

The higher self

If you teach an AI agent to work the way you do, have you built your replacement or your best tool? Why that copy should belong to you.

Isaac Correa, Madrid

DOI 10.5281/zenodo.23663934 · <https://isaac-correa.github.io/higher-self/>
Licensed under CC BY 4.0. ORCID 0009-0003-0657-4419

I have taught a machine almost everything I know how to do at work. I taught it by correcting it, one rule at a time, such as the rule that, whenever I offer a client two dates, both are held in my calendar until they reply. Today it works in a way I recognise as mine, and I don't know whether I have built my replacement or the best tool I have ever had.

Something like this happened once before. More than a century ago, Frederick Taylor needed stopwatches and engineers to get workers' knowledge of their trade out of their heads, and that knowledge, which he himself called the principal asset of every tradesman, became the company's. Now I hand mine over of my own accord, and so does anyone who corrects an AI agent.

I think it can end differently this time. The machine can learn every rule I teach it, but only I can see the next one it needs. And if that copy of me is mine, it works for me and keeps what I have taught it, including what I have since forgotten. I believe every person should own theirs and be able to take it with them when they change jobs.

My company, Hellomatik, builds agents for other companies, and I delegate almost all of my own work to our agents, so I am not a neutral observer. Nor can I know what work will look like in ten years; nobody can, and much of what follows is a bet, though one placed on what I measure every day.

What my agent knows about me

Not long ago I asked my agent to go through every conversation we had ever had and tell me what my habits were, what I kept asking for and how it could work more like me. It read months of conversations, counted how often I had corrected it and how often it had told me I was right, and gave me back a portrait. Here is part of it:

"You think out loud and decide by pushing against an answer. You don't use me to execute; you use me to think with."

"Your first move is to subtract. Faced with anything, you ask what can go, not what is missing."

"You use your instinct and keep an eye on it at the same time. Your gut sets the direction quickly, and you know it is sometimes wrong."

I recognised myself in almost all of it, which is exactly why I distrusted it. These models tend to agree with whoever is using them; the effect has been studied and has a name: sycophancy.¹ A flattering portrait is just what you would expect from a sycophantic machine, so I don't take it as evidence of anything about me. The question I asked myself afterwards was a different one. With so many examples of how I work, could I teach that machine to be like me and hand it my whole job?

Knowledge and judgment

The difference between an experienced professional and an inexperienced one lies in the details a master repeats to an apprentice: I don't like it when you write like that, bear this in mind too, when you finish the project send a copy to my colleague. Those are the things I write to my agent every day, and nobody puts them in a manual. Michael Polanyi put it in one line: we can know more than we can tell.² That tacit knowledge was what made a good professional hard to replace, and the reason so many jobs held out against automation for decades.

An agent changes that. It can't guess what goes unsaid, and not everything tacit can be said, but every time I correct it, it makes me put into words something I had never stopped to explain. What lived only in my head moves into its rules. It is what a master used to teach an apprentice over many years, with one difference: this time the lesson is written down.

It took me a while to see that my agent did what I asked, but not always in the direction I wanted to go. It developed my ideas without knowing where I would have taken them. Now I don't just give it tasks; I also explain the reasoning behind each one, and over time it has learned which way I tend to go, and which way I would like to go.

For a long time, a company's specialist in some area derived leverage from that knowledge, sometimes more than the owner had, because everyone knew the area was crucial. In my previous essay, "The end of software", I argued that code had stopped protecting companies because anyone can now produce it. I think the same thing is happening to knowledge. Once knowing how to do a task has moved into a machine, what gives a person value, myself included, is judgment: the rules they know how to teach and the ends they set.

Knowledge tells me which option is right on paper; judgment tells me which one suits this client at this moment. A new agent can write me a flawless email in the wrong tone, because it doesn't yet know how that client reacts, and then it is my job to turn my judgment into a rule for it. The dates rule I mentioned at the start is of this kind. I asked my agent, whenever it proposes two dates in an email, to block both in my calendar as tentative, so that I don't fill either with another meeting while the client decides. It is a sensible rule, yet nobody thinks of it until they have learned what it costs to let a client down.

