

Estimating AI productivity gains from Claude conversations

Published:

November 5, 2025

Author:

Alex Tamkin and Peter McCrory

Acknowledgements:

Kevin Troy, Ruth Appel, Keir Bradwell, Sarah Heck, Jack Clark, Jared Mueller,
Jennifer Martinez, Sarah Pollack, Francesco Mosconi, Stuart Ritchie, Jared Powell,
Carl De Torres, Lara Karadogan, David Saunders, Ben Merkel

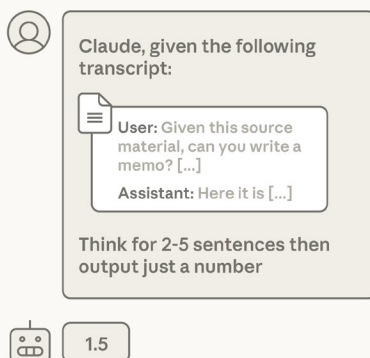
Overview

What do real conversations with Claude tell us about the effects of AI on labor productivity? Using our privacy-preserving [analysis method](#), we sample one hundred thousand real conversations from [Claude.ai](#), estimate how long the tasks in these conversations would take with and without AI assistance, and study the productivity implications across the broader economy. Based on Claude's estimates, these tasks would take on average about 90 minutes to complete without AI assistance, and Claude speeds up individual tasks by about 80%.

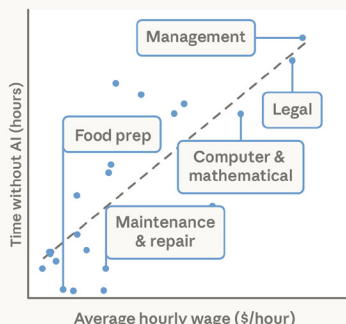
Extrapolating these estimates out suggests current-generation AI models could increase annual US labor productivity growth by 1.8% over the next decade – roughly twice the run rate in recent years. But this isn't a prediction of the future, since we don't take into account the rate of adoption or what might happen once AI models improve further.

Estimating Claude's productivity impacts

We use Claude to estimate how long it would take people to do real-world tasks with and without AI assistance



Tasks associated with higher-wage occupations (e.g. lawyers, programmers, or managers) tend to take more time



The est. time savings suggests an additional 1.8% annual labor productivity growth, assuming full adoption in 10 years

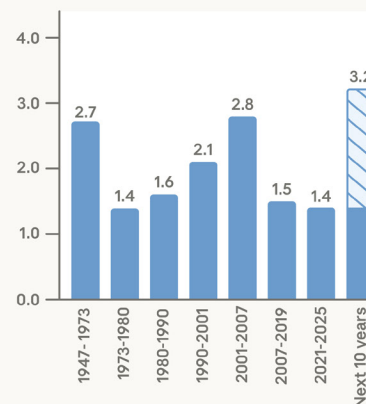


Figure 1: An overview of our method and some of our main results. See below for how we validate Claude's estimates, the assumptions we make, and limitations of our analysis.

Our analysis has limits. Most notably, we can't account for additional time humans spend on tasks outside of their conversations with Claude, including validating the quality or accuracy of Claude's work. But as AI models get better at time estimation, we think our methods in this research note could become increasingly useful for understanding how AI is shaping real work.

Here's a more detailed summary of our results:

- **Across one hundred thousand real world conversations, Claude estimates that AI reduces task completion time by 80%.** We use Claude to evaluate anonymized Claude.ai transcripts to estimate the productivity impact of AI. According to Claude's estimates, people typically use AI for complex tasks that would, on average, take people 1.4 hours to complete. By matching tasks to O*NET occupations and BLS wage data, we estimate these tasks would otherwise cost \$55 in human labor.
- **The estimated scope, cost, and time savings of tasks varies widely by occupation.** Based on Claude's estimates, people use Claude for legal and management tasks that would have taken nearly two hours, but for food preparation tasks that would have taken only 30 minutes. And we find that healthcare assistance tasks can be completed 90% more quickly, whereas hardware issues see time savings of 56%. This doesn't account for the time that humans might spend on these tasks beyond their conversation on [Claude.ai](https://claude.ai), however, so we think these estimates might overstate current productivity effects to at least some degree.
- **Extrapolating these results to the economy, current generation AI models could increase annual US labor productivity growth by 1.8% over the next decade. This would double the annual growth the US has seen since 2019, and places our estimate towards the upper end of [recent estimates](#).** Taking as given Claude's estimates of task-level efficiency gains, we use standard methods to calculate a 1.8% implied annual increase in US labor productivity over the next ten years. However, this estimate does not account for future improvements in AI models (or more sophisticated uses of current technology), which could significantly magnify AI's economic impact.
- **As AI accelerates some tasks, others may become bottlenecks:** We see large speedups for some tasks and much smaller ones in others, even within the same occupational groups. Where AI makes less of a

difference, these tasks might become bottlenecks, potentially acting as a constraint on growth.

- This gives us a new lens for understanding how AI’s economic impacts over time, which we will track going forward as part of our [Economic Index](#): Computing these estimates based on real-world Claude conversations gives us a new lens to understand AI productivity. This complements other approaches, like lab studies in narrow domains, or government statistics which provide more coarse-grained insights. We will track how these estimates change over time to get an evolving picture of these issues as capabilities and adoption continue to progress.

Introduction

As part of the [Anthropic Economic Index](#), we have documented how people use Claude across different tasks, industries, and places. We’ve captured the **breadth** of uses—how people use Claude for legal, scientific, and programming tasks—but not their **depth**. How substantial are the tasks for which people use Claude, and how much time does Claude save them?

The current version of the Economic Index can’t capture this **within-task heterogeneity**—for instance, it can’t distinguish report-writing tasks that take five minutes from those that take five days, or financial modeling tasks that take an afternoon from those that take a few weeks. This makes it difficult to assess AI’s economic effects: a software developer might use Claude to write ten pull requests in a day, but if nine are minor documentation updates and one is a critical infrastructure change, simply counting the number of these tasks performed with Claude misses the point.

Not only that, but as model capabilities improve, we want to understand whether they do higher-value work. To understand how AI is reshaping work and productivity, we need to know not just which tasks Claude handles, but how substantial those tasks and time savings are.

Several groups have begun conducting randomized controlled trials to measure productivity gains in narrow domains, including [software engineering tasks](#), [writing](#), and [customer service](#). METR’s work on

[measuring AI's ability to complete long tasks](#) has demonstrated that AI systems can independently tackle extended, multi-step challenges. But these evaluations consider a narrow set of problems, rather than broad real-world use. To assess AI's overall impact on the economy, we need a way to analyze hundreds or thousands of real-world AI applications.

