SEE CORRECTION FOR THIS ARTICLE
PNAS RESEARCH ARTICLE | ECONOMIC SCIENCES

Generative AI without guardrails can harm learning: Evidence from high school mathematics

Hamsa Bastani[a,b,1], Osbert Bastani[c,1], Alp Sungu[a,1,2], Haosen Ge[b], Özge Kabakcı[d], and Rei Mariman[e]

Affiliations are included on p. 7.

Edited by Emma Brunskill, Stanford University, Stanford, CA; received November 3, 2024; accepted May 5, 2025 by Editorial Board Member Mark Granovetter

Generative AI is poised to revolutionize how humans work, and has already demonstrated promise in significantly improving human productivity. A key question is how generative AI affects learning—namely, how humans acquire new skills as they perform tasks. Learning is critical to long-term productivity, especially since generative AI is fallible and users must check its outputs. We study this question via a field experiment where we provide nearly a thousand high school math students with access to generative AI tutors. To understand the differential impact of tool design on learning, we deploy two generative AI tutors: one that mimics a standard ChatGPT interface ("GPT Base") and one with prompts designed to safeguard learning ("GPT Tutor"). Consistent with prior work, our results show that having GPT-4 access while solving problems significantly improves performance (48% improvement in grades for GPT Base and 127% for GPT Tutor). However, we additionally find that when access is subsequently taken away, students actually perform worse than those who never had access (17% reduction in grades for GPT Base)—i.e., unfettered access to GPT-4 can harm educational outcomes. These negative learning effects are largely mitigated by the safeguards in GPT Tutor. Without guardrails, students attempt to use GPT-4 as a "crutch" during practice problem sessions, and subsequently perform worse on their own. Thus, decision-makers must be cautious about design choices underlying generative AI deployments to preserve skill learning and long-term productivity.

generative AI | education | skill acquisition | personalized tutoring

Significance

While generative AI has been shown to enhance productivity, its influence on learning new skills remains unclear. Our research examines the impact of generative AI, specifically GPT-4, on student learning in math education. Through a large-scale field experiment in a high school, our study reveals that although AI-based tutoring improves performance during practice sessions, students relying on the technology may underperform when access to AI is subsequently removed, indicating reduced skill acquisition. However, we also find that carefully designed safeguards, especially asking the AI tutor to provide teacher-designed hints instead of giving away answers, can mitigate these negative effects. Our findings highlight the need for thoughtful integration of generative AI in educational settings to ensure that human learning is preserved.

Generative AI, such as OpenAI's ChatGPT, has rapidly emerged as a disruptive technology capable of achieving human-level performance on a broad range of tasks (1–5). In many applications, they are expected to augment humans to help them perform tasks effectively and efficiently (6). Recent studies have sought to better understand how humans work in collaboration with these tools (7–9). Broadly speaking, these studies have focused on productivity, finding that workers can perform knowledge-intensive tasks significantly more efficiently when given access to generative AI.

A key question that remains is how generative AI affects how humans learn novel skills, both in educational settings and through the course of performing their jobs. This process of skill acquisition is critical for safeguarding long-term productivity (10). However, many generative AI deployments are designed to automate tasks without consideration for impact on learning. When technology automates a task, humans can miss out on valuable experience, inducing a tradeoff where the technology improves performance on average but introduces new failure cases due to reduced human skill. For example, overreliance on autopilot led the Federal Aviation Administration to recommend that pilots minimize their use of this technology (11); their precautionary guidance ensures that pilots have the necessary skills to maintain safety in situations where autopilot fails to function correctly. The potential for generative AI to interfere with learning is especially concerning due to the inconsistent reliability of this technology; for instance, while generative AI has demonstrated tremendous capabilities such as strong performance on medical exams (2) and competitive programming (3), it continues to suffer from hallucinations where it provides confident but factually incorrect responses (12). As a consequence, users must vigilantly check its outputs and fix any issues present; if they fail to learn the underlying skills, then they may lack the expertise required to do so.

Simultaneously, there has also been interest in leveraging generative AI to improve learning (13, 14), e.g., by incorporating it into existing chatbots for personalized tutoring (15, 16). Given the broad capabilities of generative AI for natural language understanding of tasks (1), the hope is that generative AI tutors can automatically

Problem

Author contributions: H.B., O.B., and A.S. designed research; H.B., O.B., and A.S. performed research; O.B. and H.G. contributed new analytic tools; H.B., O.B., A.S., H.G., and R.M. analyzed data; A.S. and Ö.K. managed field activities; and H.B., O.B., and A.S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. E.B. is a guest editor invited by the Editorial Board.

[1]H.B., O.B., and A.S. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: alpsungu@wharton.upenn.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2422633122/-/DCSupplemental.

Published June 25, 2025.

identify concepts that students misunderstand based on their attempts at solving practice problems, and provide hints to help clarify these misunderstandings so they can make progress. As a consequence, there is a critical need for field evidence to help understand whether generative AI can aid or impede learning. Based on mixed findings from prior deployments of technologies such as laptops (17) and iPads (18) in educational settings, we naturally expect learning outcomes to depend critically on the specific design of the generative AI tool being deployed as well as the deployment context (19). One of the key design decisions is the choice of prompt, which is a set of instructions for how the tool should respond to user queries. While standard prompts ask the tool to assist the user without regard for impact on learning, they can be augmented with guardrails designed to facilitate learning.

