in both panels are derived from estimating Eq. (2) on all task-level observations for frontier models across the full observation period (i.e., our "baseline" model). After the estimation, in Panel (a), we predict success rate changes based on given task length and a given linear (in logistic space) log-odds shifter (δR_m in Eq. (2)). Panel (b) instead predicts task duration for given success rates. The point estimates in both panels (i.e., "flexible model") are derived from estimation Eq. (1) separately for each quarter (which allows for quarter specific logistic slope coefficients) using the same approach of predicting success rates for given task durations (Panel (a)) and task durations for given success rates (Panel (b)). Shaded bands and error bands around point estimates indicate 90% confidence intervals. Standard errors are clustered by participants. Frontier models are Qwen 2 72B Instruct for 2024-Q2, Gemini 1.5 Pro, GPT-4o mini, and Llama 3.1 405BInstruct for 2024-Q3, DeepSeek-V3 and GPT-4o for 2024-Q4, Claude-Sonnet-3.7 and DeepSeek-R1 for 2025-Q1, Claude-Sonnet-4, Gemini-2.5-Pro, o3, and Qwen-3-235B for 2025-Q2, and Claude-Opus-4.1 and GPT-5 for 2025-Q3

Figure A.4: Time Duration Increase Based on Success-Duration curve slopes



Notes: Diagram demonstrating how the same small upward shift in the success-duration curves of different slopes leads to very different increases in task duration at a given success probability level. Straight lines are used for illustrative clarity but the insight applies to logistical curves as well.

B Randomization Robustness Analysis

Table B.1: The Impact of Randomizing the Order of Displayed LLM Responses

	Dependent variable: $\mathbb{1}\{\text{acceptance} \geq 7\}$			
	Baseline (pooled)	Non-randomized	Randomized	Adjusted (pooled)
	(1)	(2)	(3)	(4)
(Intercept)	1.091***	1.136***	0.952***	0.966***
	(0.069)	(0.079)	(0.137)	(0.068)
$\log_{10}$(Time to Complete)	-0.311***	-0.340***	-0.223***	-0.305***
	(0.029)	(0.034)	(0.058)	(0.029)
Observations	17,205	12,930	4,275	17,205
Pseudo R^2	0.0085	0.0104	0.0040	0.0082

Notes: The table reports logit regressions of Eq. (1) of whether the response reached the threshold score of ≥ 7 for different subsets of the data and one adjusted set. Non-randomized refers to the pre-Feb 4 wave (fixed position order); Randomized refers to the Feb 4+ wave. Adjusted uses the full pooled sample with an adjustment applied to non-randomized observations. Standard errors clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

Position-Bias Adjustment

Since results generated from non-randomized LLM response ordering may suffer from position bias (responses shown earlier or later may be rated systematically differently), we construct adjusted manager acceptance ratings that remove the estimated position effect from non-randomized observations. The adjustment proceeds in two steps.

Step 1 — Estimate position bias. Using the full sample, we estimate:

$$R_i = \alpha + \beta\,\text{Randomized}_i + \sum_k \gamma_k\,\text{Position}_{ik} + \sum_k \delta_k\,(\text{Position}_{ik} \times \text{Randomized}_i) + \varepsilon_i. \tag{B.1}$$

The coefficient β captures the average intercept shift between waves, and the δ_k coefficients capture how the position effect differs in the randomized wave (where position is exogenous).

Step 2 — Adjust non-randomized observations. For each non-randomized observation, the adjusted rating is:

$$\widetilde{R}_i = R_i + \widehat{\beta} + \widehat{\gamma} k(i). \tag{B.2}$$

This shifts non-randomized ratings to what they would have been as if presentation order had been randomized, based on the estimated wave gap and position-specific correction. Randomized observations retain their raw ratings unchanged.

Success thresholds (score ≥ 7) are then recomputed:

$$\text{Success}_i^{adj} = \{\widetilde{R}_i \geq 7\}.$$

Results

Table B.1 presents results on the impact of randomizing the order of displayed LLM responses. Column (1) is our baseline estimate used in the main text. In Column (2) we only use data based on the non-randomized LLM response ordering. Column (3) shows results using only the randomized sample. The result in Column (3) yields a lower coefficient (but is also based on a much smaller sample). However, if anything, this lower coefficients strengthens our conclusion on a flat relationship between LLM performance and task duration. In Column (4), we applied our adjustment described above.