This report takes a first step toward that goal. It uses Claude to estimate how much time it would take a human to complete the tasks that Claude handles, compares that to how long Claude and the human took together, and thereby calculates how much time the AI has saved. While AI models lack context about users' expertise, workflows, and constraints, we find that model-estimated times show promising accuracy for a dataset of software engineering tasks, relative to both human-estimated completion times and time-tracked outcomes.

In what follows, we present our methodology for estimating task-level time savings, validate our approach against ground-truth data, and then use these estimates to assess which tasks and occupations show the largest productivity gains from AI. We then explore what our task-level estimates imply about aggregate productivity as AI begins to be adopted throughout the economy.

Estimating task length and time savings

Using our [privacy-preserving analysis system](#), we analyzed 100,000 conversation transcripts from Claude.ai (Free, Pro, and Max tiers) to measure the length and time savings of tasks Claude handles. We generated two core estimates for each task:

- **Time estimate without AI:** The hours a human professional would need to complete the task without AI assistance
- **Time estimate with AI:** The amount of time it took to complete the task with AI assistance

We used Claude to generate these estimates for each conversation. Following our [Economic Index methodology](#), we then aggregated these individual chat conversations to tasks in the [O*NET](#) taxonomy by taking the median of time

estimates for each task. This allowed us to explore how such time estimates vary across tasks and occupations within the economy. Classification prompts are in the Appendix.

Analyzing real-world transcripts enables us to account for intra-task variation. For instance, even if the overall share of designing manufacturing equipment tasks stays fixed, transcript-level information lets us see whether people tackle more complex, longer-timescale projects (or attain greater time savings) with AI over time. Our Economic Index will track how these estimates evolve over time, and share aggregate datasets that researchers can use to make their own forecasts and conclusions.

Validation

Estimating task duration is [notoriously difficult for humans](#). AI models have an even more difficult job, since they lack crucial context about the broader context of tasks (though we expect this context to increase over time as features like [memory](#) and [external integrations](#) become more comprehensive). To assess whether Claude’s estimates are informative, we conducted two validation analyses.

Self-consistency testing: First, we assess whether Claude produces stable estimates of task lengths across different conversation samples, or across variations in our prompts.

We create multiple prompt variations—for example, asking about an “employee with appropriate skills” versus a “skilled professional”—to assess how sensitive estimates are to the way the prompt is phrased. We analyze 1,800 conversations with each variant, where users consented to share these conversations with us, and computed correlations across prompt variants. The results showed strong self-agreement, with log-scale correlations of $r=0.89-0.93$ across variants.

External benchmarking: Self-agreement doesn’t matter much if a model’s predictions don’t correspond well to reality. To check this, we tested Claude’s time estimation capabilities against a dataset of thousands of real-world [software development tasks](#) gathered from JIRA tickets for

Log-scale analysis of human time estimates ($r=0.892$)

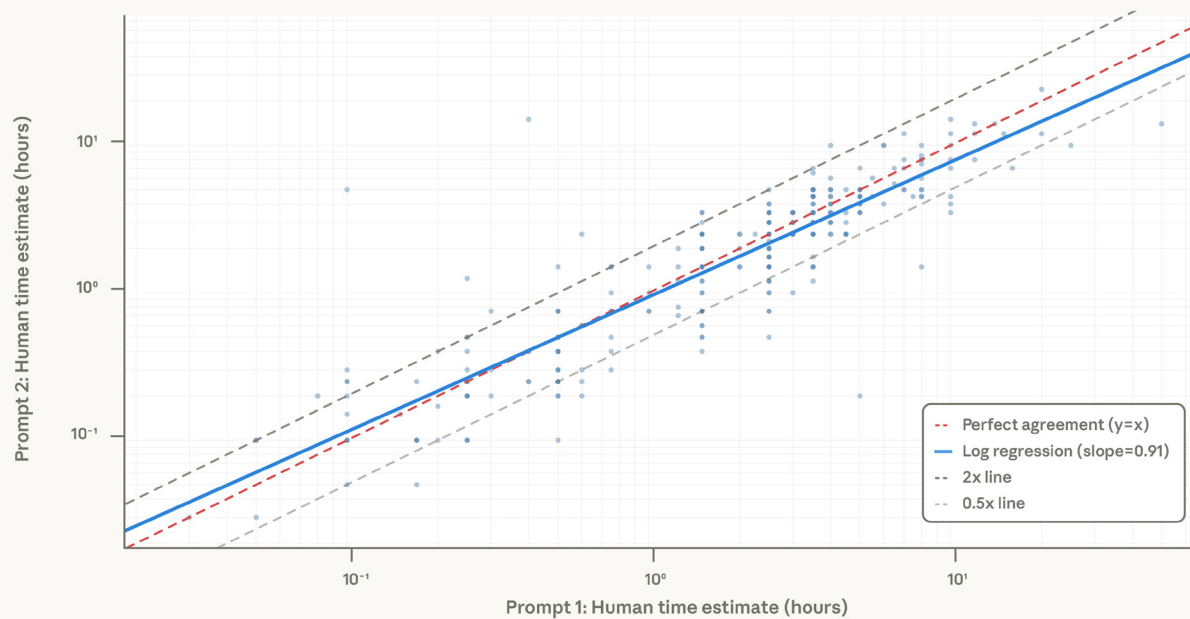


Figure 2: Claude's estimated human completion times show high correlation across prompt variations. Prompt 1 asks Claude to estimate the time it would take an “employee with appropriate skills” to complete and Prompt 2 asks about a “human worker” who is “competent in the relevant field.” The two prompts show a log-scale correlation of 0.89, indicating high agreement. Analysis performed on [Claude.ai](#) transcripts where users have consented to share them with us for research purposes.

open-source repositories, with both developer estimates and actual tracked completion times.

This is a very challenging task for Claude, given that Claude receives only the title and description of the JIRA tickets, while the human developers have full context on the codebase and the ticket, and have seen how long similar tasks take to complete. On a subset of 1000 tasks from this benchmark:

- Human developers themselves achieved $\rho=0.50$ Spearman correlation with actual times, and a Pearson correlation of $r_{\log}=0.67$ on the log values, indicating a moderate-strength correlation (higher is better for both values).

- Claude Sonnet 4.5 achieved $\rho=0.44$ and $r_{\log}=0.46$
- Claude Sonnet 4.5 with ten examples of tasks and their ground-truth time lengths showed a worse $\rho=0.39$, but improved $r_{\log}=0.48$

This analysis suggests that Claude’s estimates provide directional information that is only slightly worse than software developers’ own estimates. However, we observe that Claude’s estimates are much more compressed than humans—predicting comparatively long times for shorter tasks, and vice versa—and are overall more prone to overestimates. This suggests that the actual differences in task lengths across tasks may be larger than we report, and that actual task lengths may be slightly shorter. Overall, these findings demonstrate that model predictions have meaningful correlation with real-world outcomes, at least in this domain, making them useful for comparing one task to another or tracking changes over time. We also observe higher correlation from Claude Sonnet 4.5 compared to Claude Sonnet 4, suggesting that these estimates may continue to improve with model capabilities.