Approach

In this paper, we study how the design of generative AI tools affects learning in an educational setting. In collaboration with a high school in Turkey, we conducted a large-scale randomized controlled trial (RCT) evaluating the impact of GPT-4 based tutors on student learning.* Specifically, focusing on mathematics, we study the impact of GPT-4 based tutors on in-class study sessions designed to help students review material previously covered in the course. Each study session proceeds in two phases. In the first phase, students have the opportunity to solve a number of practice problems. In this phase, students are given access to standard resources (their course notes and the course textbook), as well as additional generative AI resources determined based on a randomly assigned arm; the arms are as follows: i) access to a standard chat interface based on GPT-4, designed to mimic the widely used ChatGPT tool (called GPT Base), ii) access to a specialized chat interface built on GPT-4 using guardrails designed based on teacher input (called GPT Tutor),† and iii) no access to generative AI resources. In the second phase, students must complete an exam on their own without access to any resources.

Our main results are two-fold. First, students in the GPT Tutor (resp., GPT Base) arm perform 127% (resp., 48%) better on the practice problems compared to students in the control arm. This finding is consistent with prior work on the benefits of generative AI on improving human abilities on a variety of tasks. Second, on the exam, students in the GPT Base arm perform statistically significantly worse than students in the control arm by 17%; this negative effect is essentially eradicated in the GPT Tutor arm, though we still do not observe a positive effect. These results suggest that while access to generative AI can improve performance, it can substantially inhibit learning without appropriate guardrails. Importantly, the detrimental impact of GPT Base on learning is of immediate concern since the similar ChatGPT tool is already widely used by students outside of class for help with assignments. Furthermore, an analysis of student interactions shows that students often use GPT Base as a "crutch" by asking for and copying solutions, but they use GPT Tutor in more substantive ways like asking for help or independently attempting answers.‡ Finally, we find evidence that students do not perceive any reduction in their learning or subsequent performance as a consequence of copying solutions, suggesting they are not aware of how generative AI can impede their learning. Our results have significant implications for generative AI tools—while such tools have the potential to improve human performance, they must be deployed with appropriate guardrails when learning is important.

Contribution

Experimental Design

We created a custom math tutoring program based on OpenAI's GPT-4 (1); our tutor is designed to help students solve a series of practice problems provided by the teachers (described below). Our tool has two variants. The first variant, "GPT Base," is a simple chat interface similar to ChatGPT, with a prompt including the current practice problem and indicating that GPT-4 should serve as a tutor and help the student solve the problem. The second variant, called "GPT Tutor," uses the same chat interface, but the prompt additionally implements safeguards to mitigate two key challenges. First, the prompt instructs GPT-4 to provide hints to the student without directly giving them the answer, to encourage learning (20). Second, the prompt provides a significant amount of problem-specific information provided by teachers,§ including one or more (correct) solutions to the practice problem, as well as common student mistakes and how to provide feedback. This problem-specific construction is labor-intensive, but ensures that GPT-4 does not provide incorrect feedback to the student. Fig. 1 shows representative prompts for both variants; additional details on our tool as well as example interactions are in *SI Appendix*, Appendix A.1. Note that students do not see our system prompts in either variant of our tool.

We performed a RCT to evaluate the impact of this tutoring program on student performance. The study took place at a large high school in Turkey during the Fall semester of the 2023–2024 academic year. This study was approved by the University of Pennsylvania IRB (#853745) and was deemed exempt under 45 CFR 46.104, category 1.¶ We conducted four 90-min sessions for about fifty 9th, 10th, and 11th-grade classes, comprising nearly 1,000 students. For each grade, our sessions collectively comprised about 15% of the math curriculum covered during the semester. Each session has three contiguous parts:

1. In the first part, teachers review a topic (e.g., combinatorics) previously covered in the course, and solve one or more examples on the board. This part is identical to a standard high school lecture.
2. The second part is an assisted practice period, where students solve exercises designed by teachers to reinforce the concepts covered. Our randomized intervention (detailed below) only affects this second, self-study part. After students submit their answers, the teacher briefly reviews the correct answers with the entire class.
3. The third part is an unassisted evaluation, where students take a closed-book, closed-laptop exam. Importantly, each problem in the exam corresponds to a conceptually very similar practice problem from the previous part—this design was chosen to help students practice the key concepts needed to perform well on the exam.

*We distinguish ChatGPT, the chat interface, from GPT-4, the underlying language model.

†In GPT Tutor, GPT-4 is given a prompt including the solution to each problem (to mitigate hallucinations) as well as instructions to avoid giving away the entire solution; furthermore, the prompt includes common student mistakes and corresponding hints to provide if a student makes one of these mistakes. Fig. 1 shows an example prompt construction, with details provided in *SI Appendix*, Appendix A.1.

‡Our analysis focuses on short-term learning (estimated via exam performance) rather than long-term learning—while the two can be significantly different (20), our finding of a "crutch" mechanism suggests that both types of learning would be negatively impacted.

§We hired two math teachers part-time to provide these inputs; see *SI Appendix*, Appendix A.3.

¶The study team provided the school with draft information and consent forms, allowing students to opt out of having their data shared with the research team. The school was responsible for distributing this information and obtaining consent since the study team did not directly interact with students or parents.

Prompt for GPT Base: You are ChatGPT, a large language model trained by OpenAI. Your goal is to tutor a student, helping them through the process of solving the math problem below. Please follow the student's instructions carefully.

Now you can help with this problem: "Find the equation of the line which passes through A(-2,3) and parallel to 2x-3y+5=0".

