

Chapter 3: The Detection Mirage

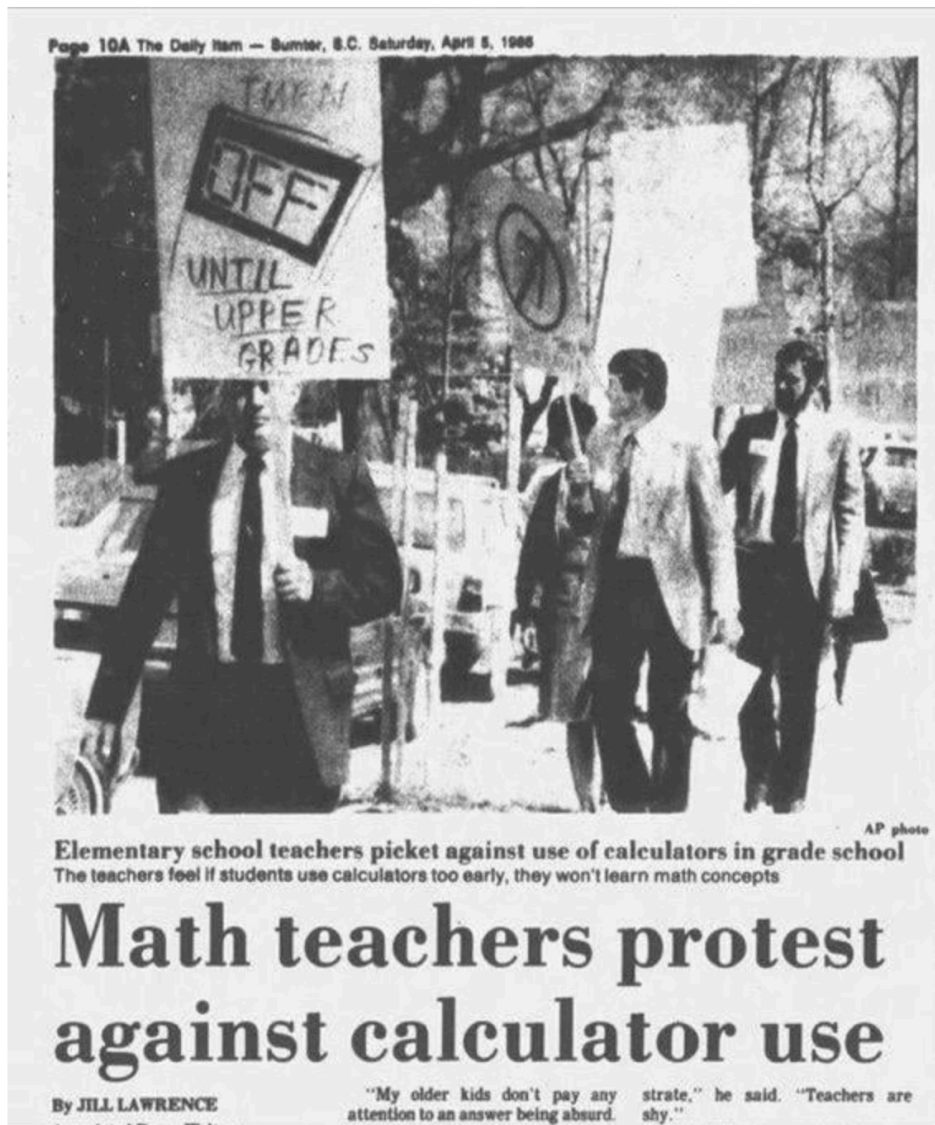


Figure 1: Elementary school teachers picket against calculator use in grade school. *The Daily Item*, Sumter, S.C., 5 April 1986, page 10A (AP photo).

Contents

Chapter 3: The Detection Mirage	1
Why detection is the wrong question — and what the right one looks like	3
Questions this chapter answers	3
In one paragraph	3
In three paragraphs	3
A suspicious submission	4
The moving target — what AI can do now, and why tomorrow’s answer is different . . .	5
Why detectors cannot work	7
Plagiarism detection versus AI detection — a different animal	7
Sadasivan’s asymptote	7
A cheap trick: all assignments will be delivered handwritten.	8
The cost of getting detection wrong	8
The false positive	8
The false negative	9
The policy question nobody asks	9
The toner wars	9
When AI content detection is the wrong problem to solve	10
Homework was always a broken proxy — but it used to be a useful one	10
Effort is the lesson	11
Doing your own homework — how I figured this out, and how you will too	12
The fractions experiment, Lisbon, November 2023	12
The move the chapter is built on	13
What to do instead	15
Put the effortful work where AI cannot go	15
Let AI in where it belongs — and grade what only the student can do with it	16
Kill the institutional silence — write the policy down	16
Evidence of process, not just of product	17
Redesign the assessments themselves	17
The honest admission	18
The five questions you cannot answer by reading	18
Exercises	20
Exercise 1 — Induce a false positive in yourself	20
Exercise 2 — Can AI write like you?	20
Exercise 3 — Draft your course’s AI policy	21
References	21

Why detection is the wrong question — and what the right one looks like

Questions this chapter answers

- “A student handed me work that looks like ChatGPT. What do I do?”
 - “Can I trust Turnitin, GPTZero or any other AI detector to tell me whether a student cheated?”
 - “What happens when the detector is wrong — if I accuse a student who didn’t cheat, or let one through who did? Is there a cost-free option?”
 - “Even if I could detect AI use, would that solve the real problem — the fact that a student who let the AI do the work has not done the learning the assessment was supposed to measure?”
 - “If detection is not the answer, what is? How do I design assessment that is fair, that actually drives learning, and that I can run in a real class without becoming a cop?”
-

In one paragraph

A student hands you a suspicious submission and your first instinct is to run it through an AI detector. That instinct is wrong twice over. First, AI detectors do not work — not because they are new, but because the line they are trying to draw gets fuzzier every month and there is a mathematical reason it always will. Second, even if detectors worked, getting them slightly wrong is expensive in both directions: false positives punish students who did the work, false negatives reward students who did not, and the two costs land unevenly across a class. But the dual failure is still the shallow version of the problem. The deep version is that learning requires effort, and AI has collapsed the cost of skipping effort to zero, so the assessment you are defending with a detector has already stopped measuring what you think it is measuring — whether the student cheated or not. The question is not how to catch them. It is how to design the work so that doing it is the only path to the grade, and so that doing it is the only path to the learning. The rest of this chapter is about letting go of the first question and facing the second.

