What ChatGPT Still Can't Do (But We Might Do With It): Hubert Dreyfus and Extended-Mind Cyborgs

Accepted for publication in Katlin Pulk and Riina Koris (eds.). 2025. Generative AI in Higher Education: The Good, the Bad, and the Ugly. Edward Elgar Publishing. Cheltenham, Glos.

2. What ChatGPT still can't do (but we might do with it): Hubert Dreyfus and extended-mind cyborgs

Wayne Martin and Deidre Williams

In the early 1970s, Hubert Dreyfus published a controversial book called *What Computers Can't Do* (Dreyfus 1972). Two decades later, he published *What Computers Still Can't Do* (Dreyfus 1992). Dreyfus died in 2017, before the widespread availability of generative AI tools using large language models. But it is instructive to consider how his "Critique of Artificial Reason" relates to the latest generation of AI, and what light the new tools cast on the structure and limitations of his critique. We begin with a review of the historical record, clarifying what Dreyfus did, and what he did not say, about the limits of machine intelligence. We adjudicate the controversy over his assessment of the prospects for computer chess and provide an analysis of the master argument upon which his critique of AI relied. In assessing the significance of that critique, we turn our attention to Deep Blue, IBM's chess-playing supercomputer that defeated the then-reigning world champion in 1997. We argue that one element of Deep Blue's programming architecture points towards a neglected alternative in Dreyfus' disjunctive argument about the limits of artificial intelligence and an affinity with recent generative AI tools. We introduce the concept of *extended-mind cyborgs* to reflect on the opportunities and limitations of programs such as ChatGPT.

2.1 FACT-CHECKING

Hubert Dreyfus was the most prominent philosophical critic of AI of his generation. A few weeks before Dreyfus' death in 2017, his long-time philosophical sparring partner, Daniel Dennett, gave an interview to the BBC. We begin with an extract from that interview:

> **Jim Al-Khalili (BBC)**: At this point, and having taken on Quine as a student back in Harvard, your next target was Hubert Dreyfus.
> **Daniel Dennett**: That's right. I mean he – unlike some other philosophers, unlike his Berkeley Colleague John Searle – he was prepared to put his

> money where his mouth was in effect and say: "Right; the computers will never play good chess because they can't have intuition." […]
>
> **Jim Al-Khalili**: And of course later, quite embarrassingly, Dreyfus was himself beaten by a computer.
>
> **Daniel Dennett**: Oh yeah, to the great cheers of the AI crowd.[1]

Back in the summer of 1964, Dreyfus, fresh from filing his PhD dissertation at Harvard, spent three months as a researcher at the RAND corporation, the research and public policy think tank based in Santa Monica. His job was to research philosophical issues raised by the prospect of "thinking machines." His findings were presented at a seminar that summer at RAND, and his report was subsequently published (amid considerable controversy) under the RAND imprint. Its title was a clear provocation: *Alchemy and Artificial Intelligence* (Dreyfus 1965).

Dreyfus' RAND report was widely reported at the time; indeed in June, 1966, it was the lead item in "The Talk of the Town" in *The New Yorker*. The *New Yorker* quoted Herbert Simon's notorious 1957 prediction that a computer would win the world chess championship within ten years. It then went on to say: "We've just come across a lovely piece by Hubert L. Dreyfus, a professor at the Massachusetts Institute of Technology, which says computers can't, and won't."[2]

It is worth noting an ambiguity in the *New Yorker* report. Dreyfus is reported as saying that "computers can't, and won't." But the scope of this remark is unclear. Does it mean that computers can't and won't win the world chess championship – *ever*? Or that they can't and won't win the world chess championship *within the next ten years*? The ambiguity makes a difference between an inaccurate prediction and an accurate one. Two decades later this sort of ambiguity had vanished from the press reports, and the position attributed to Dreyfus had taken on an uncanny specificity. Seemingly reputable news sources reported that "In the mid-'60s, scientist Herbert [sic.] Dreyfus bluntly asserted that no computer would ever be able to beat a 10-year-old. He ate his words when an MIT undergraduate wrote a program that beat Dreyfus himself" (Levy 1997; see also Lehmann-Haupt 1984).

