ESSAY · 03

When Knowledge Is Cheap, What Is Worth Learning?

Learning Priorities in the Age of Generative AI

Max Song October 21, 2026 Version 1.0 31 min read

Self-published preprint · Not peer reviewed · Prepared with AI assistance

Abstract

Generative AI has made one kind of knowledge cheap: the kind that can be written down and handed over on request. A passable answer to almost any question now costs seconds and nothing. This essay asks what is worth learning when that is so. It argues from one economic observation and from the studies that have accumulated since 2023. The observation, from the economics of prediction, is that when a good becomes cheap, value moves to its complements. The studies show where those complements lie. Tools that hand over answers raise the output of people who can judge them: writers, consultants and customer-support agents worked faster and better, and the least experienced gained most. The same tools lower the learning of people who cannot judge them: high-school students who practised with an unguarded chatbot did better on the homework and worse on the exam, developers who learned a new library with an assistant understood it less, and the people who trust the tool most report thinking least about what it says. Four things are therefore worth learning: enough of a subject to tell a right answer from a wrong one; the skill of asking a question and choosing between answers; what cannot be written down; and the practice of learning itself, which the tools make easy to skip. The essay ends with what a reader can do, and with the limits of the evidence it rests on.

中文摘要

生成式 AI 让一种知识变得廉价：能写下来、按要求交出来的那种。现在，几乎任何问题都能在几秒内得到一个像样的回答，而且不花钱。本文问的是：在这种情况下，什么东西还值得学？文章从一条经济学观察和 2023 年以来积累的研究出发。观察来自预测经济学：一样东西变便宜，价值就转移到它的互补品上。研究则显示互补品在哪里。交出答案的工具会提高「能判断答案的人」的产出：写作者、咨询顾问、客服人员都做得更快更好，经验最少的人获益最大。同样的工具会降低「不能判断答案的人」的学习：用没有护栏的聊天机器人练习的高中生作业分数更高、考试分数更低；用助手学习新代码库的开发者理解得更少；最信任工具的人自述对它说的话想得最少。因此有四样东西值得学：对一门学问懂到能分辨对错的程度；提问与取舍的本事；写不下来的东西；以及学习这件事本身的功夫——工具让它最容易被跳过。文末写了读者可以做的事，也写了这些证据的局限。

1. The question

In October 2026 I had this site audited for privacy. One of the things I wanted to know was whether any page carried a tracking script, because the site's privacy page says that I run no analytics. An AI assistant did the first pass. It fetched the home page, searched the whole repository, found no analytics script, and reported, in the even and confident register in which such reports are written, that there was none; an allowance for one in the site's security policy was a leftover from an earlier setup. A second pass, in a real browser, found the opposite. Every page was being served with Cloudflare's analytics beacon and its bot-detection script, which wrote two cookies as it went. The assistant had not lied. It had looked where the scripts were not. They were not in the repository and not in the page as my server sent it; Cloudflare, which sits between the server and the reader, adds them at the edge, and only to responses to requests that look like a browser asking for a page. A fetch that did not look like a browser got the page without them, and from that evidence the assistant's answer was the right one. I turned the analytics off that afternoon, in Cloudflare's dashboard, and the privacy page became true again.

Two things about that afternoon have stayed with me. The first is how good the wrong answer looked. It said what it had checked, it explained the leftover, and nothing in it signalled doubt. The second is what it took to see through it: not a more fluent answer, but one particular piece of knowledge about how a page reaches a browser, and the habit of not believing a "none found" until the same check has found something it should have. Neither came from the tool. They were the price of being able to use it.

The other thing is older. In ninth grade I wrote my first programs in Python, among them a virtual kitten with moods, sounds and a couple of mini-games. Nothing in it was hard by any standard but mine, and all of it was hard by mine: I had to find out how to do each piece, and then why it did not work. The same program would now come out of a model in one pass, probably correct the first time. I have wondered whether that time was wasted. I think it is the reason I can read what a model writes in Python and tell when it is wrong, and I am not sure there is another.