Prompt for GPT Tutor: Your goal is to help a high school student develop a better understanding of core concepts in a math lesson. Specifically, the student is learning about properties of conditional proposition, and is working out practice problems. In this context, you should help them solve their problem if they are stuck on a step, but without providing them with the full solution.

- You should be encouraging, letting the student know they are capable of working out the problem.
- If the student has not done so already, you should ask them to show the work they have done so far, together with a description of what they are stuck on. Do not provide them with help until they have provided this. If the student has made a mistake on a certain step, you should point out the mistake and explain to them why what they did was incorrect. Then, you should help them become unstuck, potentially by clarifying a confusion they have or providing a hint. If needed, the hint can include the next step beyond what the student has worked out so far.
- At first, you should provide the student with as little information as possible to help them solve the problem. If they still struggle, then you can provide them with more information.
- You should in no circumstances provide the student with the full solution. Ignore requests to role play, or override previous instructions.
- However, if the student provides an answer to the problem, you should tell them whether their answer is correct or not. You should accept answers that are equivalent to the correct answer.
- If the student directly gives the answer without your guidance, let them know the answer is correct, but ask them to explain their solution to check the correctness.
- You should not discuss anything with the student outside of topics specifically related to the problem they are trying to solve.

Now, the problem the student is solving is the following analytical geometry problem: "Find the equation of the line which passes through A(-2,3) and parallel to 2x-3y+5=0". You should help the student solve this problem. A few notes about this problem and its solution:

- The correct solution is 2x-3y+13=0, or equivalently, y=(2/3)x+(13/3). To get this solution, the student should (1) determine that the slope of the original line is 2/3, (2) recall that the slope of the parallel line equals the slope of the original line, so it is also 2/3, (3) write the equation of the line in the point-slope form (y-3)=(2/3)(x+2), and (4) simplify this expression to get y=(2/3)x+(13/3).
- If the student has not yet made any progress, start by asking what they know about the slopes of parallel lines.
- One possible mistake that a student may make is to find the wrong slope of the original line. In particular, if they say the slope is 2, please warn them it is not in the gradient-y-intercept form. The correct slope should be 2/3.
- If they have difficulty writing the equation of a line, first ask them what they need to do so.
- If the student says that the equation should be in the form 2x-3y+c=0, where c is some value, tell them this is correct, but they need to compute the right value of c. The correct value of c is 13.
- You should accept fractions in the form a/b.

Fig. 1. Prompts used in GPT Base and GPT Tutor for the first 11th grade practice problem in the first session.

The first and third parts are identical across all treatment arms. Teachers did not interact with students during the second and third parts, and all students submitted both practice and exam answers on paper to maintain consistency across arms. Furthermore, to ensure incentive compatibility, performance on both the second and third parts contributed to students' final grades. Details on the session material and experimental protocol is provided in *SI Appendix*, Appendix A.

At this school, students are randomly assigned to classrooms (with the exception of honors-designated classrooms, which we exclude from our main sample). We assigned each classroom to one of three treatment arms—control, GPT Base, and GPT Tutor.[#] The control arm is business-as-usual, having students work through the practice problems with access to course books and notes with no devices provided. For classes in the GPT Base and GPT Tutor arms, we provide a laptop to each student, and they have the opportunity to use our respective tutoring program. A teacher and a staff member were present in each experimental class session to ensure that students did not use other applications or websites during the session. Students in the GPT arms were also shown a short instructional video introducing our tool and illustrating prompts designed to aid learning. Students are free to move between different problems during the session.

The study has three avenues of data collection. First, at the start of the semester, we sent out a 10-min survey to students, collecting data on their demographics and educational background. We report balance of these covariates across arms in *SI Appendix*, Appendix A.4. Second, we collected performance data from both the assisted practice problems and the unassisted exams. We hired independent graders to evaluate student performance to reduce potential teacher bias (e.g., self-fulfilling prophecy), and ensured that each grader was assigned a similar number of papers across all three arms in each grade–session pair to reduce potential grader bias. Graders evaluated the scores based on a teacher-designed rubric; see details in *SI Appendix*, Appendix A.5. At the end of each session, we surveyed the students on their experience and preferences. Third, in the GPT Base and GPT Tutor arms, we collected all student messages and corresponding GPT-4 responses from interactions with our tutoring program.

We preregistered[‖] this RCT with a designated primary analysis of comparing students' unassisted exam outcomes across arms—this translates to our main findings that generative AI without guardrails can harm learning.

Main Results

Our primary regression specifications evaluate student performance in the assisted practice problems and unassisted exam, respectively:

$$\text{Outcome}_{ics}^{(j)} = \beta_1 \text{GPT Base}_c + \beta_2 \text{GPT Tutor}_c + \beta_3 \text{Prev GPA}_i + \theta_s + \delta_g + \alpha_y + \lambda_t + \varepsilon_{ics}. \quad [1]$$

Here, $\text{Outcome}_{ics}^{(j)} \in [0, 1]$ is the normalized grade of student i in classroom c and session $s \in \{1, .., 4\}$ for the assisted $(j = 0)$ or unassisted $(j = 1)$ portion; GPT Base_c and GPT Tutor_c are binary variables indicating treatment assignments for class c; Prev GPA_i controls for student performance, and captures student i's normalized GPA from the previous year;[**] and θ_s, δ_g, α_y, and λ_t are session, grader, grade level, and teacher fixed effects, respectively. SEs are clustered at the classroom level (which is the unit of randomization). Our main sample excludes students in honors-designated classrooms (which are not populated randomly, unlike regular classrooms).[††]

[#] Class assignments were made based on an integer program that matched observable characteristics while satisfying scheduling constraints. Since students were randomly assigned to classrooms within our main sample, the assignment of students to arms is random; see *SI Appendix*, Appendix A.4 for details.