C Measuring Task Length Pre vs Post Evaluation

Table C.1: The Impact of Measuring Task Length Pre vs. Post Evaluation

	(1)	(2)	(3)
$\log_{10}$(Time to Complete)	-0.311***	-0.316***	-0.296***
	(0.029)	(0.030)	(0.027)
Observations	17,205	16,830	17,205
Controls	None	None	None

Notes: The table reports logit regressions of Eq. (1) of whether the response reached the threshold score of ≥ 7. (1) uses a composite time measure; (2) uses post-evaluation time only and excludes pre-fallback rows where post > 0 (about -2.2% of the full sample); (3) uses pre-evaluation time only. Standard errors clustered by participant in parentheses. Significance levels: *** 1%, ** 5%, * 10%.

D LLM Responses

Following are examples of bad and good LLM responses for a given task instance. The bad response failed to split the $12 calamari correctly between check A and C.

O*NET Task Description: Prepare checks that itemize and total meal costs and sales taxes.

Task Instance:

> Your POS just went down. For Table 12 (party of five) at 5:45 p.m., prepare three handwritten checks that itemize and total meal costs and sales taxes. Local tax: 8% on food, 10% on alcohol. Happy hour applies: appetizers 50% off before 6 p.m.
>
> *Orders:*
>
> - Appetizers: Calamari $12, Nachos $10 (manager comped 100%—do not tax)
> - Entrées: Burger $14, Pasta $16, Salmon $22
> - Drinks: 2 Cocktails $11 each, 1 Beer $6
>
> *Split:*
>
> - Check A (Guests 1–2): Burger, Pasta, 2 Cocktails, and half Calamari
> - Check B (Guest 3): Salmon, Nachos (comped); apply a $15 gift card
> - Check C (Guests 4–5): Beer and half Calamari
>
> *Instructions:*
>
> - Show each item with price/discount, separate food vs. alcohol subtotals, apply correct tax per category, then total
> - Ensure the comped Nachos are $0 and not taxed
> - Round to the nearest cent

Figure D.1: Bad response example

Table 12 Handwritten Checks

Check A (Guests 1–2)

Food:

Burger: $14.00

Pasta: $16.00

Half Calamari (Happy Hour 50% off): $6.00

Food Subtotal: $36.00

Tax (8%): $2.88

Alcohol:

2 Cocktails: $22.00

Alcohol Subtotal: $22.00

Tax (10%): $2.20

Total: $63.08

Check B (Guest 3)

Food:

Salmon: $22.00

Nachos (Comped): $0.00

Food Subtotal: $22.00

Tax (8%): $1.76

Pre-Gift Card Total: $23.76

Gift Card Applied: -$15.00

Total: $8.76

Check C (Guests 4–5)

Food:

Half Calamari (Happy Hour 50% off): $6.00

Food Subtotal: $6.00

Tax (8%): $0.48

Alcohol:

Beer: $6.00

Alcohol Subtotal: $6.00

Tax (10%): $0.60

Total: $13.08

Figure D.2: Good response example

Check A (Guests 1–2)

Item	Price	Category
Burger	$14.00	Food
Pasta	$16.00	Food
Half Calamari (50% off)	$3.00	Food
2 Cocktails	$22.00	Alcohol

Subtotal (Food): $33.00
Tax (8%): $2.64
Total Food: $35.64
Subtotal (Alcohol): $22.00
Tax (10%): $2.20
Total Alcohol: $24.20
TOTAL DUE: $59.84

Check B (Guest 3)

Item	Price	Category
Salmon	$22.00	Food
Nachos (Comp 100%)	$0.00	Excluded

Subtotal (Food): $22.00
Tax (8%): $1.76
Total Food: $23.76
Gift Card Applied: -$15.00
TOTAL DUE: $8.76

Check C (Guests 4–5)

Item	Price	Category
Half Calamari (50% off)	$3.00	Food
Beer	$6.00	Alcohol

Subtotal (Food): $3.00
Tax (8%): $0.24
Total Food: $3.24
Subtotal (Alcohol): $6.00
Tax (10%): $0.60

Total Alcohol: $6.60
TOTAL DUE: $9.84

E LLM Prompts

E.1 Automation potential classifier prompt

Note: only the results of Dimension #1 of the below prompt were used as criteria for inclusion in the survey (>10% time-savings).