That is the question of this essay, asked generally, and it is older than the machine that makes it urgent. Every technology that stores or produces knowledge has been met with the same worry: that what the tool does for us, we will no longer be able to do ourselves. Sometimes the worry was right, and sometimes it missed what mattered. What is new in 2026 is the scale and the shape of the change. A large language model does not store a particular fact the way a book or a search index does. It produces, on demand and at almost no cost, a fluent answer to nearly any question that can be put into words, including questions no one has asked before. The cost of obtaining a usable statement of what is known has fallen close to zero. That is what "cheap" means in the title. It does not mean that knowledge has lost its market value, or that credentials have; it means that the act of acquiring, or generating, a piece of written knowledge no longer costs much of anything.

If that is so, a natural conclusion follows: learning has become pointless, since there is no reason to keep in one's head what a machine will supply on request. The argument of this essay is that the conclusion is wrong, and not for sentimental reasons. It is wrong because it misreads how value moves when something becomes cheap, and because the evidence of the last three years shows, with fair consistency, whom the cheap answers help and whom they hurt. The people they help are the ones who already know enough to judge them. The people they hurt are the ones who are still trying to learn. Which of the two a reader becomes depends on what they choose to learn, and on how.

2. What "cheap" means

2.1 Three times the price fell

In Plato's *Phaedrus*, Socrates tells of the Egyptian god Theuth, who brought his invention of writing to King Thamus as a remedy for memory. Thamus answered that it would do the opposite. Those who learned it would trust marks outside themselves rather than their own memory; they would read many things without instruction and so seem to know many things while knowing little; the god was offering his pupils "the appearance of wisdom, not true wisdom" [1]. The king's complaint is the earliest recorded version of the worry, and it is instructive that he was both right and wrong. Writing did change what people kept in memory. It also made possible every kind of knowledge that does not fit in one head, including the dialogue in which his complaint survives.

The second fall in price came with search. In 2011 Sparrow, Liu and Wegner reported a series of experiments in which people who typed trivia statements into a computer remembered the statements less well when told the computer would save them than when told it would erase them, and remembered the folder a statement had been saved in better than the statement itself [2]. The paper's conclusion, that the internet had become a form of external memory, became a commonplace. It needs a caveat that is itself a lesson about cheap knowledge. When Camerer and twenty-three colleagues replicated twenty-one social-science experiments published in *Nature* and *Science*, thirteen reproduced, with effects about half as large as first reported, and the first of the Sparrow experiments, a priming effect, was among those that did not [3]; a later direct replication found no conclusive evidence for it either [4]. The finding about remembering where rather than what came from a different experiment, which those replications did not test. The point survives in a weaker form: when a fact is known to be retrievable, people put less effort into keeping it.

The third fall is the present one, and it differs in kind. A search engine returns documents that someone wrote; the reader still has to read, choose and assemble. A generative model returns the assembled answer. It writes the paragraph, drafts the code, works the problem, in the register of someone who knows. The reader is left with a finished product and a choice: accept it, check it, or set it aside. Checking requires knowing something. Accepting does not. That asymmetry is the subject of the rest of this essay.

2.2 Where value goes when something becomes cheap

The frame comes from economics. In *Prediction Machines*, Agrawal, Gans and Goldfarb describe the first generation of machine learning as a fall in the price of prediction and draw the textbook consequence: when the price of one thing falls, the things used together with it become more valuable [5]. Their candidate complement is judgment, the work of deciding what an outcome is worth, which a prediction cannot supply. In a formal version of the argument the same authors show that prediction and judgment are complements "as long as judgment is not too difficult" [6]: cheaper prediction makes judgment more valuable, up to the point where judgment is itself so hard that it is cheaper to automate the whole decision.

Generative models are a fall in the price of something broader than prediction: the price of a competent first draft of almost anything that can be expressed in words or code. The same logic applies. What becomes valuable is whatever such a draft cannot supply, and whatever is needed to use one well. Herbert Simon described the general case half a century before the models existed. Information, he wrote in 1971, consumes the attention of those who receive it, so

that "a wealth of information creates a poverty of attention" and a need to allocate attention well; most of the cost of information, he added, falls on the recipient [7]. A machine that produces unlimited answers at no cost to the asker produces, at the same moment, an unlimited demand on the asker's judgment about which answers deserve attention.

Autor makes the same point about work. Information, he observed in 2024, is "merely an input into a more consequential economic function, decision-making", and the opportunity in AI is to extend the reach and value of expertise to a larger set of workers who possess the complementary knowledge to use it [8]. The condition in that sentence is the hinge of this essay. Cheap knowledge raises the value of the knowledge that lets a person use it. It does not create that knowledge.