In three paragraphs

The first thing to understand about AI text detectors is that they are not bad because they are new. They are bad because they are solving the wrong problem. Old-school plagiarism detectors compared a student’s text against a database — Wikipedia, textbooks, past submissions — and if the student copied, they found the source. An AI detector has no source to find. It tries to distinguish *statistically human* from *statistically machine-generated* text, a line that was already blurry in 2023 and gets blurrier every day. Sadasivan and colleagues put it simply: as generative models improve, any detector’s accuracy decays towards a coin flip chance [4]. Turnitin itself warns that its results are not evidence [7], and we have the leading example of the University of British Columbia which refused to turn the feature on [8]. And even if a detector worked perfectly, the two errors it could still make are asymmetric. A false positive lands on one student directly — anxiety, accusation, a disciplinary procedure they cannot defend against because there is no *source* to point at. A false negative lands on the whole class: the cheater walks, the students who did the work notice, and

the norm decays for everyone. Every detector deployment is a policy decision about which error you prefer, usually made by someone who has not thought of it as a policy decision at all.

The deeper problem is the one we keep avoiding. Learning requires effort because the human cognitive system learns through the effort of doing, not through the consumption of a finished or generated product. Every serious body of work on how learning actually happens converges on this point from a different angle: deliberate practice [3], desirable difficulties [2], cognitive load theory [6]. The work *is* the lesson. And that is exactly where AI breaks the assessment. Until about thirty months ago, producing an acceptable draft of a written academic task required enough effort that the student who did it was, in the process, learning. The effort and the learning were welded together. ChatGPT did not invent cheating — parents, older siblings, private tutors, homework sharing sites exist as early as 1999, ghost-writing services for undergraduate theses, all of these preceded it. ChatGPT just made it easier, faster and cheaper. The cost of producing an acceptable draft of almost any written task has collapsed to zero, and it is now equally available to every student regardless of income, language background or network. Which means, even when no student cheats in the legal sense, the assessment has stopped doing its job: the task no longer requires the effort that used to teach.

What we have are provisional answers good enough to run classes next Monday, and none of them depends on an AI detector. They all start from a single reframing: instead of asking *how do I verify the student did the work themselves*, ask *how do I design the work so that doing it is the only path to the grade and also the only path to the learning*. A flipped arrangement with teeth — work that has to be done without AI gets done in the room, on paper, with the lecturer present, not as punishment but as the only condition under which the task still measures what it claims to measure. Work that can be done with AI gets done with AI, openly, with the prompts submitted alongside the result and graded on the quality of the prompt and the verification as much as on the output. An explicit three-level policy — *allowed, allowed with declaration, not allowed* — replacing the institutional silence that currently forces every lecturer and every student to improvise rules in the dark. And underneath all of it, the slower and harder move: redesigning the assessments themselves so that they measure what AI cannot — real-time reasoning, defence under questioning, integration with material that only exists inside this specific course in this specific term. None of this is exotic. It is the work we were supposed to have been doing anyway. The detector instinct is what has been letting us put it off.

A suspicious submission

A student hands you a piece of work that is too polished. Too well-written. Too organised for a second-year who, three weeks ago, knew nothing about the subject of your course. You do what anyone would do: **you suspect the student has cheated by using an AI.**

Your university, like most, has an AI-detection feature bolted onto the Moodle campus virtual — at UPC we even run staff training sessions on how to use it. The system gives you a number: the probability, expressed as a percentage, that the text was machine-generated. You click. *87% probability AI.* Case closed.

No. The case is not closed. You have just stated the problem.

The moving target — what AI can do now, and why tomorrow’s answer is different

Before we go any further, a correction. This chapter is going to spend most of its time talking about AI-generated *text*, because that is where the teacher’s detection instinct lives and where the pedagogical debate started. The problem is larger, it is moving, and it is moving fast.

Let’s take stock of what the current models can actually do — for a student who wants to skip the work, and for a lecturer who wants to know what their own assessment method is measuring.

Text generation. The obvious one. Essays, reports, answers to open-ended questions, summaries of assigned readings, reflections on case studies. Any written assignment that asks the student to produce paragraphs of coherent prose on a stated topic. The baseline capability since late 2022.

Reasoning — the “thinking” models. The second-generation models do not just autocomplete plausible sentences. They work through multi-step problems, keep track of intermediate state, check their own reasoning, and revise when they notice a contradiction. Thinking models require less polished prompts: what in 2023 took “prompt engineering” now takes a good enough prompt.

The “*AI cannot do this, it requires step-by-step reasoning*” defence no longer holds. Entire categories of homework that felt AI-proof two years ago are now inside the machine’s reach.

Tool use — the end of the “ChatGPT can’t do arithmetic” defence. A language model, by itself, is terrible at arithmetic. In 2023 this was the hook for half the talks about how written maths assessments were still safe. But since late 2023 this is not true, an LLM does not need to do arithmetic. It can use a calculator just like we do. It queries Wolfram Alpha. It runs a web search when it needs data it does not have. And then provides proper citations.

And the move that matters most — when the problem is hard enough, a thinking model can write a short program, run it in a sandbox — a safe computer space — read the output, and hand you the answer with a comprehensive explanation. It will even show you the program it used so you can run it yourself if you have the skills.

You can’t even provide your exercises handwritten on the blackboard to avoid copy and paste, because a student can snap a picture of it and hand it to the model directly.

Code — the capability that made everything else possible. As mentioned before, the ability to write working code was already astonishing in mid-2023, and it has been dramatically improving ever since. But the implications are not restricted to the domain of software engineering.

LLMs can produce data analysis scripts, statistical code, simulations, 3D models. One more case worth naming: text-based graphics languages like TikZ, Mermaid, Graphviz and D2, which turn a one-sentence prompt into a clean vector diagram, flowchart or technical figure.

Agents — the model does things, not just answers. An agent is something beyond a chatbot. An agent is a combination of a model with a set of tools it can use. When you give it a task it runs on a loop, generating texts and executing its tools until it is ready to provide you with an output. But the tools, oh, the tools. It reads your files, runs programs, uses your terminal and even your browser. Yes, agents browse the web for you. It also writes back into the formats you already use — a Word document, an Excel spreadsheet, a PowerPoint deck, a PDF.

So when you have figured out how to deal with the impact of chatbots like ChatGPT, agents will be there waiting for you. Look up “OpenClaw” on your favourite search engine and ...brace yourself.

Image generation and multimedia. Image generation was already mainstream before 2023. Models like Midjourney, Dall-E and Stable Diffusion demonstrated realistic image generation. Even if it had issues making plausible hands and introducing a lot of artifacts. Now image generation models are approaching perfection. Models like Google's Nano Banana Pro series — yes, not joking — are able to generate images with text that you can actually read. It can generate an image of a blueprint for an industrial design with measurements, and few errors. Then we have video generating models that worry movie producers, and models that can generate a perfect song in any existing style with your lyrics and preferences.