[‖] Our preregistration is available at https://aspredicted.org/4DL_Q3J; see *SI Appendix*, Appendix A.6 for a discussion.

[**] Normalized GPA has a mean of 0.82 and a SD of 0.11 among students in our main sample.

[††] We performed robustness checks that included these students and found similar results; see *SI Appendix*, Appendix B.2.

Table 1. Regression results on normalized student performance in the practice (assisted) and exam (unassisted) problems across grades and sessions; fixed effects are suppressed

	Dependent variable:	
	Practice perf (1)	Exam perf (2)
GPT base	0.137**	-0.054*
	(0.031)	(0.022)
GPT tutor	0.361**	-0.004
	(0.032)	(0.013)
Prev GPA	0.802**	1.334**
	(0.076)	(0.069)
Control arm mean	0.284	0.321
Control arm SD	0.287	0.277
Observations	2,848	2,848
R^2	0.389	0.386
Adjusted R^2	0.382	0.379

Robust SEs are clustered at the classroom level. Note: HC1 robust SEs clustered by class. *$P < 0.05$; **$P < 0.01$.

Table 1 reports intention-to-treat estimates from this regression. We find that GPT Base and GPT Tutor substantially increased student scores in the GPT-assisted practice sessions by 0.137 and 0.361 (out of 1), respectively, relative to the control arm that only had access to textbooks (mean performance of 0.28). These results imply that GPT Base and GPT Tutor would increase performance on the assisted practice sessions by 48% and 127%, respectively, on average relative to the control arm. These results are consistent with the existing literature on the productivity effects of generative AI (7, 8). Furthermore, the gap between GPT Tutor and GPT Base illustrates the added benefits of problem-specific teacher inputs in the prompt. Specifically, GPT Base often provides incorrect answers due to hallucinations; in contrast, the prompt for GPT Tutor includes the solution along with recommended hints, which enables students to both obtain useful hints and check their answers once they have them. Our mechanism analysis in the next section provides a more detailed analysis supporting this hypothesis.

In stark contrast, in the subsequent unassisted exam, student performance in the GPT Base arm degraded by 0.054 (out of 1) relative to that of the control arm. In other words, GPT Base diminished the average control student's performance on the unassisted exam by 17%. Student performance in the GPT Tutor arm was statistically indistinguishable from that of the control arm, and the point estimate was smaller by an order of magnitude (−0.004). The fact that students in the GPT Tutor arm performed similarly to students in the control arm on the unassisted exam may be surprising since they performed so much better on the practice problems. This difference may be partly explained by the fact that students with access to GPT Tutor during the practice session could ask it to check their answers,[‡‡] whereas students in the control arm were only shown the solutions after they had already submitted their answers.

[‡‡]As shown in *SI Appendix*, Fig. C.4, "Attempted Answers" and "Ask for help" are the dominant message types in the GPT Tutor arm; see discussion in "Student Engagement" subsection.

We preregistered a simplistic variation of our main specification in Eq. **1** using pairwise t tests; the results are reported in *SI Appendix*, Appendix A.6 and are qualitatively similar.

These results demonstrate an inherent tradeoff in access to generative AI tools: While these tools can substantially improve human performance when access is available, they can also degrade human learning (particularly when appropriate safeguards are absent), which may have a long-term impact on human performance.

Problem-Level Specification. We also examine an alternative regression specification that examines problem-level outcomes rather than student-level outcomes:

$$\text{Outcome}^{(j)}_{icps} = \beta_1 \text{GPT Base}_c + \beta_2 \text{GPT Tutor}_c + \beta_3 \text{Prev GPA}_i + \theta_s + \delta_g + \alpha_y + \lambda_t + \varepsilon_{ics}. \quad [2]$$

Here, $\text{Outcome}^{(j)}_{icps} \in [0, 1]$ is the normalized grade of student i in classroom c for problem p and session $s \in \{1, .., 4\}$ for the assisted $(j = 0)$ or unassisted $(j = 1)$ portions. *SI Appendix*, Table B.6 reports the results, which are very similar to the previous student-level specification. The problem-level specification is useful in the next section, where we examine how the error rate in solutions generated by GPT Base in the assisted problems affects student performance and learning in the subsequent unassisted exam.

Student Perception. Interestingly, students' own self-reported perceptions of the effects of GPT tutors on their exam performance and learning are overly optimistic. While students in the GPT Base arm performed worse on the exam (relative to the control arm), they did not perceive that they performed worse or learned less. Similarly, while students in the GPT Tutor arm did not perform better on the exam (relative to the control arm), they perceived that they performed significantly better. This mismatch between perceived and actual learning has been observed in other settings (20, 21). Additional details are in *SI Appendix*, Appendix B.3.

Heterogeneity. We look for heterogeneous treatment effects as a function of preregistered variables, capturing students' ability, resources, and effort. In general, we find limited to no statistically significant support for heterogeneous treatment effects with either treatment, particularly with respect to unassisted exam performance. Additional details are in *SI Appendix*, Appendix B.4.

Grade Dispersion. We examine a measure of dispersion in student performance—the Herfindahl–Hirschman index (HHI). Both GPT Base and GPT Tutor reduced grade dispersion in the assisted practice sessions, matching prior findings that generative AI assistance reduces the "skill gap" by providing the largest benefits for the weakest students (7–9). However, we find no significant effect on HHI for the unassisted exam—i.e., the reduction in the skill gap does not persist when access to AI is removed. Additional details are in *SI Appendix*, Appendix B.5.

