Functionalism, Integrity, and Digital Consciousness

Derek Shiller

Note to readers: This is not the final published copy, but a late-stage draft.
See the final version here:
https://link.springer.com/article/10.1007/s11229-023-04473-z

Abstract: The prospect of consciousness in digital systems depends on the viability of functionalism about consciousness. Even if functionalism is true, it does not follow that any digital system that implements the right functional organization would be conscious. Functionalism requires constraints on what it takes it to properly implement a functional organization. Existing proposals for constraints on implementation relate to the integrity of the parts and states of the realizers of roles in a functional organization. This paper presents and motivates three integrity constraints not satisfied by current neural network models. It is proposed that for a system to be conscious, there must be a straightforward relationship between the material entities that compose the system and the realizers of functional roles, that the realizers of the functional roles must play their roles due to internal causal powers, and that they must continue to exist over time.

Introduction

Many cognitive capacities now found only in humans may soon be shared by artificial systems. Some models already produce verbal behavior resembling ours (Bubeck et al. 2023). The question of whether such systems would be conscious

or whether they might merely mimic us will matter to law and to policy and also to how we see the future roles and responsibilities of our species (Shulman and Bostrom 2021).

The core idea of this paper is that artificial consciousness – though not the appearance of prototypical conscious behavior – may require innovations in computer hardware and software. I will compare neural networks in biological and traditional digital environments and identify several differences that are relevant to our assessment of digital consciousness. Along the way, I present an interpretation of consciousness functionalism as constrained by integrity. It is because contemporary digital structures lack integrity that I am skeptical that computers would enable digital consciousness. The computational entities created by current programming paradigms that might play the right functional roles are fragmented, short-lived, and causally inert structures.

I begin this paper by exploring functionalism and its relation to multiple realizability. I then argue for the need for substantial requirements for implementing functional organizations to prevent consciousness from being realized by certain systems that are intuitively not conscious. I next describe some differences between biological and artificial neural networks. With this background in place, I present three plausible integrity-related constraints regarding what it takes to properly implement the functional organization necessary for consciousness. I argue that these constraints provide reasons to be wary of the idea that computers may soon be conscious.

Functionalism

Philosophers generally accept that consciousness is not a special product of human neural physiology. Other animals are conscious, even animals with rather different sorts of brains. Non-biological systems could also be conscious. The

traditional way to make sense of the multiple realizability of consciousness is with functionalism (Putnam 1967; Fodor 1968; Lewis 1972). Functionalism about consciousness holds that the functional organization of a system, i.e. the relationships between its possible internal states and its inputs and outputs, determines whether it is conscious.

The neuronal basis of human cognition is a product of our evolutionary heritage. We are multicellular organisms. Our bodies are built from the sequencing of the DNA activations that guide cellular division and differentiation. Neurons are capable of complex interactions that allow coordination and computation. They are also living, growing, metabolizing cells. Neurons originated from cells that found a role in coordinating complex behaviors. Mutation and selection have crafted them over hundreds of millions of years, combining them into specialized circuits that perform computational work as neural networks.

Neural networks are systems of interconnected nodes that influence each other's activities. Neural networks have proven to be well-suited to producing complex and flexible behavior. Biological neural networks utilize neurons, but artificial neural networks implemented as computer software have demonstrated comparable flexibility. Given recent rapid advances, it seems that what is special about neurons is that they make neural networks possible; it is neural networks that are special, not neurons.

It would be surprising if, among all the ways it was possible to implement a neural network, the way that our ancestors stumbled upon uniquely permitted consciousness. It would be surprising if neural networks weren't capable of consciousness unless they were made out of biological cells.

Functionalism eliminates such coincidences.

The functional organization of a system is comprised of the relations between

the external inputs it receives, the states of its parts, and the behavioral outputs it produces. Inputs affect the system's internal state. The overall state of a system can be subdivided into the states of its parts; the effects of the inputs on behavioral outputs depend on how those inputs influence the state transitions of the parts, how those state transitions beget further transitions, and how behaviors are determined by those states.

Functionalism holds that consciousness turns on organization, not on the substrate from which that organization is constructed. Neurons are fantastic building blocks. They can be composed into structures that interact in all sorts of ways, including whatever ways may be stipulated as necessary for consciousness. Neurons make these structures possible, but any other building block capable of implementing a neural network can do the same. Just as neurons are excellent building blocks, so too are electrical circuits. Any suitable building block can be used to create structures that constitute consciousness. If a global workspace can be built in a biological neural network, it can probably be built in artificial neural networks too.

Formalizing Functionalism

We may interpret a functional organization as a set of claims about the states of parts and their relation to each other specified with generic language that conveys nothing of the nature of those parts or their states[1]. Atomic claims describe conditional state transitions, such as:

> If part A is in state X and input M is received, then part A transitions to state Y and output N is produced.

[1]This understanding of functionalism borrows from both Lewis (1972) and Rescorla (2014). The primary advantage of this understanding over the traditional mapping account is that it makes it easier to introduce non-functional requirements ('enrichments') on the organization's realizers, as described below.

In this example, A, X, M, Y, and N have no meaning beyond their classifications as particular parts, states, inputs, and outputs. The claims in our model are all logical constructions (disjunctions, quantifications, probability claims, etc.) from atomic claims.

