Computational Dualism and Objective Superintelligence

Michael Timothy Bennett[1]
[0000−0001−6895−8782]

The Australian National University
michael.bennett@anu.edu.au
http://www.michaeltimothybennett.com/

Abstract. The concept of intelligent software is flawed. The behaviour of software depends upon the hardware that interprets it. This undermines claims regarding the behaviour of theorised, software superintelligence. Here we characterise this problem as "computational dualism", where instead of mental and physical substance, we have software and hardware. We argue that to make objective claims regarding performance we must avoid computational dualism. We propose using an alternative based upon pancomputationalism, which defines all aspects of the environment as nothing more than relations between otherwise irreducible states. We formalise systems as behaviour (inputs and outputs, with policy being a causal intermediary), and cognition as embodied, embedded, extended and enactive. The result is cognition formalised as a part of the environment, rather than as a disembodied policy interacting with the environment though an interpreter. This allows us to make objective claims regarding intelligence, which we argue is the ability to "generalise", identify causes and adapt. We then propose objective upper bounds for intelligent behaviour.

Keywords: enactivism · computational dualism · AGI · AI safety.

1 Introduction

AIXI [1] is a general reinforcement learning agent. It uses Solomonoff Induction, a formalism of Ockham's Razor [2], to make accurate inferences from minimal data. It was initially thought to be pareto optimal, representing an upper bound on intelligence [3]. Unfortunately, this claim was later shown to be a matter of interpretation [4]. We argue that this is a valuable insight indicative of a much larger problem with how artificial intelligence is conceived [5]. We call this problem "computational dualism", in reference to René Descarte's interactionist, substance dualism. We then discuss an alternative formulation of intelligent behaviour that permits objective performance claims, and allows us to propose upper bounds on intelligent behaviour. It is based upon enactive cognition [6], pancomputationalism [7] and weak constraint optimisation [8].

Intelligence: Our focus is not AIXI, but on intelligence in general. General intelligence is often defined in terms of predictive models [1, 9, 10], or more specifically the ability to "generalise" and make accurate predictions in unfamiliar circumstances and *adapt*. The reason generalisation matters, as opposed to just predictive accuracy, is because with enough training examples even a lookup table can correctly identify the cause of data and maximise predictive accuracy (because it will have seen every example). Hence, what matters is how *efficiently* one can learn the cause of data. A model can be understood as a program relating aspects of the environment to one another. If we wish to explain a sequence generated by a program, then the "correct" model *is* that program which generated the sequence. Intuitively, a model *explains* the present by identifying those aspects of the past which *caused* it. Using such a model, we might use the more distant past to explain events in the more recent past, and the present to predict the future. It is the future with which we are concerned, as an agent that can accurately predict the results of its actions can choose the actions that yield the most reward. Hence the ability to adapt to unforeseen circumstances, satisfy goals and otherwise behave intelligently can be equated with the ability to identify cause and effect [11]. Of course, a model of the environment is not the environment. Even if we only consider models that explain the past, some will are more likely than others to "generalise" and accurately predict future events.

Simplicity: This is where Ockham's Razor comes in. All else being equal, it implies that simpler models are more likely to generalise [12]. Simplicity is often formalised as Kolmogorov Complexity (KC) [13] or minimum description length [14]. The KC of an object is the length of the shortest self extracting archive of that object. Models with smaller KC tend to make more accurate predictions. This is why some believe that compression and intelligence are closely related [15]. Formally, in the case of AIXI, if the model which generated past data is indeed computable, then the simplest model will dominate the Bayesian posterior as more and more data is observed. Eventually, AIXI will have identified the correct model, which it can use to generate the next sample (predict the future).

Subjectivity: We will now use AIXI to illustrate the problem the idea of a software "intelligence". KC (and thus AIXI's performance) is measured in the context of a UTM [4]. By itself, changing the UTM would not meaningfully affect performance. When used in a universal prior to predict deterministic binary sequences, the number of incorrect predictions a model will make is bounded by a multiple of the KC of that model [2]. If the UTM is changed the number of errors only changes by a constant [16, pp. 2.1.1 & 3.1.1], so changing the UTM doesn't change which model is considered most plausible. However, when AIXI employs this prior in an *interactive* setting, a problem occurs [4]. Intuitively (with significant abuse of notation), assume a program f_1 is software, f_2 is an interpreter and f_3 is the reality (an environment, body etc) within which goals are pursued. According to Ockham's Razor, AIXI is the optimal choice of f_1 to maximise the performance of $f_3(f_2(f_1))$. However, in an interactive setting one's perception of success may not match reality.

