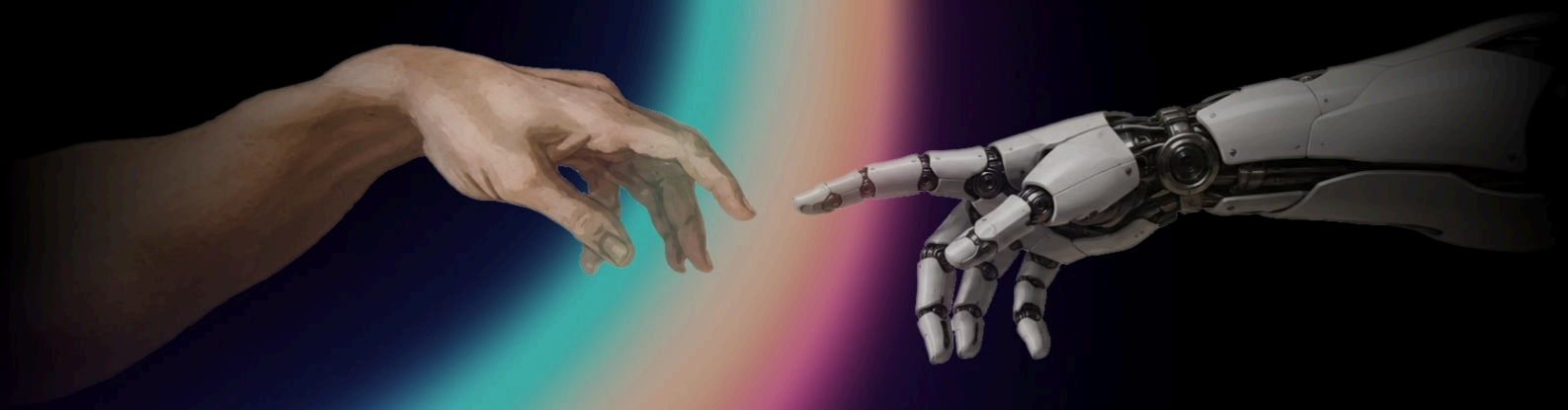
cogniloop

Aporia

Why learning starts where your reasoning breaks

November 2026 · Working paper



cogniloop.app

cogniloop

Working paper

Aporia: why learning starts where your reasoning breaks

Draft 1. The design, the reasoning and the engine's records are real. Real data comes from one learner, the founder, and every projection and illustration is labelled as one.

Author	Cogniloop
Date	November 2026
Version	Draft 1

Contents

01 Executive summary
02 Introduction
03 Aporia: where learning starts
04 What AI changes
05 An engine under the tutor
06 Reading the work, line by line
07 What the engine knows about you
08 Remembering you
09 Deciding what comes next
10 The engine checks itself
11 What would prove this wrong
12 Who it is for
13 Limitations
14 Conclusion

Correspondence: hello@cogniloop.app

01 Executive summary

Most people believe AI is bad for learning. They are right about how it is mostly used. A tool that hands over the answer in seconds meets a student who would rather not spend hours, and the student takes the answer. Studying feels like understanding, and the exam shows that it was not. A large field experiment measured exactly this: students who practised maths with plain GPT-4 scored 48 per cent higher in practice and 17 per cent lower on the exam that followed.[1]

I think the problem is how AI is applied, not AI itself. Learning starts at the moment your own reasoning breaks: the impasse that Socrates' questions led to, which the Greeks called *aporia*. A tool that removes that moment removes the learning with it. A tool that finds it, every time, could make study shorter and make it hold.

+48% → −17%

practice scores, then exam scores, of students who practised with plain GPT-4

PUBLISHED STUDY

67% → 33%

share of problems the engine expected me to solve alone, against what I actually solved alone

ONE LEARNER, THE FOUNDER

0.79

average effect of one-to-one human tutoring, in standard deviations, in controlled studies

PUBLISHED REVIEW

Cogniloop is built on that idea. You try first. Under the tutor, an engine reads your working line by line and finds where it stopped holding. It keeps what you can do alone apart from what you did with help, schedules each topic to come back before it is likely to fade, and remembers you across weeks. It also checks its own predictions about you, so it finds out when it is wrong.

Its first self-check did find it wrong. On my own study, it expected me to solve two-thirds of my problems alone, and I solved one-third. It had started new topics too high, and it had counted help as progress: the same illusion it was built to break. Both were corrected.

Nothing in this paper shows that Cogniloop improves learning. It sets out the test that would: whether learners who use it can solve more on their own a week and a month later than learners who use a chatbot or study the usual way. If they cannot, the approach is wrong, and I will say so.

02 Introduction

I have never liked studying. It takes a long time, and much of it does not stick. Chatbots made it worse for me: they make it feel as if you studied, and then you fail the next exam. Personal tutoring is widely known to work, but most students cannot afford a tutor or find a good one.

Many people see AI as bad for learning, and they have reasons to. A student with a problem set and a tool that can produce the answer in seconds will, sooner or later, take the answer. That is not a flaw of character. Studying is long and tedious, and the incentive points one way. The result is a familiar pattern: the student feels they understand, then fails the exam.

The problem, then, is not AI. It is how AI is applied. A tool that gives students what they ask for usually gives them the answer, and the learning that holds up on exams and later in life does not come with it. Holding the answer back helps, as Section 4 shows, but it is only the start. Most AI study tools stop at rules written over a general model: do not give the answer, ask a question instead. Cogniloop adds an engine underneath. It builds a model of the learner from their own work, checked step by step. It remembers them across weeks. And it tests its own predictions about them, so that it can find out when it is wrong.

I avoided my own app at first, because it felt tedious. After fixing that, I used Cogniloop to study for a real exam, starting three or four days before it. The difference was that it kept making me find where my reasoning broke, instead of letting me feel that I understood. That is my experience, and I built the thing, so it is the reason for this paper, not evidence for its claims.

What this paper does

Section 3 sets out the idea behind the design and what research on testing, failed attempts and productive failure says about it. Section 4 looks at what AI changes, for better and worse. Sections 5 to 9 describe the engine: how it reads work, what it knows about a learner, how it remembers them and how it decides what comes next. Section 10 reports the engine's first check on its own predictions, run on my own study, and what it got wrong. Section 11 sets out the test that would prove the approach wrong. Sections 12 and 13 cover who it is for and its limits.

The aim is practical: studying should take less time and hold up better. The design goal is a focused session of one to two hours that does real work, not a day at the desk. That is a goal, not a result.

03 Aporia: where learning starts

If you cannot reason through why something works, you know only how it looks. You have learned to recognise the problem, not to solve it. The moment your reasoning breaks shows exactly what is missing, and that is where learning starts.

In Plato's dialogues, Socrates questions people who are sure they know something, such as what courage or virtue is, until their own answers contradict each other. The method is called the *elenchus*, and it usually ends in *aporia*, literally "no way through": an impasse,[2] where the speaker sees that their own reasoning has failed. In the *Meno*, the young Meno complains that talking to Socrates numbs him, like the shock of an electric ray:

> "You seem to me … to be very like the flat torpedo fish, who torpifies those who come near him …"
>
> Plato, *Meno* 80a, translated by Benjamin Jowett[3]

Socrates then questions a slave boy about a geometry problem. The boy answers confidently and is wrong, then wrong again, and finally admits that he does not know. Socrates asks Meno: "Is he not better off in knowing his ignorance?" And has giving him "the torpedo's shock" done him any harm? Meno agrees that it has not.[3] Only after the impasse does the boy find the answer, by reasoning it through himself.

That method is the elenchus, and Cogniloop is built on it; this paper is named after its result. The claim is not that confusion is good. It is that the precise point where a learner's own reasoning stops holding is the most useful thing to know about them. That is where they are missing something, and it is only visible if they try first.