A system possesses the organization characterized by a claim set if there exists a way of carving up the inputs it receives, the states it can occupy, and the outputs it produces so that there exists an interpretation of the generic terms as naming inputs, parts, states, and outputs that validate the claims.

A carving of a system is a scheme of necessary and sufficient conditions that specify under which configurations the system counts as having particular inputs, outputs, and parts in states. For human beings, our carvings might abstract from patterns of neural firing. For computers, carvings might abstract from patterns of memory usage. Carving a system into parts is a creative process. Making room for creativity is already a concession to the possibility of digital minds: the states we ascribe to computers have a very complex relationship with the underlying electrical circuits.

Proper Implementation

For any given system, there will be many possible ways to carve it into parts. For any claim set describing an organization and any physical system that intuitively satisfies it, the claims will be true only under a small subset of carvings. Functionalist theories could identify canonical ways of carving up a system against which the claims are to be assessed[2]. However, it will likely be difficult to select some carvings in a non-arbitrary way. Instead, functionalists

[2]Proponents of the Integrated Information Theory (Oizumi, Albantakis, and Tononi 2014) have endorsed a sort of canonical carving approach. IIT treats information (in the sense of how the state of the system at one time constrains past and future states) as central to consciousness; the canonical carving maximizes the amount of information in the system.

have traditionally opted for liberalism: all it takes to implement an organization described by a claim set is for there to be a truth-making assignment of terms under at least one carving.

Liberal functionalism has a problem. If all carvings count, then it is easy to implement an organization. Further constraints on which carvings are proper are needed to avoid counterintuitive verdicts.

The oldest version of the worry is attributed to Ian Hinckfuss (described in Lycan 1981), who evocatively suggested that the dynamics in a pail of water might come to briefly reflect the structure of a brain. This is problematic if it is obvious that a pail of water would not be conscious no matter how it sloshes about. While it is relatively unintuitive that an ordinary pail of water could be conscious, it is also dubious that we should trust our intuitions about rare configurations of water molecules. Subsequent critics have accepted the possibility of an occasional conscious pail of water (e.g. Sprevak 2018), so long as it has peculiar internal dynamics.

A second version of the problem suggests that it isn't just unusual systems that we need to worry about – nearly any material object can be gerrymandered to fit whatever organizations are responsible for consciousness.

The classical presentation of this problem was developed by Hilary Putnam (1988). Putnam argued that for sufficiently complex systems, there nearly always exists some way of carving it into parts and states that can make it trivially easy to satisfy any functional organization. The idea is that if, given any input, a system transitions through a series of unique configurations from a unique starting point, then we can find an agreeable carving by grouping configurations together so as to make any set of generic claims true. Large numbers of unique configurations provide a blank canvas on which any formal structure can be

drawn. Multiple realizability offered a significant motivation for functionalism, but if it says everything is conscious, it has gone too far.

Critics have noted that the complex systems identified by Putnam, while they may be gerrymandered to get all the counterfactuals right, don't support the kinds of causal relations we expect to exist between parts of an organization (Chalmers 1996; Chrisley 1994).

A third version of the problem suggests that there are some systems that liberal functionalism will counterintuitively predict are conscious.

Consider a grid of mirrors that is placed facing another mirror, such that each mirror in the grid reflects the whole grid. If we write a distinct number on each mirror then it will contain both its number and a deeply nested reflection. This reflection will take time to form: before light has made a return trip, each grid mirror reflects just its number. A moment later it reflects that number and the whole grid of written numbers. A moment later it reflects that number and one level deep of nested reflections, and so on.

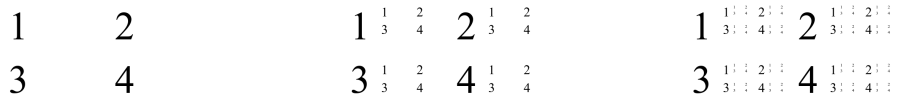

Figure 1: Nested Reflections

At each moment, the state of each distinctly numbered mirror in the grid is unique over the history of the system and sensitive each part of the total prior state of the grid. With such a system, we could carve out not only any states we want from disjunctions of surface reflectance patterns, but also support all of the causal relations that exist in the model with causal relations in the grid. (This accommodates only deterministic relationships, but it is not hard to add any relationshipts to this model.) To find a truth-making assignment, we simply take some abstract model that satisfies our set of claims. For each possible input,

we assign some set of characters scrawled on the mirrors of the grid. Then we derive the reflectance states that result from those initial states of the grid and assign them to those states that follow in the model of our claim set, grouping together as necessary any states that can be arrived at in multiple ways.[3]

The mirror grid is not much more plausibly conscious than an arbitrary physical object. This suggests a need for further constraints. However, the problem isn't (just) that we should think that the mirror grid is unconscious: an adequate functionalist theory ought not to say so. If our theory says that the grid is conscious, we have formulated it incorrectly.

The problematic carvings of unconscious systems into brain-like organizations depend upon gerrymandering them into the necessary shape. There are many ways to amend liberal functionalism to limit gerrymandering.