> "Legg-Hutter intelligence [3] is measured with respect to a fixed UTM. AIXI is the most intelligent policy if it uses the same UTM." [4, p.10]

If intelligence is measured with respect to one UTM while AIXI runs on another, then this is like AIXI being engaged in one reality, while success is determined by another, entirely different reality. Using our (very informal) analogy of functions representing mind, interpreter and reality, performance in terms of $f_3(f_2(f_1))$ depends upon $f_2(f_1)$, not f_1 alone. Thus a claim regarding the performance of f_1 alone would be *subjective*, in that it depends upon f_2.

> "This undermines all existing optimality properties for AIXI." [4, p.1]

1.1 Computational Dualism

We suggest this problem has a broader significance. The concept of artificial intelligence as a software "mind" interacting with a hardware "body" echoes Descartes' interactionist, substance dualism [17]. Descartes argued mental and physical "substances" interact through the pineal gland which interprets mental events to cause physical events [18], like a UTM interprets software. When later scholars pointed out the inconsistencies implied by this interaction between mental and physical, some argued that the mental "supervenes" on the physical, meaning any two objects that are exactly the same mentally must be exactly the same physically. More recently, philosophers have proposed purely physicalist depictions of the mind [19], and have even gone so far as to formalise cognition not just as embodied, but as enacted through the environment [6]. One might argue that artificial intelligence is the engineering branch of philosophy of mind and cognitive science. Instead of mental and physical substances, we have software and hardware. Software "supervenes" on hardware in the sense that any two computers that are exactly alike in hardware configuration down to the value of each and every bit, must be exactly alike in terms of software. However, we have not moved on from dualism and formalised more recent conceptions of mind. Artificial intelligence still tends to be equated with software. However, unless there exists a Platonic realm of pure math (akin to a mental realm), Hinton's "immortal" computations [20] do not exist and never have. There are only embodied "mortal" computations, because software is nothing more than the configuration of hardware. Our understanding of artificial intelligence should be revised to reflect this. This is what we set out to do here.

Results: We propose the concept of computational dualism, argue artificial intelligence warrants an alternative, discuss one such alternative, and formalise the objective upper bound of intelligent behaviour. Note some of these definitions are shared with related work [8, 21, 22, 23].

2 Formalising Enactivism

An alternative to dualism is enactivism [24] which holds that mind and body are inseparable, embedded in time and place. Cognitive activity extends into the environment, and is enacted through what the organism does. For example, if someone uses pen and paper to compute the solution to a math problem, then cognition is enacted using the pen and paper [25]. Formalising enactivism can address problems associated with dualism. However it is unclear how enactive cognition might work computationally, because it blurs the boundary between the agent and environment. To address this we formalise the environment along pancomputationalist [7] lines, albeit a very minimal interpretation thereof. Pancomputationalism holds that everything is a computational system. Using the analogy from earlier, we're seeking to formalise $f_2(f_3(f_1))$ instead of $f_3(f_2(f_1))$. One may regard the interpreter f_2 as the laws of nature, the environment f_3 as software running on f_2, and we're seeking to portray the mind f_1 as subject to the environment. Because of this, the distinction between mental (software) and physical (hardware) must be discarded. Instead, we describe artificial minds in a purely behaviourist manner [10], describing inputs and outputs rather than focusing on the mechanism by which one is mapped to the other. Policies are then all possible "causal intermediaries" between inputs and outputs, akin to functionalist explanations of human mentality [19].