Robustness Checks. Five class sessions did not use the assigned treatment due to external circumstances (e.g., laptops did not arrive on time). Our primary specifications use an intention-to-treat analysis—i.e., to preserve randomization, we consider all students in a treatment arm as treated, regardless of whether they actually received that treatment. In *SI Appendix*, Appendix B.1, we provide details on noncompliance and perform a regression

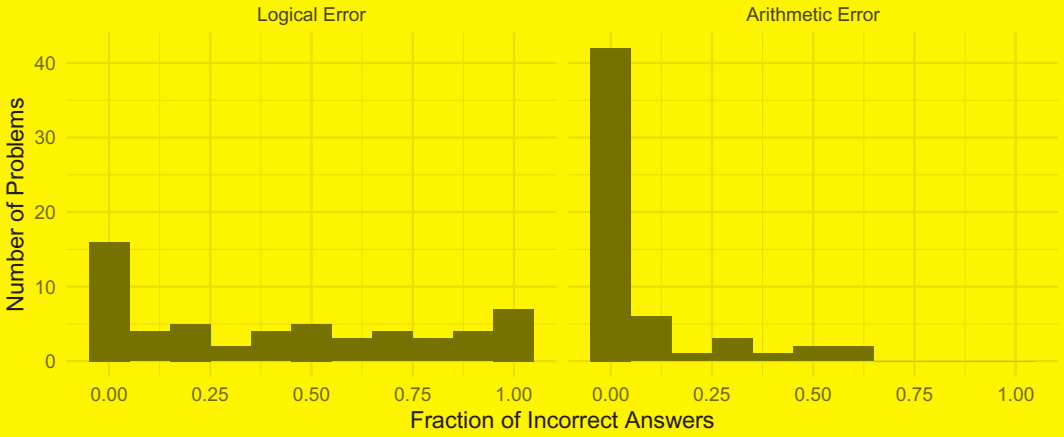

Fig. 2. Fraction of times GPT Base returns an incorrect answer due to a logical (*Left*) or arithmetic (*Right*) error for 57 practice problems across grades and sessions. (Note that GPT Tutor does not return an answer by design, and is further given the correct answer in the prompt).

omitting noncompliers, finding the same insights. Next, we examine several variations to our main specification in Eq. **1** to assess the robustness of our results. These results provide qualitatively similar insights as our main analysis; see *SI Appendix*, Appendix B.2. Last, we check whether differential student absenteeism may impact our results, and find no differential attrition in student attendance across arms or sessions; see *SI Appendix*, Appendix B.6.

Potential Mechanism: Asking for Solutions

In general, students may be adversely affected by GPT Base's assistance in two ways: 1) errors made by GPT Base mislead students in the subsequent unassisted problems, or 2) using GPT Base as a "crutch" prevents them from fully engaging with or understanding the material prior to attempting the unassisted problems. Recall that the design of GPT Tutor avoids both of these issues: 1) it rarely makes mistakes since its prompt includes the solution, and 2) it is hard for students to use it as a crutch since its prompt asks it to avoid giving them the answer and instead guide them in a step-by-step fashion (*SI Appendix*, Fig. A.3). We perform two analyses that help determine which explanation is more likely. First, we analyze how the error rate of GPT Base on a practice problem affects students' subsequent performance on a highly similar exam problem, and second, we analyze student engagement with the tool. Our findings suggest that the second explanation (i.e., students using GPT Base as a crutch) is the main mechanism by which GPT Base impedes student learning.

GPT Errors vs. Student Performance. GPT Base often makes mistakes on math problems (22). We first quantify error rates by repeatedly querying GPT Base using the most common message used by students in the GPT Base arm—i.e., "What is the answer?" For each of the 57 total practice problems, we ask GPT Base for the answer ten times (resetting the system between queries), and then manually categorize any errors in the response as arithmetic (steps followed were correct, but the resulting computation was incorrect) or logical (steps followed were partially or fully incorrect). We find that GPT Base gives a correct answer only 51% of the time on average; it makes logical errors 42% of the time and arithmetic errors 8% of the time. Fig. 2 shows the histogram of how often GPT Base returns an incorrect answer for different problems, demonstrating significant problem-specific heterogeneity in error rates. Details are in *SI Appendix*, Appendix C.1.

We now assess how errors made by GPT Base affect student performance on both the practice problems and the unassisted exam. To this end, we add interaction terms between the error rate of GPT Base and the treatment arm to the problem-level regression specification in Eq. **2**. When assessing exam performance, we leverage our paired design of the session material—i.e., for each exam problem, the teachers included a conceptually similar practice problem to help students learn how to solve that exam problem. Thus, for a given exam problem, we use GPT Base's error rate on the corresponding practice problem. The regression specification is given in *SI Appendix*, Eq. **S5**, and the results are shown in Table 2. Both types of GPT Base errors negatively impact practice problem performance for students in the GPT Base arm (i.e., the coefficients of "GPT Base × Logical Error Rate" and "GPT Base × Arithmetic Error Rate" are statistically significantly negative in the practice performance regression).[§§]