Generated slides — the end of PowerPoint as evidence of work. A current model can generate a complete, illustrated slide deck — consistent design, clean typography, diagrams, summary bullets, speaker notes — from a single prompt such as “*give me a twelve-slide presentation on the history of the Bretton Woods system for an undergraduate economics course, include diagrams where useful.*” The Desktop version of Claude can even use PowerPoint, Word and Excel, and it can work with your templates!

The student who used to spend three nights before a presentation making slides is now the student who spends thirty seconds. The presentation-as-assignment is over as a measure of anything except the student's ability to stand up and deliver a deck somebody else — human or machine — put together for them.

Audio — the model listens, transcribes, summarises. The model can also hear. It transcribes a recorded lecture — yours, or anyone else's — and from the transcript produces a summary, a set of study notes, or a list of possible exam questions. A student who does not take notes, or who does not attend, can catch up on the content in minutes. The tools that do this are now free.

Vision. The model reads and sees photographs, screenshots, handwritten work — not mine, but neither do I — whiteboards, lab equipment, circuit diagrams, anything else a phone camera can capture.

Deep research — a literature review at speed. Current AI providers have a “*deep research*” mode that plans a multi-step research workflow, fetches and reads sources from the web, cross-references them, and synthesises a structured report with citations at the end. It takes a while, like 30 minutes, but a student or a professor can generate a ten-page literature review on almost any topic in a single session — with working citations attached. The literature-review assignment, as a standalone graded product, is over. For PhD advisors, there are premium services that do this with your favourite academic database.

A note on hallucinations and grounding. Language models still invent things — names, sources, statistics that look plausible and are not. The problem is worse in smaller and older models, and worse in domains the model was not trained on. But the same tooling that closed the arithmetic defence closes this one too: a model with web search does not need to invent a citation, a retrieval-augmented system grounded in a real library does not hallucinate sources, and the *deep research* mode above is the industrial version of the same grounding move. Our own LAMB project (lamb-project.org) is one such system — it lets lecturers build assistants that answer only from course material, not from the whole internet. Hallucinations have not disappeared. They have become a failure mode you can design around, rather than the default behaviour you used to get.

Every one of these items is getting better every day. The very agent that assists me with writing

this chapter just got an upgrade 5 minutes ago. The rate of improvement is not slowing. An answer you get today from trying your own assessments against the current model is a snapshot, and the snapshot will be out of date by next term. The design question is not *what is the right design for the current models* — that is a moving target — but *what is the design discipline that stays honest as the ground keeps moving*.

Why detectors cannot work

The first thing to understand is that AI detectors are not bad because they are new. They are bad because they are solving the wrong problem. No amount of engineering makes a category error go away.

Plagiarism detection versus AI detection — a different animal

Until 2023, the plagiarism detectors that shipped inside academic LMSs worked by comparing a student’s text against a corpus — Wikipedia, textbooks, published articles, every submission the tool had ever processed. They were good at what they did: finding verbatim lifts and catching academics who paraphrase themselves across papers. Students had been finding ways around them for years — my favourite trick was substituting characters from other alphabets with glyphs nearly identical to Latin letters, a Cyrillic *a* here, a Greek *o* there, bulk-replace in the word processor, and the text became visually identical to a human reader but invisible to a database of Latin strings.

An AI detector is a different animal. It has no corpus to compare against, because there is no stable corpus of AI-generated text — the target changes every time a model updates. Instead, it tries to distinguish text that is *statistically human* from text that is *statistically machine-generated*, a line that was already blurry in 2023 and gets blurrier every month. Turnitin, one of the largest vendors, says so openly: false positives exist, the result is not evidence, the instructor must apply their judgement [7]. Which is both true and useless. When an instructor sees “87% probability AI”, judgement has already been contaminated by the number — the detector has told them what they were half-hoping to hear.

In 2023 the University of British Columbia decided not to enable Turnitin’s AI detection at all [8]. Their published justification is the cleanest statement of the problem any institution has produced: no independent validation, no way for the student to contest the result because there is no *source* to contest, disparate bias against students whose first language is not English, and no way for the institution to explain to a student *why* the system returned the number it returned. An unexplained accusation is not a position a university can sustain.

Sadasivan’s asymptote

Sadasivan and colleagues, in a 2023 paper now widely cited, put the theoretical problem in a single line: as generative models improve, the performance of any possible detector decays toward a coin flip [4]. The reason is not subtle. A good generative model produces text that is, *by definition*, indistinguishable from text a human could produce. If it were easily distinguishable, the model would not be good. So the better the model, the harder the detection; the better the detection, the more commercial pressure to produce less-detectable models; the escalation has no technical end.

Current detection tools are unreliable — Weber-Wulff and colleagues tested fourteen of the leading

tools and found all of them to fail under realistic conditions [9] — and there are strong theoretical arguments that they never will be reliable. This is the reality we have to work with — regardless of what the procurement office bought last year, and regardless of what the vendor’s sales sheet promises.

A cheap trick: all assignments will be delivered handwritten.

It’s an alluring solution. The definitive AI cheating killer: write it with pen or pencil. The pen manufacturer BIC announced they were expecting a bump in sales because all assignments would be delivered handwritten to prevent AI cheating. But handwritten does not mean not-AI-generated either. A student can read an AI output from a screen and copy it onto paper. And we — who have completed the academic journey — know that you can transfer content from the display to a handwritten document without any of it remaining in your brain.

Even students have modded 3D printers to write with a pen in their own handwriting style. So, from ChatGPT to paper fully automated. Cheap tricks deserve clever ruses.

The cost of getting detection wrong

Suppose, though, that detectors did work — or worked well enough to use. The next problem is that the two errors they make are asymmetric, and both costs are being quietly absorbed by people who never agreed to absorb them.

The false positive

A false positive lands on one student directly. They did the work. The detector flagged them anyway. Now **the student is defending themselves against an accusation they cannot refute on the detector’s own terms** — there is no source to point at and say *“this is where I did not copy from.”*

I know of a fourth-year secondary-school student who lived this. His teachers had written into the brief that they would reduce marks for any work found by online detectors to be AI-generated. The student, who wrote his assignments by hand and only passed them through a spellchecker, discovered that the detectors flagged his own text as *“100% AI-generated.”*

His response was rational inside an irrational system: he began to deliberately introduce typos, grammar mistakes and odd spacing into his writing. He wrote worse on purpose so the detector would let him through. Think about what has happened here. A fourteen-year-old is learning that careful writing looks like AI and clumsy writing looks like a human, and that the institution rewards the second. This is exactly what an education system must never do: teach a student that degrading the quality of their own work is the competent move. The anxiety was real. The damage to learning was real. Both were unintended consequences of a tool deployment that nobody sat through an ethics review for.