So what is fact and what is fiction in all this? Start with a fact. It is true that Dreyfus played and lost to a chess-playing machine at MIT in 1967. The computer ran a program called MacHACK, which was developed by Richard Greenblatt, who had indeed been an MIT undergraduate (Papert 1968, 1.5.1). What about the allegedly "blunt assertion" about 10-year-olds? The source for this myth comes early in the RAND report, in Dreyfus' discussion of the NSS chess-playing program developed in 1958 by Newell, Shaw and Simon. The discussion there follows a pattern that recurs though much of Dreyfus' early

writings on AI. He quotes verbatim predictions made by the NSS team, and then adds this report:

> In fact, in its few recorded games, the NSS program played poor but legal chess, and in its last official bout (October, 1960) was beaten in 35 moves by a 10-year-old novice. (Dreyfus 1965: 6)

Somehow, in the reporting, a singular factual claim about a particular, past, dated event was transformed in the lore into a universal claim about future possibilities.

What about Dennett's claim? Did Dreyfus ever say, "Right; the computers will never play good chess"? This characterization of Dreyfus' position circulated early (see, e.g., Papert 1968: I-6; Nilsson 1969: V-51; Toffler 1970: 187), and Dreyfus responded to it (e.g., Dreyfus 1966: 13; Dreyfus 1972: 223, n 45). One particularly memorable rebuttal came in a joint appearance with Dennett on live national television, the day after Deep Blue had defeated Kasparov.[3] Dennett was in a gloating mood:

> It seems to me that right now is a time for the skeptics to start moving the goal posts. […] A hundred and fifty years ago Edgar Allan Poe was sure in his bones that no machine could ever play chess, and only 30 years ago so was Hubert Dreyfus, and he said so in the earlier edition of his book. Then he's changed his mind, and, as he says, it's – this is really no surprise.

In response, Dreyfus read out the frequently misreported passage from the RAND report, then added:

> I've had to put up for 35 years with this story that I said computers could never play chess. In fact, I said from the beginning it's a formal game, and of course, computers could play, in principle, could play, world-champion chess.

So far, we have been fact-checking Dreyfus' critics. So it's only fair to fact-check Dreyfus as well. Did he actually say from the beginning that "of course, computers could, in principle, play world-champion chess?" To answer this question, we need to turn to the logic of his critique.

2.2 A DISJUNCTIVE CRITIQUE OF ARTIFICIAL REASON

Dreyfus' critique of AI operated with a master-argument that he put to work in a number of case studies. The master-argument is what we'll refer to as an *argument-by-diagnosis*, and it operated in two cycles. The first cycle was descriptive, documenting a recurring pattern. In a series of different

domains (game-playing, puzzle-solving, translation, pattern-recognition …), AI research had produced striking early successes. These successes in turn inspired confident predictions and promises about what would be possible in the near future, but the predictions turned out to be inaccurate and the promises went unfulfilled. This was accompanied by what Dreyfus controversially described as "signs of stagnation" (Dreyfus 1965: 9).

The second cycle in the argument was diagnostic, and comprised two key claims. Significantly, only one of these was about computers; the other claim was about *human* intelligence. The first claim concerned the early AI successes. In the domain of game-playing, early programmers had successfully programmed computers to consistently win or draw at simple games like tic-tac-toe, to play at a high level in more complex games like checkers, and to execute end-games in complex games like chess. Dreyfus' thesis was that these early successes came by focusing on games (or parts of games) that were amenable to an approach that he described as "counting out" – considering possible moves and replies, and then selecting the move with the greatest probability of success. Dreyfus' second thesis was about human beings. Applied to the domain of game-playing, his thesis was that human players rely on methods of information-processing that cannot be reduced to "counting out" (Dreyfus 1965: 21–24).

Taken together, these two claims generate Dreyfus' skeleton explanation of the recurrent pattern of success and failure in AI research. The confident promises of AI researchers had been predicated on the assumption that the approaches that had proved effective in early tests could be scaled up to tackle the more difficult challenges. But those challenges take us into domains where human beings rely on other forms of information processing. Now this certainly does not prove that computers will not succeed at chess by counting-out methods alone. But neither was it intended to provide such a proof. It was offered as a diagnosis (an argument to the best explanation) for a distinctive pattern of success and failure in early AI R&D.