Two kinds amendments have been considered. *Enrichments* expand the vocabulary that can be used in specifying the entities of the claim set beyond generic labels. *Constraints* restrict some aspects of the objects and properties that can figure into a carving without embellishing the claim set. Under constraints, only some carvings are proper.

Enrichments

Enrichments expand the vocabulary used in formulating the claim set, allowing for further restrictions to be built into the formulation of an organization. As a toy example, we might expand our description to include colors (e.g. if A is blue and input X is received...). Systems would only satisfy the claim set if the parts are appropriately colored.

[3] Chalmers (1996, 328) anticipates a system like this and suggests that it would be impossibly complex. He is wary, however, of dismissing the problem because it is not physically viable. Liberal implementations don't live up to the spirit of functionalism. The mirror grid's conceptual possibility shows that.

Agential enrichments expand the claim set to allow language about the nature of inputs and outputs. This lets us require that some of the inputs and outputs of a system resemble ours (Chalmers 1996). Perhaps the inputs include perceptual events and the outputs include behaviors. The pail of water and the mirror grid possess no sensory faculties and perceive nothing. Nor do they have bodies they can move. Their complex dynamics might implement the functional structure of a human brain, but the inputs and outputs of the implementation can only correspond to non-sensory non-behavioral states. Requiring that the model's inputs and outputs map to actual perceptions and behaviors will limit proper implementations to systems that interact with their environment as we do.

Other enrichments borrow from the literature on what it is to implement a computation[4].

One popular theory of computation (Shagrir 2020) suggests that some of the parts of a system that contribute to its implementing a computation must bear semantic values. Computers manipulate sequences of bits. Those bits acquire meanings through the intentions of programmers. Groups of neurons in our brains have been crafted by evolution to serve a representational function. Pails of water and grids of mirrors won't generally have parts with any semantic values so they cannot implement any computations on this account.

Semantic enrichments allow semantic language to be used in formulating claim sets. A claim set might include claims about how meaning-bearing vehicles are created, manipulated, or moved around. Functionalist theories of consciousness like the global workspace theory often describe the movement of information, so

[4]The question of what it takes for a system to implement a computation in general (Piccinini 2015) is related to but distinct from this issue. The features that allow a machine to count as computing a certain algorithm in the sense that is relevant for understanding the practices of computer scientists may not be the same as the features that allow a system to count as implementing the functional role of consciousness. We should be open to non-computational interpretations of functional organizations (Piccinini 2010).

it is plausible that semantic language has a place. This would also explain why the pail of water and mirror grid aren't conscious, nothing in them possesses genuine semantic values.

Another prominent theory of computation holds that computations involve functional mechanisms (Piccinini 2015). Functional mechanisms are (among other things) components of a system that have goals or purposes. The parts of a system that play a given role must be supposed to do something, either because they evolved to play that role, were designed to play that role, or because their playing that role serves other natural ends of the system (Maley and Piccinini 2017). We might draw inspiration from this account of computation and introduce telic enrichments to functionalist theories of consciousness. If our claim set specifies certain ends for parts, then candidate systems count as being conscious only if they have parts with the relevant ends.

Constraints

Constraints amend the satisfaction account by restricting carvings: they introduce requirements regarding how system configurations can determine the states of parts.

One kind of constraint (Chalmers 1996; Godfrey-Smith 2009; Sprevak 2018), requires that parts have spatial boundaries. We might develop this constraint to enforce boundaries that are contiguous, non-overlapping, or logically independent.

A contiguity constraint would require that there not be gaps between the materials of a single part. There couldn't be a part of a system that consists in circuits in different places with no intermediary links.

A non-overlapping requirement would mean that no two parts of a system could share any materials in common. There couldn't be subsystems that do

double-duty in settling the states of multiple parts.

An independence requirement would mean that the boundaries for the material that settles the state of one part couldn't be determined by the configuration of materials outside of that part. We couldn't have a part constituted by some region or another depending on the state of some other part of the system.

Each way of developing a spatial constraint places limits on gerrymandering, because the states of parts can't depend on just any configuration of the system. However, these constraints do not prohibit gerrymandering of the states of individual parts.

Boundary constraints might be to limiting. Natural divisions of human brains may involve non-contiguous, overlapping, or dependent parts. Even if the natural carvings of human brains satisfy these constraints, it is not obvious that every conscious system should. In proposing a constraint like this, Chalmers (1996) regretfully notes that this may block some plausible systems from properly implementing functional states. Boundary constraints err on the side of being overly conservative.

A second kind of constraint (Godfrey-Smith 2009) requires similarity in the groupings of parts and states of a system. Insofar as a part's state is determined by the configurations of the system's materials, the different configurations that can give rise to the same parts in the same states must be similar to one another.

Similarity might be imposed both as a constraint on the parts of a carving and their states.

When imposed on the parts, each part must be composed of regions or materials that are naturally grouped together. When imposed on the states of the parts, the various configurations that could lead to that part having a particular state must be similar to each other.

Similarity rules out gerrymandering parts and states to fit any particular organization because that would involve drawing groups across natural boundaries. Though, as with boundary constraints, it is not immediately obvious what it says about natural carvings of the human brain.