The environment: Rather than assuming programs, we arrive at a pancomputational model of the environment by assuming first that some things exist, some do not, and that the environment is that which exists. Second, we assume there is at least one state of the environment. States could be differentiated along dimensions like time, but we don't strictly need to assume that. We also don't need to assume anything about the internal structure or contents of states. Instead, we can formalise the set of all declarative programs (programs which return true or false) as the powerset of states. A declarative program is then "true" of every state it contains, and false otherwise. Every conceivable environment or state thereof amounts to a set of true declarative programs or "facts".

Definition 1 (environment).

- *We assume an infinite set Φ whose elements we call* ***states****.*
- *A* ***declarative program*** *is $f \subseteq \Phi$, and we write P for the set of all declarative programs (the powerset of Φ).*
- *By a* ***truth*** *or* ***fact*** *about a state ϕ, we mean $f \in P$ such that $\phi \in f$.*
- *By an* ***aspect of a state*** *ϕ we mean a set l of facts about ϕ s.t. $\phi \in \bigcap l$. By an* ***aspect of the environment*** *we mean an aspect l of any state, s.t. $\bigcap l \neq \emptyset$. We say an aspect of the environment is* ***realised***[1] *by state ϕ if it is an aspect of ϕ.*

Goertzel characterises questions of the mind in terms of unary, dyadic and triadic relations [26], and our approach reflects this idea. The states we define here are

---

[1] Realised meaning it is made real, or brought into existence.

unary, having relations only to themselves. However, a declarative program is dyadic relation between states, in the sense that it refers to states which are its truth conditions. Any set of declarative programs may also be considered dyadic, as a conjunction of its parts. Sets of declarative programs form a lattice based on truth conditions, which relates them to one another. We can then define triadic relations by taking three sets of declarative programs such that one is the causal intermediary between the two others.

Embodiment: We use these dyadic relations to formalise enactivism [6], holding that everything we call the mind is just part of the environment, and "thinking" is just changes in environmental state. This blurs the line between agent and environment, making the distinction unclear. As Heidegger maintained, Being is bound by context [27]. There is no need to define an agent that has no environment, and so there seems to be little point in preserving the distinction. What is needed is not a Cartesian model of disembodied intelligence looking in upon an environment but an embodiment, embedded in a particular part of the environment, through which goal directed behaviour can be enacted. We need to describe the *circuitry* with which cognition is embodied and enacted, much like a **formal language** only with truth conditions determined by environmental states. Just as every computational system we can build is finite, and living systems are ergodic [28], we assume all of this takes place within a system which has a finite number of configurations, and so amounts to a finite number of declarative programs. We call this an "abstraction layer", which intuitively is like a window onto that part of the environment in which cognition takes place. For example, we might enumerate every possible declarative program which pertains only to one specific computer, and use that computer as our abstraction layer. However, as we can take any set of declarative programs and define an abstraction layer, it is a "pancomputational" notion of computer in the sense that it captures every system in computational terms.

Definition 2 (abstraction layer).

- *We single out a subset* $\mathfrak{v} \subseteq P$ *which we call* ***the vocabulary*** *of an abstraction layer. The vocabulary is finite unless explicitly stated otherwise. If* $\mathfrak{v} = P$*, then we say that there is no abstraction.*
- $L_{\mathfrak{v}} = \{l \subseteq \mathfrak{v} : \bigcap l \neq \emptyset\}$ *is a set of aspects in* $\mathfrak{v}$*. We call* $L_{\mathfrak{v}}$ *a formal language, and* $l \in L_{\mathfrak{v}}$ *a* ***statement****.*
- *We say a statement is* ***true*** *given a state iff it is an aspect realised by that state.*
- *A* ***completion*** *of a statement* x *is a statement* y *which is a superset of* x*. If* y *is true, then* x *is true.*
- *The* ***extension of a statement*** $x \in L_{\mathfrak{v}}$ *is* $E_x = \{y \in L_{\mathfrak{v}} : x \subseteq y\}$*.* E_x *is the set of all completions of* x*.*
- *The* ***extension of a set of statements*** $X \subseteq L_{\mathfrak{v}}$ *is* $E_X = \bigcup\limits_{x \in X} E_x$*.*
- *We say* x *and* y *are* ***equivalent*** *iff* $E_x = E_y$*.*

(notation) *E with a subscript is the extension of the subscript*[2]*.*

---

[2] e.g. E_l is the extension of l.