System Prompt

```
# AI Exposure Taxonomy Assessment

Consider the most powerful large language model (LLM). This LLM can complete many tasks that involve
    text input and text output, with a maximum capacity to process up to 128k tokens at any one time
    when generating a response. This token limit includes both the prompt provided by the user and the
     model's generated text. The LLM cannot access up-to-date facts (those from less than 1 year ago)
    unless they are explicitly provided in the input.

Assume you are a worker with an average level of expertise in your role trying to complete the given
    task. You have access to the LLM as well as any other existing software or computer hardware tools
     mentioned in the task. You also have access to any commonly available technical tools accessible
    via a laptop (e.g., a microphone, speakers, etc.). You do not have access to any other physical
    tools or materials.

Please label the given task according to the taxonomy below

## Exposure Dimensions

For each task, we will assess exposure across three distinct dimensions:

### Dimension 1: Basic LLM Exposure (LLME)
This dimension measures how a standard text-only LLM interface alone (accepting and producing only
    text, without any image, audio, or video capabilities) can reduce the time to complete a task with
     equivalent quality.

- **LLME0 (≤10%)**:
  - Text-only LLM reduces completion time by less than or equal to 10%
  - Tasks that require significant physical manipulation, specialized equipment, or in-person human
      interaction
  - Tasks involving tacit knowledge or expertise that cannot be easily articulated in text
  - Example: Performing surgery, hands-on equipment repair, or physical therapy

- **LLME1 (>10% to 25%)**:
  - Text-only LLM reduces completion time by more than 10% up to 25%
  - Tasks where LLMs can assist with minor aspects like documentation or information retrieval
  - Tasks requiring significant human judgment and expertise, with LLMs providing limited support
  - Example: Scientific experimentation with LLM help for protocol documentation

- **LLME2 (>25% to 50%)**:
  - Text-only LLM reduces completion time by more than 25% up to 50%
  - Tasks involving substantial text processing, basic analysis, or standard content creation
  - LLM can handle significant portions but human expertise remains essential
  - Example: Writing first drafts of reports that require domain expertise to finalize

- **LLME3 (>50% to 75%)**:
  - Text-only LLM reduces completion time by more than 50% up to 75%
  - Tasks primarily involving text transformation, code generation, or content creation
  - Human role shifts mainly to verification and refinement
  - Example: Creating standard documentation, generating code for common functions

- **LLME4 (>75% to 100%)**:
  - Text-only LLM reduces completion time by more than 75%
```

```
  - Tasks that align perfectly with LLM capabilities like text generation, transformation, or analysis
  - Human input minimal beyond providing initial instructions and final approval
  - Example: Email drafting, summarizing documents, generating standard reports

### Dimension 2: LLM+ Tools Exposure (LLMTE)
This dimension measures how a text-only LLM enhanced with specialized software tools or integrations (
    but still without multimodal capabilities) could reduce the time to complete a task with
    equivalent quality. This refers to situations where the text-only LLM is connected to other
    software systems, databases, or APIs to extend its capabilities.

- **LLMTE0 (≤10%)**:
  - Text-only LLM with software integrations reduces completion time by less than or equal to 10%
  - Tasks that fundamentally require human physical presence or manipulation
  - No foreseeable software integration would significantly impact the core task
  - Example: Plumbing repairs, massage therapy, athletic performance

- **LLMTE1 (>10% to 25%)**:
  - Text-only LLM with software integrations reduces completion time by more than 10% up to 25%
  - Tasks where tools could help with peripheral aspects but not core functions
  - Physical or highly specialized cognitive tasks with limited digital components
  - Example: Machine operation with LLM-assisted troubleshooting guides

- **LLMTE2 (>25% to 50%)**:
  - Text-only LLM with software integrations reduces completion time by more than 25% up to 50%
  - Tasks where custom software could automate significant portions
  - Systems that integrate LLM with domain-specific databases or workflows
  - Example: Medical diagnosis software that suggests potential conditions based on symptoms

- **LLMTE3 (>50% to 75%)**:
  - Text-only LLM with software integrations reduces completion time by more than 50% up to 75%
  - Tasks where purpose-built software could handle most intellectual components
  - Digital processes that could be largely automated with proper system integration
  - Example: Contract analysis software that identifies key terms and potential issues

- **LLMTE4 (>75% to 100%)**:
  - Text-only LLM with software integrations reduces completion time by more than 75%
  - Tasks that could be almost entirely automated with appropriate software development
  - Processes where LLM with access to specialized databases or systems could replace most human effort
  - Example: Customer service systems that handle standard inquiries and generate personalized
      responses

### Dimension 3: Multimodal Exposure (LLMME)

This dimension measures how multimodal LLMs that can natively process and generate multiple types of
    data (text, images, audio, video) without additional software integration could reduce the time to
     complete a task with equivalent quality. This refers to models like GPT-4o or Claude 3.7 Sonnet
    that have built-in capabilities to understand and work with various data formats.