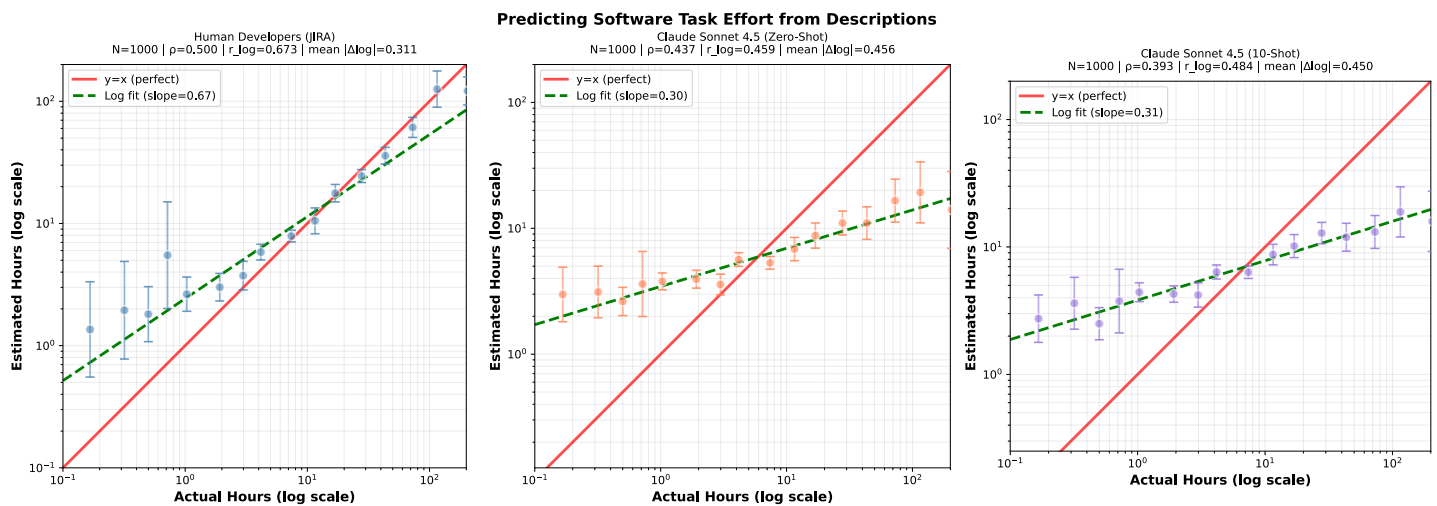


Figure 3: Correlation of actual time spent on software engineering tasks with developer and Claude estimates. Left: correlation with developers’ initial time estimates with the final time-tracked outcomes. Developers are familiar with the full codebase and understand the full context behind the request and how long similar tasks have taken. Middle: correlation with Claude Sonnet 4.5’s estimates, given just the task title and description of the JIRA ticket. Right: Correlation with Claude Sonnet 4.5’s estimates, given 10 examples in the prompt to calibrate on. Overall, Claude’s estimates have similar directional correlation to developers: Spearman’s $\rho=0.44$, compared to $\rho=0.50$ for developers, though Claude significantly overestimates short tasks and underestimates long ones. Axes are log (base 10) scaled. Error bars are 95% CIs per bin.

Results

We first use the methods above to estimate task-level savings, then aggregate these into estimates of economy-wide effects.

Task-level savings

Example AI-accelerated tasks

Vocational education teachers, postsecondary

Develop curricula and plan course content and methods of instruction.

4.5h \$33/h \$149 96%

Task time Hourly wage Task cost Time savings

Library science teachers, postsecondary

Compile bibliographies of specialized materials for outside reading assignments.

2.5h \$41/h \$101 93%

Task time Hourly wage Task cost Time savings

Office Clerks

Troubleshoot problems involving office equipment, such as computer hardware and software.

0.3h \$22/h \$7 56%

Task time Hourly wage Task cost Time savings

Social science research assistants

Prepare tables, graphs, fact sheets, and written reports summarizing research results.

2.8h \$31/h \$84 91%

Task time Hourly wage Task cost Time savings

Executive secretaries and executive administration

Prepare invoices, reports, memos, letters, financial statements, and other documents.

1.2h \$37/h \$46 87%

Task time Hourly wage Task cost Time savings

Agricultural science teachers, postsecondary

Advise students on academic and vocational curricula and on career issues.

1.6h \$47/h \$76 83%

Task time Hourly wage Task cost Time savings

Financial analysts

Interpret data on price, yield, stability, future investment-risk trends, economic influences, and other factors affecting investment programs.

0.8h \$58/h \$43 80%

Task time Hourly wage Task cost Time savings

Educational, guidance, school, and vocational counselors

Provide students with information on such topics as college degree programs and admission requirements, financial aid opportunities, trade and technical schools.

0.4h \$34/h \$14 75%

Task time Hourly wage Task cost Time savings

Secretaries and administrative assistants

Compose, type, and distribute meeting notes, routine correspondence, or reports.

0.3h \$23/h \$7 61%

Task time Hourly wage Task cost Time savings

Figure 4: Claude's estimated task time, average hourly wage of the occupation, implied task cost, and time savings for nine different tasks. Task time is estimated by having Claude predict how long a professional would take to perform the task without AI assistance. Hourly wage is derived from the Occupational Employment and Wage Statistics (OEWS) May 2024 data. Task cost is computed by multiplying the task time by the hourly wage. Time savings is computed by estimating the time the human took to complete the task and computing $1 - \text{time_with_ai} / \text{time_without_ai}$.

Example tasks demonstrate a range of time savings

Looking at individual tasks within occupations provides concrete examples of where and how AI might be delivering time savings. At the most extreme end, we see users complete curriculum development tasks that Claude thinks would take 4.5 hours in just 11 minutes. Such tasks have an implied labor cost of \$115 based on the average hourly wage of teachers.

People also use AI to save 87% of the time it would take to write invoices, memos, and other documents (at least for the type of documents Claude is asked to handle). Finally, AI saves 80% of time on financial analyst tasks like interpreting financial data for tasks that would ordinarily cost \$31 in wages.