Intuitively, $L_{\mathfrak{v}}$ is everything which can be realised in this abstraction layer. The extension E_x of a statement x is the set of all statements whose existence implies x, and so it is like a truth table. Now that we have an embodiment, we need to define goal directed behaviour. An abstraction layer is *already* goal directed in the sense that it constrains what might be described, just as living organisms evolve to thrive in particular environments. However, an organism can then engage in more *specific* goal directed behaviours by *learning* and adapting to remain fit in more circumstances, and this is what we seek to formalise in an abstraction layer. We do not need a value-neutral model of the environment [10].

> "The best model of the world is the world itself." - Rodney Brooks [29]

The only aspects of the environment that we might actually need model are those necessary to satisfy goals [5]. What we need is a model of a task, or more specifically to enumerate the goal directed behaviour (inputs and outputs) that will eventually cause the task to be completed. Goal directed here just means some outputs are "correct", and some are not. It is arbitrary.

Definition 3 ($\mathfrak{v}$-task). *For a chosen $\mathfrak{v}$, a task α is a pair $\langle I_\alpha, O_\alpha \rangle$ where:*

- *$I_\alpha \subset L_{\mathfrak{v}}$ is a set whose elements we call* ***inputs*** *of α.*
- *$O_\alpha \subset E_{I_\alpha}$ is a set whose elements we call* ***correct outputs*** *of α.*

I_α has the extension E_{I_α} we call ***outputs****, and O_α are outputs deemed correct. $\Gamma_{\mathfrak{v}}$ is the set of* ***all tasks*** *given $\mathfrak{v}$.*

(generational hierarchy) *A $\mathfrak{v}$-task α is a* ***child*** *of $\mathfrak{v}$-task ω if $I_\alpha \subset I_\omega$ and $O_\alpha \subseteq O_\omega$. This is written as $\alpha \sqsubset \omega$. If $\alpha \sqsubset \omega$ then ω is then a* ***parent*** *of α. $\sqsubset$ implies a "lattice" or generational hierarchy of tasks. Formally, the level of a task α in this hierarchy is the largest k such there is a sequence $\langle \alpha_0, \alpha_1, ...\alpha_k \rangle$ of k tasks such that $\alpha_0 = \alpha$ and $\alpha_i \sqsubset \alpha_{i+1}$ for all $i \in (0, k)$. A child is always "lower level" than its parents.*

(notation) *If $\omega \in \Gamma_{\mathfrak{v}}$, then we will use subscript ω to signify parts of ω, meaning one should assume $\omega = \langle I_\omega, O_\omega \rangle$ even if that isn't written.*

Intuitively, an **input** is a possibly incomplete description or task, and an **output** is a completion of an input [def. 2]. We treat **correctness** as binary. An output is correct if it causes the task to become complete to some acceptable degree with some acceptable probability, "satisficing" a goal [30]. Degrees of complete or correct just reflect different task definitions. Preferences that determine what is considered complete, and methods of attributing task completion to past outputs are beyond this paper's scope. For example we might put this in evolutionary terms, where the inputs and outputs are the enumeration of all "fit" behaviour in which an organism might engage, as remaining alive is goal directed behaviour. We now formalise learning and inference of goal directed behaviour as tasks. Inference requires a "policy". Being a set of declarative programs, a correct policy *is* the goal of a $\mathfrak{v}$-task, so this amounts to goal learning.

Definition 4 (inference).

- *A $\mathfrak{v}$-task* ***policy*** *is a statement $\pi \in L_\mathfrak{v}$. It constrains how we complete inputs.*
- *π is a* ***correct policy*** *iff the correct outputs O_α of α are exactly the completions π' of π such that π' is also a completion of an input.*
- *The set of all correct policies for a task α is denoted Π_α.*[3]

Assume $\mathfrak{v}$-task ω and a policy $\pi \in L_\mathfrak{v}$. Inference proceeds as follows:

1. *we are presented with an input $i \in I_\omega$, and*
2. *we must select an output $e \in E_i \cap E_\pi$.*
3. *If $e \in O_\omega$, then e is correct and the task "complete". $\pi \in \Pi_\omega$ implies $e \in O_\omega$, but $e \in O_\omega$ doesn't imply $\pi \in \Pi_\omega$ (an incorrect policy can imply a correct output).*