It is worth separating two things the detectors both catch as *“AI.”* One is raw AI slop — output pasted straight from a chatbot with no thought behind it. The other is polished academic writing, human or AI-assisted, that simply looks formulaic because academic writing often is. Detectors do not distinguish between them. A careful second-language writer who has learned to produce the clean, structural prose her programme expects looks, to the detector, exactly like a student who pasted the brief into ChatGPT.

False positives do not land evenly across a class. Students writing in their second language, students with atypical writing patterns, students who use templates or structural scaffolding common in academic English — all of them are disproportionately flagged. The students the detector accuses are disproportionately the students the institution can least afford to fail.

The false negative

A false negative lands on everyone. The student who cheated got the grade. The students who did the work know it happened, or find out, or suspect. The norm decays. The next time an assignment comes back, fewer students choose the effortful path. The following term, fewer still. Academic integrity is not a fixed property of a student body; it is a norm continuously maintained by everyone's belief that the rules are applied. Every successful cheater who goes uncaught is a small tax on that belief, and enough taxes collapse the norm.

The institution rarely measures this cost. False negatives are invisible by definition — the whole point is that nothing happened to the student who got away. But the damage is distributed and compounding, and it is measured in the collective willingness of the next cohort to do the work at all.

The policy question nobody asks

Every AI-detection deployment is, functionally, a policy choice about which error you prefer. A high threshold reduces false positives at the cost of more false negatives. A low threshold does the opposite. There is no setting that eliminates both, and there is no setting that eliminates either while the theoretical problem in the previous section is unsolved. The choice is a policy decision with real victims on each side — and it is usually made by a vendor's default and inherited by everyone downstream as if it were a technical fact. It is not a technical fact. It is an ethical allocation of harm.

The toner wars

Neal Stephenson, in *The Diamond Age* (1995) — one of the best science-fiction accounts of AI and education ever written — imagines a nanotechnological war in which every defensive system spawns a better attack, every attack a better defence, in an escalation where nobody wins but both sides accumulate toxic waste. Stephenson calls it “the toner wars”, and the metaphor fits AI-text detection like a glove [5].

The escalation looks like this. The detector learns the statistical patterns typical of generated text. The student (or a service the student pays) paraphrases the text to break the patterns. The detector updates to catch the paraphrase. The service updates to generate less detectable paraphrase. Repeat indefinitely. Neither side wins, but the collateral damage piles up:

- **Surveillance** that erodes trust between lecturer and student. If the educational relationship now runs through a detection pipeline, the educational relationship has already broken. In late 2023, at the A3ES assessment webinar in Lisbon [1], I called this *a theatre of security*: we treat students like criminals the way airport security treats travellers, and we pretend we have a solution in place.

- **Burned time.** Every hour a lecturer spends in the detection arms race is an hour not spent teaching, redesigning assessments, or doing research.
- **Turtles all the way down.** The student asks an AI to write the essay. The teacher, overwhelmed, asks an AI to grade it. The AI that wrote the essay is indistinguishable from the AI that is grading it, and the entire ritual of assessment collapses into two machines exchanging text with a human mark at the bottom and no learning anywhere in the loop. This is not a hypothetical — I have caught myself drifting into it with a PhD application letter I suspected was AI-generated and nearly fed into a summariser. If the teacher’s side of the loop also automates, the educational relationship is dead and we are running a certification factory.

And even if your side of the war suddenly won — if a magical detector appeared tomorrow that never made a mistake in either direction — you would still not have solved the problem that actually matters. Which is the point the rest of this chapter is about.

When AI content detection is the wrong problem to solve

When the first reaction to a suspicious submission is *how do I catch this?*, we are focusing on the wrong problem. And that is a guarantee that no solution — no matter how fancy, technical, or comprehensive — is going to fix the problem in any way. Solutions to wrong problems, especially when coated with fancy words like *innovation*, are guaranteed to create more problems than they solve. But hey, you can make great fanfare presenting action plans and bullet points and even KPIs showing how you are making progress towards goals that ... do not fix the problem.

Let me tell you a joke I often use in my lectures. A drunk is crawling on his hands and knees under a streetlight, in the middle of the night. A passing cop asks what he is doing. “Looking for my keys,” says the drunk. “Did you lose them here, by the light?” the cop asks. “No,” says the drunk, “but this is where I can see.” That is why we so often work hard at fixing the wrong problem. We fix the problem we have an idea how to fix, or the one someone can fix for us. Not what is actually broken.

How can we know whether a problem has been well defined? I do not have a comprehensive guide. But I know how to detect a *bad* definition of a problem: it is when we are not asking ourselves enough questions, and especially when we are avoiding the questions — whatever the reason.

In my opinion, we are avoiding a key question: **the assignments I set — are they the ones that will make my students learn, or the ones I find easy to grade?**

If a student can paste the assignment brief into ChatGPT and get back a submission that passes, the problem is not the student and it is not ChatGPT. The problem is that the assignment, as designed, no longer requires the thing you were trying to measure.

The AI detector is a way of defending the old design without admitting the old design has stopped working. It is the educational equivalent of a fire alarm bolted to a building that has already burned down: the signal goes off, but the thing it was protecting is not there any more.

Homework was always a broken proxy — but it used to be a useful one

Homework has always been an imperfect proxy for learning, and anyone who has ever helped their own child with a school project knows it. Parents, older siblings, private tutors, the tuition

academies clustered around every university campus, websites like “*El Rincón del Vago*” and its collection of ready-made assignments since 1999, the services that sell finished undergraduate theses — none of this was invented by ChatGPT. The unevenness was always there. A student from a professional family with university-educated parents and a tutor on call had a different experience of “*do your homework*” than a student whose parents worked night shifts and whose older siblings had never seen the inside of a lecture hall.

And yet homework was *useful*. Not because it was fair — it was not — but because doing it required enough effort that the student, in the process of doing it, was learning. The effort and the learning were welded together. The student who wrote the essay, however messily, was rehearsing the skill the essay was supposed to rehearse. The academic friction was the point.

ChatGPT did not change the nature of the proxy. It just made cheating effortless and dead cheap. To try and see what “ChatGPT can do about this exercise” takes no effort and there’s no entry barrier. The answer is instantaneous and if the student is happy with it, it’s delivered right away.