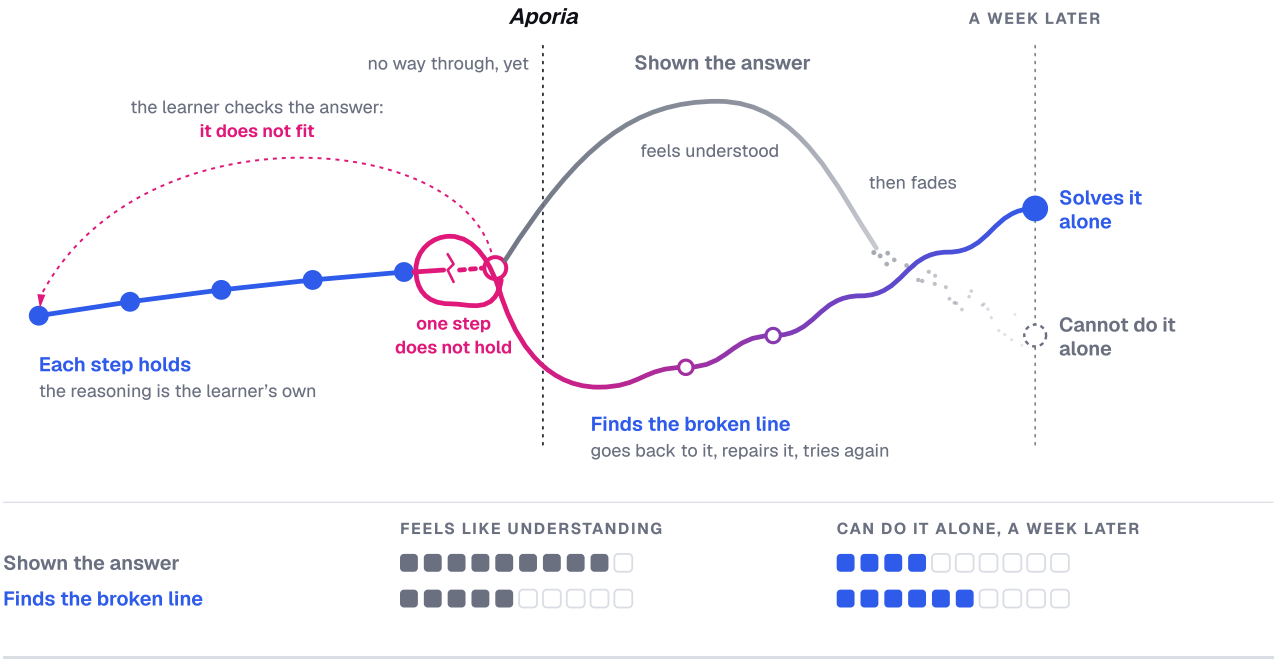


Figure 1. Being shown the answer feels like understanding; finding the broken line is what lasts. **Illustration.** A line of reasoning holds until one step does not, and the learner's own check fails: the impasse. Being shown the answer feels sure and then fades. Going back to the broken line, repairing it and trying again ends in a problem the learner can solve alone a week later. The counts below are illustrative.

Recognising is not knowing

A learner who rereads a worked solution can follow every step, and following feels like understanding. But following a solution and producing one are different skills. With the solution in front of you, the cue for each step is there too, and you cannot tell which steps you could have taken without it. Koriat and Bjork call this an illusion of competence: learners judge what they know while the answer is present, but at the test it is absent.[4]

Learners mostly do not know this, and they study accordingly. In a survey of 177 college students, 84 per cent listed rereading as a way they study, and only 11 per cent listed testing themselves.[5] In one study, even after their own test results favoured spaced study, participants rated massed study as more effective.[6] Learners tend to prefer the methods that feel productive over the ones that are.[7]

The test is the lesson

The most robust finding in this field is that trying to recall something strengthens memory more than studying it again. In a classic study, students read a short passage and then either reread it or tried to recall it, with no feedback.[8] Five minutes later, rereading had won. Two days and a week later, the order had reversed. Across 159 effects from 61 studies, testing beat restudying by an average of half a standard deviation,[9] and a major review of ten study techniques rated practice testing and spaced practice the two most useful.[10]

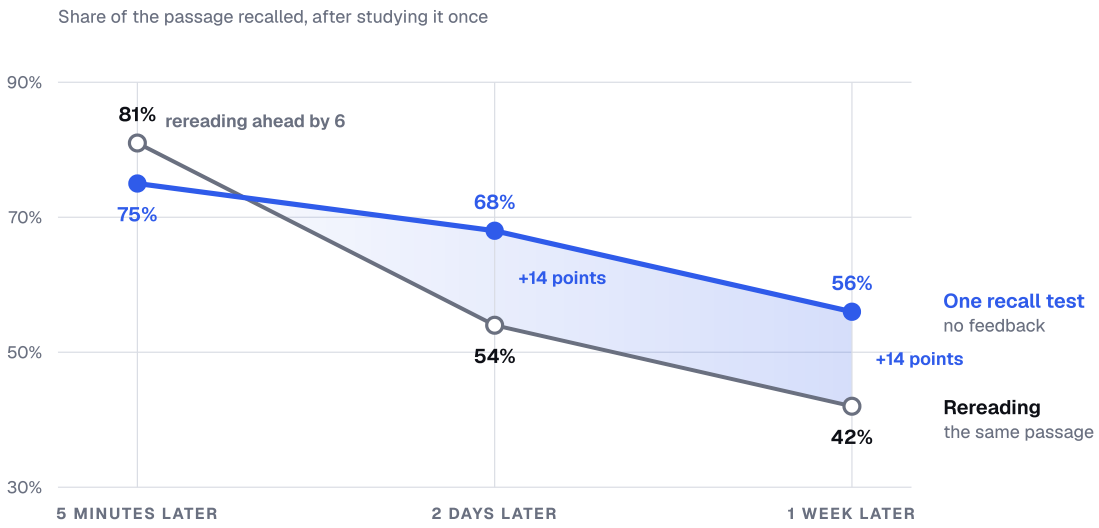


Figure 2. Rereading wins the first few minutes; one recall test wins after that. **Published data.** 120 students studied a prose passage once, then either reread it or took one recall test with no feedback. The three time points are evenly spaced, not drawn to scale.

Source: Roediger and Karpicke (2006), Experiment 1.

Failing first

The benefit does not need the attempt to succeed. When students tried to answer questions they could not possibly know before being shown the answer, they remembered those questions better than questions they had only studied.[11] Students who tried, and almost always failed, to answer questions before reading a text scored 75 per cent on those questions afterwards, against 56 per cent for students who spent the time studying instead.[12]

The same holds for whole problems. In *productive failure*, learners first attempt a problem they have not been taught to solve, and only then receive instruction.[13,14] They usually fail to find the standard solution, but the attempt prepares them to understand it. A review of 53 studies found that problem solving followed by instruction beat the reverse, with larger effects when the design was followed closely, though not for the youngest learners.[15] Bjork's name for all of this is *desirable difficulties*: conditions that make study feel harder often make learning last longer.[16] Aporia is the sharpest form of it.

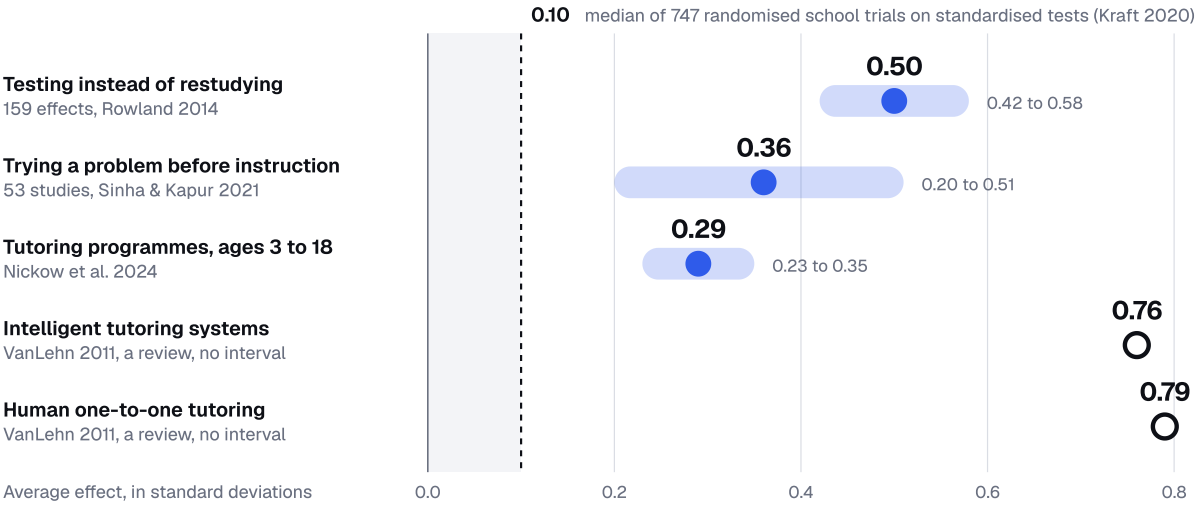


Figure 3. Trying first, testing and tutoring each show clear positive average effects. **Published data.** Average effects from published reviews, in standard deviations, with 95% intervals where the review reports one (about 0.23 to 0.35 for tutoring programmes, from the reported standard error). The comparisons differ, so the effects cannot be added. The dashed line is the median effect of 747 randomised school trials on broad standardised tests, which is not like for like: most of the effects above come from narrower measures.