Task length varies dramatically across occupations

Human time estimates show that Claude handles tasks of very different lengths depending on the occupation. In the below plots, we show averages for each occupation category among the subset of tasks that Claude is used for.¹ The average management task where Claude is used (e.g. selecting investments) is estimated to take humans 2.0 hours to complete, followed by legal (1.8 hours), education (1.7), and arts/media tasks (1.6). At the other end of the spectrum, food preparation tasks (e.g. planning or pricing menu items), installation/maintenance, and transportation tasks all take 0.3-0.5hrs on average, suggesting more circumscribed tasks, or tasks with less waiting time. Given that Claude's time estimates tend to underestimate long tasks and overestimate short tasks, it is possible that these differences might be even greater in practice.

Cost estimates amplify this variation in the impact of AI: the tasks with the longest time estimates also tend to be the tasks with the highest labor costs. We compute these cost estimates by multiplying the median time for each task by the associated occupation's average wage in the OEWS May 2024 data. The average management task would cost \$133 for a professional compared to \$119 for legal tasks and \$8 for tasks relating to food preparation and serving. Business and financial tasks average \$69 while computer and mathematical tasks average \$82.

Across all tasks we observe, we estimate Claude handles work that would cost a median of \$54 in professional labor to hire an expert to perform the work in

Claude's estimates by occupation category

Occupation Category	Time w/o AI (hours)	Avg. wage (\$/hr)	Avg. task cost (\$)	Time savings (%)
Management	2.0h ●	\$68/h ●	\$133 ●	85.1% ●
Business & financial operations	1.5h ●	\$45/h ●	\$69 ●	83.5% ●
Computer & mathematical	1.5h ●	\$56/h ●	\$82 ●	80.2% ●
Architecture & engineering	1.3h ●	\$50/h ●	\$64 ●	77.7% ●
Life, physical, & social science	1.5h ●	\$43/h ●	\$63 ●	82.8% ●
Community & social service	1.1h ●	\$30/h ●	\$33 ●	80.8% ●
Legal	1.8h ●	\$66/h ●	\$119 ●	80.7% ●
Education, training, & library	1.7h ●	\$32/h ●	\$52 ●	84.5% ●
Arts, design, entertainment, sports, & media	1.6h ●	\$37/h ●	\$58 ●	78.6% ●
Healthcare practitioners & technical	0.9h ●	\$51/h ●	\$43 ●	74.8% ●
Healthcare support	0.6h ●	\$19/h ●	\$10 ●	75.9% ●
Protective service	0.3h ●	\$29/h ●	\$9 ●	77.7% ●
Food preparation & serving	0.5h ●	\$17/h ●	\$8 ●	75.4% ●
Building & grounds cleaning & maintenance	0.6h ●	\$19/h ●	\$11 ●	75.8% ●
Personal care & service	0.5h ●	\$19/h ●	\$10 ●	61.0% ●
Sales & related	0.6h ●	\$26/h ●	\$15 ●	66.2% ●
Office & administrative support	0.7h ●	\$24/h ●	\$16 ●	67.8% ●
Farming, fishing, & forestry	0.5h ●	\$20/h ●	\$10 ●	83.3% ●
Construction & extraction	1.1h ●	\$31/h ●	\$35 ●	86.7% ●
Installation, maintenance, & repair	0.5h ●	\$30/h ●	\$13 ●	65.7% ●
Production	0.9h ●	\$24/h ●	\$22 ●	79.7% ●
Transportation & material moving	0.3h ●	\$23/h ●	\$7 ●	79.4% ●

Table: Various figures derived from Claude's time estimates for SOC major groups. Human time estimates vary substantially across occupations — People use Claude for management and legal tasks estimated to take humans around 2h unassisted, while healthcare support and food prep tasks average around a half-hour. Average hourly wage for the occupational category is retrieved from OEWS 2024 data. Average task cost is computed by multiplying each occupation's hourly wage by its median task time and computing an average weighted by each task's prevalence in our sample. Time savings are computed via $1 - \text{time_with_ai} / \text{time_without_ai}$.

each conversation. Of course, the actual performance of current models will likely be worse than a human expert for many tasks, though recent research suggests [the gap is closing](#) across a wide range of different applications.

Across major occupational groups, we observe a positive correlation between average hourly wage among tasks/occupations in our sample and the human-time-equivalent duration of the tasks Claude is asked to handle. For example, the Management and Legal occupational categories rank at the top of the classification in terms of average hourly wage—aligning with Claude's strengths in complex knowledge work.

Task duration vs. wage by occupation

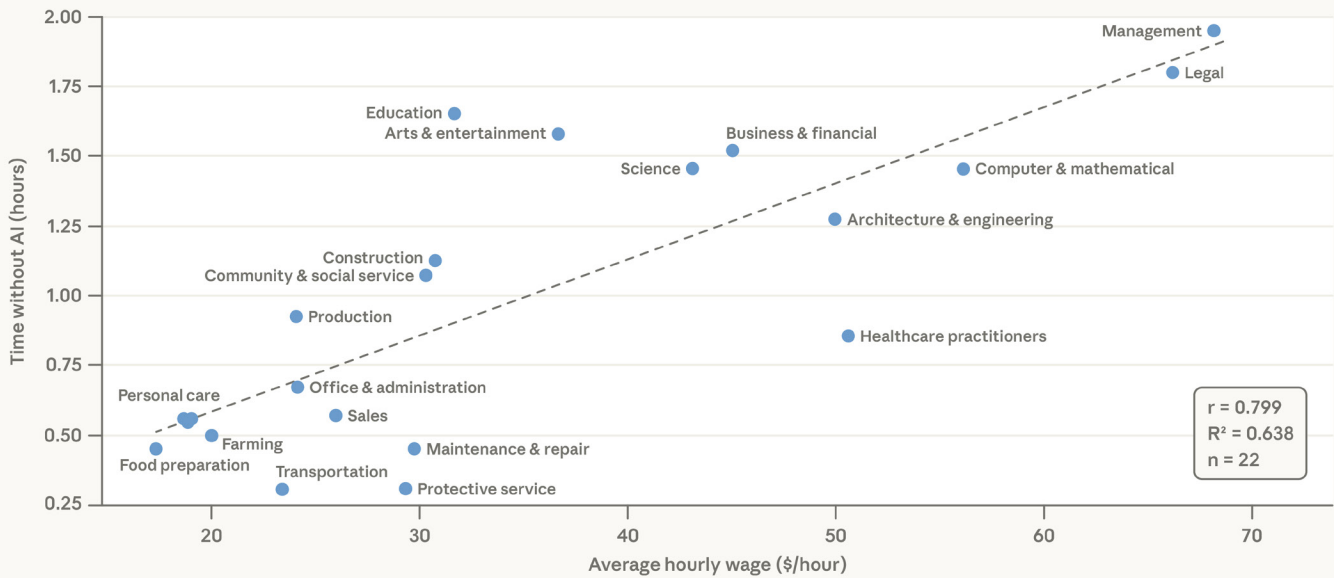


Figure 5: Correlation between average hourly wage of an occupation category and the average Claude-estimated task duration in our sample. Higher-wage occupations categories (e.g. Management and Legal) have tasks with more complex usage in our sample ($r=0.8$).