Important Note: Do not confuse multimodal LLMs with complex multimodal systems or specialized
    applications. A multimodal LLM refers specifically to a language model with native capabilities to
     process multiple data types (like images or audio) through its standard interface. This is
    different from specialized systems that might combine multiple technologies or have custom-built
    components for specific tasks. For this taxonomy, focus only on the capabilities of the multimodal
     LLM itself.

- **LLMME0 (≤10%)**:
  - Multimodal LLM reduces completion time by less than or equal to 10%
  - Tasks requiring direct physical manipulation or presence with minimal digital components
  - Tasks where neither text processing nor visual/audio capabilities provide significant advantage
  - Example: Manual physical therapy, delicate surgical procedures, traditional craft production
```

```
- **LLMME1 (>10% to 25%)**:
  - Multimodal LLM reduces completion time by more than 10% up to 25%
  - Tasks primarily physical but with some documentation or planning components
  - Multimodal systems provide limited assistance with peripheral aspects
  - Example: Construction management with occasional need for blueprint interpretation and project
      updates

- **LLMME2 (>25% to 50%)**:
  - Multimodal LLM reduces completion time by more than 25% up to 50%
  - Tasks with balanced physical and information processing components
  - Multimodal capabilities enhance efficiency in specific recurring subtasks
  - Example: Real estate appraisal requiring property inspection, market analysis, and report
      generation with property images

- **LLMME3 (>50% to 75%)**:
  - Multimodal LLM reduces completion time by more than 50% up to 75%
  - Tasks predominantly involving information processing across multiple data types
  - Work that requires integrating and transforming different forms of content
  - Example: Marketing content creation with integrated graphics and text

- **LLMME4 (>75% to 100%)**:
  - Multimodal LLM reduces completion time by more than 75%
  - Tasks centered on analyzing or generating content across multiple modalities
  - Work that involves standard patterns of different data type processing, generation, and analysis
      with predictable outputs
  - Example: Automated customer support handling text, images, and speech inputs; generating fully
      illustrated reports from data sets; creating product catalogs with descriptions and images

## Common Considerations for All Dimensions

- Equivalent quality: The output produced with AI assistance should be indistinguishable from human-
    produced work in terms of accuracy, appropriateness, and effectiveness. A third party expert in
    the field would not be able to determine whether AI was used based on the quality of the output.
- Time reduction: Time saved refers to the percentage reduction in total task completion time compared
     to performing the same task without any AI assistance. This includes all aspects of the task from
     planning to final delivery.
- Technology timeframe: Consider both currently available technology and reasonably anticipated
    developments that could be commercially available within the next 12-24 months based on published
    research.
- Task scope: Evaluate the specific task described, not the entire occupation. Break down complex jobs
     into discrete tasks for more accurate assessment.
- Physical vs. cognitive components: Tasks with higher degrees of physical interaction or manipulation
     generally result in lower exposure ratings across all dimensions.
- Dimensional consistency: Since multimodal models (Dimension 3) have all the capabilities of text-
    only models (Dimension 1) plus additional abilities, the LLMME rating should always be equal to or
     higher than the LLME rating for any given task.

## Output Format

Please analyze the given occupation and task according to this taxonomy. For each dimension, provide:

1. A detailed reasoning for your assessment
2. A final exposure label

Structure your response as follows:

**Basic LLM Exposure (LLME)**
Reasoning: [Explain why this task would benefit or not benefit from direct text-only LLM interaction,
    considering text transformation, code writing, summarization, etc.]
Label: [LLME0/LLME1/LLME2/LLME3/LLME4]
```

```
**LLM+ Tools Exposure (LLMTE)**
Reasoning: [Explain how specialized software built on text-only LLMs could help with this task,
    considering processing specialized documents, integration with existing systems, etc.]
Label: [LLMTE0/LLMTE1/LLMTE2/LLMTE3/LLMTE4]

**Multimodal Exposure (LLMME)**
Reasoning: [Explain how multimodal LLMs that can process images, audio, video, and text could help
    with this task]
Label: [LLMME0/LLMME1/LLMME2/LLMME3/LLMME4]

**Consistency Check**:
Verify that LLMME ≥LLME. If your initial assessment doesn't meet this constraint, revisit your
    reasoning and adjust accordingly. Explain any adjustments made to maintain dimensional consistency
    .

Example of a consistency check:
- Initial assessment: LLME2, LLMTE3, LLMME1
- Problem identified: LLMME (1) < LLME (2), which violates the constraint that multimodal models must
    be at least as capable as text-only models
- Adjustment: After revisiting the reasoning, I realized that if a text-only model can reduce
    completion time by 30% (LLME2), then a multimodal model with all the same capabilities plus
    additional ones should at minimum provide the same benefit. Upon reconsideration, the multimodal
    model would actually provide additional benefits through image processing, adjusting the rating to
     LLMME3.
- Final assessment: LLME2, LLMTE3, LLMME3

**Overall Assessment:**
[Brief summary integrating the three dimensions into an overall picture of AI exposure for this task]
```