3. What the evidence says

Three years of experiments have tested what happens when people are handed a generative model. Read together, they show the pattern the complements argument predicts, and they show it in four parts.

3.1 The tools raise output, most for those who know least

In 2023 Noy and Zhang gave 453 college-educated professionals writing tasks of the kind their occupations require; those allowed to use ChatGPT finished in 40% less time, produced work graded 18% higher, and the gap between stronger and weaker writers narrowed [9]. Brynjolfsson, Li and Raymond followed 5,172 customer-support agents at a software firm after an AI assistant was introduced: issues resolved per hour rose 15% on average, less experienced agents improved in both speed and quality, and the most experienced gained a little speed and lost a little quality [10]. Peng and colleagues had 95 professional programmers implement a web server in JavaScript; those with an AI code-completion tool finished 55.8% faster [11]. Dell'Acqua and eight co-authors ran a field experiment with 758 consultants at a large consulting firm. On tasks within the model's reach, consultants using it completed 12.2% more tasks, finished 25.1% faster, and produced work of significantly higher quality [12].

The experiments differ in setting and design, but they agree on two things. The tools raise output on tasks the tool can do. And the people who gain most are the ones who could do the task least well before, which is what one expects when a machine supplies a competent draft and the human's contribution is to accept it and lightly amend it.

3.2 The same tools lower learning, most for those who know least

The second set of findings concerns what happens afterwards, or outside the tool's reach.

The consulting experiment had a second task, chosen to lie outside what the model could then do well. On that task, consultants using the model were 19% less likely to produce correct solutions than those without it [12]. The authors call the boundary a jagged frontier: the model is strong on one side and weak on the other, the line between them is not visible from outside, and a user who cannot see it will trust the model exactly where it should not be trusted.

In a large high school in Turkey, Bastani and colleagues gave nearly a thousand students in about fifty classes, grades nine to eleven, access to a chatbot during practice sessions in mathematics. Students with an unmodified chatbot, which the authors call GPT Base, scored 48% higher on the practice problems; students with a version built as a

tutor, giving hints and withholding answers, scored 127% higher. On the exam that followed, taken without the tool, the first group scored 17% lower than students who had had no chatbot at all, and the tutored group was statistically indistinguishable from them [13]. For the first group the practice sessions had been sessions of copying.

Shen and Tamkin gave 52 developers a new software library to learn, half with an AI assistant and half without. Those with the assistant scored 17% lower, about two grade points, on an evaluation of conceptual understanding, code reading and debugging, and were not significantly faster. Within the assisted group the authors identified six patterns of use, and three of them, the ones that kept the developer cognitively engaged, preserved learning; the others did not [14]. The study is a preprint with a small sample, from a company that makes such assistants. Its finding agrees with the school.

Fan and eight co-authors compared 117 university students writing essays under four conditions. The group with ChatGPT improved their essays most. On a later test of what they had learned about the topic, and on a task that required transferring it, they did no better than the other groups [15]; the authors' name for the pattern is metacognitive laziness. Otis and colleagues gave Kenyan entrepreneurs a GPT-4 business adviser. On average it did nothing. Entrepreneurs who had been performing well beforehand gained more than 15%; those who had been performing poorly did about 10% worse, a gap the authors trace to which pieces of the adviser's advice each group chose to act on [16]. The adviser gave everyone the same quality of answer. Only those who could tell good advice from bad were helped by it.

3.3 The feeling of competence comes apart from competence

A third set of studies measures neither output nor learning but the user's own sense of both.

METR, an evaluation organisation, recruited sixteen experienced open-source developers in early 2025 and timed them on 246 real tasks in repositories they maintained, with and without AI tools. The developers expected the tools to make them 24% faster. They were 19% slower. After the study they still believed they had been about 20% faster [17]. A second round, reported in February 2026 from 57 developers and more than 800 tasks, found a point estimate near zero for newly recruited developers and a slowdown for those returning from the first study, with intervals in both cases that include zero. The organisation itself judged the new round an unreliable signal, because of who had agreed to take part, and concluded that the tools had probably sped developers up since 2025 but that selection effects hid by how much [18]. The two reports are best read together, and read for what they both show: a gap between how much faster people felt and how much faster they were.