Time savings are highly uneven across occupations

Our human time and cost estimates capture the magnitude of tasks people tackle with AI. But the time savings—Claude’s estimate for how much faster work gets done with AI—reflects the productivity gains that might come from using AI for those tasks.

The median conversation experienced an estimated 84% time savings, though we see considerable variation across tasks and categories. For example, the task of checking diagnostic images only shows 20% time savings, likely because this is already a task that can be done quickly by experts without AI assistance. By contrast, the task of compiling information from reports sees approximately 95% time savings, likely because AI systems can read, extract, and cite information much more quickly than people. Overall, the distribution of time saved by task is concentrated within the 50-95% range, peaking between 80-90%.

These large time savings align with Claude’s abilities to read and write far faster than people can. However, our approach doesn’t take into account the additional work people need to do to refine Claude’s outputs to a finished state, or whether they continue iterating on the work product across multiple sessions—both of which would result in smaller time savings. Past randomized controlled trials have typically found smaller time savings, including [56%](#), [40%](#), [26%](#), [14%](#) and even [negative](#) time savings across different applications—perhaps due to these effects or because these studies examined earlier generations of models.

Distribution of time savings across tasks

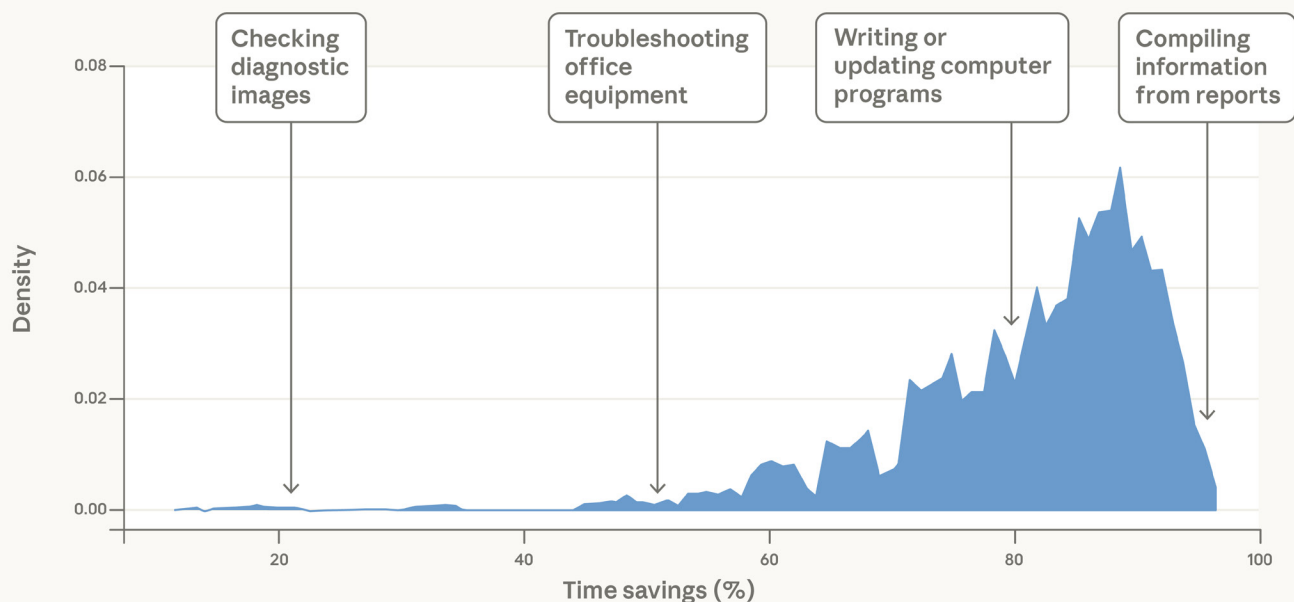


Figure 6: Density plot of time savings across O*NET tasks in our sample. We see that Claude’s estimated time savings are uneven across tasks in our sample, with most falling between 50 and 95%. The overall median savings is 81%. Time savings are computed by $1 - \text{time_with_ai} / \text{time_without_ai}$. Our estimates do not take into account the time spent refining Claude’s output outside of the chat window.

From task-level efficiency gains to economy-wide productivity effects

The above estimates capture AI-driven productivity gains at the task level.

To understand macro-level impacts, this section models how these gains could aggregate across the entire economy, assuming they play out according to Claude’s estimates.

Methodology

To estimate economy-wide productivity effects, we use Hulten’s theorem, a standard method that allows us to aggregate efficiency gains at the task-level to the broader US economy.² As in [Acemoglu \(2024\)](#)’s “baseline” approach, we model the implied increase in labor productivity as a weighted average over task-level productivity gains—a modeling choice that implicitly assumes that capital investment will increase as a result of an increase in total factor productivity (TFP) associated with AI adoption. In this framework, the implied increase TFP is the gain in labor productivity multiplied by the labor share of income.³

Task composition: For each occupation, we obtain a list of work tasks from O*NET. We then use Claude to estimate what fraction of workers’ time is spent on each of those tasks. For example, Claude estimates that programmers spend 23% of their time writing and maintaining code, 15% analyzing and rewriting programs, and smaller fractions on testing, documentation, and meetings.

Task-level productivity improvements: In the previous section, we provided estimates we can use to compute how much more quickly each task is completed with AI assistance. We take the log difference between time without AI and the time with AI to generate a productivity improvement value, and conservatively assign tasks not observed in our sample a null improvement.

Economy-wide estimate: We weight each task’s implied productivity gains by its economic importance using two factors: (i) the fraction of time that Claude estimates the occupation spends on that task (as above), and (ii) the occupation’s share of the total US wage bill (the number of people employed in that occupational category multiplied by the average wage, then divided by total wage bill across all occupations). For the total wage bill, we use May 2024 OEWS data. This approach implicitly assumes the time estimates that Claude produces represent reliable averages across

all instances of each task, and that Claude or similar AI systems will be adopted across the entire US economy.