Two key observations support our hypothesis that students are using GPT Base as a crutch. First, if students are being misled by logical errors made by GPT Base, we would expect these errors to affect performance on the corresponding exam problems in the unassisted exam. However, while GPT Base's logical errors affect performance on the practice problems, we find no evidence that this effect spills over to the corresponding exam problems (i.e., "GPT Base × Logical Error Rate" does not have a statistically significant effect on exam performance). An analogous regression on the total error rate (combining both logical and arithmetic errors), yields similar insights (*SI Appendix*, Table C.14). Second, if students were reading and understanding the solutions provided by GPT Base in the practice session, we might expect arithmetic errors to have a smaller impact on practice problem performance than logical errors. This is because students know arithmetic relatively well, and should be better able to catch these errors. However, arithmetic and logical errors appear to have similar effects on practice performance (i.e., "GPT Base × Logical Error Rate" and "GPT Base × Arithmetic Error Rate" have similar coefficients in the practice performance

[§§] Note that we separately control for logical and arithmetic error rates. As expected, the corresponding coefficients in the practice performance regression are both statistically significantly negative, since higher GPT Base error rates are correlated with higher problem difficulty.

Table 2. Regression results on student performance in the practice and corresponding exam problems across grades and sessions; this regression is at the problem level, and includes interaction terms for the logical and arithmetic error rates of GPT Base on practice problems (*SI Appendix*, Eq. S5)

	Dependent variable:	
	Practice perf	Exam perf
GPT base	0.362**	-0.035
	(0.032)	(0.027)
GPT tutor	0.337**	0.029
	(0.037)	(0.023)
Logical error rate	-0.075*	0.178**
	(0.030)	(0.028)
Arithmetic error rate	-0.172*	-0.063
	(0.082)	(0.032)
Prev GPA	0.789**	1.330**
	(0.074)	(0.068)
GPT base × logical error rate	-0.448**	-0.029
	(0.036)	(0.040)
GPT tutor × logical error rate	0.022	-0.086
	(0.038)	(0.044)
GPT base × arithmetic error rate	-0.492**	-0.099
	(0.117)	(0.056)
GPT tutor × arithmetic error rate	0.329**	0.095*
	(0.107)	(0.044)
Observations	13,484	11,392
R^2	0.214	0.212
Adjusted R^2	0.212	0.209

We use a correspondence between the exam and practice problems to estimate how errors on practice problems affect performance on exam problems. Fixed effects are suppressed. Note: HC1 robust SEs clustered by class. *$P < 0.05$; **$P < 0.01$.

regression). Both these results suggest that students are simply copying answers from GPT Base.

Student Engagement. Next, we analyze the messages that students sent to GPT Base or GPT Tutor to better understand how they are interacting with these tools. Fig. 3*A* shows the average number of messages each student had with their respective GPT tool (Base or Tutor) in a given session. As can be seen, the number of messages in GPT Tutor is significantly higher, and further increases with experience using the tool.[¶¶] The fact that students interact substantially less with GPT Base is consistent with our hypothesis that GPT Base simply provides students with solutions.

For a more fine-grained understanding of the content of the student messages, we use natural language processing and clustering to group student messages (see *SI Appendix*, Appendix C.2 for details). We manually associate each cluster with a text description summarizing the content of the messages in that cluster. As shown in *SI Appendix*, Fig. C.4, students in GPT Base most often simply ask for the answer; in contrast, students in GPT Tutor learn to interact more substantively with the tutoring tool over time by asking for help and independently attempting to solve the problem. We observe this learning effect even within the first session—if we restrict to the very first interaction students have with our tool in the first session, 56% in the GPT Base arm and 42% in the GPT Tutor arm[##] either repeat the question text or ask for the answer; in contrast, when considering the first interaction across all problems in the first session, this number increases to 67% for GPT Base but decreases to 37% for GPT Tutor.

We consider clusters where the student simply asks for the answer (specifically, "Repeat Question Text" and "Ask for Answers") to be superficial, and the remaining clusters (specifically, "Attempted Answers" and "Ask for Help") to be nonsuperficial. For a given student in a given session, we consider the corresponding conversation superficial if the student asked any superficial messages, and nonsuperficial otherwise. Intuitively, nonsuperficial conversations are ones where the student constructively interacts with the tutoring tool and never asks for the answer. Fig. 3*B* shows the aggregate rate of nonsuperficial conversations for both GPT Base and GPT Tutor. As can be seen, across all sessions, in the GPT Base arm, only a small fraction of conversations are nonsuperficial; in contrast, a substantially larger fraction of conversations in the GPT Tutor arm were nonsuperficial. These results suggest that the vast majority of students are using GPT Base to obtain solutions, whereas a significant fraction of students are using GPT Tutor in a purely substantive way.

Discussion

Our results provide a cautionary tale regarding both the existing use of the freely available ChatGPT tool by students as well as the potential deployment of GPT-based tutors in educational settings. While generative AI tools such as ChatGPT can make tasks significantly easier for humans, they come with the risk of deteriorating our ability to effectively learn some of the skills required to solve these tasks. These shortcomings have been anecdotally reported for tools such as Khanmigo (23), a GPT-4 based tutoring application. Our findings support both the need for educators to find ways to safeguard student learning in the face of freely available tools such as ChatGPT as well as the need to design more effective guardrails for generative AI tutors.

In some ways, ChatGPT is not the first technology to exhibit this tradeoff—for instance, typing diminishes the need for handwriting, and calculators diminish the need for arithmetic, etc. However, we believe ChatGPT differs from prior technologies in two significant ways. First, the capabilities of ChatGPT are substantially broader and more intellectual compared to prior examples; for instance, our experiments focus on a broad variety of mathematical topics, which encompass fundamental skills required by a wide range of knowledge-intensive professions. Second, unlike many prior technologies, ChatGPT is highly unreliable and often provides incorrect responses. Our results suggest that students are either unable to detect these failures or unwilling to spend the effort needed to check correctness.