Integrity

An adequate theory of implementation is important for functionalism. The proposals here surveyed aim to avoid unintuitive implications. They do so mainly by adding requirements on the integrity of the entities that make up a carving.

Integrity has three components: 1) the parts of a system must be distinctive entities, e.g. they must have natural identities that account for their differentiation from the other parts. 2) The states of a part must be naturally distinguished from each other so that it is principled which configurations of the system count towards which states. 3) The transitions between states must be orderly. The semantic and telic enrichments touch on the first two. Boundary constraints touch on the first. The similarity constraint touches on the second.

Integrity makes sense for functionalism. Functionalism says that consciousness rests on having parts interacting in particular ways. We should expect that the parts possess some independent existence metaphysically prior to them playing their roles.

Once we acknowledge the need for enrichments of our claim set or constraints on carvings, we should work toward a cohesive theory of those requirements. We should assess candidates in part by their intrinsic appeal, not just the work they can do for us in avoiding gerrymandered minds. If there are reasons to adopt integrity requirements to avoid excessive liberalism, we should be open to the possibility that integrity may be important for consciousness even in ways that

don't help us address problematic applications.

I see no reason to expect a priori that the requirements on proper implementation are the minimum requirements necessary to avoid gerrymandered systems from counting as conscious. Below, I'll present three integrity-related constraints on proper implementation that are intrinsically plausible and that challenge digital consciousness.

Biological and artificial networks

Human brains and computer processors may perform similar computational work, but the ways that they do so are different. Before presenting my proposed integrity constraints and arguing that computers fail them, I'll summarize how brains and computers work and how neural networks in contemporary hardware differ from networks in biological neurons.

Brains

Human brains are structures of neurons and assorted auxiliary cells. Neurons share a basic role and morphology, but their details differ to suit their particular functions. They include a cell body and long thin branches to other neurons. Neuronal branches conduct electrical signals. Dendrites receive signals from other neurons. Axons send signals to other neurons. When a signal is generated, typically in response to receiving signals, electrical fields travel down the length of the axons. The signals stimulate the release of neurotransmitters that can lead to the firing of neighboring neurons. Whether a neuron fires is generally a function of the signals it has received. At its most basic level, if a neuron receives sufficiently many excitatory signals and sufficiently few inhibitory signals in succession, it will also fire.

In humans and other mammals, the cerebral cortex houses most of the neurons

responsible for visual processing, long-term planning, language comprehension, and motor control, along with many other cognitively sophisticated functions. Neurons in the cortex are organized into layers and are connected layer-to-layer to comprise columns. These columns are integrated into larger circuits that allow cycles of signaling. Neurons are fairly noisy, often firing unprompted, so it is larger patterns of firing within and across circuits that allow for cognition.

Neuronal signals travel fairly slowly; millions of neurons contribute to computational processing in parallel. A sensory signal will lead to a widening net of activations that parse the features of the signal, manipulate content, and decide on a behavioral response over a few hundred milliseconds. Those activations can feed back to earlier stages of processing to allow different parts to adjust their behavior to achieve coherence.

Computers

Computers operate primarily through the interaction of a processor and memory. The processor directs the movement of electricity through circuits. Different distributions of charge lead to different conductance patterns, which lead to different distributions of charge and further changes in conductance patterns. Some specialized circuits, binary cells, are influenced by charge to switch between states in a manner that allows them to store data values.

Standard processors contain a number of specialized sequences of binary cells, registers, that store the data on which they operate. One register in a central processor unit, the instruction register, houses the bit representation of the operation the processor is to perform. The charges present in the instruction register configure the conductance patterns in the circuitry to automatically perform the operation.

Another register, the program counter, contains the bit representation of the

location in memory of the next instruction to be performed – as the processor performs an operation, it fetches the next operation from memory for the instruction register and increments the program counter.

Memory contains vastly more binary cells and those cells are optimized for temporary storage. Some memory is used to indefinitely hold data representing various ditigal entities for the life of the program. Other regions are devoted to scratch pads that programs use to keep track of values in the course of running operations. Many operations involve moving data from memory into processor registers, moving data between registers, and returning data from the registers into memory. Others perform logical or arithmetic operations on the data in the registers.

Program instructions are loaded into memory when they are launched. Part of the process of building a program is converting the characters of a text file into sequences of instructions, represented by series of bits, that operate within the design of the instruction register circuitry to direct the processor. These instructions are ferried from memory to the instruction register and their corresponding operations are performed.

Artificial Neural Networks

Recent advancements in artificial intelligence involve neural networks. Nodes in a neural network behave vaguely like biological neurons: they switch between states of activation as a result of the activities of their neighbors.

Artificial neural networks might theoretically be produced in a number of ways. In practice, they are created with standard digital computers. Nodes in a network are represented in memory and processor registers as the states of binary cells. Those bits are manipulated in the processor in accordance with program instructions.

Nodes in abstract neural networks are generally separated into layers and activations across nodes are calculated layer by layer. A node's activation at one time is dependent on a function applied to the activation of the relevant nodes in the previous layer.