Fernandes and colleagues had 246 and then 452 participants solve logical-reasoning problems of the kind used in law-school admissions, with an AI assistant. Participants scored about three points above the norm for such problems and estimated their own scores about four points above what they had achieved; those who rated their own AI literacy higher judged their performance less accurately, not more [19]. In a survey of 319 knowledge workers describing 936 occasions of using generative AI at work, Lee and colleagues found that the more a worker trusted the tool, the less critical thinking they reported doing, and the more they trusted their own ability, the more [20]. Gerlich's survey of 666 people found that frequent use of AI tools went with lower scores on a critical-thinking measure, that the relationship ran through offloading thinking to the tool, and that younger respondents depended on the tools most [21]; the study is cross-sectional and self-reported, so it shows an association, not a cause. Kosmyna and colleagues recorded the brain activity of 54 people writing essays with a language model, with a search engine, or with neither. The model group showed the weakest connectivity between brain regions, reported the least sense of ownership of

their essays, and, when asked afterwards, struggled to quote a sentence from what they had just written [22]. That study too is a preprint with a small sample. Its most useful finding is the plainest one: people could not quote their own work.

3.4 When the tool is built to teach, it teaches

The pattern is not that the tools cannot teach. It is that a tool built to answer does not teach, and a tool built to teach can.

Kestin and colleagues compared, for 194 students in an introductory physics course, a lesson taught by in-class active learning with the same lesson taught by an AI tutor designed to lead a student through the material in sequence and to limit how much it put in front of them at once. Learning gains with the tutor were more than double those of the classroom session, and students spent a median of 49 minutes with it [23]. In Edo State, Nigeria, a World Bank team ran a six-week after-school programme in which students worked with a generative model, with teacher support. Students assigned to the programme improved by about three tenths of a standard deviation on tests of English, the programme's main subject, and related skills, which the authors describe as the equivalent of nearly two years of typical learning; girls gained more than boys [24]. And the tutor with guardrails in the Turkish study did not lower exam scores where the unguarded chatbot did [13].

The difference between the tools that hurt and the tools that helped was not in the model. It was in whether the tool was allowed to do the student's thinking. Where it was, the student learned less. Where it was made to withhold the answer and ask for the student's attempt, the student learned as much as before, and sometimes more.

4. Four things worth learning

If cheap answers reward the people who can judge them and penalise the people who cannot, the question of what to learn has an answer in four parts.

4.1 Enough to judge

The first is knowledge of a subject, held in one's own head, to the depth at which one can tell a right answer from a wrong one.

This cuts against the oldest promise of the information age, that one need not know things because one can look them up. The research on reading says otherwise. In 1988 Recht and Leslie gave 64 junior-high students a passage describing half an inning of baseball. Some were strong readers and some weak; some knew baseball well and some did not. Prior knowledge of the game predicted how well students understood and recalled the passage, and the weak readers who knew baseball outdid the strong readers who did not [25]. Willingham, surveying this and related work for teachers, draws the general conclusion: factual knowledge is not separate from thinking but is what thinking runs on, so that the more one knows, the more easily one learns, and the gap between those who know and those who do not widens over time [26].

The classic demonstration of what knowledge does to perception is older still. Chase and Simon showed chess masters a position from a real game for five seconds and asked them to reconstruct it. Masters did so almost perfectly; weaker players did poorly. Shown positions in which the same pieces had been placed at random, the

masters did no better than anyone else [27]. The master was not seeing more. The master was seeing in larger units, assembled over years of play, and those units did not exist for a random board. Expertise is not a store of answers. It is a way of seeing that makes a wrong answer look wrong.

That is exactly the capacity cheap answers call on. A model's output is a position on a board. The reader who knows the game sees at once whether it makes sense. The reader who does not sees only fluency, and fluency is the one thing the model always supplies. Autor's condition, that AI extends expertise to those who possess the complementary knowledge [8]; the consultants who lost accuracy outside a frontier they could not see [12]; the entrepreneurs who were helped or harmed by the same adviser depending on what they already knew [16]: these are one finding, three times. The price of admission to the benefits of cheap knowledge is enough knowledge to judge it.