Labor productivity contribution by occupation

Occupation	Employment	Wage bill	Wage bill share	ln (time est. ratio)	Annual labor prod contrib.
Overall economy	154.1M	\$10.46T	100.00%	0.196	1.80%
Software developers	1.65M ●	\$239.18B ●	2.3%	1.507	0.34% ●
General and operation managers	3.58M ●	\$477.16B ●	4.6%	0.224	0.10% ●
Market research analysts and marketing specialists	0.86M ●	\$74.47B ●	0.7%	1.271	0.09% ●
Secondary school teachers, except special and career/technical education	1.07M ●	\$79.05B ●	0.8%	0.820	0.06% ●
Lawyers	0.75M ●	\$136.66B ●	1.3%	0.451	0.06% ●
Customer service representatives	2.73M ●	\$123.70B ●	1.2%	0.431	0.05% ●
Retail salespersons	3.80M ●	\$141.18B ●	1.3%	0.360	0.05% ●
Computer and information systems managers	0.65M ●	\$121.44B ●	1.2%	0.379	0.04% ●
Marketing managers	0.38M ●	\$66.03B ●	0.6%	0.697	0.04% ●
Elementary school teachers, except special	1.39M ●	\$97.24B ●	0.9%	0.384	0.04% ●

Figure 7: US economy-wide labor productivity impact: top ten occupations. Overall, Claude's estimates imply a 1.8% annualized increase (dotted line) in US labor productivity assuming current AI systems were adopted universally for all tasks we observe, driven by software, management, marketing, and customer service tasks. This corresponds to an implied 1.08% annualized increase in TFP. The average ln(time estimate ratio) represents the time-weighted productivity gain across all tasks in each occupation, where time estimate ratio = time with AI / time without AI. Labor statistics derived from OEWS 2024 data.

Findings

Assuming 10 years for AI to reach universal adoption across the US economy—and using current models—we calculate that Claude's estimates imply an annual increase in US labor productivity of 1.8%. This would nearly double the current long-term growth rate, which has averaged 2.1% per year since 1947 and 1.8% since 2019. Assuming that labor's share of total factor productivity is 0.6,⁴ this implies an overall total factor productivity increase of 1.1% per year. Given that TFP growth has tended to be less than 1% since

the early 2000s, these estimates suggest that, even broad deployment of current AI systems could cause growth to double: achieving the rates of the late 1990s, and of the 1960s and 1970s.⁵

This estimated increase in aggregate labor productivity implied by task-level efficiency gains is within the range of recent estimates of AI's potential impact on productivity, though it lies towards the upper end (Filippucci, Gal, and Schief, 2024).

Importantly, this exercise assumes that AI capabilities (and humans' effectiveness in using AI) remain the same over the next 10 years as when we took our sample. This, though, seems unlikely to hold: we think that AI will [continue to improve rapidly](#) over the coming years.

Therefore, this estimate should be taken as an exercise exploring what might happen based on current usage patterns, not a prediction of the impact on productivity that is actually most likely to happen. As we have written about

US labor productivity growth

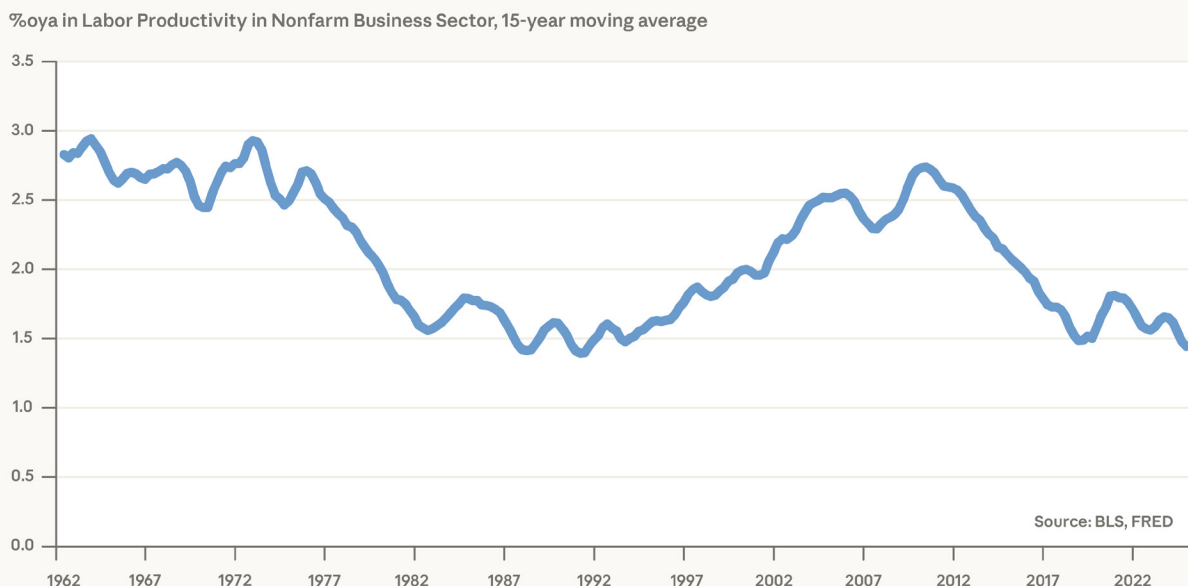


Figure 8: Labor productivity growth in the nonfarm business sector. The chart shows five year moving averages of the year-over-year percent change in labor productivity. We see a general decline from almost 3% in the 1960s to around 1.5% the last few years.

in other work, we remain extremely alert to the possibility that AI causes significant labor market disruptions, which would likely be associated with larger increases in productivity due to AI. As models progress, this could represent an approximate lower bound on the productivity effects of AI, although our estimate does not account for unevenness in adoption, which might reduce real-world productivity gains in the short term.

Reflecting the fact that [some tasks and occupations appear much more frequently in our data than others](#), we observe similar phenomena in occupations' contributions to labor productivity as well. Software developers contribute most (19%) to the total labor productivity gain attributable to AI. General and Operations Managers (about 6%), Market Research Analysts and Marketing Specialists (5%), Customer Service Representatives (4%) and Secondary School Teachers (3%) round out the top five.

In contrast, restaurants, healthcare delivery, construction, and retail contribute much less to the overall productivity effect. This is mostly because few of their tasks appear in our data—largely because these occupations have few associated tasks in our sample.

How might AI change how workers spend their time?

If workers are able to accelerate a subset of their occupational tasks with AI, the tasks where AI provides less speedup may come to represent a larger and thus more important share of those occupations' work. For example, AI might help a home inspector prepare reports, but if the inspector still has to spend the same amount of time physically traveling to the property to perform the inspection in person, this could make inspections a greater fraction of the job overall.

Figure 9 illustrates this for a few occupations. For software developers, AI speeds up the process of software development, testing, documentation, and manipulating data. But we do not currently see meaningful AI use for coordinating system installation or supervising the work of other technologists or engineers. For teachers, we see that AI assists with lesson and activity planning, but not with sponsoring extracurricular clubs or

enforcing rules in the classroom.

From a growth perspective, these observations align well with a recent observation from [Aghion, Jones, and Jones](#): “growth may be constrained not by what we are good at but rather by what is essential and yet hard to improve.”