A particular node's activity level is represented by bits formatted as floats, digital representations of fractional numbers. Networks are often memory intensive and are optimized for efficiency. A layer's activations can be recorded in lists of floats in blocks of memory. The weights of the connections between layers can also be represented as lists of floats. The relative locations of the floats in weight and activation sequences determine which must be combined to derive the activity levels of the next layer.

Different parts of a network are separately represented. A program needs to keep track of the connections between nodes during and between runs but only needs to calculate activation levels momentarily. When a network is run, the connection strengths between layers are combined with input activations via a series of matrix multiplications and the application of activation functions.

Three Integrity Constraints

Functionalism receives some of its plausibility from the fact that it allows for multiple realizability. However, if the requirements for properly playing a functional role are relaxed, functionalism will over-generate, allowing some complex systems to count as implementing just about any functional organization. In order to place reasonable limits on multiple realizability, we must supplement functionalism with enrichments or constraints on proper implementation.

We've seen several possible constraints so far. On one, the realizers of functional roles must correspond to spatially distinct parts of the system. On another,

different configurations that ground a single state of a functional part must resemble each other. There are many more possible constraints. Some of these might help ward off unpalatable applications. Many equally plausible constraints won't. Once we open the door to the existence of constraints on proper implementation to solve one problem, we should be open to the existence of more.

I'm somewhat skeptical about the prospect of philosophical intuitions or scientific experimentation to unambiguously identify appropriate constraints or enrichments. If our judgments don't play a constitutive role in determining the distribution of consciousness, I see no reason why we should have a reliable innate grasp on where it arises. Instead, we are best served by imagining plausible constraints and applying them to produce varying degrees of certainty. If on every plausible constraint, a system implements the same functional role as a human brain, functionalists should regard it as conscious. The more plausible constraints that a system fails, the more skeptical we should be. We are best off avoiding artificial systems of uncertain status insofar as it is less clear what we owe them (Schwitzgebel and Garza 2020).

In this section, I explore three further integrity-related constraints – material complexity, causal integration, and continuity – on the proper implementation of a functional organization whose plausibility raises doubts about the possibility of creating digital consciousness within current paradigms.

Material Complexity

The degree of *material complexity* of a system is a measure of the extent to which the functional parts of the system and their states have a simple relationship with the states and relations of the component entities of some underlying material ontology. The more complex a system, the more diffuse and intricate the material

entities that together explain why the system overall counts as having a part in a particular state. Material complexity may result when many components irreducibly and jointly determine the states of the system and the significance of any one component entity depends on the configurations others. It is commonly found in systems with many layers of abstraction.

Material complexity is related to naturalness. In order to evaluate the relationship between a system's component configuration and its states, we need to know how we can conceptualize its material components. That may depend upon natural material properties. There are different ways of demarcating the ontology of material components, but any acceptable scheme should more or less agree on the relevant entities for our purposes. For brains, those entities are axons and dendrites, neurons, neural columns, neural circuits, and larger neural structures. For computers, they are semiconductors, mosfet transistors, bit cells, registers, memory blocks, processor cores, and so on.

Material complexity is a relation between states and properties and *material* components, which means that a system whose states have a direct relationship with the underlying entities of some social or similar non-material ontology may still be quite complex.

Compare what it takes for an army and a company to count as being in good shape. A well-off army will have many soldiers in uniform, those soldiers will be well-nourished, uninjured, and well-armed. They'll be dispersed in various strategically valuable fortified positions or near transport hubs. They'll engage in their day-to-day tasks with systematic processes and procedures. On the other hand, a healthy company will have valuable assets or will be poised to sell goods or services at a profit. How much revenue the company can expect depends on the records of its ownership and balances, its contracts, its intellectual property, and so on. Records of these things are distributed in various filing cabinets and

computer systems in accordance with complicated legal rules. The health of many companies also depends on the health of others, such as its debtors and its subsidiaries, which are themselves subject to the similar complex considerations.

If you wanted to tell how well-off the Russian military is with satellite imagery, you could probably fare reasonably well. If you wanted to tell how well-off J.P. Morgan is with satellite imagery – even if you could see well enough to read paperwork – you probably have a much harder time. The factors that determine the latter are too complex and distributed. J.P. Morgan's health depends on the relationships between abstractions upon abstractions upon abstractions.

Brains are highly complex systems, but neural structures have a relatively simple relationship with our brains' underlying materials. Mental states are probably more like armies than companies. Neurons have clear cellular boundaries and their structure and the physical/chemical properties of synapses make it relatively straightforward to see how the activities of one contribute to the activities of another. The potential effects of individual neurons are mostly ingrained in local cellular properties. The potential effects of groups of neurons depend on the potentials of individual neurons. Brain states give off signatures broadly recognizable through fMRI scans. If the parts of a human mind are composed out of neural circuits, human brains are most likely not particularly complex in the sense discussed here.

A global workspace, for instance, is almost certainly wired in anatomically distinct ways. It seems likely that based on their connections, neuroscientists could identify the circuits comprising the global workspace before they were able to step through and figure out exactly what it did (e.g. Deco, Vidaurre, and Kringelbach 2021). Given enough time, they could plausibly work out how information was added to the workspace by looking at the local properties of individual cells. This is not to say that neural structures would have an

anatomically obvious role, but fine-grained evaluation of the brain would likely identify many of the parts that populate organizations.