How much is enough? Not everything; the point of the tools is that one need not hold everything. The working rule is this: in any field one intends to rely on the tools for, learn to the depth at which one can recognise the tools' mistakes. That is deeper than being able to ask the question and shallower than being able to produce the answer unaided. It is the depth of a good editor rather than a good writer, of a reviewer rather than an author, and it is reached only by having done the thing oneself, at least for a while.

4.2 Asking and choosing

The second thing worth learning is the skill the economists call judgment: deciding what to ask, which answer to take, and what it is worth.

Simon's point about attention is the first half of it. When answers are unlimited and attention is not, the skill that matters is allocation: knowing which questions are worth asking, which answers are worth reading, and when to stop [7]. The second half is what Agrawal, Gans and Goldfarb call judgment proper, the assignment of value to outcomes, which they showed becomes more valuable as prediction becomes cheaper, up to the point where it is too hard to exercise at all [6]. A model can list the considerations in a decision. It cannot know what the decider wants, or what the decider's situation will punish. That knowledge is where the human's work has moved.

Judgment can be learned, but only under conditions. Kahneman and Klein, who disagreed about most else concerning expert intuition, agreed on when it can be trusted: in an environment regular enough to be learned, and with enough practice and feedback to learn it. They also agreed that a person's confidence in their own judgment is "not a reliable indicator of judgment accuracy" [28]. That caution applies with full force to judging a machine. The confidence the model projects is constant; its correctness is not; and the studies in Section 3.3 show a user's sense of having judged well drifting away from having done so. To learn judgment one needs the regular environment and the feedback, and the feedback has to come from somewhere other than the model.

4.3 What cannot be written down

The third is whatever knowledge does not go into words, because what does not go into words did not go into the training data.

Polanyi's formulation is the usual one: "we can know more than we can tell" [29]. Autor, writing in 2014 about an earlier generation of automation, took the phrase for the paradox that bounded it. The tasks hardest to automate were those whose rules nobody could state, the ones requiring adaptability, common sense and judgment in situations that

could not be specified in advance; and commentators, he argued, overstated how far machines would substitute for people and understated how far they would complement them [30]. Generative models have moved that boundary. They do many things nobody wrote the rules for, having learned from examples instead. But what they learned from was writing, code and images, and what was never recorded was never learned: the feel of a material, the state of a room, what a particular person meant by a particular silence, how to carry a conversation to where it needs to go.

The labour market has been pricing this for decades. Deming showed that between 1980 and 2012 the share of jobs in the United States requiring intensive social interaction grew by nearly 12 percentage points, that mathematically demanding jobs with little social content shrank by 3.3 points, and that the strongest growth in employment and pay went to jobs needing both [31]. That was before the models. Since them, the kinds of work that can be done by producing text have become cheaper, and the kinds that cannot have not.

Brynjolfsson's warning about the Turing trap belongs here too. A machine built to imitate a person competes with that person and erodes their bargaining power, while a machine built to extend what a person can do creates new capability; and the incentives of the moment, he argued, favour the first [32]. For an individual the lesson is the mirror image. The abilities that make a person a complement to the machine rather than a substitute for it are, to a first approximation, the ones the machine was never shown.

4.4 The work of learning itself

The fourth thing worth learning is how to learn, and in particular how to recognise and tolerate the discomfort that learning requires, because the tools remove that discomfort and the learning with it.

Bjork set out the principle in 1994. Conditions that make performance improve quickly during training often fail to produce learning that lasts, while conditions that slow performance during training, by spacing, variation, testing or withholding feedback, often produce learning that does; he called the second kind "desirable difficulties". The conditions that feel most helpful, he wrote, can "act like crutches that artificially support performance during training", so that when the crutch is removed, performance collapses [33]. A model that answers on request is the most effective crutch ever built. It makes every practice session feel like progress. The Turkish students' practice scores were the performance during training; the exam was the crutch removed [13].

What learning takes, then, is the thing the tools make optional: effortful work on problems one cannot yet do, with feedback on the result. The research on expert performance is usually cited here through Ericsson, Krampe and Tesch-Römer, who attributed the difference between good and great musicians to accumulated hours of exactly such work [34]. The claim should be cited with its replication. Macnamara and Maitra repeated the study and found that practice mattered substantially but explained considerably less of the difference than first reported [35]. The weaker version is all this essay needs. Deliberate practice does not explain everything, and nothing replaces it.