So far, all of that is retrospective. But what, if anything, did Dreyfus' diagnosis imply about *future* prospects – whether for AI in general, or for computer chess in particular? In his report, Dreyfus clearly stated that "a problem can in principle always be solved on a digital computer, provided the data and the rules of transformation are explicit" (Dreyfus 1965: 70). This is of course a general claim, not a claim specifically about chess. But chess clearly meets this condition, as Dreyfus himself pointed out (Dreyfus 1965: 69). Taken together, these claims directly entail that computers can, *in principle*, play chess.

However Dreyfus also described games like chess and go as "formal/complex intelligent activities," which "*in practice* cannot be dealt with by exhaustive enumeration" (Dreyfus 1965: 79, emphasis added). In a table, he

categorized them as "uncomputable games" (Dreyfus 1965: 76, Table 1). At this point we need to proceed carefully. There is no reason to doubt Dreyfus' conclusion that chess cannot *in practice* be solved by exhaustive counting out. In a seminal early paper on computer chess, Claude Shannon famously estimated the total number of possible games of chess as being at least 10^{120}. According to Shannon, a computer operating at 1 MHz would take 10^{90} years to "solve chess" by following out the full decision-tree (Shannon 1950). Modern supercomputers are of course much faster, but nonetheless fail to make a dent in what has come to be known as *The Shannon Number*. But does it follow that chess is an "uncomputable game"? Only if exhaustive enumeration is the only way of rendering a game computable!

Dreyfus himself was keenly aware that some games could be solved without relying on exhaustive enumeration of the decision-tree. For Dreyfus, a key example of an alternate strategy was associated with the game known as *nim*. Nim is a bar game played with matchsticks – now largely forgotten except among a few math geeks and computer nerds. But nim was at various points firmly in the limelight. There had been a nim-craze in cafés in the early sixties, inspired by Resnais' 1961 surrealist film, *Last Year at Marienbad*. More importantly for our purposes: nim was the game played by one of the very first videogame computers, dubbed *Nimrod*, which made a stir at the 1951 Festival of Britain. The rules of nim are simple. In one standard variation, two players take turns removing match-sticks from an array divided into several rows. Each player can remove as many matches as they wish ($\geq$1), but only from a single row. The player who takes the last match wins.

The key point to appreciate is that nim is computable in two different ways. The decision-tree for nim is small enough that a computer (although probably not an unaided human being) could calculate the entire decision-tree, "counting out" out every possible reply to every possible move and then choosing one that yields a sure path to victory. But that was not how Nimrod played. In fact Nimrod did not do any counting out at all. Instead it used a computational shortcut. It calculated the binary XOR sum (now also known as the "nim sum") for each array with which it was presented. Then at each play, it identified a move that would yield a nim sum of zero for its opponent (Author Not Given 1951). This is a provably winning strategy for nim (Bouton 1901).

Nimrod provides an elegant example of game-computability without counting out. But it also teaches us a broader lesson about the logical form of Dreyfus' critique. In order to warrant his description of chess as an uncomputable game, Dreyfus relied on a disjunctive argument:

(1) If chess is computable in practice, then *either* it is feasible to trace the whole decision-tree *or* there is a 'nim-style' computational shortcut.
(2) It is not feasible to trace the whole decision-tree for chess.

(3) There is no 'nim-style' computational shortcut for chess.
∴ (C) Chess is not computable in practice.

Dreyfus did not offer any general argument against the possibility of a computational shortcut for chess. But one strategy to which he did devote attention was the method of "pruning the decision-tree" through heuristics. It is worth reviewing this portion of his argument, which incorporates his boldest claim about human intelligence and clarifies the grounds for his pessimism about AI.

The strategy of pruning the decision-tree for chess was in fact the main proposal in Shannon's original pioneering paper on computer chess. Shannon, who had an engineer's gift for language, dubbed this "the Type-B Programming Strategy." Both Type A and Type B programs operate with an "evaluation function" that assigns a numerical value to any board position. At each turn, Type A programs compute all variations to three moves out and use the evaluation function to rank all the resulting positions. Type-A Programs can play legal chess, but Shannon correctly predicted that play would be "both slow and weak" (Shannon 1950: 269). The strategy for Shannon's Type-B Program was to calculate deeper but on a much smaller subset of the available legal moves. His proposed method for doing so was to build in an additional layer of programming that would distinguish between "forceful" and "pointless" variations. Pointless variations are set aside to avoid wasting computational time; forceful variations are investigated in detail.