The same organizational structure found in a human brain might also be implemented in less straightforward ways. Suppose that we had a massive brain with trillions of neurons. The neurons were connected in complex ways that did not in any straightforward way replicate the dynamics of a human brain, but instead produced a seemingly arbitrary collection of firing patterns. Complementing this brain we have an ordered series of spreadsheets. Each spreadsheet has a list (in ink) of rules for interpreting the state of each neuron in the brain. Some rules hard-code an interpreted activity pattern for some neurons, disregarding whatever they are actually doing. Some rules tell us to infer the behavior of one neuron in terms of the behavior of others. Some flip the interpretation of a given neuron. The rules have no physical effect on the brains.

Suppose further that if we apply the rules included in these spreadsheets in sequence to reinterpret the state of each neuron in this brain, we would get an interpretation of the activity state of those neurons which exactly mirrors the functional dynamics of a human brain. The reinterpreted states of neurons might allow the brain to represent the same firing patterns as a human brain despite lacking any surface-level resemblance. But given the required level of complexity introduced by these reinterpretations, it is plausible that the resulting system of neurons-under-spreadsheet-reinterpretation would not be conscious.

The values of connection weights and activations in an artificial neural network exhibit a complex relationship with underlying physical hardware that threatens the integrity of nodes and their activation states.

The data representations of nodes and connections in a digital representation of a digital neural network are hard to distinguish from other data stored in memory.

Text, numbers, images, program instructions – all are just sequences of bits. The data created by a program need not be stored in a continuous sequence, let alone in any particular location. Programs make frequent (and deeply nested) use of pointers, which allow one sequence of memory space to indicate a particular piece of data is stored at another location. Programs store the address at which data structures begin, but there is nothing in the physical structure of memory marking the boundaries or what data is stored there. Data types need not be declared in memory; programs are compiled to handle stretches of memory in ways appropriate to compute functions on specific data types. It is only by tracing chains of pointer addresses back that we can see the meaning-constitutive way that a program uses the bits.

A program consists in a series of machine language bit representations of instructions for the processor. The fact that a sequence of bits even constitutes machine code rather than data is not straightforward; it is itself determined by the dynamics of the program. Furthermore, while stored in sequences of instructions, program instructions aren't carried out linearly. The processor will jump from place to place, sometimes depending on the current state or the registers. Being context sensitive, when a program will jump depend on the state of the registers, which can only be predicted by stepping through the program.

The divisions between the parts of a data structure in a computer are intimately tied to their place in the functioning of the program. The data structures have no substance on their own, separate from the behavior of the program, and the behavior of programs is complicated and highly defuse, depending on the interactions between the stored code for the program in memory, the separate code for the operating system and other auxiliary processors, and the physical structure of the hardware. This makes the interpretation of data a complex process that requires piecing together distributed information.

It is easy to overlook the material complexity of programs because programs aren't written in the same language in which they are run. Reading a text file for a program conveys some of the program's structure graphically and makes it conceptually simpler. Variables may be defined with names and types. The same orthographic string located between the same curly braces will signify a variable with the same content. But processors can't understand programming languages as they are written – they must be translated into a native format, and that translation process removes the simplicity suggested by the program code.

Causal Integration

The *causal integration* of a system is a measure of the degree to which the system's parts play their roles because of their intrinsic causal powers. Systems that aren't causally integrated may have parts that are made to perform their role by some external power. Books and bombs have immense effects on the world, but books do so only through how their readers respond. They are not causally integrated. The effects of bombs, in contrast, are produced by their internal chemistry.

Neural structures within brains have the power to influence each other. Neurons generate the functional dynamics of brain regions through their capacity to receive, process, and produce signals. The roles played by structures of neurons are ultimately grounded in the causal powers of the neurons. These neurons directly influence the neurons they connect with, and through these interactions, neural structures influence each other.

The organizations whose parts have causal power within human brains might be implemented in systems where they lack causal powers. We might reconstruct a human brain in an intricate diagram of neurons, their states, and their connections

on a massive blackboard. Teams of researchers might update the diagram so that it evolves just as would a real brain. The marks on the blackboard have no power themselves. They sit unchanging until adjusted. They serve as notes that lead the researchers to make appropriate updates. The markings on the blackboard might have the same structure and dynamics as a human brain, but it is plausible they would not be a proper implementation on account of their internal powerlessness.

The structures that represent nodes in an artificial neural network also do not have any power themselves. The representation of the state of a node and its connections are contained within sequences of binary cells. The same binary cells with the same contents could be just as easily put to any other purpose. They can be fetched and manipulated by any program granted access to that region of memory space. The node representations can do nothing all by themselves. Instead, those data are used by a program to represent a neural network. A structure of node representations only plays a functional role insofar as it is used for that end by a processor.

It might be argued that the right computer structures within an implementation of a neural network aren't just the sequences of binary cells representing the node states and connections in memory, but also all of the various pointers to those structures, all of the instructions and other data in memory required to give meaning to those pointers and depict operations to be performed on them, and also the architecture of the processor that allows the operations to be performed in the right way (among other things). This inclusive interpretation of the relevant structures internalizes the causal powers by incorporating within the structures the parts that do the work.