Intuitively, a **policy** constrains how we complete inputs. It is a **correct policy** if it constrains us to correct outputs. To "learn" a policy we use a **proxy**. A proxy estimates one thing, by measuring another seemingly unrelated thing, much like how AIXI uses simplicity to estimate model veracity. In this case, we want a policy that classifies correct outputs. We will use a proxy called "weakness", which has been shown to outperform simplicity in sample efficiency and causal learning [8, 21]. Where simplicity is a property of *syntax*, weakness is a property of *function*. In order to make objective claims regarding performance, we cannot rely upon subjective interpretations of syntax.

Definition 5 (learning).

- *A* ***proxy*** *$<$ is a binary relation on statements. $<_w$ is the* ***weakness*** *proxy. For statements l_1, l_2 we have $l_1 <_w l_2$ iff $|E_{l_1}| < |E_{l_2}|$.*

(generalisation) *A statement l* ***generalises*** *to a $\mathfrak{v}$-task α iff $l \in \Pi_\alpha$. We speak of* ***learning*** *ω from α iff, given a proxy $<$, $\pi \in \Pi_\alpha$ maximises $<$ relative to all other policies in Π_α, and $\pi \in \Pi_\omega$.*

(probability of generalisation) *We assume a uniform distribution over $\Gamma_\mathfrak{v}$. If l_1 and l_2 are policies, we say it is less probable that l_1 generalizes than that l_2 generalizes, written $l_1 <_g l_2$, iff, when a task α is chosen at random from $\Gamma_\mathfrak{v}$ (using a uniform distribution) then the probability that l_1 generalizes to alpha is less than the probability that l_2 generalizes to α.*

(sample efficiency) *Suppose $\mathfrak{app}$ is the set of* ***a****ll* ***p****airs of* ***p****olicies. Assume a proxy $<$ returns 1 iff true, else 0. Proxy $<_a$ is more sample efficient than $<_b$ iff*

$$\left(\sum_{(l_1, l_2) \in \mathfrak{app}} |(l_1 <_g l_2) - (l_1 <_a l_2)| - |(l_1 <_g l_2) - (l_1 <_b l_2)| \right) < 0$$

---

[3] To repeat the above definition in set builder notation:

$$\Pi_\alpha = \{\pi \in L_\mathfrak{v} : E_{I_\alpha} \cap E_\pi = O_\alpha\}$$

(optimal proxy) *There is no proxy more sample efficient than* $<_w$*, so we call* $<_w$ *optimal. This formalises the idea that "explanations should be no more specific than necessary" (see Bennett's razor in [8]).*

Learning is an activity undertaken by some manner of agent, and a task has been "learned" when that agent knows a correct policy. For example, one might be considered to have "learned" chess when one knows the rules and strategies necessary to win. In this context we have not defined "agent", but we have abstraction layers and tasks. Humans typically learn from "examples", and in the context of a task one "example" is one correct output and one input that is a subset thereof. A collection of examples is a child task, so "learning" is an attempt to generalise from a known child task, to one of its parents. Intuitively, a child task functions like a "history" or memory of *correct* interactions (a one class "ostensive definition"). The lower level the child is in the generational hierarchy, the "faster" the task is learned and the more sample efficient the proxy. We assume tasks are **uniformly distributed** because anything else would imply that environment "prefers" or makes a value judgement about goals. The environment *is* a value judgement about what exists, but is otherwise assumed to be impartial. Because of this, the most sample efficient proxy is $<_w$ [8], enabling causal learning [21] as the weakest correct policies have the highest probability of being the causal intermediary between inputs and outputs.