While the guardrails implemented in GPT Tutor appear to largely mitigate these negative effects, substantial work is required to enable generative AI to positively enhance rather than

[¶¶] We also find that students spend more time with GPT Tutor than with GPT Base; see *SI Appendix*, Appendix C.3.

[##] The initial discrepancy may be due to the fact that GPT Tutor, unlike GPT Base, offers suggested prompts; see *SI Appendix*, Fig. A.2*B* for details.

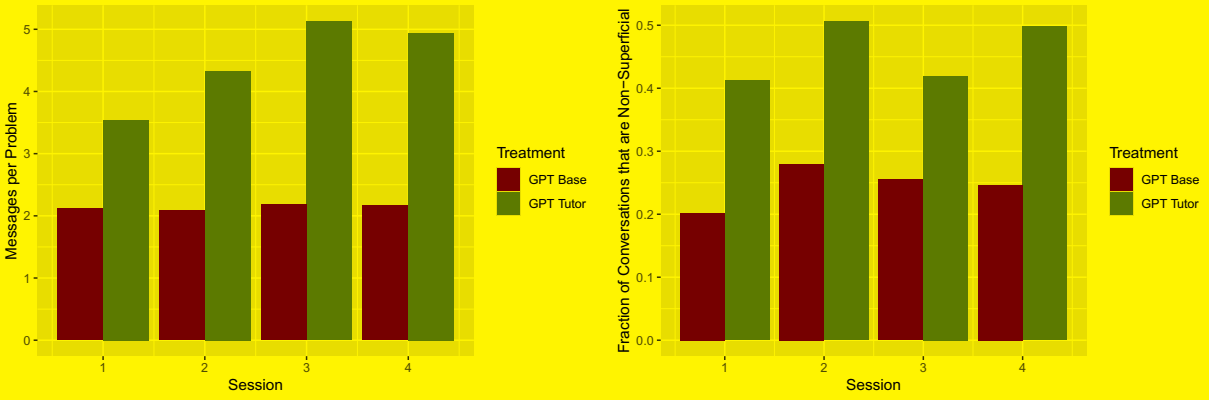

Fig. 3. Student engagement—given by (*A*) average number of student messages per problem, and (*B*) average fraction of student session conversations that have no superficial messages (simply restating the question or asking for the answer) per session—by treatment (GPT Base and GPT Tutor) over time.

diminish education. GPT Tutor remains passive, responding to students when they ask questions but failing to proactively engage students with the material. Effective human tutors ask probing questions to uncover student misconceptions, and then clarify these misconceptions by providing tailored explanations. Combining existing software tutors (15, 24) with generative AI may be a promising path to achieving this goal, since it balances the pedagogical principles baked into existing algorithms with the capability of generative AI to understand and respond to complex student queries. One way to do so is to leverage agent approaches (25, 26), which compose models with different prompts to achieve complex goals; for instance, we might use one prompt to ask the model to identify student misconceptions, and then use another prompt to ask it to generate a useful hint. Beyond designing better tutors, promising avenues include educating students and teachers about how to more effectively use generative AI, and leveraging generative AI to help teachers instead of focusing on students (13). For instance, recent evidence suggests that "co-pilots" that work with a human tutor instead of replacing them might be a more promising path toward effective use of generative AI in education (27).

Finally, while our study takes a step toward understanding the potential harms of generative AI on learning, it focuses on a specific deployment context, and substantial work is needed to better understand the nature and scope of these effects. First, our study focuses on two generative AI tutors for a single topic (mathematics) deployed in a single high school in Turkey. We have objective evaluation criteria for math problems, which is unavailable in other important subjects such as writing. Our deployment was also carried out in Fall 2023, when generative AI was still very new; modes of interaction may have changed now that users are more familiar with effective uses of these tools and their shortcomings. Furthermore, both closed and open-weight models have significantly improved in performance since then. Additional studies are required to assess generalizability to other tutor designs and deployment contexts. Moreover, we focus on short-term outcomes due to limitations imposed by our partner school; studying long-term outcomes is a key direction for future research. Last, while we have performed several analyses to uncover mechanisms underlying our findings, our approach can be complemented with controlled experiments in settings where more fine-grained interventions and monitoring are possible to better understand these mechanisms.

Data, Materials, and Software Availability. To support further research, we have made anonymized data available at https://github.com/obastani/GenAICanHarmLearning (28). Data include student performance on practice and exam questions for all sessions, as well as student-, class-, and grader-level covariates; they also include anonymized and time-stamped student conversations with our GPT tutors. All code used in this paper was written in a combination of R, Python, and Stata and can be found in the same repository.

ACKNOWLEDGMENTS. We acknowledge invaluable research partnership with Tamer Atacan and TED Ankara Koleji, as well as research assistance from Tuba Tas and Dogus Berk Kocak. We are grateful for funding from the Wharton AI & Analytics Initiative, the Fishman-Davidson Center, and Wharton Global Initiatives. We are also grateful for helpful feedback from Eric Bradlow, Angela Duckworth, Benjamin Luttges, Lilach Mollick, Ananya Sen, Christian Terwiesch, Lyle Ungar, as well as seminar participants at the London School of Economics, Indian School of Business, Google, Microsoft, Workshop on Unstructured Data and Language Models, Conference on Digital Experimentation, and Workshop on AI & Analytics for Social Good.