Type-B programming proved to be a fruitful strategy. One reason for its fecundity was that it established a research program. Type-B programmers adopted a strategy that later came to be known as *cognitive simulation*. After all, Shannon reasoned, human chess-players don't consider all possible moves. They focus on a small subset of promising moves. They clearly have the ability to distinguish promising variations from pointless ones without counting out plies and replies. So the aim of the research program was to *capture what chess-players already know* in the form of what came to be called "heuristics" – programmable rules that provide a preliminary sorting of promising from pointless moves.

A number of the most important philosophical points in Dreyfus' critique of AI concerned heuristics, which was a topic that continued to occupy him in the research that he and his brother (Stuart Dreyfus) later undertook for the US Air Force Office of Scientific Research (Dreyfus and Dreyfus 1980). One of their observations had in fact already been made by Shannon, who noted the important difference of a logical kind between these heuristic rules and the rules that determine legal moves or the rules used in computation by exhaustive enumeration. The rules of chess hold without exception. But the heuristic that tells you (for example) not to sacrifice a Queen for a Pawn or that "rooks should be

placed on open ranks" are not like this at all. In Shannon's words, they only hold "other things being equal (!)" (Shannon 1950, 261, punctuation original).

Dreyfus went further. He argued, first, that one component in human intelligence consists in knowing when heuristic rules are to be broken. That, indeed, is one of the hallmarks of expertise. In the work for the Air Force, he went on to distinguish the behavior of novices, who do follow explicit rules (and perform poorly), from the behavior of experts, whose performance (they argued) transcends anything that can be captured in fully explicit rules. Applied to chess, their conclusion was that an expert's ability to recognize+ forceful possibilities and disregard pointless ones derives from a form of perceptual attunement to particularities of the chess board itself (Aristotle would have called it *phronesis*), born of thousands of hours of skill-development. It does not arise through, and cannot be captured in, general heuristic formulae operating over symbolic representations.

Let's pause to take stock. What we have found is that Dreyfus did, in one sense, "put his money where his mouth was" by describing chess as an "uncomputable game." However this did not mean that he thought that "computers will never play good chess." But the more important lesson here concerns the reasoning in support of the description of chess as an uncomputable game. What we have found is that Dreyfus' argument for that conclusion takes an essentially disjunctive form. From the point we have reached, we can identify at least three disjuncts. In order to be confident in the conclusion, we would need to be sure that chess is not computable by exhaustive counting out, is not computable by a nim-style computational shortcut, and is not computable by a "Type-B" hybrid that combines heuristic pruning and targeted counting out.

So why did Dreyfus describe chess as an uncomputable game? We suggest that he was rightly confident that there was no feasible method of exhaustive enumeration, and that he thought any alternative approach would have to rely on heuristics. Those heuristics essentially admit of exceptions, precluding the sort of "foregone conclusion" winning strategy that is available for nim. But more importantly, Dreyfus thought that the research program associated with Type-B programming was doomed. Expert players can indeed distinguish forceful from pointless variations. And they can do so without wasting their time counting-out nodes on a decision-tree. But according to Dreyfus, experts will never render that know-how explicit in the form of programmable heuristics, because the knowledge and intelligence that they exercise in such circumstances is not inscribed in rules. It is worth noting, once again, the intricate intermixing here of claims about computers and claims about us.

2.3 EXTENDED-MIND CYBORGS

What lessons might be drawn from all this in thinking about the current generation of AI tools? Allow us to approach this question by considering one further point about computers and chess.

It is often said that Deep Blue triumphed over Kasparov through its sheer brute-force calculations deep into the decision-trees of the games in which it competed. But this is at best a partial truth. It is true that Deep Blue used its enormous computational power within the broad strategic parameters that Shannon had mapped out: it combined an evaluation function with pruning to focus its searches along forceful pathways. But in executing this strategy, Deep Blue incorporated a technique that neither Shannon nor Dreyfus had anticipated. The crucial factor: Deep Blue's memory banks stored 700,000 past games played by grandmasters.

