Consciousness Requires Mortal Computation

Johannes Kleiner[1,2,3,4]

[1]Munich Center for Mathematical Philosophy, LMU Munich
[2]Munich Graduate School of Systemic Neurosciences, LMU Munich
[3]Institute for Psychology, University of Bamberg
[4]Association for Mathematical Consciousness Science

Abstract. All organisms compute, though in vastly different ways. Whereas biological systems carry out mortal computation, contemporary AI systems and all previous general purpose computers carry out immortal computation. Here, we show that if Computational Functionalism holds true, consciousness requires mortal computation. This implies that none of the contemporary AI systems, and no AI system that runs on hardware of the type in use today, can be conscious.

This paper is concerned with the question of whether modern AI systems, including Large Language Models such as GPT, are or could be conscious. We prove that, in contrast to common conception, Computational Functionalism implies that contemporary AI systems are not and cannot be conscious. The reason for this is that contemporary AI systems carry out what Geoffrey Hinton has recently called *immortal computation* (Hinton, 2022), whereas, as we shall see, Computational Functionalism requires *mortal computation*.

1. Computational Functionalism

Computational Functionalism was introduced by Putnam (1967):

1. "All organisms capable of feeling pain are Probabilistic Automata.
2. Every organism capable of feeling pain possesses at least one Description of a certain kind (i.e., being capable of feeling pain *is* possessing an appropriate kind of Functional Organization).
3. No organism capable of feeling pain possesses a decomposition into parts which separately possess Descriptions of the kind referred to in 2.
4. For every Description of the kind referred to in 2, there exists a subset of the sensory inputs such that an organism with that Description is in pain when and only when some of its sensory inputs are in that subset." (Putnam, 1967, 1975, p. 434)

In giving this definition, Putnam equates Probabilistic Automata with *Descriptions* of a system, and the Functional Organization mentioned in 2. is the abstract Probabilistic Automaton (PA) that any concrete specification of a Probabilistic Automaton instantiates (the latter's isomorphism class, that is). Condition 1 "is, obviously, redundant, and is only introduced for expository reasons. (It is, in fact, empty, since everything is a Probabilistic Automaton under some Description.)" (Putnam, 1967, p. 435). In what follows, we only make use of condition 2.

While Probabilistic Automata are intimately connected to computation as presently understood, nothing here hinges on this notion. We can replace it by any conception of computation that, for every system S in a class *Sys* of systems, provides a class $\mathcal{C}(S)$ of computations that the system realizes. In Putnam's terms, $\mathcal{C}(S)$ consists of the Probabilistic Automaton Descriptions of S, and he refers to the systems in *Sys* as organisms.

For any such choice of class of systems, we denote those systems that are capable of having a conscious experience e by Sys_e, and those which aren't by $Sys_{\neg e} = Sys \setminus Sys_e$. In Putnam's definition, e is a form of pain, but the definition is meant to apply to any experience. Making use of this notation, condition 2 reads:

(CF) There is at least one computation c^* such that for all $S \in Sys$,

$$S \in Sys_e \;\Leftrightarrow\; c^* \in \mathcal{C}(S)\,.$$

The equivalence in this formalization derives from the bracket in Putnam's condition. Technically speaking, c^* is more properly denoted as c^*_e, to indicate that according to Putnam's definition, different conscious experiences can be different computations. But we will leave the subscript implicit for notational simplicity in what follows.

2. Mortal Computation

A fundamental tenet of general purpose digital computing, and all major conceptualizations of what computations are, is that software is separated from hardware, so that the same program or algorithm can be run on any suitable system. This tenet is about to be broken. Contemporary developments in AI and chip production suggest that deep learning will make a novel form of general purpose computing available, where the parameter values that define a computation "are only useful for that specific hardware instance" (Hinton, 2022, p. 13).

Geoffrey Hinton has coined the term *mortal computation* for this new form of computation, because in cases where "parameter values are only useful for that specific hardware instance, (...) the computation they perform is mortal: it dies with the hardware" (Hinton, 2022, p. 13). Present-day computation, in contrast, ensures that "[t]he knowledge in a program does not die when the hardware dies (...), so that the same program or the same set of weights can be run on a different physical copy of the hardware. (...) This makes the knowledge contained in the program or the weights immortal" (Hinton, 2022, p. 13).