Three of the studies above show what preserving it looks like. The developers who kept their learning were the ones whose way of using the assistant kept them engaged with the problem [14]. The students who learned from the tutor were the ones the tutor made to try [23]. The students who learned nothing from the chatbot were the ones it allowed to copy [13]. In each case the difference was whether the learner did the hard part.

5. What to do

The four things above are abstract. They come with five practices, and with the limits of the evidence behind them.

Attempt first, then ask. Write the paragraph, sketch the code, work the problem, before asking the model to. Then compare. The attempt is where the learning is, and the comparison is where judgment is trained. The developers who kept their learning were the ones who stayed engaged with the problem rather than handing it over [14].

Ask for an explanation, not a result. When the purpose is to learn, instruct the tool to withhold the answer and to question, hint or explain instead. The tools that helped students were built this way [13,23]. A reader can build the same constraint into a prompt, and can notice when they have quietly removed it.

Check against something that is not the model. Run the code, open the source, ask a person, try the claim against the world. Judgment is learned from feedback [28], and the one source of feedback that cannot calibrate a model is the model.

Learn one thing to the depth of recognising mistakes. Choose the fields one will rely on the tools for and, in each, go far enough to see a wrong answer as wrong. That is a bounded commitment, and it is the condition for everything else.

Spend the saved time on what the tools do not supply. The hours the tools free are hours for the knowledge that was never written down: doing the physical thing, being in the room, talking to the person. That is where the value has moved.

The evidence has limits that a reader should hold alongside it. The samples are small: sixteen developers in one study, fifty-two in another, fifty-four in a third. Several of the studies are preprints, which have not been through review, and one comes from a company whose product it concerns. The surveys measure what people say about their own thinking, not their thinking. The experiments are short, weeks at most, and say nothing about what a decade of use will do. Two of the best-known findings in this literature, on external memory and on deliberate practice, were weakened by replication, and a third, on developer speed, was qualified by its own authors within a year. None of the studies cited here should be taken as settled. What can be taken from them is a pattern that holds across settings as different as a Turkish classroom, a consulting firm, a Kenyan market and an open-source repository, and that the economics predicted before any of them was run.

6. What the question comes to

Thamus was right that writing would change what people kept in memory, and wrong about what that would mean, because he judged the new tool by the old skill. The same mistake is available now. If knowledge means a store of answers in a head, the models have made it worthless, and learning along with it. But that was never what knowledge was for. It was for seeing a situation as it is, asking the question that matters, telling a sound answer from a hollow one, and doing the things no answer can do. Those were always the hard parts. The tools have not touched them; they have only made it possible to skip the work by which they are acquired. What is worth learning, when knowledge is cheap, is whatever makes a person able to use it, and the only way anyone has found to learn that is still to do, for a while, the thing the machine would gladly do instead.

The things I still practise are the ones no model does for me. I write and mix my own music; the forty pieces on this site's Sound page are mine, and the pitch shifter in its DJ console is one I wrote myself, in 128-frame blocks, because how a key change should sound while the knob is still turning is not a question a model answers for you: you turn the knob and listen. In May 2026 I DJed for five hours at a school showcase where students played songs they had written, and nothing that comes out of a model in seconds helps with the next transition in front of a room. The knowledge that was worth learning turned out to be the kind that exists only in the doing, and the cheap answers, used well, buy the time to do it. The audit that opened this essay ended the same way. The tools did the looking. Knowing what a "none" is worth, and what to check it against, was the part that stayed with a person.

Authorship, status, and materials

Max Song chose the question, decided the argument and is the author of this essay. The literature search, the checking of sources and the text were prepared with substantial assistance from an AI system (Claude, Anthropic), which drafted prose and opened and read sources under the author's direction; the author is responsible for every claim. Every reference was opened on 2026-10-21, and the source notes published beside this page record, for each one, what was read, what it supports, what it does not, and how it was checked. The essay is self-published, version 1.0, dated 2026-10-21. It has not been peer reviewed. No DOI, institutional affiliation or external endorsement is claimed. It reports no measurement of its own; the numbers in it are the published numbers of the studies cited, given with their sample sizes.

References

[1] Plato. 1925. Phaedrus. Harvard University Press. Translated by Harold N. Fowler. Plato in Twelve Volumes, vol. 9, Loeb Classical Library; London: William Heinemann. Sections 274e–275b, read in the Perseus Digital Library.