Accelerated vs. bottleneck tasks

How AI could reshape time allocation within different jobs

Software developers		Retail salespersons	
Accelerated	Bottlenecks	Accelerated	Bottlenecks
Develop and direct software system testing and validation procedures, programming, and documentation. 86% potential time savings Currently ~11% of weekly time	Coordinate software system installation and monitor equipment functioning to ensure specifications are met. No AI usage data Currently ~4% of weekly time	Recommend, select, and help locate or obtain merchandise based on customer needs and desires. 80% potential time savings Currently ~12% of weekly time	Compute sales prices, total purchase, and receive and process cash or credit payment. No AI usage data Currently ~10% of weekly time
Modify existing software to correct errors, allow it to adapt to new hardware, or to improve its performance. 76% Potential time savings Currently ~22% of weekly time	Supervise the work of programmers, technologists and other engineering and scientific personnel. No AI usage data Currently ~2% of weekly time	Answer questions regarding the store and its merchandise. 80% potential time savings Currently ~6% of weekly time	Ticket, arrange, and display merchandise to promote sales No AI usage data Currently ~5% of weekly time

Customer service representatives		Secondary school teachers	
Accelerated	Bottlenecks	Accelerated	Bottlenecks
Confer with customers by telephone or in person to provide information about products or services, take or enter orders, cancel accounts or obtain details of complaints. 73% potential time savings Currently ~38% of weekly time	Check to ensure that appropriate changes were made to resolve customers' problems. No AI usage data Currently ~5% of weekly time	Advise students on academic and vocational curricula and on career issues. 93% potential time savings Currently ~1% of weekly time	Initiate, facilitate, and moderate classroom discussions. No AI usage data Currently ~3% of weekly time
Keep records of customer interactions or transactions, recording details of inquiries, complaints, or comments, as well as actions taken. 47% potential time savings Currently ~9% of weekly time	Determine charges for services requested, collect deposits or payments, or arrange for billing. No AI usage data Currently ~5% of weekly time	Plan and conduct activities for a balanced program of instruction, demonstration, and work time that provides students with opportunities to observe, question, and investigate. 82% potential time savings Currently ~5% of weekly time	Establish and enforce rules for behavior and procedures for maintaining order among students. No AI usage data Currently ~2% of weekly time

Figure 9: Four different occupations along with “accelerated” tasks that show large potential time savings, and potential “bottleneck” tasks that do not appear in our sample. For example, software engineers see large estimated time savings in developing and debugging software, but not in supervising programmers. Weekly time fractions are estimated by Claude (see previous section).

Limitations

Our approach has several limitations that we think warrant further research on this topic:

- **Claude’s predictions are imperfect and we lack real-world validation of Claude’s time estimates:** AI systems are imperfect predictors, and can’t see activity that happens after the user finishes their interaction with the model. While we expect these estimates will improve with models capabilities, using model estimates introduces a significant source of noise. While our estimates show that models are approaching human performance at estimating task times, and humans are far from perfect themselves, we lack real-world data to validate the estimates that Claude provides.
- **Task taxonomy limitations:** Real jobs are more complex than an O*NET task list, and the time allocations we estimate for each task are only approximate. Many important aspects of work—tacit knowledge, relationships, judgment under uncertainty—don’t appear in these formal task descriptions, and the connections between tasks may matter just as much or more to productivity as the time savings for those tasks in isolation. While we show large predicted time savings for individual tasks, [a recent randomized controlled trial](#) studying end-to-end software features did not see time savings due to AI.
- **Structural assumptions:** In our calculations above, we compare the time it would take a professional to complete a given task without AI to the time it took with AI. But this could either understate the productivity gains – since it takes additional resources we’re not accounting for to hire an employee and communicate context, and possibly overstate it, if the quality of the AI’s work is worse than a human’s.
- **Restructuring of organizations:** Historically, the largest productivity gains for individual firms have followed from [restructuring business operations](#) to adopt new technologies. Our model can help predict the effects of such a restructuring, but it cannot predict how companies might decide to restructure, or how quickly this process might happen.
- **The role of innovation:** Technological innovation is [the engine](#) of economic growth. Our model does not capture how AI systems could

accelerate or even automate the scientific process, nor the effects that would have on productivity, growth, and the structure of work.

- **Limited data:** Our dataset is derived from [Claude.ai](#) conversations only. This sample is not representative of the full spectrum of AI uses, and there's likely some selection effect where the instances of tasks people use Claude for are the ones they think Claude will be most useful. Additionally, due to our finite sample size, we likely miss some less common AI tasks.

The measurement infrastructure we develop here enables continuous tracking of the effect of AI on time savings at large scale. As models improve and better methods address these limitations, we can re-estimate these time savings and identify how these capability improvements translate into broader economic impacts. We expect to track these changes in the months and years ahead.

Conclusion

Claude handles tasks of widely varying complexity—from simple food preparation questions that would take a few minutes to complete, to complex legal and management tasks that would take multiple hours. But what is the aggregate effect of this work?

Based on Claude's time estimates per task (and assuming universal adoption over the next 10 years), we find that current AI use of current models implies a potential increase in US labor productivity of 1.8% per year—a doubling of the recent rate of labor productivity growth. Based on current AI use, these gains would be concentrated in technology, education, and professional services, while retail, restaurants, and transportation sectors would see minimal impact. We'll be tracking these changes over time as part of our Economic Index as model capabilities, products, and adoption continue to progress.

These productivity gains come from making existing tasks faster to complete. Historically, though, transformative productivity improvements—from electrification, computing, or the internet—came not from speeding up old tasks, but from fundamentally reorganizing production. In futures

like these, AI not only makes implementing features faster, but companies restructure meetings and code review to validate and ship those features faster, whether using AI or through other means.

Our framework could be used to help estimate the effects of such restructuring, but it cannot predict which changes will occur, or how quickly. An important direction for future work is understanding this question—to get a better understanding of when and how firms are reorganizing themselves around emerging AI capabilities. The answer will determine when AI makes the jump from providing significant but bounded productivity boosts, to representing the kind of structural transformation that has historically defined technological revolutions.