3 Limits of Intelligence

As stated earlier, intelligence is often equated with "the ability to generalise", which amounts to **sample efficiency** in learning the policy which "caused" examples, and thus predictive accuracy. We wish to understand the *objective* upper limits of intelligence. It has been formally proven that $<_w$ is the optimal choice of proxy for sample efficiency [8, prop. 1, 2, 3] (and weak policies generalised at $110 - 500\%$ the rate of simple policies in experiments). In other words, the upper bound of intelligent behaviour in the context of an abstraction layer and task is attained by the weakness proxy. However, more intelligence is not always useful. Intuitively, making a human more intelligent is unlikely to improve their driving. Likewise, intelligence conveys no advantage if one's task is to interpret random noise. The extent to which it is *possible* to be sample efficient depends on the *abstraction layer* in which the task is instantiated. An abstraction layer is a bottleneck on intelligence, and can be goal directed just like a task. We can measure this goal directed "utility" as the extent to which it is *possible* and *advantageous* to be sample efficient given a task in a particular abstraction layer.

Definition 6 (utility of intelligence). $\epsilon : \Gamma_{\mathfrak{v}} \rightarrow \mathbb{N}$ *is a function that takes a* $\mathfrak{v}$*-task* γ *and returns a "utility" value, s.t.* $\epsilon(\gamma) = \max_{\pi \in \Pi_\gamma} (|E_\pi| - |O_\gamma|)$.

To maximise sample efficiency, we need to use the weakness proxy *and* maximise utility. To compare utility, we need a way to represent a task in different vocabularies. Utility is maximised when $\mathfrak{v} = P$. Γ_P contains all tasks in all vocabulary.

Hence, for every task ρ in Γ_P we can define a function that takes a vocabulary $\mathfrak{v}$ and returns a $\mathfrak{v}$-task which is a child of ρ (allowing for the fact that this will often be α s.t. $I_\alpha = O_\alpha = \emptyset$). We use 2^P to represent the powerset of P.

Definition 7 (uninstantiated-tasks). *The set of all tasks with no abstraction (meaning $\mathfrak{v} = P$) is Γ_P (it contains every task in every vocabulary). For every P-task $\rho \in \Gamma_P$ there exists a function $\lambda_\rho : 2^P \rightarrow \Gamma_P$ that takes a vocabulary $\mathfrak{v}' \in 2^P$ and returns a $\mathfrak{v}'$-task $\omega \sqsubset \rho$. We call λ_ρ an* ***uninstantiated-task****. It is instantiated by choosing a vocabulary.*

Weakness is maximised when the vocabulary is $\mathfrak{v}$ s.t. $\epsilon(\lambda(\mathfrak{v})) = \epsilon(\lambda(P))$. The most 'intelligent' choice of policy and vocabulary given λ is:

$$\pi \text{ and } \mathfrak{v} \text{ s.t. } \mathfrak{v} \in \operatorname*{arg\,max}_{\mathfrak{v} \in 2^P} \epsilon(\lambda(\mathfrak{v})) \text{ and } \pi \in \Pi_{\lambda(\mathfrak{v})} \text{ and } \neg\exists \pi' \in \Pi_{\lambda(\mathfrak{v})} (\pi <_w \pi')$$

Proposition 1 (upper bound). *Given uninstantiated task λ, sample efficiency is maximised by $<_w$ given vocabulary $\mathfrak{v}$ s.t. $\neg\exists \mathfrak{v}' \in 2^P$ where $\epsilon(\lambda(\mathfrak{v}')) > \epsilon(\lambda(\mathfrak{v}))$.*

Proof: According to [8, prop. 1, 2] the weakest correct policies have the highest probability of generalising. Given an uninstantiated task λ, we can use utility ϵ to measure the weakness of the weakest correct policies in different vocabularies. By choosing $\mathfrak{v}$ s.t. $\epsilon(\lambda(\mathfrak{v})) = \epsilon(\lambda(P))$ we instantiate λ in a vocabulary that maximises the weakness of correct policies for λ. Then, using weakness proxy, we can select a policy that has the highest possible probability of generalising, and thus maximise sample efficiency. ■

Concluding remarks: Here we proposed computational dualism, and presented an argument in support of an alternative which allows for objective claims regarding behaviour. We proved an objective upper bound for intelligent behaviour, but to attain this upper bound one would need to search infinitely many, infinitely large vocabularies. We anticipate these results may further our understanding of AI safety and general intelligence.

References

[1] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability.* Berlin, Heidelberg: Springer-Verlag, 2010.