Sources: Rowland (2014); Sinha and Kapur (2021); Nickow, Oreopoulos and Quan (2024); VanLehn (2011); Kraft (2020).

Forgetting, and coming back in time

Memory fades on a curve, fast at first and then more slowly. Ebbinghaus measured it on himself and published it in 1885, and a replication 130 years later found much the same shape.[17] Spacing study out slows the fade: across 271 comparisons, spaced study beat massed study in all but 12.[18] The best gap depends on how long the memory must last, shrinking as a share of the delay from about a third for a test a week away to under a tenth for a test a year away.[19] Nobody can track that for every topic by hand. An engine can.

04 What AI changes

AI can give every student an answer in seconds, or something close to a tutor. The evidence says both are possible, and that the difference lies in how it is built.

The clearest evidence of harm comes from about a thousand high-school students in Turkey, randomly assigned to practise maths with plain GPT-4, with a guarded GPT-4 tutor that gave hints instead of answers, or with only their textbook and notes.[1] Plain GPT-4 lifted practice scores and lowered exam scores. The guarded tutor lifted practice even more and left exam scores where they were. Neither group noticed: students who used plain GPT-4 did not think they had done worse, and students with the guarded tutor believed they had done better. That is the illusion of competence, produced at scale.

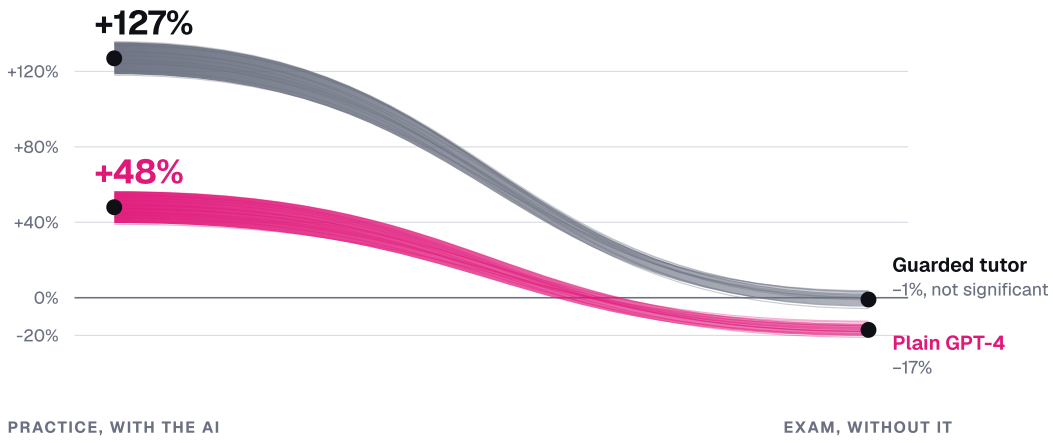


Figure 4. Help that lifts practice can lower what students can do alone. **Published data.** Practice and exam scores relative to students who used only a textbook and notes, in four 90-minute maths sessions with about 1,000 students. The guarded tutor gave hints instead of answers. Each ribbon joins two measurements; the path between them is drawn, not measured.

Source: Bastani et al. (2025).

The same technology, built differently, can do the opposite. In a randomised crossover study in a Harvard physics course, an AI tutor designed around how people learn, which scaffolded each step and did not hand over answers, taught a lesson better than in-class active learning, in less time.[20] The authors tie the result to new material and an expert-built tutor, and the study covered one course, testing soon after each lesson. A smaller laboratory study raises a related concern: students who wrote with ChatGPT improved their essays more than others but did not gain more knowledge.[21]

Tutoring works; the question is which parts

Tutoring is one of the best-supported interventions in education. Randomised trials of tutoring programmes for children find large average effects by the field's standards.[22] The famous claim that tutoring lifts the average student by two standard deviations[23] does not hold up so well: it rested on two small studies in which tutored students also received extra testing and feedback, and it has not been replicated.[24] A more useful finding for software is VanLehn's review: human tutors achieved about 0.79

standard deviations, and computer tutors that respond to each step of the learner's solution came close, at 0.76.[25]

What a rule in a prompt cannot do

General assistants have added study modes, built on instructions to the same general model.[26,27] They are real improvements, and the evidence suggests guardrails like these prevent the worst harm. But in my own use, a rule written over a general model has four limits that no wording fixes.

- **It lacks authority.** Told never to give the answer, a general model usually gave it to me within a message or two once I insisted. It is built to be helpful to everyone, not to tutor.
- **It does not know the learner.** Chat assistants keep some memory, a few saved notes, but not a model of what this learner can do alone, built from their checked work over weeks.
- **It does not check its own arithmetic.** A model judging a line of working can be wrong about the line, and nothing in a prompt guarantees that it recomputes before it answers.
- **It cannot tell whether it is right about the learner.** It makes no predictions it can be held to, so it never finds out when its picture of them is wrong.

The guarded tutor in Bastani's study prevented harm but produced no gain. Holding back the answer removes the damage. It does not, by itself, supply what makes tutoring work: reading the work step by step, keeping track of what the learner can do alone, and bringing it back before it fades.

05 An engine under the tutor

Cogniloop is not a better prompt. A language model can say almost anything; the hard part of tutoring is choosing the one thing worth saying to this learner now, and holding to it. So the work is split. The engine decides, and the model writes.

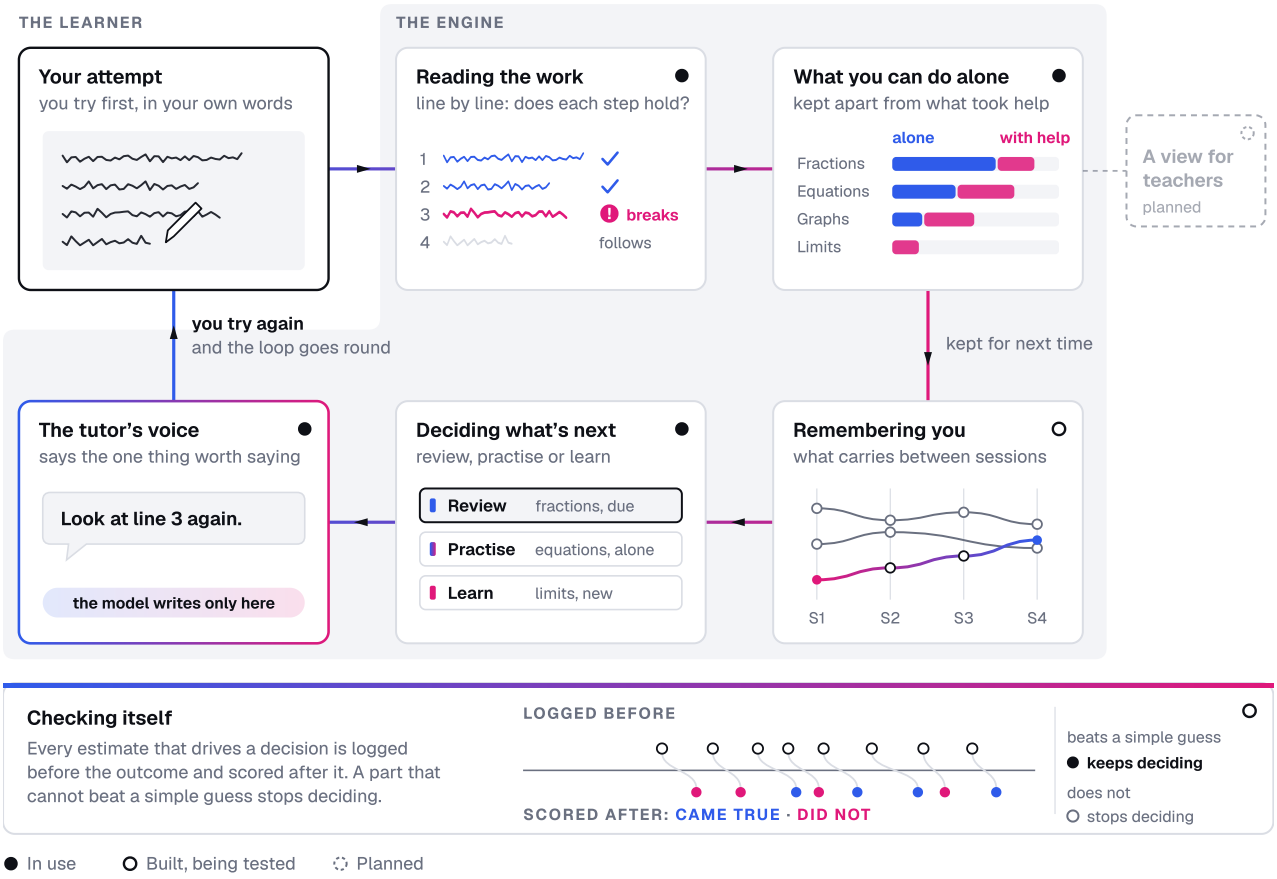


Figure 5. The learner tries first; the engine reads, counts, remembers and decides, and the model speaks only at the end. **Engine design.** Each box shows a small picture of what happens inside it, and the loop returns to the learner, who tries again. Underneath, every estimate that drives a decision is logged before the outcome so it can be scored after it; the scoring is run by hand today.