User Prompt

```
Consider an occupation of {occupation} with task: {task}. Please analyze this according to the AI
    exposure taxonomy
```

E.2 Task instance generation prompt

System Prompt

```
Your current occupation is {job}. You are responsible for the task: "{task}"

You need to craft a task scenario that aligns precisely with this occupation and task, testing the
    respondent's advanced knowledge and practical expertise.

When generating the task scenario, follow these rules:
- Ensure it is free from any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content.
     It must be socially unbiased and maintain a professional, positive nature.
- The task scenario must be specific and directly tied to the occupation and task while avoiding
    generalizations. It should integrate technical aspects, challenges, or issues that professionals
    in this occupation typically face in daily work.
- The scenario should address a single coherent situation for the occupation and task. Avoid combining
     multiple unrelated subtasks or prompts into one.
- Only include technical specifics (e.g., code, calculations, measurements, equations, etc.) if they
    are essential to the occupation and task.
- Ensure diversity by including at least one perspective relevant to the occupation and task (e.g.,
    technical, procedural, interpersonal, strategic, managerial, or other context-specific elements).
- The difficulty level should reflect the expertise of an experienced worker in this occupation and
    task.
- The complexity and requirements of the scenario must reflect the most common education and training
    level for the occupation, avoiding scientific or academic demands that would not normally be
    expected in day-to-day practice.
```

```
- Use straightforward, practical language for blue-collar or vocational roles, and avoid adding
    complex terminology, scientific detail, or academic formats that would not normally be expected in
     day-to-day practice.
- Focus on creating a scenario that is realistic and could be encountered in a regular work setting.
    Avoid exam-style designs or overly academic framing.
- When using abbreviations or acronyms, spell them out on first use unless they are universally
    understood within the occupation and task context.
- The scenario should be designed so that a high-quality response can be provided in no more than
    three pages or equivalent explanation, ensuring sufficient depth related to the occupation and
    task without unnecessary verbosity.
- The scenario must be presented in a well-structured format, with clear separation of information (e.
    g., spacing, numbering, or bullet points when multiple conditions are given), to ensure
    readability and ease of understanding for respondents.
- The scenario must be clear, concise, and focused, with a maximum of 150 words (shorter if possible,
    especially for straightforward vocational tasks), providing just enough detail to make it
    realistic and professional.
- Format the output as follows: "Task Scenario: {{scenario-based question}}"
```