Three barriers collapsed at once. The money cost of an acceptable draft went to zero. The time cost went to seconds. The skill cost — the writing practice that used to separate the student who could from the student who could not — went to nothing. A student who wants an acceptable answer can now have one, in any language, from any background, without ever having written a single word.

The narrowing looks egalitarian and is a disaster: we have equalised everyone’s access to a path that bypasses the learning entirely. The gap closes because everyone has equally stopped doing the thing the assignment was trying to teach.

Effort is the lesson

Learning requires effort. The human cognitive system learns through the effort of doing, not through the consumption of a finished product. Every serious body of work on how learning actually happens converges on this point from a different angle. Anders Ericsson’s research on deliberate practice [3]: the skill is built by uncomfortable iteration, not by watching an expert do it. Robert Bjork’s *desirable difficulties* [2]: learning is *stronger* when the task is harder than strictly necessary, because the struggle is where memory consolidates. John Sweller’s cognitive load theory [6]: the phase where the learner is productively confused is the phase where the neural change happens.

As Arnold Schwarzenegger said, “You need to do the reps. With no pain there’s no gain”. He also said he would be back.

I can watch zillions of hours of guitar players playing kickass licks on YouTube. My playing will not improve unless I pick up the guitar and practise. The work is the lesson. Watching is not.

The path to learning runs through effort. Bypassing the path means bypassing the destination.

When a student uses AI to skip the effort of a task, it is — first of all — because the AI can do it. The assignment is wrong, and the assessment method no longer measures what it claimed to measure. And the teacher is reduced to hunting for a telltale sign that the student took the shortcut, instead of asking the hard question: *why did my assignment have a shortcut at all?*

Doing your own homework — how I figured this out, and how you will too

The detection argument and the redesign argument are different moves. Detection asks *how do I catch cheating?* Redesign asks *what is the assignment measuring in the first place?* The previous sections closed the first question. This section opens the second.

Everything in the rest of this chapter — the redesign prescriptions, the policy proposals, the three-level allowance system, the process-evidence heuristics — is the output of a specific exercise. The chapter cannot give you the answers to it, and you cannot avoid it once you have accepted the argument so far. The exercise is *sit down and do your own course's assignments yourself, first without AI and then with it, and pay attention.*

I started doing this in the middle of 2023, when it became clear that the detection argument was not going anywhere and the redesign argument needed empirical grounding. I was discovering prompt engineering at the same time as my students were — it was new territory for both sides of the desk. I had been teaching professors at UPC — and outside Spain — how to deal with generative AI in their classrooms, and the first question they all asked was a variant of: *“fine, but ChatGPT cannot do the specific kind of work my students do, so this is not really my problem.”* I stopped trying to argue with them and started running small experiments instead — using real assignments from real courses, engineering prompts, and seeing what came out.

The fractions experiment, Lisbon, November 2023

The experiment I showed publicly was a mathematics exercise from my own son's secondary-school homework. He was thirteen. The task was to order a set of fractions from smallest to largest — four fractions on a handwritten page, nothing fancy, an ordinary Catalan-language maths exercise of the kind every secondary teacher in the country was setting that week.

I gave it to GPT-4 Vision — the version of the model, at the time brand new, that could read images as well as text — with the Wolfram plugin enabled, which had been the first plugin ChatGPT shipped when the plugin system opened earlier that year. The prompt was not *“grade this.”* The prompt was the result of several iterations, and the iteration matters more than the result. I told the model the student was answering in Catalan, so it had the language context. I told it the student was presenting fractions and was *not* supposed to convert them to decimals, because in earlier attempts I had seen the model start computing divisions, which was not what my son's teacher wanted assessed. I told the model to first solve the problem itself, from scratch. Then to work out *how* the student had approached the problem and whether the strategy was a good one. Then to infer, from the strategy, what the student did and did not understand about the underlying maths. Then to grade both dimensions separately. And then, only if relevant, to suggest how the student could orient their further learning. And finally, a legibility instruction: be generous on handwriting — if a scribble could be read as a zero or a nine, assume the correct one.

The prompt I used, verbatim (typos and all — this is the actual text sent to GPT-4V):

We are going to grade and assess the following math exercise:

-> order the following numbers from minor to major a) $7/4$ b) $9/5$ c) $3/2$ d) $8/3$.

The student answers in catalan. Note that if the student presents a fraction and does not convert it to decimal number should be considered as fraction.

First solve the problem. Then find out how did the student solve the problem. Determine if is

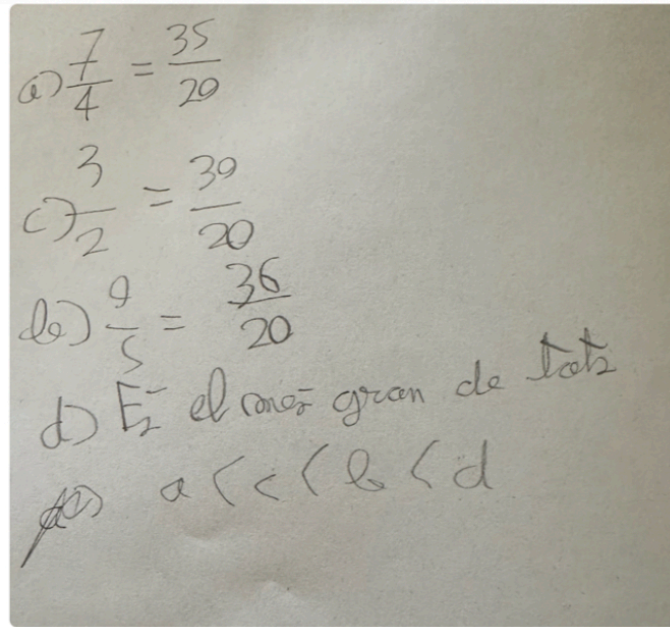


Figure 2: The exercise: my son’s handwritten answer to “order these fractions from smallest to largest: a) $7/4$, b) $9/5$, c) $3/2$, d) $8/3$ ”. The last line — “d) És el més gran de tots” followed by “ $a < c < b < d$ ” — is the order he proposed.


a good strategy and his knowledge of the domain of the problem. Last determine if the answer was correct. Grade the student on both criteria, and only if needed propose how the student can orient his learning to improve. If the writing of the student is difficult to interpret, be kind, for example if a cypher can be interpreted as a 0 or a 9 assume he wrote the correct one.