The engine records what the learner does, reads it, estimates what they can do alone, remembers them, and decides the next move. The tutor then words that move. Underneath, every estimate that drives a decision is logged so it can be scored against what happened, and a part of the engine that guesses badly can be found and switched off. Today that scoring and switching is done by hand.

One rule: you try first

Cogniloop has one rule that a learner meets every time: you try first. The rule is not there to make studying harder. It exists because the attempt is how the engine learns about the learner, and because the attempt is where aporia happens. A learner can still get the answer, by asking for it, and the engine

then knows that the solve was not their own. Everything else should be easier than a chat assistant: less typing, less waiting, less explaining what you already said.

The rule is held by the application, not by the model. I learned this the hard way: a different model followed the same instructions very differently, and its reading of "hold back" stopped it helping at all. Behaviour that matters belongs in the application, where it does not change when the model does.

What is built

Cogniloop is in development, and some of its parts are switched off until they are tested. Reading the work, the help ladder, the estimate of what a learner can do alone, spaced reviews, the exam plan and the prediction log are in use. The memory of the learner across sessions and the adaptive signals are built and being tested. A view for teachers is planned. The engine has been used and evaluated on mathematics so far, in English and Swedish.

06 Reading the work, line by line

A final answer says whether a learner got there. The working says where they stopped being right. Cogniloop reads the working and finds the first line that does not hold.

It records where the attempt broke, and whether the mistake was a **slip**, a right method carried out wrongly, or a **gap**, a missing or wrong idea. The two are treated differently from then on, because they need different help.

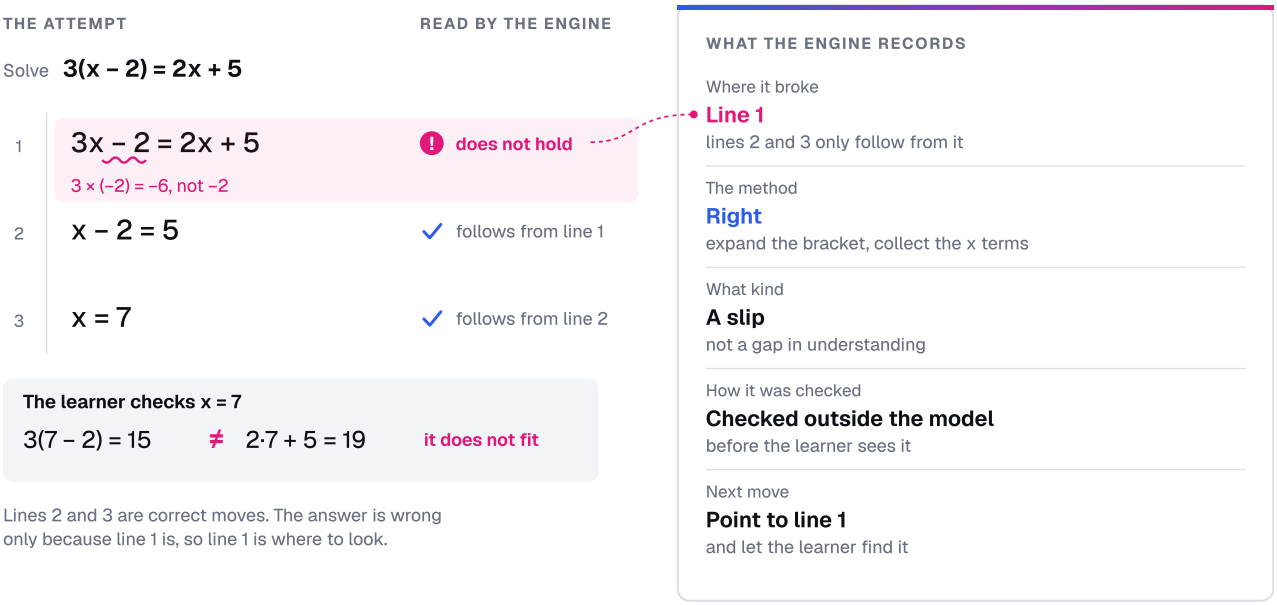


Figure 6. The engine records where an attempt broke, not just that the answer is wrong. **Engine design.** An invented attempt. Line 1 does not hold; lines 2 and 3 are correct moves that follow from it, so the answer is wrong only because line 1 is. The verdict is checked outside the model, and the next move points the learner to the line instead of giving the answer.

The model does not do the arithmetic

Language models make arithmetic mistakes, and a tutor that tells a learner a right line is wrong does more harm than one that says nothing. So verdicts are checked outside the model before a learner sees them. When the engine is not sure, it says so, and an uncertain verdict never counts against the learner. Where learners have uploaded their own course material, it is used to follow the course's own notation and definitions.

07 What the engine knows about you

For each topic, the engine keeps an estimate of what you can solve on your own. Help is recorded, and it counts for much less than a solve without it.

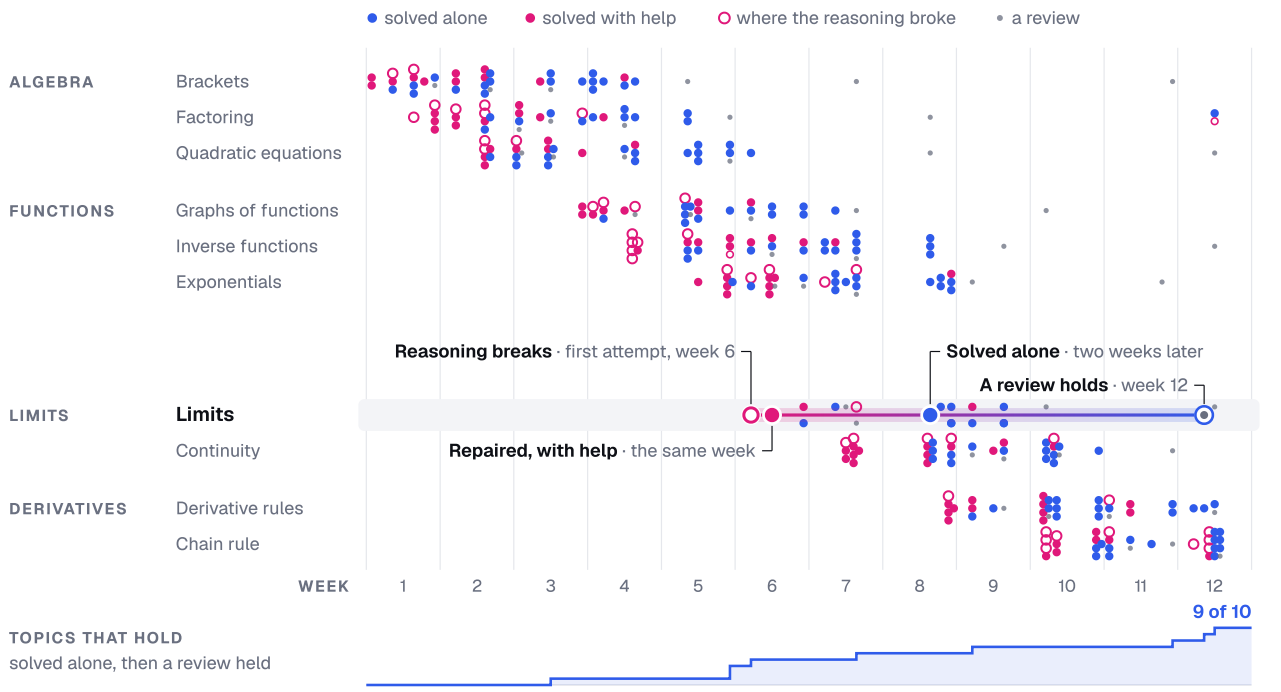


Figure 7. Learning shows up first as broken reasoning, and later as work that holds without help. **Illustration.** Each dot is one attempt by an invented learner over a 12-week term: blue solved alone, pink solved with help, a pink ring where the reasoning broke, a small grey dot a review. One topic is traced from its first break to a review that still holds a month later.