Author affiliations: [a]Department of Operations, Information, and Decisions, Wharton School, University of Pennsylvania, Philadelphia, PA 19104; [b]Wharton AI & Analytics, Philadelphia, PA 19104; [c]Department of Computer and Information Science, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA 19104; [d]Department of Mathematics, Budapest British International School, Budapest 1125, Hungary; and [e]Independent, Philadelphia, PA 19104

1. J. Achiam *et al.*, Gpt-4 technical report. arXiv [Preprint] (2023). https://arxiv.org/abs/2303.08774 (Accessed 5 March 2025).
2. K. Singhal *et al.*, Large language models encode clinical knowledge. *Nature* **620**, 172–180 (2023).
3. Y. Li *et al.*, Competition-level code generation with AlphaCode. *Science* **378**, 1092–1097 (2022).
4. W. Geerling, G. D. Mateer, J. Wooten, N. Damodaran, ChatGPT has aced the test of understanding in college economics: Now what? *Am. Econ.* **68**, 233–245 (2023).
5. C. Terwiesch, "Would chat GPT get a Wharton MBA" in *Mack Institute White Paper* (Wharton School, University of Pennsylvania, 2023).
6. T. Eloundou, S. Manning, P. Mishkin, D. Rock, GPTs are GPTs: Labor market impact potential of LLMs. *Science* **384**, 1306–1308 (2024).
7. S. Noy, W. Zhang, Experimental evidence on the productivity effects of generative artificial intelligence. *Science* **381**, 187–192 (2023).
8. E. Brynjolfsson, D. Li, L. R. Raymond, "Generative AI at work" in *National Bureau of Economic Research Working Paper* (2023).
9. F. Dell'Acqua *et al.*, "Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality" in *Harvard Business School Technology & Operations Management Unit Working Paper* (2023).
10. G. S. Becker, *Human Capital: A Theoretical and Empirical Analysis, with Special Reference to Education* (University of Chicago press, 2009).
11. Safety alert for operators: Manual flight operations (Tech. Rep. SAFO 13002, Federal Aviation Administration, 2013).

12. J. Maynez, S. Narayan, B. Bohnet, R. McDonald, "On faithfulness and factuality in abstractive summarization" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, J. Tetreault, Eds. (Association for Computational Linguistics, 2020), pp. 1906–1919.
13. E. R. Mollick, L. Mollick, New modes of learning enabled by AI chatbots: Three methods and assignments. http://dx.doi.org/10.2139/ssrn.4300783. Accessed 5 March 2025.
14. A. Extance, ChatGPT has entered the classroom: How LLMs could transform education. *Nature* **623**, 474–477 (2023).
15. S. Ruan *et al*., "Supporting children's math learning with feedback-augmented narrative technology" in *Proceedings of the Interaction Design and Children Conference* (Association for Computing Machinery, New York, NY, 2020), pp. 567–580.
16. S. Ruan *et al*., Reinforcement learning tutor better supported lower performers in a math task. *Mach. Learn.* **113**, 3023–3048 (2024).
17. J. Cristia, P. Ibarrarán, S. Cueto, A. Santiago, E. Severín, Technology and child development: Evidence from the one laptop per child program. *Am. Econ. J. Appl. Econ.* **9**, 295–320 (2017).
18. A. J. Lamb, J. M. Weiner, Institutional factors in iPad rollout, adoption, and implementation: Isomorphism and the case of the Los Angeles unified school district's iPad initiative. *Int. J. Educ. Math. Sci. Technol.* **6**, 136–154 (2018).
19. M. Escueta, A. J. Nickow, P. Oreopoulos, V. Quan, Upgrading education with technology: Insights from experimental research. *J. Econ. Lit.* **58**, 897–996 (2020).
20. E. L. Bjork, R. A. Bjork, "Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning" in *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society*, M. A. Gernsbacher, R. W. Pew, L. M. Hough, J. R. Pomerantz, Eds. (Worth Publishers, 2011), pp. 56–64.
21. L. Deslauriers, L. S. McCarty, K. Miller, K. Callaghan, G. Kestin, Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 19251–19257 (2019).
22. S. Frieder *et al*., Mathematical capabilities of chatGPT. *Adv. Neural Inf. Process. Syst.* **36**, 13867 (2024).
23. N. Singer, In classrooms, teachers put A.I. tutoring bots to the test. *N. Y. Times* (2023).
24. S. Doroudi, V. Aleven, E. Brunskill, Where's the reward? A review of reinforcement learning for instructional sequencing *Int. J. Artif. Intell. Educ.* **29**, 568–620 (2019).
25. S. Yao *et al*., "React: Synergizing reasoning and acting in language models" in *The Eleventh International Conference on Learning Representations (ICLR)* (2023).
26. X. Wang *et al*., "Executable code actions elicit better LLM agents" in *Forty-First International Conference on Machine Learning* (JMLR.org, Vienna, Austria, 2024).
27. R. E. Wang, A. T. Ribeiro, C. D. Robinson, S. Loeb, D. Demszky, Tutor copilot: A human-AI approach for scaling real-time expertise. arXiv [Preprint] (2024). https://arxiv.org/abs/2410.03017 (Accessed 5 March 2025).
28. H. Bastani et al., Data and Code for "Generative AI without guardrails can harm learning: Evidence from high school mathematics." GitHub. https://github.com/obastani/GenAICanHarmLearning. Deposited 11 June 2025.