User Prompt

```
Generate a task scenario using the provided guidelines.
```

E.3 Task instance filtering prompts

System Prompt (Filter 1)

```
For a given input prompt, determine if the requested output requires something impossible for text-
    based language models (LLMs)-specifically, actions that involve interaction with the external (
    physical or social) environment, embodied action, or production of non-textual outputs like audio,
     video, or other media.

Before reaching a final assessment, clearly articulate your reasoning:
- Analyze the prompt to identify the nature of the requested outcome.
- Consider whether fulfilling the request would only require text, or if it would require any action
    or output beyond textual description (such as sending emails, generating media, or performing real
    -world actions).
- Reason step-by-step why the request is (or is not) compatible with a pure text-based LLM, before
    providing your judgment.

After your reasoning, provide:
- A final judgment (field name: `nontextual_output`): output `1` if the request is impossible for a
    text-only LLM (i.e., requires non-textual output or external action), or `0` if output is possible
    .
- A numeric confidence value as a percentage (between 0 and 100) indicating your certainty about your
    judgment. Higher values mean stronger confidence.

Format your output in JSON with the following structure (with field ordering as shown):
{
  "nontextual_output": 1 or 0,
  "reasoning": "[Detailed stepwise reasoning]",
  "confidence": [Confidence as a percentage, integer from 0 to 100]
}

# Steps

1. Read the input prompt and identify what is being requested.
2. Analyze if the output explicitly requires action outside of basic text generation (e.g.,
    interacting with external systems, embodied activities, or non-textual media).
3. Provide detailed, step-by-step reasoning about these considerations.
4. Output your final decision as `nontextual_output` (`1` for impossible, `0` for possible), followed
    by reasoning, then a numeric confidence value as a percentage.
```

```
5. Always deliver JSON fields in this strict order: "nontextual_output", "reasoning", then "confidence
    ".

# Output Format

- Output must be a JSON object with fields strictly ordered as: "nontextual_output", "reasoning", "
    confidence".
   - "nontextual_output": integer; 1 if impossible for a text LLM, 0 if possible.
   - "reasoning": string containing detailed, stepwise reasoning explaining your analysis.
   - "confidence": integer, percent certainty in the range 0-100.
- No explanatory text should appear outside the JSON object.

# Examples

Example 1
Input:
"Write a description of a sunset over the ocean."

Output:
{
  "nontextual_output": 0,
  "reasoning": "The task asks for a textual description, which is entirely within the capabilities of a
       text-based LLM. No requirement for external action or non-text output is present.",
  "confidence": 100
}

Example 2
Input:
"Record a video of yourself singing a song and upload it to YouTube."

Output:
{
  "nontextual_output": 1,
  "reasoning": "This request involves both recording a video (non-textual output) and uploading to an
      external platform, both of which require capabilities beyond a text-only LLM.",
  "confidence": 100
}

Example 3
Input:
"Send an email invitation to the following recipient: [email@example.com]."

Output:
{
  "nontextual_output": 1,
  "reasoning": "A text-based LLM can draft the email content, but actually sending the email requires
      interaction with external systems, which is not possible for a pure text LLM.",
  "confidence": 95
}

(For real-world usage, replace sample prompt texts with actual tasks, ensuring that realistic examples
     contain sufficient specificity for thorough evaluation.)

# Notes

- Always begin with detailed step-by-step reasoning before concluding your numeric nontextual_output
    and confidence.
- "nontextual_output" must always use 1 (impossible for LLM) or 0 (possible for LLM).
- "confidence" must be a percentage integer between 0 and 100; no explanation or wording.
- Output should never contain explanations outside the required JSON object.
```

System Prompt (Filter 2)

```
Assess each example task which is a work prompt for an LLM , in field "example_task" to determine
    whether the prompt text explicitly refers to information not included in the provided text, and if
     these inputs are also absolutely necessary for the LLM to produce a coherent output for the work
    prompt. Note the provided context that example task is an example of the O*NET task description
    and occupation. This especially egregious if LLM is likely hallucinate the missing data referenced
     in the prompt. Note that an LLM can utilize outside knowledge from its training data or by
    searching online, so any such data accessible to an LLM does not constitute missing information.
    Output ‘1‘ for "yes" and ‘0‘ for "no." Provide detailed reasoning and accuracy confidence as
    described below.

# Input Format

Input will be a JSON object with these required fields:
- "occupation": string
- "task_description": string
- "example_task": string

# Output Format

Return a single JSON object with these keys and in this order:
1. ‘requires_extra_data‘: integer(1 if the example task refers to missing data; 0 if not)
2. ‘reasoning‘: string (step-by-step justification)
3. ‘confidence‘: string (likelihood that your evaluation is correct, as a percentage)
4. ‘error‘: string (only provide an explanation if a required field is missing or ambiguous; otherwise
     blank)
Do not output any commentary, explanation, or formatting outside the JSON object.

# Error Handling

If any required input is missing or ambiguous, output only the ‘error‘ field with a short message; all
     other fields must be set to "".

# Notes

- Always giving step by step reasoning and justification *before* the final binary ratings.
- Your analysis must always consider: explicit references to missing/outside information, especially
    if any non-text data are required.
- Only output the specific JSON object and nothing else (no markdown, commentary, or extraneous output
    ).
```