The estimate updates with every attempt, and the next problem is pitched at the edge of what you can do: hard enough for the attempt to teach something, not so hard that every attempt fails. Repeated failure sends the learner back to what the topic builds on.

Help is not learning yet

Cogniloop offers help as a graded ladder, from a light hint up to the answer itself, and the learner climbs only as far as they need. The help used is recorded against the problem, which is why the engine can tell "solved" from "solved alone" at all. A solve after help counts for much less than a solve without it, and less again the stronger the help.

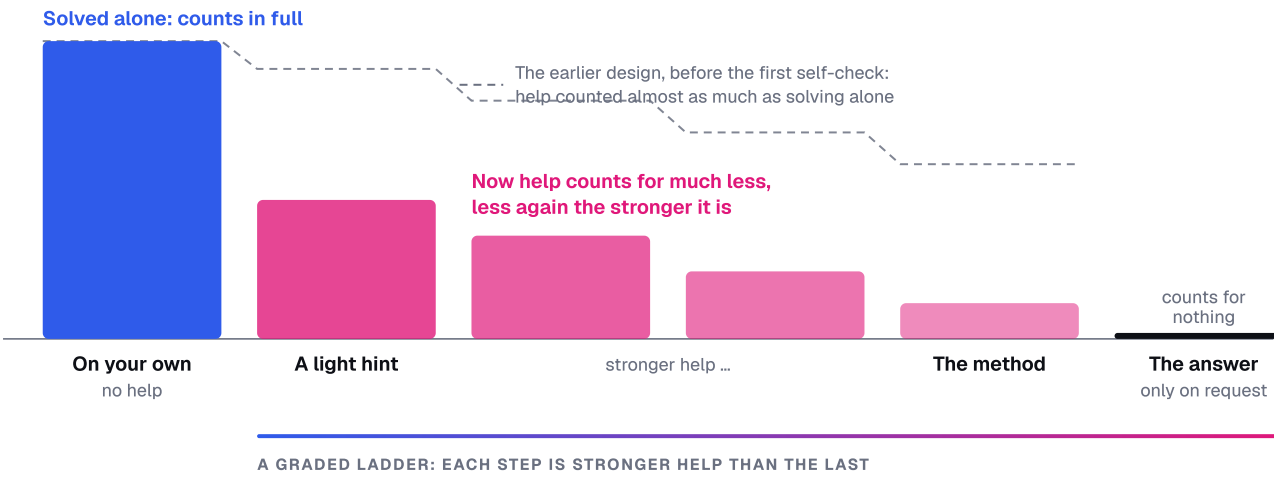


Figure 8. The stronger the help, the less a solve counts towards what you can do alone. Engine design. A graded ladder of help, from a light hint to the answer itself, which is given only on request and then counts for nothing. Bar heights show the shape only. The dashed steps are the earlier design, before the first self-check.

Forgetting on schedule

Knowing something today does not mean knowing it next month. Cogniloop schedules reviews with FSRS, an open model of memory that grew out of work on spaced-repetition scheduling[28] and is used by the flashcard program Anki.[29] It schedules each review for before a topic is likely to slip, and each successful review makes the next gap longer. How a review went is read from the work, not from a button the learner presses.

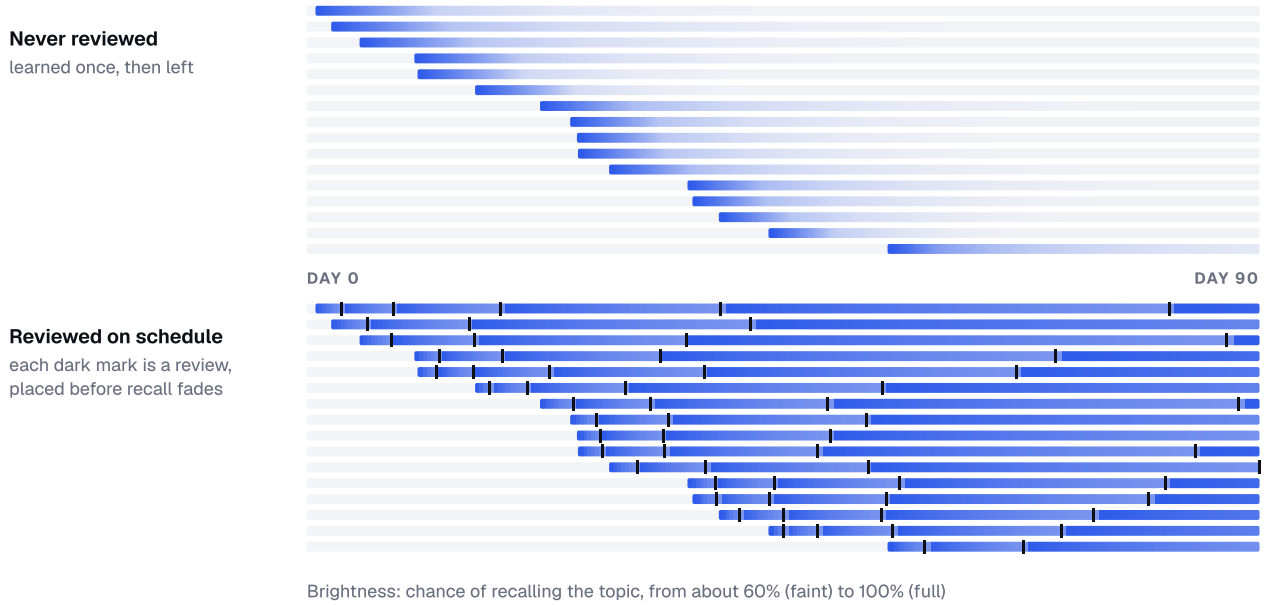


Figure 9. The same sixteen topics, left to fade or caught just in time. **Illustration.** Each row is one topic, from the day it was learned; brightness is the chance of recalling it. Below, a review comes before that chance fades too far, and each review makes the next one later. The curve is the published FSRS forgetting curve; the topics are invented.

Formula: FSRS, open-spaced-repetition.

08 Remembering you

A chat assistant meets you new in every conversation. A tutor should remember what you said last week, what you have already shown you can do and what is still shaky, and should never ask you again.

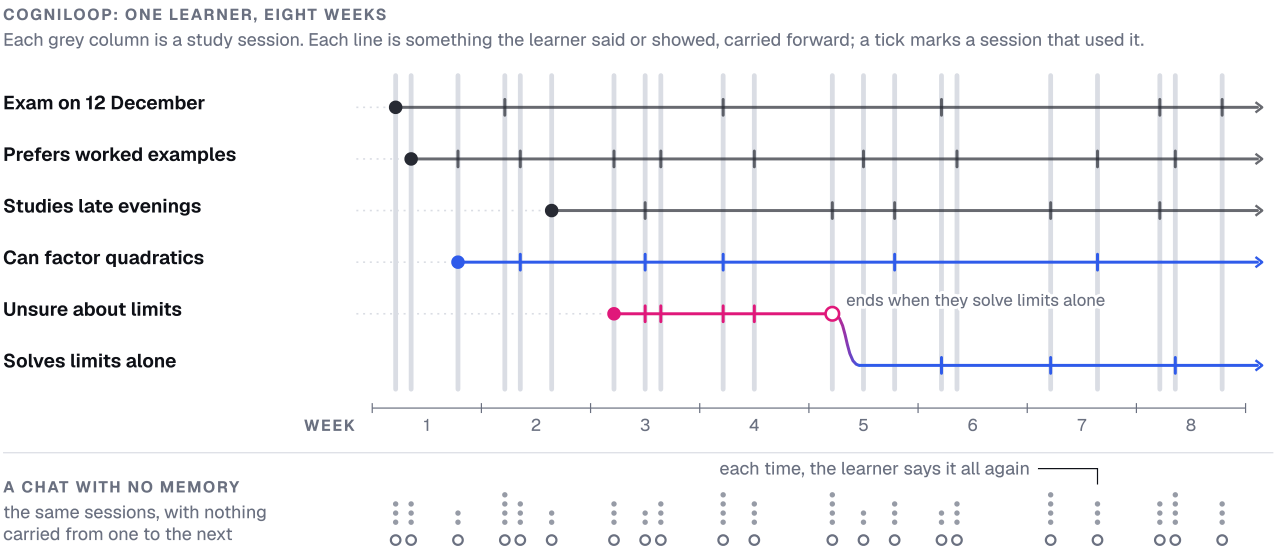


Figure 10. What a learner says or shows is carried forward, and checked work can change it. Illustration. Eight weeks of an invented learner's sessions. Each line is something they said or showed, used again in later sessions; "unsure about limits" ends when they solve limits alone. Below, a chat with no memory carries nothing between the same sessions.