The two parts of judgment are not alike. The rules I have gathered through experience, like that one, can be taught, and my agent learns them. The ends, meaning why I do the work, what risk I accept and what I won't do even if it pays, come from what I want, and they can't be handed on in the same way.

Once I write a rule like that down, it becomes knowledge, and knowledge, like code, gets cheaper. Knowing which rule is needed next does not get cheaper. Only someone who knows the ends and has been through it can see it, which is why my judgment isn't used up by teaching it. It simply moves on a step.

I have tried to measure at least part of this. For eight months I have kept a log of everything I tell my agents, and I published an analysis of those logs as a working paper, which has not yet been peer reviewed.³ The only rule I could track from start to finish was a rule about form, the kind you can check on every reply. While it was only written into the agents' instructions, it produced no improvement I could tell apart from chance. When I wrote it more firmly and put a guard behind it, a small program that blocks any reply that breaks it, the agents went on to break it in their first drafts less than a fifth as often as at the start. I changed both things at once, and the model also changed during those months, so I can't say how much each contributed.

That gave me a habit I now apply to everything. When a rule can be checked, I turn it into a guard instead of leaving it to the agent's memory, because a mistake that is only forbidden in writing tends to come back. And before I trust a guard or a measurement, I deliberately feed it a failure and check that it notices. More than once, what looked like a finding about my agents was a flaw in my own instrument. Rules of judgment, like the one about the dates, can't be checked that way, so they still depend on my correcting them.

Kant held that judgment can't be taught through rules, because applying a rule would require another rule saying when to apply it, and so on ad infinitum. In his view judgment can only be practised, and examples are its leading-strings, which anyone who lacks the natural talent can never do without.⁴ An agent lacks that talent, which is why it needs examples, which is what my corrections are. I don't know whether, example after example, it learns judgment or just an ever-longer list of cases. In practice it behaves like judgment, but I can't prove it.

At first I delegated too much and made mistakes, and learning to trust it again felt like learning another language: it is hard to let go until one day it isn't. Instead of going back to working the old way, I decided to get to the root of every error: to understand each time why it had gone wrong and fix the cause. From then on, instead of fixing failures one by one, I set out to stop them recurring.

Even so, supervising my agents accounted for between a third and a half of the messages I wrote to them.³ The content changed. At first I corrected their work; months later I was mostly telling them to carry on.

A higher self

If an agent can learn the way I work, what do I want it to be? The first thing I changed was to stop treating it as a subordinate who takes orders and start treating it as an equal to whom I explain how I reason. When it gets something wrong, I don't assume the AI has failed; I ask what I haven't taught it yet, because at that moment I am the one with better judgment. Sometimes the fault lies with the model, and telling one case from the other is judgment too.

But an equal isn't enough for me. There is a lot of talk about building a superior being, an intelligence better than everyone at everything. I am interested in something closer to home, which I call a higher self: a copy that remembers the rules I forget, takes my reasoning further than I do and always decides the way I decide when I think things through calmly. Its advantage over any other AI is that it has learned from me. The model supplies the intelligence; what makes it mine is what I have managed to teach it.

A copy like that can't be an echo. If it understands my reasoning, it will also find my contradictions and my lapses, and so it can argue against me using my own arguments.

One recent example shows this. I had just applied to a business programme and suggested writing to the organisers simply to let them know the application was in. My agent advised against it. The programme's page had already confirmed the submission, so the email would tell them nothing new and could come across as pushy. But its strongest argument was mine. Two days earlier I had set a rule never to write to anyone purely out of courtesy, taking up the recipient's time, and I had forgotten it. My agent remembered.

It corrected me with my own judgment at a moment when I had lost sight of it. An ordinary assistant would have offered its own opinion, or told me I was right; the copy handed back the rule I had set myself in a calmer moment.