The interesting question for our purposes is about what Deep Blue did with this vast store of information. One point is worth noting immediately: it stored this information symbolically, in a modified form of the binary representation of chess positions that Shannon had proposed. Moreover, its operations over those symbolic representations were implementations of fully explicit algorithmic rules. But we should not therefore conclude that there was nothing new in all this. Confronted with a board position when playing against Kasparov, Deep Blue searched to see whether the position had occurred in any of the stored games. If it had, Deep Blue checked to see what move had been made from that position, and how that earlier game had ended. It also took into account a number of other factors in a weighted formula, including the frequency with which that position had been met with that reply, the rankings of the players who had played it, and whether the move was annotated in "box score" reports of the games in which it had occurred. In its evaluation function, Deep Blue then awarded a bonus to moves that scored highly in this weighted sum. In some cases, where a particular move had been played frequently on paths that ultimately led to victory, Deep Blue followed the proven track without relying on any "counting out" or heuristics at all – reserving precious computation time for later in the game (Campbell 1999; Hsu 1999).

Where should we place this 'extended book' technique against Dreyfus' disjunctive alternatives? It does not involve counting out plies and replies; indeed it was specifically designed to reduce the need for that laborious form of computation, and in some cases to dispense with it altogether. But neither does it implement a nim-style computational shortcut that would render counting out entirely superfluous. Should we categorize it as an example of Type-B Programming? Only at the risk of obscuring its most distinctive features.

What Dreyfus' alternatives did not take into account was a *machine* technique for exploiting *human* intelligence in way that did not involve a detour through propositionally formulated generalizations that lose the fine resolution associated with expert performance.

To appreciate the point, think of the records in Deep Blue's extended book as a *material sedimentation of human intelligence*. Each individual entry records a discrete instance of intelligent behavior by an individual human being; taken as a whole they comprise not only a *representation* but also an accumulated *by-product* of collective human intelligence. Those 700,000 recorded games in turn hearken back to millions upon millions of unrecorded games, by masters and grandmasters, mid-level players, and novices alike, in which gambits and stratagems were tried and refined, sometimes succeeding and sometimes failing. Expert human players build on that history in two different ways. They build on it explicitly when they study earlier games – although of course no human being could possibly study the whole set. They also build on it implicitly. Lessons learned through thousands of hours of play, deploying strategies refined by hundreds of years of competition, come to attune their awareness, allowing them to respond intelligently to patterns they perceive on the board even without being able to articulate fully the reasons behind their selections. Aside from the sheer fact of its victory, the most important fact about Deep Blue was that it managed to harness, in a set of explicit rules operating over fully explicit data, know-how which expert humans access and deploy in quite a different manner.

In this sense, Deep Blue's extended book technique marks a 'neglected alternative' outside the scope of Dreyfus' disjunctive critique, a novel form of collaboration between computer and machine intelligence, and a whole that is greater than its parts. In the extended book, human and machine intelligence each made a distinctive contribution. Human players built up expertise, relying on forms of human intelligence that (if Dreyfus was correct) transcend algorithmic symbol manipulation. Along the way they developed practices and techniques for recording their intelligent behavior. Machine intelligence, itself developed by human beings, was then deployed to interrogate that data and to extract guidance from it in determining how to navigate specific decision-situations. A human being then moved a piece of wood. We think of the resulting system, drawing conjointly on human and machine intelligence in a way that neither could manage without reliance on the other, as an *extended-mind cyborg*.[4]

The resulting technique differs not only from 'counting out' or a nim-style computational shortcut, but also from Type-B heuristics. Heuristics begin from the chess wisdom that expert players are able to articulate and formulate as general, 'other things being equal' rules-of-thumb. *Generating* heuristics involves abstracting from exceptions, so *using* heuristics well requires

an exercise of intelligence in determining when to set them aside. IBM's extended-mind cyborg was not hobbled by these limitations. Crucially, its operations were built directly on the sedimented by-product of intelligent *behavior*, not on the expert's reflections on or generalizations over that behavior. Indeed in any particular instance, neither man nor machine nor cyborg may be able to explain why the endorsed move was preferred. They thereby fall outside Dreyfus' disjunction, and represent a collaborative way for computers to draw on just the kind of inexplicit know-how that he considered a hallmark of human intelligence.