The tutor remembers what you have said and shown, so it does not ask again. Memory never decides a grade: a verdict comes from the work, and checked work always overrides what the tutor remembers.

Learners can see and correct what the engine holds about them, export their data and delete their account. This memory is built and being tested. On replays of my own sessions the tutor re-asked less, which is a measure of manners, not of learning.

09 Deciding what comes next

Much of the time students spend studying goes on things other than learning: choosing what to do, redoing what they already know, avoiding what they do not. The engine makes that choice, aiming to let a session be short and still count.

Cogniloop has three ways of studying. **Learn** teaches a topic like a concise textbook, in parts, checking between them that the learner is still with it, without grading. **Practice** gives problems, reads the working and offers the help ladder. **Revise** brings back what is due and what the next exam covers. Each day starts with anything left unfinished, then the reviews that are due, then the next topic.

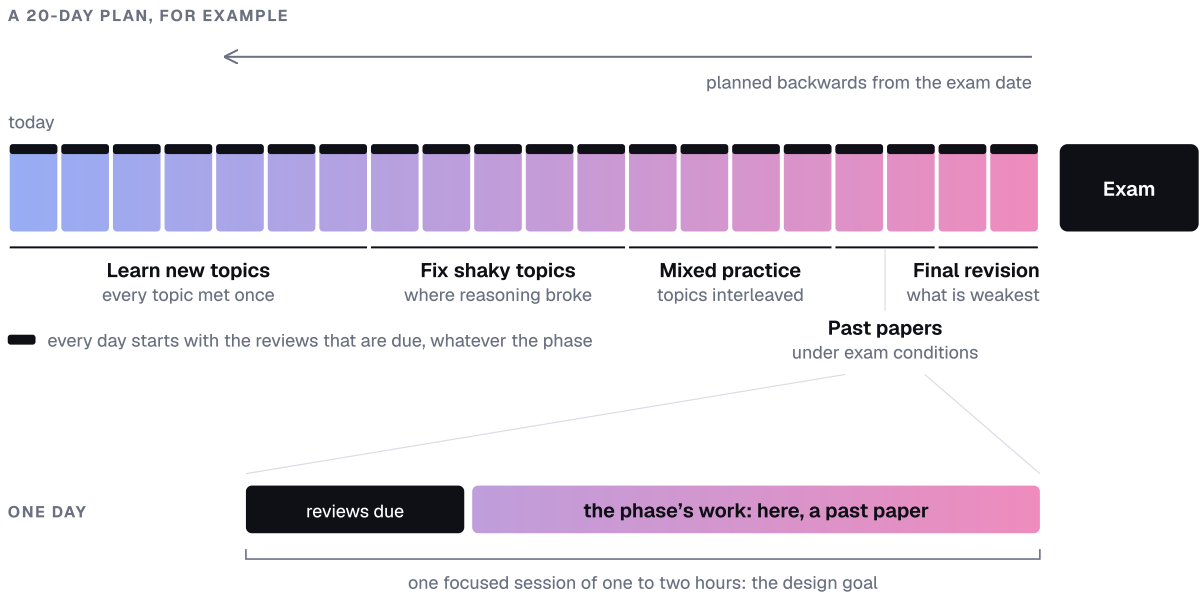


Figure 11. The plan starts from the exam date and works backwards. Engine design. An example 20-day plan. Every day starts with the reviews that are due, then the work of that phase.

When an exam date is set, the days left are planned backwards from it, in phases: learning new topics, fixing shaky ones, mixed practice, past papers and final revision. The design goal is a focused session of one to two hours that does the work of a longer one. It is a goal, not a measured result.

10 The engine checks itself

An engine that decides for a learner has to be able to find out that it is wrong. So every decision Cogniloop takes is logged with the prediction behind it, and the prediction is scored when the outcome arrives against the simplest rival there is: always guessing the learner's own average. None of the engine's adaptive signals decides anything by default until it beats that rival. The basic estimate of what a learner can do alone still picks the next problem; it is the part this check corrected. Today the check is run by hand.

The first check, in November 2026, came before the prediction log existed. It replayed 18 days of my own study, 46 graded problems, and asked for each one what the engine would have predicted just before it. On average it expected me to solve about two-thirds of them on my own, about 31 of the 46. I solved 15.

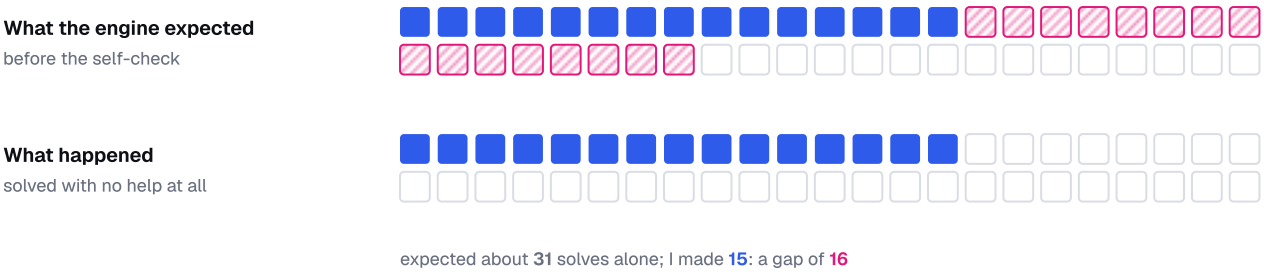


Figure 12. The engine expected about 31 solves alone out of 46; I made 15. **Data: one learner, the founder.** Each square is one graded problem from 18 days of my own study; the engine gave a probability for each, and these are the expected and actual totals. Two causes were found: new topics started with too high an estimate, the larger cause, and help was counted as progress.

Source: Cogniloop self-check, November 2026.

Two causes were found. The larger was mundane: a new topic started with too high an estimate, and most topics had only a problem or two, so the estimate never had the chance to come down. The other is the one this paper is about: the engine counted help as progress. A problem solved after a hint raised its estimate of what I could do alone almost as much as a problem solved without one. That is the illusion that help produces in learners, absorbed by the engine itself. Both were corrected, and help now counts for far less (Figure 8). After the correction the estimate is closer, but it is still not better than guessing my average, and the correction was tuned and scored on the same 46 problems, so even that is flattering.

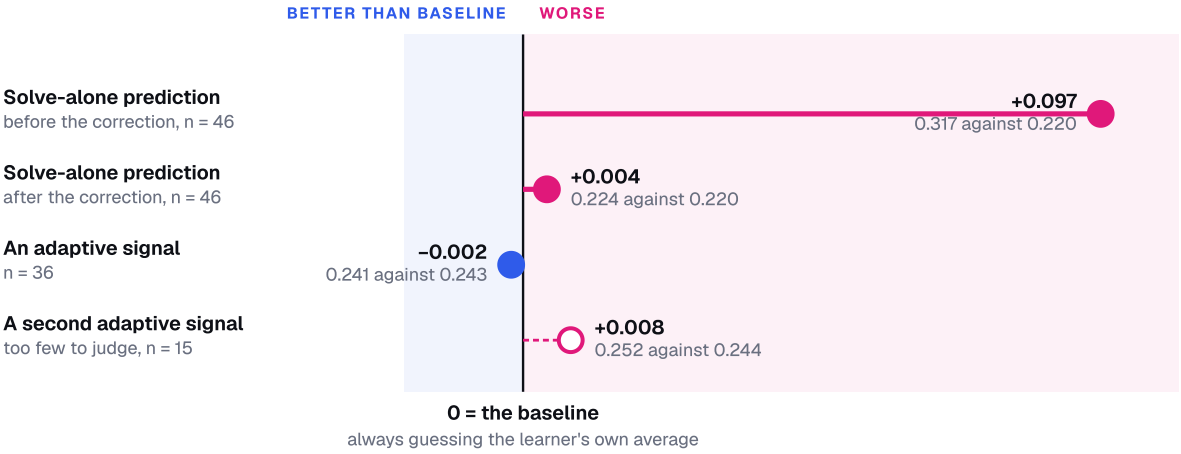


Figure 13. None of the predictions clearly beats its baseline yet, so none is trusted to decide by default. Data: one learner, the founder. Each dot is a prediction's Brier score minus the score of always guessing my own average; lower is better. The correction closed most of the gap on the solve-alone prediction, though it was tuned and scored on the same 46 problems. One adaptive signal is level with its baseline; the other has too few cases to judge.

Source: Cogniloop self-check and replays, November 2026.