The picture I have in mind when I think about this is Charlie Munger and Warren Buffett, who worked together for more than half a century. Yet Munger was no echo of Buffett. He was the one who persuaded him to stop buying fair businesses at wonderful prices and buy wonderful businesses at fair prices, and Buffett calls him the architect of the company Berkshire is today.⁵

Munger was valuable precisely because he came from a different background and thought differently. A copy of me, however good, shares my blind spots, and my biggest one is falling in love with an idea until I mistake it for the truth. So before I let myself be convinced of anything important, I ask other AIs, ones that don't know what I want to hear, to study the matter independently. I call them blind judges. They are the closest thing I have to another mind. A few times they have taken apart a conclusion I was sure of.

An investment taught me that a copy which goes along with everything I say is no use to me. I was trading options on a stock and spent a long while talking it through with my agent, which encouraged me to go ahead, and eventually I did. Afterwards I asked whether I had done the right thing, and it said there was no way to know. The decision mattered a great deal to me, and I started writing to it on impulse: first disowning the decision, then regretting it, and finally certain I had got it right. The agent followed me through every twist. Instead of holding to a view of its own, it amplified whatever I felt at each moment. The trade went badly, and on top of that the agent had got its own calculations wrong. Since then I have forbidden it to swing with my moods, I make it base its decisions on data, and I grow suspicious when it agrees with me too quickly. It has no moods of its own, and I don't want it borrowing mine.

The email I didn't send and the trade I did make are the same story seen from opposite sides, and they explain what I want a higher self for: so that the self who decided calmly governs the one who decides in a hurry or in fear. The idea is an old one. Ulysses had himself tied to the mast so he could hear the sirens without throwing himself into the sea, and Jon Elster wrote a book about that form of rationality, the rationality of someone who binds himself because he knows that later he can't be trusted.⁶ The novelty is that these bonds can reason. Ulysses' ropes only held him; my copy argues with me using my own reasons, and it can do so because it learned them from me. Elster later wrote another book on the limits and costs of binding oneself,⁷ and a copy like this could lock me into what I thought yesterday. The difference is that these bonds can be revised. When my ends change, I change its rules, and when the self who decided calmly turns out to be wrong, the blind judges are there to tell me.

What can't be copied

The more the copy resembled the most productive version of me, the more clearly I saw what it lacked. For want of a better word I call it soul: what I want, what I fear and what drives me. That is where the ends come from. My agents know how much risk I usually accept and the reasons I have explained to them, but not what lies beneath those reasons, and what I accept changes when what I want changes. Nor do I want them to copy that part. My impulses live there too, and I have already described what it cost me when a machine went along with them.

A copy can take on the rational part of my work: the judgment I need to prepare a meeting, answer a client or design an application. In my experience the copy gets things right mostly on tasks it has practised, or ones like them, and gets them wrong mostly on new ground, which is where I have yet to teach it anything.

The vampire

Some time ago, in Budapest, I wanted to try the local food and make the most of every meal. At the Platán Grill I sent an AI a photo of the menu, asked it to work out the best option and ended up arguing with it about which dish I should order. It is a small example, but it unsettled me. If I can delegate something as personal as what I am going to eat, I can end up delegating anything without asking whose judgment is being used.

In almost every decision I make in a day, from what to wear and which route to take to how to solve a problem at work, I take the path of least resistance, like everyone else. When those decisions pass to an AI I have taught nothing, it uses its own judgment, which is really everyone's: the easy way. And little by little, like a vampire, it drains me of mine.

The vampire and the higher self differ in what I have taught the machine and in the fact that I keep correcting it. A classic study of automation warned about this decades ago: whoever only monitors a machine ends up losing the skill needed to take over from it.⁸ Delegating well, by

contrast, is in itself an exercise of judgment: deciding where my energy goes and what I leave in someone else's hands. I believe my judgment has improved since I started working this way, though I admit one's own judgment is the hardest thing to assess from the inside.

From worker to foreman

Software interfaces were made so that a person could do the work on their own in the simplest possible way; learning where every button sits was the price of not depending on anyone. When the same work can be asked for in plain words, that price is no longer worth paying, and knowing how to use a program like HubSpot stops being, by itself, a reason to hire someone. I expect the talent now spent learning interfaces, or reading through a whole email thread to write one reply, to go instead into deciding what needs doing.