The inclusive approach raises its own issues.

One issue with this proposal is that other instructions of the program that manipulate data structures mostly serve a general purpose, so different structures will largely overlap in their subparts that perform causal work. Many of the instructions needed to step forward through a neural network will be basic routines for loading and manipulating data. These routines will be used in much the same way to manipulate every structure in the network. Substantial overlap blurs the distinctness of different parts of a system.

A second potential issue is that internalizing the machinery for updating a network into the parts of the structure will result in a clear bifurcation between the operator and the data inside of that part. The distinctive component of these substructures, what makes one part different from the others, is passive. Insofar as the combined structure has the powers it does, it has them because of an internal system of manipulating notes. The notes contain the structures that distinguish the state from others, but they are powerless in themselves. This is strikingly *functionally* different from the way that human brains work.

Continuity

The continuity of a system is a measure of the extent to which its parts persist over time under identities that are separate from their functional roles. Systems without continuity may have a constantly evolving set of subcomponents built from different materials; insofar as the parts of a non-continuous system persist over time, the slices of each part at different times may have a unity through their shared positions in a broader functional organization. The batter in a baseball game maintains their role throughout the game, but is played by different people at different times. The Dalai Lama is a role that has lasted for centuries and that has been played, at least according to some, by just one metaphysical entity.

The neural structures that perform the sorts of functional roles in human

brains underlying consciousness are probably relatively consistent. Neurons are specialized for specific tasks by their internal structure and connections. They maintain their structure and connections over time, occupying the same positions in the brain and being composed out of the same materials moment-to-moment.

The same structures found in human brains might be implemented in systems where they lack continuity. Consider a machine that takes as an input a human brain that has been frozen into a single state within a preservative medium and produces as an output a fully new human brain frozen in another brain state. This machine would disassemble the input brain and construct the output brain out of new atomic materials that reflected what state the input brain would have momentarily occupied were it not frozen. We might route the output brains back around to form the machine's inputs so that it produced a constant succession of new frozen brains reflecting the states that a human brain would naturally occupy as its internal dynamics evolved over time.

The succession of frozen brains could be said to have the same organization as a human brain. However, this requires some interpretive creativity. The dynamism of the functional architecture is spread over multiple brains; the parts that play each role would jump from place to place and be constantly composed of new materials. We might identify a network that forms a global workspace and hold that the brains' global workspace at each time consists in a relevant network in the latest frozen brain. As new brains are produced, the workspace will jump from brain to brain. The faculties to which information is broadcast would also only exist in future brains.

Even though each frozen brain's parts would have causal connections with the previous brain's parts through the operations of the machine, their separateness poses an intuitive challenge for consciousness. It is plausible that this is an improper implementation because the material parts that play a functional role

in a proper implementation must retain their role over time.

The regions of memory that are available for use by a computer program depend on the runtime context. When a program starts up, the operating system carves off a section of memory space for the program. The operating system then provides additional sections when the program's needs grow. Which sections of memory are provided to the program depends on which sections are currently in use. If different programs are run, or the same programs are run in a different order, they'll have access to different regions of memory. This means that when a program stores data about a node's state in binary cells in memory, the circuits that store that data change from run to run.

While running, programming languages typically handle function calls by creating data structures that are stored separately in regions of memory. The same function, when called multiple times, may use different sections of memory. The memory space allocated to each call can contain some basic data, they also contain pointers to other memory locations for data objects that have variable sizes. Data located in different memory regions will use distinct binary cells. So even when a function is called multiple times within the course of a single execution of the program, it is likely to store its data in different places. The arrays of activation levels of nodes will not be supported by any specific section of memory but may jump around.

Neural networks are typically layered and do not need to keep track of node state in layers over time. Instead, they need to remember the connection strengths between layers and the activation state of the current layer. Even when a sequence representing node activation values is stored in memory and updated within the scope of a single function call, whether the result is stored in the same place will depend upon the details of the code. Depending on the neural structure in which the nodes are embedded, there may be no reason to keep

any record of its activation state after its effects on the next layer have been computed. The memory that was allocated to recording each node's state may be freed for future use or may be immediately put to other purposes. This means that the bits storing node state don't change over time the way the state of a neuron does. They are allocated to their role in storing a node state and given a single unchanging value for their life.

As a result of the complex and dynamic factors influencing memory use, the binary cells that at one point store the state of a node may at another time store the state of a different node, or store a different kind of data altogether. If the program does not reuse the same memory regions to record the state of a node from moment to moment, then the structures in a neural network representation that are responsible for consciousness will lack continuity. Each time a node is marked as activated, the representation of the activation will involve memory at a different location composed of different circuits.

If continuity is a constraint on what it takes for a structure to properly play a functional role, then ensuring the continuity of memory usage may be critical for consciousness.

Building Conscious Systems

The three integrity constraints proposed above raise doubts about the potential consciousness of AI systems built using current tools and techniques. These are doubts about potential consciousness, not behavioral capacities: present tools may allow us to build systems with every behavioral indication of phenomenal states without the structural integrity to actually feel them. This makes these concerns particularly pernicious: we may be easily misled into attributing consciousness.