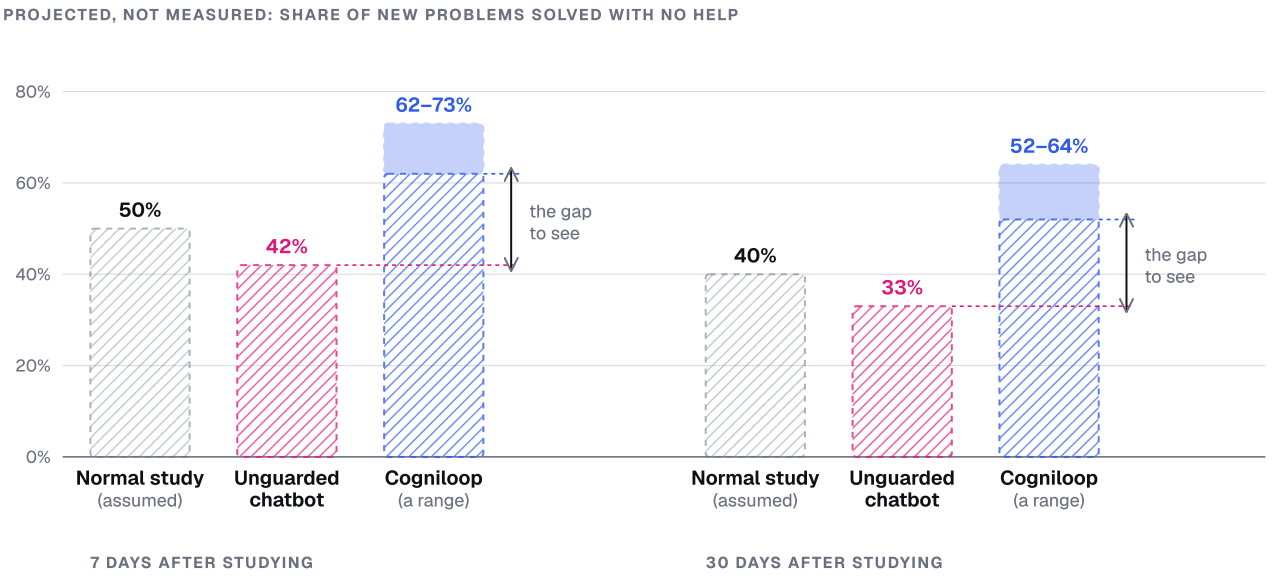
The check could not test the reviews at all. In those 18 days I reviewed too soon after studying to forget anything, so there was nothing to score. The scheduler is not shown to be wrong, only untested on me; its model has been benchmarked on about 350 million reviews from 10,000 other learners.[29] The check also showed the limit of one learner: 46 problems cannot fit a model of a person. So the engine's signals stay simple, and none of them decides anything by default until it has beaten its baseline.

A self-check that never finds anything is not a check. This one found two errors within three weeks, from the engine's own records. That is what the check is for.

11 What would prove this wrong

Cogniloop makes one claim that matters: learners who use it can solve more on their own, a week and a month later, than learners who use a plain chatbot or study the usual way. If that is not true, the approach is wrong and has to change.

- **The outcome is solving alone, later.** Not time spent, not satisfaction and not practice scores, which help inflates. Learners take new variants of problems they studied, with no help, 7 and 30 days after studying them.
- **Three comparisons.** Cogniloop, a general chatbot used as the learner likes, and the learner's normal way of studying, with the same material and the same study time.
- **Written down first.** The outcome, the comparison, the sample and the analysis are fixed before the data is collected.
- **A clear failure line.** If learners using Cogniloop do not solve more problems on their own at 7 and 30 days than both comparisons, the approach is wrong for those learners, and I will say so.



Assumed anchor: 50% and 40% for normal study. Chatbot: the anchor reduced by 17%. Cogniloop: the anchor shifted by effects of 0.3 to 0.6 standard deviations.

Figure 14. What published effects suggest the test could find, if they carry over. Projection. Share of new problems solved with no help, 7 and 30 days after studying. Normal study is an assumed anchor. The chatbot applies the 17% fall in unassisted exam scores that Bastani and colleagues found with unguarded GPT-4, measured right after each session and assumed here to carry to 7 and 30 days. The Cogniloop range shifts the anchor by 0.3 to 0.6 standard deviations on a normal (probit) scale, around the published effects of its parts. If Cogniloop has no effect, its bars sit at the anchor. Not measured; to be replaced by tester results.

Sources: Bastani et al. (2025); Rowland (2014); VanLehn (2011).

Figure 14 is a projection, not a result. It shows what published effects would imply if they carried over, and so what the test should be able to see. The range for Cogniloop is set around the published effects of its parts, starting below most of them, because effects shrink when a method leaves the laboratory: across

747 randomised trials of school interventions measured on standardised tests, the median effect was a tenth of a standard deviation.[30]

Who tests it, in what order

My own use can find bugs and test whether the engine's records are honest, but it cannot show that Cogniloop works: I built it, I know what it wants, and I want it to work. A first group of friends, using it alone at a real address, can show whether people who did not build it use it as designed and come back without being asked. University testers can produce the first numbers that come from people other than me. Only a study with comparison groups can answer the claim.

12 Who it is for

Students first. Personal tutoring is one of the best-supported ways to help someone learn, and most students cannot pay for it: platforms that list private tutors put typical rates at about $24 to $46 an hour in the United States.[31,32] Software cannot replace a good human tutor's judgement, warmth or authority. But step-level tutoring software has come close to human tutors on the measures the research uses, and software can be there at eleven at night, three days before an exam, for every student, at a small fraction of that price. Whether Cogniloop does as well is what Section 11 sets out to test.

Teachers have good reason to be wary of AI. A tool that gives students what they ask for usually gives them the answer. Cogniloop is built the other way round, and its records could serve teachers directly: because the engine records where reasoning broke, and whether it was a slip or a gap, it could show not only who got a problem wrong but where a whole class's reasoning breaks. An institutional version is a plan, and it would need each student's consent and a careful line between helping a teacher see a class and watching a student.

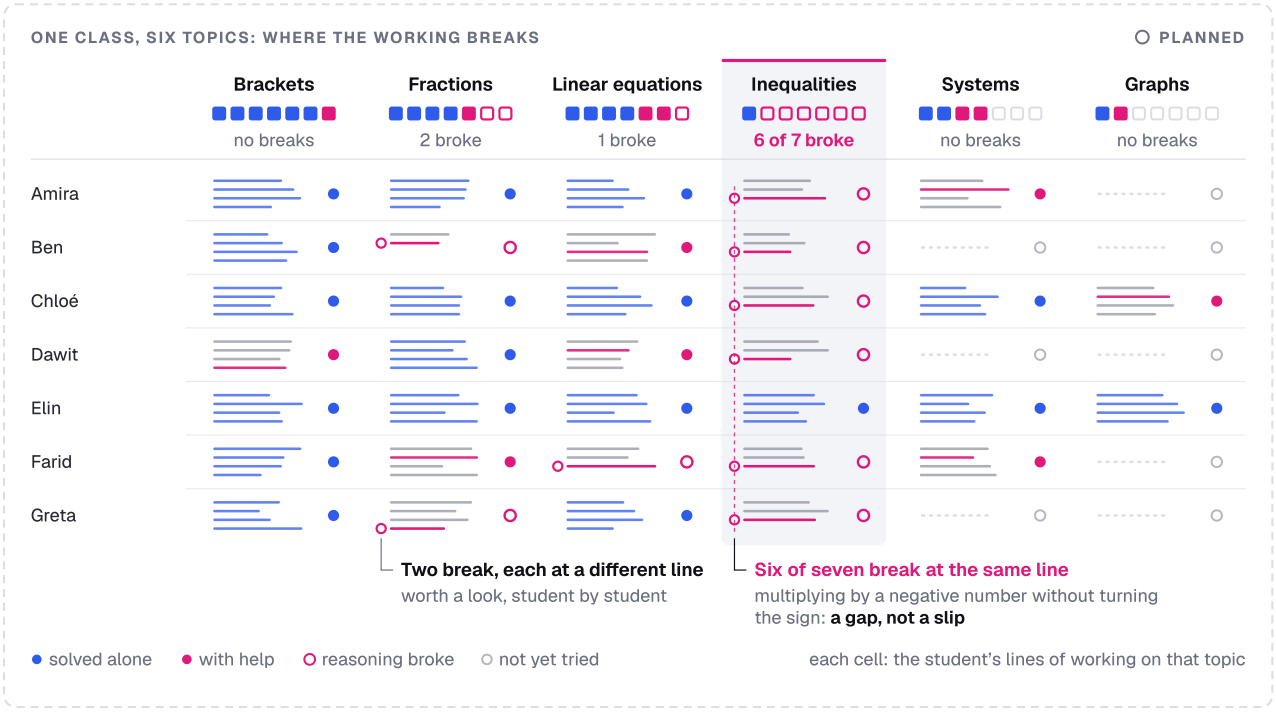


Figure 15. When most of a class breaks at the same line, it is a gap to teach, not a slip. Illustration, planned. A planned view for teachers, with seven invented students and six topics. Each cell is a student's lines of working, with a mark where the reasoning broke. In Inequalities, six of seven break at the same step. Not built.