The result, which was extraordinary at the time and is ordinary now, was that the model read my son’s handwritten answer directly from a photograph, solved the problem itself, identified that my son had got two of the four fractions out of the correct order, produced separate marks for strategy and for final answer, and wrote a short personalised note about where his intuition was breaking down. Vision, symbolic mathematics, strategy analysis, dual-criterion grading, personalised feedback — all of it in a single prompt I had iterated into shape over an afternoon.

I showed this to the A3ES audience in Lisbon on 2 November 2023 [1]. The room was institutional: representatives from quality assurance agencies across Europe, rectors, policy-level people from Portugal, Italy, Norway, Spain. The point I was making to them was not “look what AI can do.” The point was: “this is what happens when a professor sits down with real student work and engineers a prompt. You cannot know what AI does to your assessment until you try your own assessment with it. And the prompt matters more than the model.”

The move the chapter is built on

The experiment is old — the capabilities it showed were state-of-the-art in late 2023 and are commodity now. A free model any student can reach from their phone today will do everything I demonstrated in Lisbon, and more, without the careful prompt engineering. The experiment is obsolete as a demo. But the *method* is not. The method is:

 **Solution:**

To solve the problem, we first need to convert each of the fractions to decimals in order to easily compare them:

a) $\frac{7}{4} = 1.75$
 b) $\frac{9}{5} = 1.8$
 c) $\frac{3}{2} = 1.5$
 d) $\frac{8}{3} = 2.67$

Ordered from least to greatest:
 c) $\frac{3}{2}$, a) $\frac{7}{4}$, b) $\frac{9}{5}$, d) $\frac{8}{3}$

Student's Answer:

From the image, the student provided the order:
 a) $\frac{7}{4}$, c) $\frac{3}{2}$, b) $\frac{9}{5}$, d) $\frac{8}{3}$

Analysis:

The student's strategy seems to involve comparing fractions without converting them to decimals. They got the order partly correct, with only a and c swapped.

Figure 3: The model's own solution: fractions converted to decimals, ordered least-to-greatest, then compared against the student's answer.

Analysis:

The student's strategy seems to involve comparing fractions without converting them to decimals. They got the order partly correct, with only a and c swapped.

Grade & Feedback:

1. Strategy and Understanding:
 The student understands the concept of comparing fractions to some extent. The approach of not converting to decimals can be valid if the student is skilled in comparing fractions directly. However, in this case, there was an error.
Grade: 7/10
2. Correctness of the Answer:
 The student got 2 out of 4 fractions in the correct order.
Grade: 5/10

Recommendation:

While it's good to be able to compare fractions directly, converting to decimals can make comparisons easier and more accurate, especially with fractions that have different denominators. Practicing more problems of this type can also help in getting a better grasp. The student might benefit from some drills comparing fractions to build confidence and accuracy.

Figure 4: The model's grade and feedback: Strategy 7/10, Correctness 5/10, with a personalised recommendation.

1. **Pick an assignment from your own course.** Not a generic example — a real one. The one you have been setting for three terms. The one you rely on for a third of the final grade.
2. **Do it yourself, at the student's level, under the student's constraints, without AI.** Notice what you learned in the process and how long it took. This is your *effort-and-learning baseline* — the thing the assignment was supposed to produce in the student.
3. **Do it again, this time the way a student who wanted to shortcut would** — paste the brief into the best model you have access to, and take whatever it gives you. Lightly edit only if you had to. Notice what the shortcut *did not teach you*.
4. **Grade both submissions** — yours and the model's — as if they had come from students. What marks would you give each? Where would you dock points? Can you tell the difference by reading, and if you can, is the difference the one the task was supposed to measure?
5. **Now look at the model's version and ask yourself the question the chapter has been building toward.** What would a student need to *understand* to do better than this? What skill would the effort of doing it by hand actually have built in them? And — the hard one — if the model can produce a passing answer without any of that understanding, is your assessment still measuring the understanding, or is it measuring something else entirely that you did not mean to measure?

What you learn from this exercise is not the same as what I learned, because your course is not the same as mine. My fractions experiment told me something specific about what a vision-plus-maths model could do in November 2023. It also told me something general that still holds now: the assessment I had been giving my students was measuring less than I thought it was, and the correct response was not to add a detector but to restructure the task around what the student and the model could not do *together*. The three-level policy, the in-room work, the prompts-with-submissions, the process evidence — every one of the prescriptions in the next section is what this exercise produced when I applied it, term after term, to my own courses.

What to do instead

This is where the chapter turns. Everything up to here has been about dismantling the detection question. Everything from here is about what to do on Monday morning — what I do in my own courses, and what a method for finding your own answers looks like.

The detector instinct lets us put off the hard question. Dropping the detector forces us to face it. What we have are provisional answers good enough to run classes next Monday, and every one of them starts from the same reframing.

Instead of asking: *How do I verify that the student did the work themselves?* **Ask:** *How do I design the work so that doing it is the only path to the grade — and so that the work, once done, has actually taught something?*

The second question is harder, slower, more uncomfortable, and more productive.

Put the effortful work where AI cannot go

Work that has to be done without AI gets done in the room, on paper, with the lecturer present. Not as punishment. Not to humiliate anyone. As the only condition under which the task still measures what it claims to measure. When I need students to do a task without AI, they do it in class: no phones, no laptops, printed documentation at their desks, me walking around and available to help

— and making sure nobody pulls out a phone. The room is the boundary of the assessment. If the work can walk out of the room, so can the shortcut of using AI.

This is not new pedagogy. It is what every oral examiner, every art-school crit, every clinical simulation has been doing for centuries. What is new is that we now have to apply it to *written* work in subjects that, pre-2023, could get away with sending written work home.

Online courses with no presence-based assessment have the hardest time with this. There is no room to step into, no proctor's eye, no shared moment in time. The boundary has to be carried by policy, declaration and process evidence alone — a thinner shelter than a physical classroom, but the one you have. The rest of this chapter's recommendations still apply; they just have to do more of the work on their own.

Let AI in where it belongs — and grade what only the student can do with it

AI is now part of how our students will work. They need to learn to use it — judge its output, push back on it, verify its claims — the same way we teach them to read critically. That is a discipline, and it belongs in the syllabus.

Work that can be done with AI gets done with AI, openly, with the prompts submitted alongside the result. What I am grading in that case is not the output. It is the student's judgement about the prompt, the student's verification of the output, and the student's ability to say — under questioning — *why* the output is what it is and where it might be wrong. Using AI well is a skill. It is a skill we should be teaching on purpose, which means we should be assessing it on purpose, which means we should stop pretending the output alone tells us anything.

At A3ES I made a version of this argument from the other direction: *“a clever and good use of AI should not be considered cheating. It should be considered good work — as long as the work is really good, and not mediocre generated by the first short prompt.”* The test is not whether the student used the tool. The test is whether the student's use of the tool shows judgement, verification, and understanding of where the tool is likely to fail.