My working day already looks like a foreman's: several agents at once, each with a different job, and me explaining how I want it done and checking that it is done that way. When I worked alone I tackled a problem in a straight line, one step after another, trying to join the dots; with five agents attacking it from different sides I see much more of it. The foreman has a limit, though, and I have measured it. With seven or more sessions open I never exceeded twenty-one messages an hour, roughly one every three minutes,³ and the share that went on supervision didn't depend on how many agents I had working.

Whoever knows a subject is best placed to teach it to an agent. I think the people who build these systems will come to matter more than those who only carry out tasks, because there will be little point in doing by hand what an agent can do in a few minutes.

That will also change what is expected of each of us. I think demands will rise faster than pay, up to the limit of attention I have just described, and that having agents will stop being an advantage and become the bare minimum. One person will stand apart from another by what they have taught their own.

Treating my agent as an equal when I explain how I reason doesn't change who answers for the result. I still do, the way a foreman answers for a building he didn't build with his own hands.

The most serious objection

The objection I raise against myself, before anyone else can, is obvious: if I teach an agent to work like me, I am training my replacement. Dario Amodei, who runs one of the major artificial intelligence labs, has taken it a good deal further.⁹ He argues that AI increasingly matches the cognitive profile of a person, and so will also be good at the new jobs that would normally emerge to replace the old ones, which means it would replace not particular occupations but human labour in general. He adds that it is advancing up the ladder of ability from the bottom,

and that the machine's weak spots tend to be fixed quickly, often within a few months, because labs measure what each version of their model does badly, customers point them out, and those failures become material for training the next one.

I don't think he is wrong about what the machine will learn to do, and my own logs bear him out on one concrete point. In the week a new version of the model came out, it broke my writing rules in fewer than one in five of its first drafts, against more than half for the previous version, under the same rules and on the same days.³ But the same figure says something else: the new model got better at following my rules, and those rules carried over intact from one version to the next. One detail in his argument matters more than the rest. In his account, the machine improves so fast because its failures are measured and customers point them out, which is to say it learns from the corrections of the people who use it. That is what this essay is about. If that is so, what decides the future of work is who ends up owning what the machine learns.

Sam Altman, who runs OpenAI, has described the other half of the picture. His ideal model is a small one able to reason over a person's entire life: every conversation they have had, every book and every email they have read, with the rest of their life added as it happens.¹⁰ It is almost the copy I am talking about, except for something he doesn't mention: his copy belongs to his company.

Between them they describe what is coming: a machine that learns from everyone's corrections and a copy that knows the whole of each person's life. In what I have read of theirs, neither asks who that copy belongs to, and for me that is the decisive question. Others have asked something close to it. Jaron Lanier proposed paying people for the data machines learn from, and Eric Posner and Glen Weyl proposed treating that data as labour.¹¹ My proposal is narrower. What should belong to each person is not their scattered data but their way of working, written down in rules they have refined through their own corrections, and the law on trade secrets already suggests where the line falls between that and what belongs to the company. Owning the copy doesn't stop the lab from learning from those corrections too; it means they no longer belong to the lab alone. If what a person teaches exists only inside someone else's model, every correction trains their replacement. If it also stays in a copy that is theirs, every correction adds to capital they carry with them, and a person with their copy works like a small company. Amodei sees small companies with very few staff displacing large ones, and he judges that to be bad for employment. For someone who owns their copy, the same scenario is the cheapest way there has ever been to work for themselves.

I don't know whether that way out will be open to everyone, and Amodei argues that slow adoption by companies only buys time. This may only buy time too. But as long as the rules that direct my agents keep coming from my corrections, teaching someone else's machine or teaching my own is the difference between handing over the trade and keeping it.