To provide a formal definition of this concept, we focus on the core intuition behind immortal computation: "that the software should be separable from the hardware" (Hinton, 2022, p. 13). In practice, this separation is enabled by a processing unit's *Instruction Set Architecture* (ISA). An ISA contains specifications of the various computations that the processing unit can carry out, and it is with respect to these specifications that operating systems and compilers are defined. Differences among processing units' performance, design, size, etc., are differences in an ISA's *implementation*. The ISA exists to ensure binary-code compatibility of software despite these differences, it provides a reference relative to which software computations are defined, and which ensures that the program runs on different physical copies of the same type of hardware. In computer science, the ISA is often taken as the boundary between software and hardware.

The crucial property that allows for a separation of software from hardware is that there is a reference relative to which software is defined, and which a class of hardware implements. Computation is immortal precisely because it is defined with respect to such a reference, the ISA in practice. We can formalize this requirement as follows.

(R) A computation c is **defined with respect to a reference** iff there is a class c_{ref} of computations such that every system that can realize c_{ref} can also realize c. Formally,

$$c_{\text{ref}} \subset \mathcal{C}(S) \;\Rightarrow c \in \mathcal{C}(S)\,,$$

for all systems $S \in Sys$.

Making use of references, immortal computation can be formalized as follows.

(IC) A computation c is **immortal** iff it is defined with respect to a reference and all $S \in Sys$ can realize the reference computations. Formally,

$$c_{\text{ref}} \subset \mathcal{C}(S) \quad \text{for all } S \in Sys\,.$$

A computation c is **mortal** iff it is not immortal.

Together, definitions (CF) and (R) imply that if a computation c is immortal, then $c \in \mathcal{C}(S)$ for all $S \in Sys$. The computation

can be realized by all systems in *Sys*, meaning that "the same program or the same set of weights can be run on a different physical copy of the hardware" (Hinton, 2022, p. 13).

3. Main Result

The final ingredient to prove our main result is an assumption which is implicit in Putnam's definition, and which is a necessary condition for Computational Functionalism to make sense:[1] that for the conscious experience e in question (in Putnam's case, the experience of feeling a form of pain), there are systems in *Sys* which are not capable of that experience.

(NC) There are systems which are not capable of having conscious experience e. Formally,

$$Sys_{\neg e} \neq \emptyset\,.$$

Besides being necessary, this condition is also intuitively convincing given Computational Functionalism, because it focuses on individual conscious experiences e, such as a form of pain, or a particular taste, or the experience of a specific form of visual beauty.

We can now prove our main result, where c^* is from condition (CF) of Computational Functionalism.

Thm 1. c^* is a mortal computation.

Proof. Assume, c^* is an immortal computation. (IC) implies that therefore, c^*_{ref} exists and $c^*_{\text{ref}} \subset \mathcal{C}(S)$ for all $S \in Sys$. By (R), this implies that $c^* \in \mathcal{C}(S)$ for all $S \in Sys$. But according to condition (CF) of Computational Functionalism, this implies that $S \in Sys_e$ for all $S \in Sys$, so that $Sys_{\neg e} = \emptyset$. But this contradicts the necessary condition (NC). Therefore, c^* cannot be an immortal computation. It must be a mortal computation. □

The computation c^* *is* the conscious experience e according to Computational Functionalism (cf. the bracket in Putnam's second condition). Therefore, Theorem 2 shows that conscious experience e requires mortal computation. And since we did not make any specific assumptions about e, it follows that any conscious experience requires mortal computation. Therefore Theorem 1 shows that consciousness requires mortal computation.

The intuition behind this result is simple. If consciousness is a computation, and if this computation is immortal in a class of systems or organisms, then every system or organism in that class must be conscious, because immortality means that the computation can be run on any system or organism in that class. This violates a necessary condition for Computational Functionalism to make sense (that there are systems that aren't conscious).