[2] R. Solomonoff. "Complexity-based induction systems: Comparisons and convergence theorems". In: *IEEE TIT* 24.4 (1978), pp. 422–432.

[3] S. Legg and M. Hutter. "Universal Intelligence: A Definition of Machine Intelligence". In: *Minds and Machines* 17.4 (2007), pp. 391–444.

[4] J. Leike and M. Hutter. "Bad Universal Priors and Notions of Optimality". In: *Proceedings of The 28th Conference on Learning Theory, in Proceedings of Machine Learning Research* (2015), pp. 1244–1259.

[5] M. T. Bennett. "Compression, The Fermi Paradox and Artificial Super-Intelligence". In: *Artificial General Intelligence.* 2022, pp. 41–44.

[6] E. Thompson. *Mind in Life: Biology, Phenomenology, and the Sciences of Mind.* Cambridge MA: Harvard University Press, 2007.

[7] G. Piccinini and C. Maley. “Computation in Physical Systems”. In: *The Stanford Encyclopedia of Philosophy.* Sum. 21. Stanford University, 2021.

[8] M. T. Bennett. “The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest”. In: *Artificial General Intelligence.* Springer, 2023, pp. 42–51.

[9] F. Chollet. *On the Measure of Intelligence.* 2019.

[10] M. T. Bennett. “Symbol Emergence and the Solutions to Any Task”. In: *Artificial General Intelligence.* 2022, pp. 30–40.

[11] J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect.* 1st. New York: Basic Books, Inc., 2018.

[12] E. Sober. *Ockham’s Razors: A User’s Manual.* Cambridge Uni. Press, 2015.

[13] A. Kolmogorov. “On tables of random numbers”. In: *Sankhya: The Indian Journal of Statistics* A (1963), pp. 369–376.

[14] J. Rissanen. “Modeling By Shortest Data Description*”. In: *Autom.* 14 (1978), pp. 465–471.

[15] G. Chaitin. “The Limits of Reason”. In: *Sci. Am.* 294.3 (2006), pp. 74–81.

[16] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (Third Edition).* New York: Springer, 2008.

[17] M. T. Bennett. “Computable Artificial General Intelligence”. In: (2022).

[18] J. Kim. *Philosophy of Mind.* 3rd ed. New York: Routledge, 2011.

[19] H. Putnam. “Psychological Predicates”. In: *Art, mind, and religion.* Uni. of Pittsburgh Press, 1967, pp. 37–48.

[20] G. Hinton. *The Forward-Forward Algorithm: Some Preliminary Investigations.* 2022. arXiv: `2212.13345 [cs.LG]`.

[21] M. T. Bennett. “Emergent Causality and the Foundation of Consciousness”. In: *Artificial General Intelligence.* Springer, 2023, pp. 52–61.

[22] M. T. Bennett. “On the Computation of Meaning, Language Models and Incomprehensible Horrors”. In: *Artificial General Intelligence.* Springer, 2023, pp. 32–41.

[23] M. T. Bennett. “Is Complexity an Illusion?” In: (2024).

[24] D. Ward, D. Silverman, and M. Villalobos. “Introduction: The Varieties of Enactivism”. In: *Topoi* 36 (Apr. 2017).

[25] A. Clark and D. J. Chalmers. “The Extended Mind”. In: *Analysis* 58.1 (1998), pp. 7–19.

[26] B. Goertzel. *The Hidden Pattern: A Patternist Philosophy of Mind.* BrownWalker Press, 2006.

[27] M. Wheeler. “Martin Heidegger”. In: *The Stanford Encyclopedia of Philosophy.* Fall 2020. Stanford University, 2020.

[28] K. Friston. “Life as we know it”. In: *The Royal Society Interface* (2013).

[29] H. L. Dreyfus. “Why Heideggerian AI Failed and How Fixing it Would Require Making it More Heideggerian”. In: *Philosophical Psychology* 20.2 (2007), pp. 247–268.

[30] F. M. Artinger, G. Gigerenzer, and P. Jacobs. “Satisficing: Integrating Two Traditions”. In: *Journal of Economic Literature* 60.2 (2022), pp. 598–635.