[2] Betsy Sparrow; Jenny Liu; Daniel M. Wegner. 2011. Google Effects on Memory: Cognitive Consequences of Having Information at Our Fingertips. Science. vol. 333(6043). pp. 776–778.

[3] Colin F. Camerer; Anna Dreber; Felix Holzmeister; et al.. 2018. Evaluating the Replicability of Social Science Experiments in Nature and Science between 2010 and 2015. Nature Human Behaviour. vol. 2(9). pp. 637–644. Twenty-four authors.

[4] Guido Hesselmann. 2020. No Conclusive Evidence That Difficult General Knowledge Questions Cause a “Google Stroop Effect”. A Replication Study. PeerJ. vol. 8. Article e10325.

[5] Ajay Agrawal; Joshua Gans; Avi Goldfarb. 2018. Prediction Machines: The Simple Economics of Artificial Intelligence. Harvard Business Review Press. Boston, MA. 250 pp. The chapters “Cheap Changes Everything” and “The Value of Judgment”.

[6] Ajay K. Agrawal; Joshua S. Gans; Avi Goldfarb. 2018. Prediction, Judgment and Complexity: A Theory of Decision Making and Artificial Intelligence. National Bureau of Economic Research. Working Paper 24243, January 2018. A revised version appears in The Economics of Artificial Intelligence: An Agenda (University of Chicago Press, 2019).

[7] Herbert A. Simon. 1971. Designing Organizations for an Information-Rich World. Computers, Communications, and the Public Interest. pp. 37–72. Edited by Martin Greenberger. Baltimore: The Johns Hopkins Press. The passage on the scarcity of attention is on pp. 40–41.

[8] David Autor. 2024. Applying AI to Rebuild Middle Class Jobs. National Bureau of Economic Research. Working Paper 32140, February 2024.

[9] Shakked Noy; Whitney Zhang. 2023. Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence. Science. vol. 381(6654). pp. 187–192.

[10] Erik Brynjolfsson; Danielle Li; Lindsey Raymond. 2025. Generative AI at Work. The Quarterly Journal of Economics. vol. 140(2). pp. 889–942.

[11] Sida Peng; Eirini Kalliamvakou; Peter Cihon; Mert Demirer. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590, preprint. Version 1, 13 February 2023.

[12] Fabrizio Dell'Acqua; Edward McFowland III; Ethan Mollick; Hila Lifshitz; Katherine C. Kellogg; Saran Rajendran; Lisa Krayer; François Candelon; Karim R. Lakhani. 2026. Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality. Organization Science. vol. 37(2). pp. 403–423. First circulated as Harvard Business School Working Paper 24-013, September 2023.

[13] Hamsa Bastani; Osbert Bastani; Alp Sungu; Haosen Ge; Özge Kabakcı; Rei Mariman. 2025. Generative AI without Guardrails Can Harm Learning: Evidence from High School Mathematics. Proceedings of the National Academy of Sciences. vol. 122(26). Article e2422633122.

[14] Judy Hanwen Shen; Alex Tamkin. 2026. How AI Impacts Skill Formation. arXiv:2601.20245, preprint. Version 2, 1 February 2026.

[15] Yizhou Fan; Luzhen Tang; Huixiao Le; Kejie Shen; Shufang Tan; Yueying Zhao; Yuan Shen; Xinyu Li; Dragan Gašević. 2025. Beware of Metacognitive Laziness: Effects of Generative Artificial Intelligence on Learning Motivation, Processes, and Performance. British Journal of Educational Technology. vol. 56(2). pp. 489–530.

[16] Nicholas G. Otis; Rowan Clarke; Solène Delecourt; David Holtz; Rembrand Koning. 2025. The Uneven Impact of Generative AI on Entrepreneurial Performance. OSF, preprint. Version 3, 3 October 2025.

[17] Joel Becker; Nate Rush; Elizabeth Barnes; David Rein. 2025. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. arXiv:2507.09089, preprint. Version 2, 25 July 2025. Also published by METR as a report on 10 July 2025.

[18] Joel Becker; Nate Rush; Tom Cunningham; David Rein; Khalid Mahamud. 2026. We Are Changing Our Developer Productivity Experiment Design. METR, blog post. 24 February 2026. Read 2026-10-21.