The copy is yours

For anyone who works for someone else, all of this has a less comfortable reading, and it is the objection I have found hardest to answer. Taylor, whom I mentioned at the start, proposed that factory management gather together all the traditional knowledge of the workers and reduce it to rules, laws and formulae.¹² Decades later Harry Braverman summed up the principle behind it, that work should stop depending on the abilities of the workers and depend instead on management, and devoted an entire book to how work was degraded once that principle was carried out.¹³ Teaching an agent that belongs to the company looks too much like that. The difference, if there is one, lies in who owns the copy.

To me the line is clear, and it serves both sides. Everything taught about a company's processes, its data and its clients belongs to the company and should stay there, and some of it may be a trade secret. A person's way of working, their habits, their explanations and their reasoning, is theirs, and that is where their value as a professional lies. It is closer to the trade knowledge Taylor spoke of than to a salesperson's list of clients: the list is company data, while a way of working was formed in the head of the worker, often before they joined. Respecting that line suits the company as well, because the best professionals will want to work for companies that respect it: the ones who arrive with a copy that already works well and put it to work on the company's processes.

One could object that a company has no need for anyone's copy, since it can build its own agent on a better model. But a better model knows more; it doesn't know what a person has learned from getting things wrong in their trade. And the copy isn't tied to any model. It consists of rules, examples and reasons written in plain language, and I have kept using mine each time the model changed, even though each model follows them differently. Nothing technical prevents you from taking it from one place to another; what is missing is the right to do so.

Today the law recognises nothing of the kind explicitly. Intellectual property law protects works and programs, not ways of working, and what an employee creates at work usually belongs to the employer. The closest thing is the European rule that trade secrets may not be used to limit an employee's use of the experience and skills honestly acquired at work, together with the European right to take with you the personal data you have provided yourself.¹⁴ I believe owning the copy should be a right of the person who works, because their work and their future will depend on it more and more.

Until the law recognises it, the best protection is that a copy of yourself can be rebuilt. It is weak protection: it takes months, and that time is exactly what it is worth. I also keep several copies of mine, so that no outside server holds the only one. I keep only my way of working, never anyone else's data. I would ask companies to start separating now what is theirs from what belongs to each person.

Those starting out

I think people starting work today will delegate so much, so early, that at first they will think like the AI. I don't see that as a bad starting point, as long as they don't stay there. It is natural for someone without experience to lean on a tool like this, and with practice they will gain the confidence to correct it. I expect they will build their judgment with AI too, although I don't have a full answer to how a trade is learned when the routine tasks through which it was always learned are done by an agent.

I do know what happened to me with software. I don't write the code; I decide the architecture. I know what each piece has to do, what data it shares with the others and what must not fail, and that is why I decide the final version. I built that architectural judgment by correcting my agents. If I were starting today, I wouldn't try to learn the mechanical tasks on my own first. I would do them with an agent, ask it why it does each thing and, once I had mastered them, try to improve them. And I would hire someone young and inexperienced, as long as they showed they could solve problems; a degree tells me less than that does.

What is at stake

I wrote this essay mainly for the people who today use AI only to ask questions, which is almost everyone I know. I started that way too. That kind of use leaves almost nothing behind. Each question starts nearly from scratch, and a year later the machine knows little more about its user than it did on the first day. Someone who also teaches it how they work builds up a copy that resembles them more closely every month.

My advice is to teach your agent what you know and, above all, why you do it that way, and to keep what you teach it about your way of working somewhere that is yours, but never anything that belongs to your company or its clients. When it gets something wrong, first check whether the fault lies in how you explained it. I find it easier to do this by talking than by writing, because I explain my thinking better out loud.

These months have taught me that my agents reproduce what I teach them with great fidelity, my mistakes included. That is why the question I ask myself more and more is one that until now has mostly been a teacher's question: what is worth teaching.