The relevance of this to the current generation of AI tools should now be visible. Indeed in retrospect, one would not be far wrong in describing Deep Blue's extended book as an early example of a "large language model," encoded in the distinctive language of binary chess notation. By contemporary standards, the model may not have been all that large, but the IBM programmers broke important new ground in developing novel ways of exploiting it, using an algorithm to project the next symbol in a string in a context.

We find it helpful to think about tools like ChatGPT not so much as intelligent machines (their claim to that title is certainly open to dispute) but rather as machine components in extended-mind cyborgs. They can do what they cannot do on their own by being embedded in partnerships and teams that are equipped with human intelligence. The human partners play a role in producing and curating the corpora upon which they train, and then in formulating the prompts that produce responses from the machine. Neither are capable on their own of producing the novel products that they can produce together.

What might Dreyfus have had to say about ChatGPT? He would certainly have been keen to document all the many things that it *still can't do*, and he would have taken a mischievous delight in the tasks on which it spectacularly fails, and in its well-documented inability to learn from its own mistakes. But if the arguments that we have advanced here are correct, then he would also have needed to revisit his critique of artificial reason to recognize not only new computability techniques but also a novel form of collaboration between machine and human intelligence.

For our part, we are also interested in what generative AI can and cannot do. Since OpenAI's public release of ChatGPT in November, 2022, we have regularly probed its abilities in part by asking it to construct natural deduction proofs, or to tease out the assumptions at work in a philosophical argument. So far, at least, deductive logic does not seem to be its strongest suit.[5] Ultimately, however, what interests us more are the questions about how best to act in concert with tools like ChatGPT and how to prepare ourselves for the new extended-mind cyborgs that lie over the horizon.

REFERENCES

Author Not Given, *Faster than Thought: The Ferranti Nimrod Digital Computer* (London: Ferranti, 1951).

Bouton, Charles, "Nim: A Game with a Complete Mathematical Theory," *Annals of Mathematics* 3:2 (1901) 35–39.

Campbell, Murray, "Knowledge Discovery in Deep Blue: A Vast Database of Human Experience can be Used to Direct a Search," *Communications of the ACM* 42:11 (1999), 65–67.

Clark, Andy and Chalmers, David, "The Extended Mind," Analysis 58:1 (1998), 7–19.

Dennett, Daniel, *Darwin's Dangerous Idea: Evolution and the Meanings of Life* (London: Penguin, 1995).

Dreyfus, Hubert, *Alchemy and Artificial Intelligence* (Santa Monica, California: RAND, 1965).

Dreyfus, Hubert, "Statement from Prof. Dreyfus," *ACM SICART Newsletter* (1966), 13.

Dreyfus, Hubert, *What Computers Can't Do: A Critique of Artificial Reason* (New York: Harper and Row, 1972).

Dreyfus, Hubert, *What Computers Still Can't Do: A Critique of Artificial Reason* (New York: Harper and Row, 1992).

Dreyfus, Hubert, and Dreyfus, Stuart, *A Five-Stage Model of the Mental Activities in Directed Skill Acquisition* (Berkeley: UC Berkeley Operations Research Center, 1980); USAF Office of Scientific Research Contract F49620-79-C-0063.

Hsu, Feng-Hsiung, "IBM's Deep Blue Chess Grandmaster Chips," *IEEE Micro* 19:02 (1999), 70–81.

Lehmann-Haupt, Christopher, "Hackers as Heroes," *New York Times*, 24 Dec., 1984; Section 1, page 15.

Levy, Neil, "Man Vs. Machine," *Newsweek*, 4 May, 1997.

Nilsson, Nils, *Problem-Solving Methods in Artificial Intelligence* (Palo Alto, CA: Stanford Research Institute Artificial Intelligence Group, 1969).

Papert, Seymour, *Artificial Intelligence Memo No. 154: The Artificial Intelligence of Hubert Dreyfus: A Budget of Fallacies* (Cambridge, MA: Massachusetts Institute of Technology Project MAC, 1968).

Shannon, Claude, "Programming a Computer for Playing Chess," *Philosophical Magazine* Series 7, Vol. 41, No. 314 (March 1950), 256–275.

Toffler, Alvin, *Future Shock* (New York: Bantam Books, 1970).