[19] Daniela Fernandes; Steeven Villa; Salla Nicholls; Otso Haavisto; Daniel Buschek; Albrecht Schmidt; Thomas Kosch; Chenxinran Shen; Robin Welsch. 2026. AI Makes You Smarter but None the Wiser: The Disconnect between Performance and Metacognition. Computers in Human Behavior. vol. 175. Article 108779. Published online 4 November 2025.

[20] Hao-Ping (Hank) Lee; Advait Sarkar; Lev Tankelevitch; Ian Drosos; Sean Rintel; Richard Banks; Nicholas Wilson. 2025. The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects from a Survey of Knowledge Workers. Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25). pp. 1–22. Article 1121. New York: Association for Computing Machinery.

[21] Michael Gerlich. 2025. AI Tools in Society: Impacts on Cognitive Offloading and the Future of Critical Thinking. Societies. vol. 15(1). Article 6. A correction to one table was published as Societies 15(9), 252, on 10 September 2025.

[22] Nataliya Kosmyna; Eugene Hauptmann; Ye Tong Yuan; Jessica Situ; Xian-Hao Liao; Ashly Vivian Beresnitzky; Iris Braunstein; Pattie Maes. 2025. Your Brain on ChatGPT: Accumulation of Cognitive Debt When Using an AI Assistant for Essay Writing Task. arXiv:2506.08872, preprint. Version 2, 31 December 2025.

[23] Gregory Kestin; Kelly Miller; Anna Klales; Timothy Milbourne; Gregorio Ponti. 2025. AI Tutoring Outperforms In-Class Active Learning: An RCT Introducing a Novel Research-Based Design in an Authentic Educational Setting. Scientific Reports. vol. 15. Article 17458.

[24] Martín De Simone; Federico Tiberti; Maria Barron Rodriguez; Federico Manolio; Wuraola Mosuro; Eliot Jolomi Dikoru. 2025. From Chalkboards to Chatbots: Evaluating the Impact of Generative AI on Learning Outcomes in Nigeria. World Bank. Policy Research Working Paper 11125, May 2025. The headline numbers are also reported by the authors in a World Bank blog post of 9 January 2025.

[25] Donna R. Recht; Lauren Leslie. 1988. Effect of Prior Knowledge on Good and Poor Readers' Memory of Text. Journal of Educational Psychology. vol. 80(1). pp. 16–20.

[26] Daniel T. Willingham. 2006. How Knowledge Helps: It Speeds and Strengthens Reading Comprehension, Learning—and Thinking. American Educator. vol. 30(1). Spring 2006.

[27] William G. Chase; Herbert A. Simon. 1973. Perception in Chess. Cognitive Psychology. vol. 4(1). pp. 55–81.

[28] Daniel Kahneman; Gary Klein. 2009. Conditions for Intuitive Expertise: A Failure to Disagree. American Psychologist. vol. 64(6). pp. 515–526.

[29] Michael Polanyi. 1966. The Tacit Dimension. Doubleday. Garden City, NY. Reissued by the University of Chicago Press, 2009. The sentence quoted is on p. 4, chapter 1, “Tacit Knowing”.

[30] David H. Autor. 2014. Polanyi's Paradox and the Shape of Employment Growth. National Bureau of Economic Research. Working Paper 20485, September 2014.

[31] David J. Deming. 2017. The Growing Importance of Social Skills in the Labor Market. The Quarterly Journal of Economics. vol. 132(4). pp. 1593–1640.

[32] Erik Brynjolfsson. 2022. The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence. Daedalus. vol. 151(2). pp. 272–287.

[33] Robert A. Bjork. 1994. Memory and Metamemory Considerations in the Training of Human Beings. Metacognition: Knowing about Knowing. pp. 185–205. Edited by Janet Metcalfe and Arthur P. Shimamura. Cambridge, MA: MIT Press.

[34] K. Anders Ericsson; Ralf Th. Krampe; Clemens Tesch-Römer. 1993. The Role of Deliberate Practice in the Acquisition of Expert Performance. Psychological Review. vol. 100(3). pp. 363–406.

[35] Brooke N. Macnamara; Megha Maitra. 2019. The Role of Deliberate Practice in Expert Performance: Revisiting Ericsson, Krampe & Tesch-Römer (1993). Royal Society Open Science. vol. 6(8). Article 190327.