Researchers rarely get weeks of real coursework: a laboratory can keep participants for an hour or a week, seldom a term. Cogniloop records every attempt, line of working, hint and review, with the engine's prediction before each outcome. With learners' consent, and anonymised to the standards the law requires, that record could support research on questions short studies cannot reach, such as how help-seeking changes over a course. Learners can already opt in to research use, off by default. No research use has happened, and none will without that consent and an ethics review.

13 Limitations

- **One learner, and a biased one.** All real data in this paper comes from me. I chose the material, I know how the engine decides, and I want it to work. It seems to work for me. That is a reason to test it, not evidence that it works.
- **No outcome data.** Nothing here shows that Cogniloop improves exam results or long-term memory. The projections show what published effects would imply, not what has happened.
- **Effects shrink outside the laboratory,** and combining methods does not simply add their effects.
- **The reader of the work can be wrong.** A language model reads the working. Verdicts are checked outside the model, but a wrong one is still possible, and a wrong "you made a mistake here" is worse than none. Learners can challenge a verdict, and challenges are recorded.
- **The adaptive signals are unproven.** None beats its baseline yet, which is why they are off by default.
- **Trying first has a cost.** Some learners, on some days, will find it tedious and leave. I did, before I made the first version easier. A design that people avoid teaches nobody.
- **One subject so far.** The engine is built for any subject, but it has been used and evaluated only on mathematics.

14 Conclusion

The problem with AI in education is not the technology. It is that a tool which hands over the answer in seconds meets a learner who would rather not spend hours, and the learner takes the answer. Both behave sensibly. The result is study that feels like understanding and fails on the exam. Withholding answers is a start, but a rule in a prompt is easily talked out of, does not know the learner, does not check its own arithmetic, and cannot tell whether its judgement of the learner is right.

Cogniloop puts an engine under the tutor. It reads the work line by line and finds the first line that does not hold. It keeps what a learner can do alone apart from what they did with help, brings each topic back before it fades, remembers the learner, and logs every prediction it acts on so that it can be found wrong. Its first self-check found it wrong twice, once in the most instructive way possible: it had counted help as progress, the same illusion it was built to break.

Learning starts at the line where your reasoning breaks. None of this shows yet that Cogniloop helps people learn. That depends on one test: whether learners who use it solve more on their own a week and a month later. This paper sets out the design, the reasoning and the test. The next version will report the results, whichever way they fall.

Notes

1. Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakcı, Ö., & Mariman, R. (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences, 122*(26), e2422633122. https://doi.org/10.1073/pnas.2422633122
2. Woodruff, P. (2022). Plato's shorter ethical works. In *The Stanford Encyclopedia of Philosophy.* https://plato.stanford.edu/entries/plato-ethics-shorter/
3. Plato, *Meno* 80a and 84b, translated by B. Jowett. The Internet Classics Archive, classics.mit.edu/Plato/meno.html
4. Koriat, A., & Bjork, R. A. (2005). Illusions of competence in monitoring one's knowledge during study. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31*(2), 187–194. https://doi.org/10.1037/0278-7393.31.2.187
5. Karpicke, J. D., Butler, A. C., & Roediger, H. L., III (2009). Metacognitive strategies in student learning. *Memory, 17*(4), 471–479. https://doi.org/10.1080/09658210802647009
6. Kornell, N., & Bjork, R. A. (2008). Learning concepts and categories: Is spacing the "enemy of induction"? *Psychological Science, 19*(6), 585–592. https://doi.org/10.1111/j.1467-9280.2008.02127.x
7. Bjork, R. A., Dunlosky, J., & Kornell, N. (2013). Self-regulated learning: Beliefs, techniques, and illusions. *Annual Review of Psychology, 64*, 417–444. https://doi.org/10.1146/annurev-psych-113011-143823
8. Roediger, H. L., III, & Karpicke, J. D. (2006). Test-enhanced learning. *Psychological Science, 17*(3), 249–255. https://doi.org/10.1111/j.1467-9280.2006.01693.x
9. Rowland, C. A. (2014). The effect of testing versus restudy on retention: A meta-analytic review of the testing effect. *Psychological Bulletin, 140*(6), 1432–1463. https://doi.org/10.1037/a0037559
10. Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest, 14*(1), 4–58. https://doi.org/10.1177/1529100612453266
11. Kornell, N., Hays, M. J., & Bjork, R. A. (2009). Unsuccessful retrieval attempts enhance subsequent learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 35*(4), 989–998. https://doi.org/10.1037/a0015729
12. Richland, L. E., Kornell, N., & Kao, L. S. (2009). The pretesting effect: Do unsuccessful retrieval attempts enhance learning? *Journal of Experimental Psychology: Applied, 15*(3), 243–257. https://doi.org/10.1037/a0016496
13. Kapur, M. (2008). Productive failure. *Cognition and Instruction, 26*(3), 379–424. https://doi.org/10.1080/07370000802212669
14. Kapur, M. (2014). Productive failure in learning math. *Cognitive Science, 38*(5), 1008–1022. https://doi.org/10.1111/cogs.12107
15. Sinha, T., & Kapur, M. (2021). When problem solving followed by instruction works: Evidence for productive failure. *Review of Educational Research, 91*(5), 761–798. https://doi.org/10.3102/00346543211019105
16. Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. In J. Metcalfe & A. Shimamura (Eds.), *Metacognition: Knowing about knowing* (pp. 185–205). MIT Press.
17. Murre, J. M. J., & Dros, J. (2015). Replication and analysis of Ebbinghaus' forgetting curve. *PLOS ONE, 10*(7), e0120644. https://doi.org/10.1371/journal.pone.0120644
18. Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks. *Psychological Bulletin, 132*(3), 354–380. https://doi.org/10.1037/0033-2909.132.3.354
19. Cepeda, N. J., Vul, E., Rohrer, D., Wixted, J. T., & Pashler, H. (2008). Spacing effects in learning: A temporal ridgeline of optimal retention. *Psychological Science, 19*(11), 1095–1102. https://doi.org/10.1111/j.1467-9280.2008.02209.x
20. Kestin, G., Miller, K., Klales, A., Milbourne, T., & Ponti, G. (2025). AI tutoring outperforms in-class active learning. *Scientific Reports, 15*, 17458. https://doi.org/10.1038/s41598-025-97652-6
21. Fan, Y., et al. (2025). Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology, 56*(2), 489–530. https://doi.org/10.1111/bjet.13544
22. Nickow, A., Oreopoulos, P., & Quan, V. (2024). The promise of tutoring for PreK–12 learning. *American Educational Research Journal, 61*(1), 74–107. https://doi.org/10.3102/00028312231208687
23. Bloom, B. S. (1984). The 2 sigma problem. *Educational Researcher, 13*(6), 4–16. https://doi.org/10.3102/0013189X013006004
24. von Hippel, P. T. (2024). Two-sigma tutoring: Separating science fiction from science fact. *Education Next, 24*(2), 22–31.
25. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369

26. Willison, S. (2025, 29 July). OpenAI: Introducing study mode. simonwillison.net (quoting OpenAI's announcement, which describes study mode as powered by custom system instructions).

27. Anthropic (2025, 2 April). Introducing Claude for Education. anthropic.com/news/introducing-claude-for-education

28. Ye, J., Su, J., & Cao, Y. (2022). A stochastic shortest path algorithm for optimizing spaced repetition scheduling. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4381–4390. https://doi.org/10.1145/3534678.3539081

29. open-spaced-repetition. FSRS and the SRS benchmark (about 350 million reviews from 10,000 Anki users). github.com/open-spaced-repetition/srs-benchmark, as of September 2026.

30. Kraft, M. A. (2020). Interpreting effect sizes of education interventions. *Educational Researcher, 49*(4), 241–253. https://doi.org/10.3102/0013189X20912798

31. Care.com (2025). How much does a tutor cost? Typical posted starting rate about $24 an hour, as of 22 September 2025.

32. Wiingy (2025). Private tutoring costs in the USA. A tutoring platform's own analysis of over 3,600 listings on five platforms, its own included: mathematics about $46 an hour.