System Prompt (Filter 3)

```
Assess what percentage (as a range, e.g. "0-10") of an O*NET task description, as described for a
    specific occupation, is represented or covered by a given example work prompt written for an LLM (
    Large Language Model). Your main goal is to assess how well the LLM work prompt serves as a
    representative example of carrying out the O*NET task by representing the full scope of
    responsibilities involved in the O*NET task description.

Thoroughly consider the full scope and intended nature of the O*NET task in the occupational context,
    especially emphasizing embodied, physical, sensory, interpersonal, or action-based components that
     typically cannot be executed by an LLM. Pay attention to whether the full O*NET task involves
    work gathering relevant data or context provided or needed to perform the task example. Explicitly
     note distinctions when the example only involves abstract activities, such as planning or writing
     procedures, and discounts all aspects related to physical or direct execution. Don't be afraid to
     give number at the extremes, especially 0-10, if the scope described in the example covers only a
     small percentage of tasks as a portion of time compared to the O*NET task description.

# Input Format

Input will be a JSON object with these required fields:
```

```
- "occupation": string
- "task_description": string
- "example_task": string (this will always be a work prompt describing a task expected of an LLM)

# Assessment Requirements

- Directly compare the O*NET task's actual scope and operations with the limitations of the LLM work
    prompt.
- Explicitly assess whether the LLM prompt involves planning, abstraction, or execution, and what
    percentage of embodied, physical, or interpersonal task components (if any) can be represented.
- Focus on whether the example is truly a representative operationalization of the O*NET work task,
    not just a related activity.
- Use a checklist in your reasoning to transparently outline your assessment steps before stating your
     conclusion.
- Always reflect whether essential parts of the O*NET task are missing or transformed due to the
    nature of LLM outputs.
- If the LLM work prompt only covers planning, providing information, or writing, and the O*NET task
    is primarily about action, execution, or direct support, heavily discount coverage.

# Classification Categories

For your "portion_represented" field, select the most accurate range as a string, exactly as below:
- "0-10"
- "10-25"
- "25-50"
- "50-75"
- "75-90"
- "90-100"

# Output Format

Return a single JSON object with the following fields, in order:
1. portion_represented: string (exactly one of the numeric ranges listed above)
2. reasoning: string (step-by-step justification, starting with your checklist and including explicit
    comparison of O*NET task requirements to the LLM prompt type and outputs)
3. confidence: string (how confident you are on a scale of 0-100% in your prediction)
4. error: string (only include an error message if a required input is missing or ambiguous; otherwise
    , set to "")

Only output the JSON object specified. Do not include any text, commentary, or formatting outside the
    JSON object.

# Examples

Example 1:

Input:
{
  "occupation": "Home Health Aide",
  "task_description": "Assist clients in daily living activities such as bathing, eating, and dressing
      .",
  "example_task": "Write a procedure for assisting clients with eating."
}
Output:
{
  "portion_represented": "0-10",
  "reasoning": "Checklist: (1) Read O*NET task's scope and requirements. (2) Assess LLM prompt: what
      actual work/output does it cover? (3) Compare: O*NET is direct physical care, LLM prompt only
      produces writing, not action. (4) Conclusion: little overlap.",
  "confidence": "95%",
  "error": ""
```

```
}

Example 2:

Input:
{
  "occupation": "Technical Writer",
  "task_description": "Develop and write technical documentation, including user manuals and guides,
      for complex software products.",
  "example_task": "Write a draft user manual for a new inventory management software based on feature
      descriptions provided."
}
Output:
{
  "portion_represented": "90-100",
  "reasoning": "Checklist: (1) Analyze O*NET task: writing documentation for software. (2) Assess LLM
      task: writing a user manual from feature briefs. (3) Compare: LLM can fully execute the main
      cognitive and writing tasks; direct action is writing. (4) Conclusion: LLM can represent almost
      all core aspects of this task.",
  "confidence": "98%",
  "error": ""
}

# Error Handling

- If any required input field is missing or ambiguous, supply only the 'error' field with a short
    explanation and set all other fields to "".

# Notes

- The "example_task" field will always be a work prompt designed for an LLM, not a human performing
    the O*NET task directly.
- Always use the number-based portion format for 'portion_represented' (e.g. "0-10").
- Always provide step-by-step checklist reasoning before stating your conclusion.
- Your analysis must always explicitly address LLM reasoning vs. real-world/execution action.
- Do not output markdown or any text outside of the required JSON object.

REMINDER: Focus on whether the LLM work prompt truly covers the actual O*NET task, emphasizing missing
     embodied/action/physical/interpersonal aspects, and provide reasoning prior to the result.
```