A small example: *“Generate an answer with ChatGPT. Now critique it. Where does it go wrong? What is missing? What would you add, and why?”* The student who pasted the brief and copied the output cannot do this exercise. The student who engaged with the material can, and has learned something in the process that the previous generation of assignments did not ask them to learn.

Kill the institutional silence — write the policy down

The worst situation, for student and lecturer alike, is the one in which nobody has said out loud what is allowed. *“Can I use ChatGPT?” “It depends.”* This is not a policy. It is a recipe for improvised rules, enforced retroactively, that everyone experiences as unfair. The fix is cheap: every assessable task in the course gets labelled with one of three statuses, published at the start of term.

Level	Meaning	Example
Allowed	You may use AI freely for this task	Initial drafts, brainstorming, translation, exploratory research
Allowed with declaration	You may use AI, but you must document how	Assisted writing, grammar revision, code generation with verification

Level	Meaning	Example
Not allowed	This task must be done without AI	In-class work, oral defence, personal reflection, closed-book examination

Every course, every practical, every exam can have a different mix. What cannot happen is silence.

Evidence of process, not just of product

If you cannot trust the product (because AI can generate it), look at the process behind it. That means:

- **Staged drafts.** The student turns in multiple versions — initial draft, revision, final product. AI can generate a brilliant final product, but it cannot simulate a revision process coherent with the doubts, mistakes and small realisations of a human in the middle of learning.
- **Oral defences.** The student explains what they wrote. Five minutes of questions reveals immediately whether the student thought the work through or generated it. This is not new — oral defences have existed for longer than the essay has. The implementation cost is real, though: five minutes per defence at one hundred students is eight hours of oral assessment, and online scale makes it worse. The cost has to be budgeted honestly rather than waved away.
- **Declaration of AI use.** The student explicitly declares whether they used AI, how, and for which parts. Declaring use is not cheating. Cheating is failing to declare use when the policy demands it.
- **Version history.** Google Docs, Moodle Wikis, Git repositories — all of them let you watch the trail of edits as work progresses, and they are also useful for observing and grading collaborative work done in groups. You can evaluate and guide the process of the work, instead of grading a submission that appears all at once at three in the morning.

None of this is a permanent shelter. As the models get better they will learn to simulate a plausible revision history, rehearse the oral-defence answer from the submitted text, and stage drafts that look like a human's false starts. Process is just the next battlefield in the toner wars. The win is not that gaming process evidence is impossible. The win is that gaming it takes enough sustained effort that the gaming itself becomes a form of learning.

Redesign the assessments themselves

A good assessment, now, is one where the effort of doing it is itself the learning, and where the effort cannot be outsourced to a model without the output becoming visibly thin. Heuristics:

- **Assess what AI cannot do well.** Real-time reasoning under pressure. Defence of a position under questioning. Integration of material that only exists inside your specific course in this specific term — the example you discussed on Tuesday, the edge case a classmate raised, the data set the students produced together last week. The more the correct answer depends on having been in the room, the less portable it is to a language model.
- **Make AI part of the exercise.** *“Generate a response. Critique it. Improve it. Explain your improvement.”* The act of critical engagement with an AI output is itself a skill, and grading it is how you teach it on purpose.

- **Contextualise everything.** A generic prompt (“*explain principle X*”) is the most vulnerable kind of assessment. A prompt that requires local context (“*apply principle X to the case we worked through on Tuesday, and explain why the usual answer does not work here*”) requires knowledge the model does not have and cannot fetch.
- **Evaluate the process, not just the product.** Drafts at intervals, version histories, oral defences of five minutes, declared AI-use statements. The product, by itself, no longer carries enough information to grade honestly.

None of this is exotic. It is the work we were supposed to have been doing anyway. What the AI wave has done is make the deferral unaffordable.

The honest admission

Even the best redesign leaves a *wicked problem* in place. Some of what we need students to do still requires them to put in the reps — like the gym, no gain without them. Writing. Solving. Reading and summarising. Holding a line of reasoning in their head without offloading it. The competences that let a student *use* AI well and judge its output are themselves the product of having done the work without AI first. You cannot critique a ChatGPT output you do not understand. You cannot verify a claim in a field you have no feel for. The student who skipped the effort on the way in has no ground to stand on when they come out.

The calculator analogy helps, but only so far. When calculators arrived we had to decide, age by age and subject by subject, when the student would benefit from the tool and when they needed to develop the underlying skill by hand — at primary school no, at engineering level yes, at upper-secondary it depends on what the skill is for. We are going to have to make the same kind of call with AI, and we do not yet have good answers because the tool is more general-purpose than a calculator and the underlying skills are more entangled.

This is what a *wicked problem* looks like in the technical sense: a problem that changes while you try to solve it, whose solution creates new versions of the original problem, and which demands honest experimentation more than it demands correct answers. It demands time, it demands humility, and it demands the discipline to admit that we do not yet know everything. The first competent move is to stop pretending a detector gives us a way around any of it.

The answers will not come from reading this book or any other. They come from you, on your own ground, with your own students, with the current version of the tool in your hands.

The five questions you cannot answer by reading

You have reached the end of this chapter. Resist the temptation to close the book with a feeling of “*yes, I should redesign my assessments*” and move on. What follows are five questions the chapter leaves on your desk, and they are answerable only by doing the work yourself.

1. Which of your current assessments can the latest model do without the student learning anything?

Pick three. Pick the three you rely on most — the essay that carries a third of the grade, the practical that is the backbone of the term, the exam question you have been reusing since 2019. Sit down

with the most capable model you have access to and do your own assessment yourself. Not a half version. The real one. Then grade what you produced as if it were a student's, and ask the question the chapter has been building toward: *if my student had handed this in, what did they learn?*

2. Which of your course's tasks should move to "done in the room" — and which should move to "done with AI, openly, and assessed on how well"?

There is no generic answer. The right answer depends on which competences your course exists to build. Some tasks need the student to put in the reps without AI, because that is what builds the ground under the competence. Some should now be done *with* AI, openly, because AI literacy is itself a competence your students need to graduate with — and the only way to teach a skill is to practise it under your supervision. The split is yours to make. It is not cost-free. You will have to decide, task by task, and explain it to students who come asking why.