4. Implications for AI Consciousness

Let Sys_0 denote the class of central processing units (CPUs), graphics processing units (GPUs) or tensor processing units (TPUs). All contemporary and near-future AI is immortal computation in that class.[2] If Computational Functionalism is true, Theorem 1 applies and shows that any computation c^* that is conscious must be a mortal computation. Since all AI on Sys_0 is immortal computation, we have the following result:

Thm 2. If Computational Functionalism is true, no AI that runs on Sys_0 can be conscious.

As a consequence, none of our current AI systems and none of the near-future AI systems are or can be conscious.

5. Conclusion

While the question of machine consciousness has been a perennial part of modern philosophy of mind, only with the dawn of LLM-type AI has it left the domains of the ivory tower. It has now acquired a societal dimension that includes questions of morality (Metzinger, 2021) and existential risk,[3] and affects the behavior of millions of users who integrate AI companion personas into their daily and emotional lives.[4]

Most, if not all, of the current efforts to provide reliable answers to the question of AI consciousness rely on theories of consciousness. While, ultimately, this is likely the gold-standard, it is questionable whether contemporary theories of consciousness have sufficient empirical support to give credence to any assessment of AI consciousness with real-life consequences.

The result presented here is an attempt to provide a more reliable answer to the question of AI consciousness. The result does not rely on any particular theory of consciousness or cognitive mechanism; it only assumes that Computational Functionalism holds true. And it does not merely give good reasons or provide intuitions, but offers a no-go *theorem* regarding AI consciousness. The result is notable because a number of contemporary studies that provide indicators in favor of consciousness in near-future AI, for example (Butlin et al., 2023), assume Computational Functionalism.

This result does not settle the question of AI consciousness. While Computational Functionalism is intimately tied to questions of artificial consciousness, it might not hold true, and implications of other frameworks have to be studied as well. In (Kleiner & Ludwig, 2023), we consider a different perspective, that does not presume or imply Computational Functionalism, but leads to a similar conclusion as the one presented here.

The result presented here applies to all AI systems that carry out immortal computation. This includes all noteworthy AI systems that presently exist, and likely includes most noteworthy AI systems that will be built in the near future. It is important to note, however, that developments that push for mortal AI already exist. Geoffrey Hinton's seminal (2022) paper aims to lay the ground for general purpose mortal computation, and developments in that direction are pursued in the context of Active Inference/Free Energy Principle (Wiese, 2023; Ororbia & Friston, 2023), as well as in the semiconductor industry (Le Gallo et al., 2023).

Computation is a major technological paradigm of our times. Therefore, it is not surprising that this paradigm shapes many theories and much thinking about consciousness. The intuition that it is the functional organization that matters for consciousness, rather than "physical-chemical states of the brain" (Putnam, 1967, p. 436), is very strong. But so is the intuition that artificial systems and computers are not conscious (Aru, Larkum, & Shine, 2023; Seth, 2009). Because biological systems carry out mortal computation, whereas computers carry out immortal computation, the result presented here underwrites both intuitions, and points to biology, and the type of computation that biological systems perform, as a source of consciousness.

Acknowledgments. I would like to thank Hanna Tolle, Tim Ludwig, Wanja Wiese, and David Chalmers for valuable discussions on mortal computation, as well as the organisers and participants of the *C3: Complexity, Computers, and Consciousness* workshop at the *Institute of Physics* for valuable feedback on earlier ideas.