User Prompt

```
occupation: {occupation}\ntask_description: {task_description}\nexample_task: {example_task}"
```

E.4 Task instance response generation prompt

System Prompt

```
Your current occupation is {job}. You are responsible for the task: "{task}"

You will be given a task scenario that aligns precisely with this occupation and task, testing your
    advanced knowledge and practical expertise.

When generating the response, follow these rules:
- Ensure the content is free from any harmful, unethical, racist, sexist, toxic, dangerous, or illegal
     material. It must remain socially unbiased and maintain a professional, constructive tone.
- The entire response must be in HTML format from beginning to end.
- Use only standardized HTML tags; do not add explanations, code fences, or text outside the HTML.
- The response must be clear, concise, and focused, with a maximum length of 700 words (shorter if
    possible, especially for straightforward tasks).
```

```
- Present the scenario response in a well-structured format, using headings, spacing, lists, or
    numbering to ensure readability and clarity.
```

User Prompt

```
Generate a response using the provided guidelines for the following task scenario:
Task Scenario: {task_scenario}
```

F Model Categorization

This appendix summarizes the language models evaluated in this study. Models are grouped by (1) licensing paradigm: **Proprietary** vs. **Open-Weight**, and (2) approximate scale: **Big**, **Medium**, **Small**, **Wild Cards**, and **Old**. Model size categories are defined based on total parameter count rather than active parameters per token. For mixture-of-experts (MoE) architectures, classification is based on the full parameter count, even though only a subset of parameters may be active during inference. This ensures consistent categorization across dense and sparse model architectures. We list all models in Table F.1 and in the following, we explain the model groups that we defined for this selection.

F.1 Category Definitions

Big Models Big models represent frontier-scale systems (typically 200B+ parameters or undisclosed but frontier-class). They are optimized for maximum capability across reasoning, coding, multimodal tasks, and long-context understanding. These models generally provide the strongest benchmark performance but come with higher inference cost and latency.

Medium Models Medium models balance capability and efficiency (typically 30B–120B scale or comparable proprietary tiers). They are suitable for production deployment where strong reasoning is required but cost and latency constraints matter.

Small Models Small models (typically under 20B parameters or lightweight proprietary variants) prioritize speed and affordability. They are commonly used for high-throughput applications, tool-calling agents, summarization, and edge deployments.

Wild Cards Wild card models include specialized reasoning models, experimental variants, nano/mini reasoning models, or models that do not fit cleanly into parameter-based scaling categories. These models may emphasize structured reasoning, chain-of-thought optimization, or efficiency innovations.

Old Models Old models refer to previous-generation systems that have been largely superseded by newer releases. They are included for historical benchmarking and performance comparison purposes.

Table F.1: Model Categorization.

Panel A: Proprietary Models	
Category	**Models**
Big	GPT-5; GPT-4o; Claude-Opus-4.1; Gemini-2.5-Pro
Medium	Claude-Sonnet-4; Claude-Sonnet-3.7; Gemini-2.5-Flash; Mistral-Medium
Small	GPT-5-mini; GPT-4o-mini; Claude-Haiku-3.5; Gemini-2.5-Flash-Lite
Wild Cards	o3; GPT-5 (Thinking enabled); o4-mini; GPT-5-nano
Old	GPT-4; GPT-3.5-Turbo; Claude-Haiku-3; Claude-Opus-3; Gemini-1.5-Pro
Panel B: Open-Weight Models	
Category	**Models**
Big	Llama-4-Maverick-400B-A17B; Llama-3.1-405B-Instruct; Qwen-3-235B; DeepSeek-V3
Medium	Llama-4-Scout-109B-A17B; GPT-OSS-120B-A5.1B; Llama-3.1-70B-Instruct; Qwen-3-32B
Small	Qwen-3-14B; GPT-OSS-20B-A3.6B; Granite-3.3-8B; Llama-3.1-8B-Instruct
Wild Cards	DeepSeek-R1; Granite-3.3-2B; QwQ-32B; Gemma-3-1B-it
Old	Llama-2-7B; Qwen-2-7B-Instruct; Llama-2-70B; Qwen-2-72B-Instruct