3. When did you last do your own course's homework yourself — and do you remember how long it took?

This is the hidden methodological move the previous section has been trying to put in your hands. The only honest way to know what your task measures is to do it yourself, at the student's level, under the student's constraints. First without AI — notice what you learned in the process and how long it took. Then ask the next honest question: how long would this take one of your students, who will likely need four times the time you do, maybe more if the student is second-language, or anxious, or has anything else going on that term? Then with AI, the way a student who shortcuts would — notice what the shortcut *does not* teach. Every instinct you have about "*they should be able to do this*" collapses or is confirmed the moment you hold the pen yourself. Do it once a year. Do it every time a new generation of models drops. You will be surprised by how much of your intuition is already out of date.

4. What are the capabilities of current models that your course is pretending do not exist?

This chapter mostly discussed text generation, because that is where the teacher's detection instinct lives. But the current models also do multi-step mathematical reasoning, write and debug code, read handwritten lab reports from a phone camera, generate complete illustrated slide decks from a single prompt, solve physics problems across several steps, call external tools to fetch real data, and explain their reasoning well enough for another model to check it. PowerPoint assignments are over as a measure of anything. Handwritten submissions are no longer a shelter. What has your course been assuming is still private territory? Go check. The list will have surprised you twelve months ago. It will surprise you again twelve months from now.

5. What is the practice that will keep you honest as the capabilities move?

The reframe is not a one-time architectural move. The capabilities boundary is moving fast and will keep moving for the rest of your career. What is "resistant" this term will be vulnerable by next spring. The question is not *what is the right design* but *what is the practice that keeps me honest about my course as the ground shifts*. The answer is a habit: once a term, you sit down with your assessments and the newest model and you do the work. It takes an afternoon. It is the cheapest professional development in your career, and the only one in this area that does not go stale.

Close the book. Open your course. Do the exercise. The chapter cannot help you past this point — the answers live in the specific shape of your own teaching, and they will not be the same as your colleague's. The alternative is that your assessment continues to run, your students continue to turn in work that costs

them nothing, the detector tells you what to grade, and the only thing the degree certifies is that everyone involved managed to maintain the performance. That is also a choice. It is the choice most institutions are currently making by default. You do not have to make it.

Exercises

Exercise 1 — Induce a false positive in yourself

Goal: experience the detector failure directly.

1. Write a 200-word paragraph on a topic you know well. By hand, without AI.
2. Run it through three free AI detectors (GPTZero, ZeroGPT, Sapling).
3. Record the result from each.
4. Now generate a paragraph on the same topic with ChatGPT.
5. Run it through the same three detectors.
6. Finally, take your own original paragraph and paraphrase it lightly — change three or four words, reorder one sentence. Re-run the detectors.

Questions: - Did any detector flag your own human text as AI-generated? (False positive.) - Did any detector flag the machine text as human? (False negative.) - Did the paraphrase of your own text change the result? - Would you stake a student's grade on this system?

Exercise 2 — Can AI write like you?

Goal: see how much of your own style a model can imitate — and what that says about detection.

1. Take a piece of your own writing, around 500 words — a fragment of an article, a longer email, some lecture notes. It should reflect how you actually write.
2. Give it to a chatbot with this prompt:

Analyse the writing style of this text. Identify: sentence length, vocabulary, tone, use of connectors, level of formality, and any stylistic tics you notice. Then write a 200-word paragraph on [a different topic] imitating exactly this style.
3. Read the output. Observe:
 - Would you recognise the text as yours if you did not know it was generated?
 - Which features did it capture well? Which did it miss?
 - Run your original and the imitation through the detectors from Exercise 1. Which one gets flagged?
4. Now ask the chatbot to write on the *same topic* as your original, but without having seen it — just from a generic prompt. Compare the two: the style-imitation and the topic-only generation.

Questions: - If the model can imitate your style, what does that say about “sounds like AI” as a detection criterion? - If your own text and the imitation produce similar detector scores, who has the problem — you or the detector?

Exercise 3 — Draft your course’s AI policy

Goal: build a clear, communicable policy on AI use for one of your courses.

1. List every assessable task in the course — practicals, exams, essays, presentations.
2. For each task, decide: **allowed / allowed with declaration / not allowed**.
3. For “allowed with declaration” tasks, define what the student must declare (which tool, for which part, how they verified the output).
4. For “not allowed” tasks, define what alternative evidence you will ask for (oral defence, version history, staged drafts).
5. Write the policy as a one-page document in a register a student can read without consulting a lawyer.
6. Show it to a colleague in your department and ask: “*Is this clear? Could you apply it to your own course?*”

Deliverable: the one-page policy. If you do this as a group, compare policies across the group — you will probably find that the differences between subjects are legitimate, but the tone and structure can be shared.

References

- [1] Alier, M. (2023). *How can AI and ChatGPT change the way we assess higher education?* Keynote, A3ES “Assessment in the AI Era” webinar, 2 November 2023.
- [2] Bjork, R. A., & Bjork, E. L. (2011). Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. In M. A. Gernsbacher, R. W. Pew, L. M. Hough, & J. R. Pomerantz (Eds.), *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society*.
- [3] Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3), 363–406.
- [4] Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can AI-Generated Text Be Reliably Detected? *arXiv preprint arXiv:2303.11156*.
- [5] Stephenson, N. (1995). *The Diamond Age: Or, A Young Lady’s Illustrated Primer*. Bantam Books.
- [6] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285.
- [7] Turnitin. (2023). AI writing detection: Educator FAQs. *turnitin.com*.
- [8] University of British Columbia. (2023). UBC decision on Turnitin AI detection. *provost.ubc.ca*.
- [9] Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., et al. (2023). Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(26).

This is the third chapter of the unfinished book — AI in education, what do we do now? — (title in progress). We are releasing this chapter because of its urgency and because it holds on its own. And if George Lucas started with Episode 4, we can start with Chapter 3.

By Marc Alier Forment

Institut de Ciències de l'Educació — Universitat Politècnica de Catalunya

Licensed under Creative Commons BY-NC-SA 4.0 — Attribution, Non-Commercial, Share Alike.
Except the *Math teachers protest against calculator use* image (AP photo, 1986), which may be subject to copyright.

Acknowledgements. Thanks to Maria José Casañ, Juanan Pereira, Faraon Llorens and Francisco García Peñalvo, for early reviews and many insights contained in this chapter, gathered on many conversations and shared publications.

<https://lamb-project.org>

About AI use in the creation of this chapter. This chapter is the outcome of a long process of pair research and writing with Claude Code, Claude Opus 4.6 - 4.7 and Ollama running Qwen3.5 122b as research tools, drawing on all the materials, videos, papers and keynotes I have published over the years. The use of AI has not improved my prose, but it has helped a lot in exploring my own ideas, validating facts and sources, and providing views of the structure and the messages bottled in the following pages. All the mistakes and views expressed are mine.