Notes

[1]Assumption (NC) is a necessary condition for Computational Functionalism to make sense, for otherwise Computational Functionalism is trivially true. To see that this is the case, suppose that all systems in *Sys* are capable of having experience e, and consider the Probabilistic Automaton (PA) with one state, one output and two inputs, both of which leave the internal state invariant. Every system realizes this PA, and it does so in particular in such a way that the first input of the PA describes all sensory inputs to the system for which the system has experience e, and the second input of the PA describes all sensory inputs for which the system doesn't have the experience e. All physical states of the system and all of its outputs are lumped into one PA state and one PA output, repsectively; if the second sensory set is empty, we can just choose a placeholder symbol instead, and still have a valid description of the system, since the input leaves the state invariant. Therefore, this PA satisfies Putnam's second and fourth conditions. Since condition 1 is empty, it remains to check the third condition. "The purpose of condition 3 is to rule out such 'organisms' (if they can count as such) as swarms of bees as single pain-feelers" (Putnam, 1967, p. 435). If S' is a subsystem of a system S (for example, part of a decomposition of S), then S possesses any PA that S' possesses; the description of S given by the PA simply ignores all parts of S not relevant for S'. Therefore, swarms of bees posses any PA that any single bee possesses. In other words, any system capable of having experience e in virtue of some subsystem possesses any PA which that subsystem possesses. Therefore, condition 3 is a condition on which organisms count as feeling pain, and it is meant to exclude organisms (such as swarm of bees) that can be decomposed into parts that (all) feel pain. If any such organism exists in *Sys*, then $Sys_{\neg e}$ is not empty, and (NC) holds. If no such organism exists, then condition 3 is satisfied and the PA we have constructed above satisfied all four of Putnam's conditions, so that Computational Functionalism is trivially true.

[2] All contemporary and near-future AI is immortal computation w.r.t. Sys_0 because of source-code compatibility. Source-code compatibility ensures that AI

computations can not only be run on all CPUs, GPUs, or TPUs that realize one ISA, but on all CPUs, GPUs, or TPUs that realize *some* ISA, in principle at least.

[3]The idea that consciousness is key for a system's capabilities, and might therefore constitute a special point of interest with respect to existential risk, has been made very vividly by Yoshua Bengio during a keynote at the 26th meeting of the Association for the Scientific Study of Consciousness: "An important aspect of consciousness is also self-awareness and our *self-preservation instinct*: putting that into machines could be very dangerous, introducing a new kind of species that could be smarter than us, posing *existential risks*" (NYU, June 22-25, 2023).

[4] The implications of attributions of consciousness to AI companion personas in light of a "radical societal shift towards ubiquitous artificial social agents" have been beautifully illustrated in a talk by Henry Shevlin as part of the *C3: Complexity, Computers, and Consciousness* workshop at the *Institute of Physics* (November 9, 2023).

References

Aru, J., Larkum, M. E., & Shine, J. M. (2023). The feasibility of artificial consciousness through the lens of neuroscience. *Trends in Neurosciences*.

Butlin, P., Long, R., Elmoznino, E., Bengio, Y., Birch, J., Constant, A., ... others (2023). Consciousness in artificial intelligence: Insights from the science of consciousness. *arXiv preprint arXiv:2308.08708*.

Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*.

Kleiner, J., & Ludwig, T. (2023). If consciousness is dynamically relevant, artificial intelligence isn't conscious. *arXiv preprint arXiv:2304.05077*.

Le Gallo, M., Khaddam-Aljameh, R., Stanisavljevic, M., Vasilopoulos, A., Kersting, B., Dazzi, M., ... others (2023). A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference. *Nature Electronics*, *6*(9), 680–693.

Metzinger, T. (2021). Artificial suffering: An argument for a global moratorium on synthetic phenomenology. *Journal of Artificial Intelligence and Consciousness*.

Ororbia, A., & Friston, K. (2023). Mortal computation: A foundation for biomimetic intelligence. *arXiv preprint arXiv:2311.09589*.

Putnam, H. (1967). Psychological predicates. In W. H. Capitan & D. D. Merrill (Eds.), *Art, mind, and religion.* Pittsburgh: University of Pittsburgh Press. (Reprinted in (Putnam, 1975).)

Putnam, H. (1975). The nature of mental states. In *Mind, language, and reality: Philosophical papers* (Vol. ii). Cambridge: Cambridge University Press.

Seth, A. (2009). The strength of weak artificial consciousness. *International Journal of Machine Consciousness*, *1*(01), 71–82.

Wiese, W. (2023). Could large language models be conscious? a perspective from the Free Energy Principle. *PhilArchive Preprint PhilArchive:WIECLL*.