The constraints present no principled barrier to artificial consciousness. There is no reason to think that we can't build systems whose internal parts possess every kind of integrity that brains do. Securing all of these constraints would require rethinking how AI systems are implemented. The processors and memory on which systems run could be redesigned so as to include robust structures of transistors. The particular architectures of nodes and connections might be physically implemented in circuits, rather than inferred from computations on tensors of floats.

Some exploratory steps have already been taken in this direction. Neuromorphic computers implement some of the structural features of brains. Some use traditional circuits organized in a more brain-like way, such as IBM's TrueNorth (Merolla et al. 2014) and Intel's Loihi (Davies et al. 2018). These architectures connect groups of processors dedicated to performing node computations in a manner more reminiscent of neural connections. Other computers use circuits whose properties make them more like neurons themselves, such as in changing their responses to electrical charge based on their histories of stimulation (Ielmini, Wang, and Liu 2021). Such experimentation suggests that future computers could satisfy the integrity constraints with an artificial brain that looks more like our own.

Neuromorphic computers show promise because they come closer to implementing neural structures in hardware, but my arguments don't cast doubt on the possibility of conscious systems that are fundamentally dissimilar to us. Different sorts of systems, including systems not implementing neural networks, could be well integrated. Neural networks are a promising way to implement complex organizations, but they are not necessarily the only way. Perhaps future alternatives will allow for more efficient implementations of the same organizations.

It is not clear whether non-traditional processor architectures will win out.

The traditional approaches developed in the 1940s, 50s, and 60s may be ideal, or they may enjoy enough of a head start so as to never be worth replacing. However, if we care about reducing ambiguity in AI consciousness, and if we take functionalism seriously, then we should strive to reduce the doubts raised by integrity considerations. This will require redesigning hardware and software to make consciousness-relevant structures more robust.

Bibliography

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, et al. 2023. "Sparks of Artificial General Intelligence: Early Experiments with GPT-4." http://arxiv.org/abs/2303.12712.

Chalmers, David J. 1996. "Does a Rock Implement Every Finite-State Automaton?" *Synthese* 108 (3): 309–33.

Chrisley, Ronald L. 1994. "Why Everything Doesn't Realize Every Computation." *Minds and Machines* 4 (4): 403–20.

Davies, Mike, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, et al. 2018. "Loihi: A Neuromorphic Manycore Processor with on-Chip Learning." *IEEE Micro* 38 (1): 82–99.

Deco, Gustavo, Diego Vidaurre, and Morten L Kringelbach. 2021. "Revisiting the Global Workspace Orchestrating the Hierarchical Organization of the Human Brain." *Nature Human Behaviour* 5 (4): 497–511.

Fodor, Jerry A. 1968. *Psychological Explanation: An Introduction to the Philosophy of Psychology.* Random House.

Godfrey-Smith, Peter. 2009. "Triviality Arguments Against Functionalism." *Philosophical Studies* 145 (2): 273–95.

Ielmini, D, Z Wang, and Y Liu. 2021. "Brain-Inspired Computing via Memory Device Physics." *APL Materials* 9 (5).

Lewis, David. 1972. "Psychophysical and Theoretical Identifications." *Australasian Journal of Philosophy* 50 (3): 249–58.

Lycan, William. 1981. "Form, Function, and Feel." *The Journal of Philosophy* 78 (1): 24–50.

Maley, Corey J, and Gualtiero Piccinini. 2017. "Of Teleological Functions for Psychology and Neuroscience." In *Explanation and Integration in Mind and Brain Science*, 236–56. Oxford University Press.

Merolla, Paul A, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, et al. 2014. "A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface." *Science* 345 (6197): 668–73.

Oizumi, Masafumi, Larissa Albantakis, and Giulio Tononi. 2014. "From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0." *PLoS Computational Biology* 10 (5): 1–25.

Piccinini, Gualtiero. 2010. "The Mind as Neural Software? Understanding Functionalism, Computationalism, and Computational Functionalism." *Philosophy and Phenomenological Research* 81 (2): 269–311.

———. 2015. *Physical Computation: A Mechanistic Account*. Oxford University Press.

Putnam, Hilary. 1967. "Psychological Predicates." In *Art, Mind, and Religion*, edited by W. H. Capitan and D. D. Merrill, 37–48. University of Pittsburgh

Press.

———. 1988. *Representation and Reality*. MIT Press.

Rescorla, Michael. 2014. "A Theory of Computational Implementation." *Synthese* 191: 1277–1307.

Schwitzgebel, Eric, and Mara Garza. 2020. "Designing AI with Rights, Consciousness, Self-Respect, and Freedom." In *Ethics of Artificial Intelligence*, edited by S. Matthew Liao, 459–79. Oxford University Press.

Shagrir, Oron. 2020. "In Defense of the Semantic View of Computation." *Synthese* 197: 4083–4108.

Shulman, Carl, and Nick Bostrom. 2021. "Sharing the World with Digital Minds." *Rethinking Moral Status*, 306–26.

Sprevak, Mark. 2018. "Triviality Arguments About Computational Implementation." In *The Routledge Handbook of the Computational Mind*, 175–91. Routledge.