Bibtex

If you'd like to cite this post, you can use the following Bibtex key:

```
@online{tamkinmccrory2025productivity,  
  author = {Alex Tamkin and Peter McCrory},  
  title = {Estimating AI productivity gains from Claude  
conversations},  
  date = {2025-11-05},  
  year = {2025},  
  url = {https://www.anthropic.com/TODO},  
}
```

-
- ¹ Claude is prone to produce outlier estimates of both time horizon and cost; for example, it classifies some programming tasks as taking humans years to complete or being valued at millions of dollars. While this is possible, to produce more conservative estimates we take an average of the median value for each task, weighted by the number of conversations in each task.
 - ² Hulten's theorem states that in a competitive equilibrium without distortions, the contribution to total factor productivity of micro-level productivity gains are proportional to that production factor's Domar weight to a first order approximation. A factor's Domar weight is the ratio of its value of gross output to GDP. In the task-based model presented by Acemoglu (2024) a task's Domar weight for labor-intensive tasks is equal to that task's share of the wage bill multiplied by the labor share of income. See Baqaee and Farhi (2019) for a recent treatment and extension of Hulten's Theorem. Formulaically, Hulten's Theorem states the log change in TFP is equal to the Domar-weighted sum over the log change in micro-productivities. In our case, the log change is taken as $\ln(\text{Completion time without AI}) - \ln(\text{Completion time with AI})$.

- ³ The increase in TFP is more primitive than the increase in labor productivity. Labor productivity is the ratio of output per worker and can increase due to an increase in other factors of production aside from labor even when TFP is unchanged.
- ⁴ [Acemoglu 2024](#) calculates the labor share in AI-exposed industries as 0.57; we use the economy-wide share of 0.6 for simplicity given how close it is.
- ⁵ For historical data on total factor productivity see estimates from the Federal Reserve Bank of San Francisco: <https://www.frbsf.org/research-and-insights/data-and-indicators/total-factor-productivity-tfp/>. The average growth in TFP from 2015 to 2024 was 0.7%. Twenty years earlier, the average growth rate in TFP from 1995 to 2004 was 1.6%.

Appendix

Comparison of Claude's estimates to other estimates

AI's predicted macro-level productivity gains

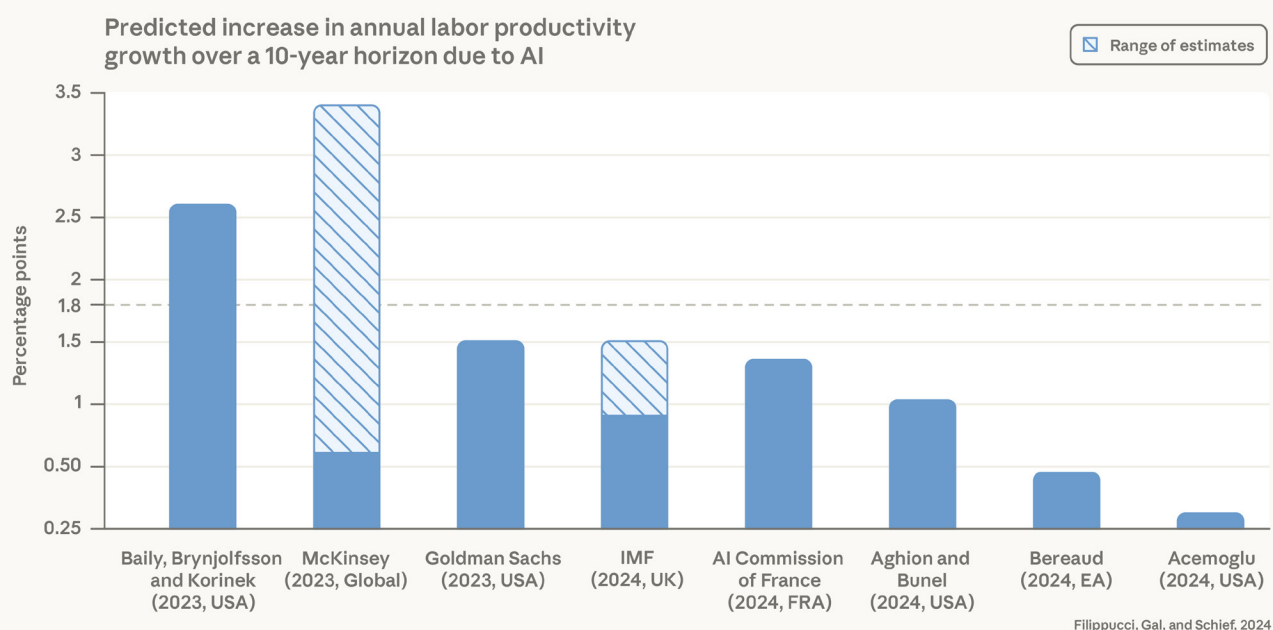


Figure 10: Predicted increase in annual labor productivity growth over a 10-year horizon due to AI. Figure reproduced from Filippucci, Gal, and Schief, 2024. The dotted line is 1.8%, derived from Claude's estimates.

Prompts used for our time estimates

Human time estimation prompt:

Human: Consider the following conversation:

```
<conversation>
{{TRANSCRIPT}}
</conversation>
```

Estimate how many hours a competent professional would need to complete the tasks done by the Assistant.

Assume they have:

- The necessary domain knowledge and skills
- All relevant context and background information
- Access to required tools and resources

Before providing your final answer, use <thinking> tags to break down your reasoning process:

```
<thinking>
```

2-5 sentences of reasoning estimating how many hours would be needed to complete the tasks.

```
</thinking>
```

Provide your output in the following format:

```
<answer>A number representing hours (can use decimals like 0.5 for shorter tasks)</answer>
```

Assistant: <thinking>

Interaction time estimation prompt:

Human: Consider the following conversation:

```
<conversation>
{{TRANSCRIPT}}
</conversation>
```

Estimate how many minutes the user spent completing the tasks in the prompt with the model.

Consider:

- Number and complexity of human messages
- Time reading Claude's responses
- Time thinking and formulating questions
- Time reviewing outputs and iterating
- Realistic typing/reading speeds
- Time implementing suggestions or running code outside of the conversation (only if directly relevant to the tasks)

Before providing your final answer, use <thinking> tags to break down your reasoning process:

```
<thinking>
```

2-5 sentences of reasoning about how many minutes the user spent.

```
</thinking>
```

Provide your output in the following format:

```
<answer>A number representing minutes</answer>
```

Assistant: <thinking>

Software development time estimation prompt:

Human: You are estimating software development tasks for open-source projects. Provide ONLY a number in hours (e.g., 0.3, 1.6, 15). Do not explain.

Task: {task}

Description: {description}:

Estimate (hours):

Assistant:

Task time estimation prompt

You are estimating how much time workers in the occupation "{occupation_title}" spend on each of their job tasks.

Below is the complete list of tasks for this occupation. For each task, estimate how many hours per week a typical worker spends on it.

Important: Don't worry about making the hours sum to exactly 40 or any specific total - we'll normalize the results afterward. Just give your best estimate for each task independently based on what seems realistic.

Tasks:
{tasks}

Return ONLY a JSON object mapping each task_id to your estimated hours per week, with no additional text, explanations, or commentary. Format:

```
{{
  "task_id_1": hours,
  "task_id_2": hours,
  ...
}}
```