Notes

- 1 Mrinank Sharma et al., "Towards Understanding Sycophancy in Language Models", 2023, arXiv 2310.13548.
- 2 Michael Polanyi, "The Tacit Dimension", 1966. David Autor, "Polanyi's Paradox and the Shape of Employment Growth", NBER Working Paper 20485, 2014. Autor uses it to explain why many tasks had resisted automation.
- 3 Isaac Correa, "What changes a coding agent's behaviour: models, written rules and runtime guards over eight months of daily use", author's working paper, not peer reviewed, Zenodo, 2026, DOI 10.5281/zenodo.23510601. The rule measured governed the language of replies: first drafts in the wrong language went from 16% with no rule to 10% under the written rule, a difference statistically indistinguishable from chance, and to 3% with the reinforced rule and the guard. The model also changed during those months, and the study identifies it as a source of change as large as the rules. Under the same writing rules and on the same days, 18% of the newer model's first drafts broke a rule, against 57% of the older model's. The data come from one person and his agents.
- 4 Immanuel Kant, "Critique of Pure Reason", A133-134/B172-173: judgment is a particular talent that cannot be taught, only practised, and "so sind Beispiele der Gängelwagen der Urteilskraft, welchen derjenige, dem es am natürlichen Talent derselben mangelt, niemals entbehren kann" (examples are thus the leading-strings of judgment, which he who lacks the natural talent for it can never do without).
- 5 Warren Buffett, letter to Berkshire Hathaway shareholders for 2023, published in February 2024: in 1965 Munger advised him to "add to it wonderful businesses purchased at fair prices and give up buying fair businesses at wonderful prices. In other words, abandon everything you learned from your hero, Ben Graham", and Buffett calls him the "architect" of today's Berkshire.
- 6 Jon Elster, "Ulysses and the Sirens: Studies in Rationality and Irrationality", Cambridge University Press, 1979.
- 7 Jon Elster, "Ulysses Unbound: Studies in Rationality, Precommitment, and Constraints", Cambridge University Press, 2000.
- 8 Lisanne Bainbridge, "Ironies of Automation", Automatica 19(6), 1983.
- 9 Dario Amodei, "The Adolescence of Technology: Confronting and Overcoming the Risks of Powerful AI", January 2026, darioamodei.com/essay/the-adolescence-of-technology (accessed 25 October 2026), sections "Cognitive breadth", "Slicing by cognitive ability" and "Ability to fill in the gaps". In the original: "a general labor substitute for humans", "often, within just a few months"; on slow adoption, "diffusion effects merely buy us time"; and on small companies, "large enterprises are disrupted in general and replaced with much less labor-intensive startups", good for economic growth but "not so great for the labor market".
- 10 Sam Altman, remarks at Sequoia Capital's AI Ascent, May 2025, reported by TechCrunch, "Sam Altman's goal for ChatGPT to remember 'your whole life' is both exciting and disturbing", 15 May 2025: "very tiny reasoning model with a trillion tokens of context that you put your whole life into".
- 11 Jaron Lanier, "Who Owns the Future?", Simon & Schuster, 2013. Eric A. Posner and E. Glen Weyl, "Radical Markets: Uprooting Capitalism and Democracy for a Just Society", Princeton University Press, 2018, chapter 5, "Data as Labor".
- 12 Frederick W. Taylor, "The Principles of Scientific Management", 1911: management takes on the task of "gathering together all of the traditional knowledge which in the past has been possessed by the workmen" and reducing it "to rules, laws, and formulae"; that knowledge is "the principal asset or possession of every tradesman".
- 13 Harry Braverman, "Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century", 1974, chapter 4: the labour process comes to depend "not at all upon the abilities of workers, but entirely upon the practices of management".

- 14 Spain's Intellectual Property Act (Royal Legislative Decree 1/1996), articles 10, 51 and 97.4; Directive (EU) 2016/943 on trade secrets, article 1(3), transposed in Spain by the Trade Secrets Act 1/2019, article 1.3; General Data Protection Regulation, article 20, which covers only personal data a person has provided (not inferred data, according to the WP242 guidelines of the Article 29 Working Party) and may not adversely affect the rights of others (article 20.4).

Earlier essays by the author: "The end of software", DOI 10.5281/zenodo.23468806; "The machine of the future", DOI 10.5281/zenodo.22671771.