Computation in Context[*]

André Curtis-Trudel

Forthcoming in *Erkenntnis*

Abstract

Unlimited pancomputationalism claims that every physical system implements every computational model simultaneously. Some philosophers argue that unlimited pancomputationalism renders implementation 'trivial' or 'vacuous', unsuitable for serious scientific work. A popular and natural reaction to this argument is to reject unlimited pancomputationalism. However, I argue that given certain assumptions about the nature of computational ascription, unlimited pancomputationalism does not entail that implementation is trivial. These assumptions concern the relativity and context sensitivity of computational ascription. Very roughly: relative to a specific, contextually salient way of regarding a physical system computationally, the claim that that system implements a specific computational model is as non-trivial as one could reasonably want.

1 The state of play

Since the late 1980s, philosophical reflection on physical computation has been animated by a skeptical argument advanced by Hilary Putnam (1987) and John Searle (1992). At issue is the notion of computational implementation. Putnam and Searle argue for *unlimited pancomputationalism*, the view that *every* physical

[*]Thanks to Chris Pincock, Andrew Richmond, Richard Samuels, and Stewart Shapiro for helpful comments and discussion. I am also grateful to audiences at ANU, the University of Cincinnati, the University of Kansas, the Israel Institute for Advanced Studies, the Ohio State University, and Lingnan University for comments and discussion. Two anonymous referees provided immensely helpful feedback on a previous version. Any mistakes that remain are my own.

Work on this article was partially supported by a Senior Research Fellowship award from the Research Grants Council of the Hong Kong SAR, China ('Philosophy of Contemporary and Future Science', Project no. SRFS2122-3H01).

system implements *every* computational model simultaneously. They conclude on the basis of unlimited pancomputationalism that computational descriptions of physical systems are 'trivial' or 'vacuous'. This argument has potentially dramatic implications, if sound. Not only would it seriously threaten computationalist programmes in the philosophy of mind, but if computational descriptions are trivial it would seem we ought to substantially curtail — if not eliminate outright — the use of such descriptions in scientific inquiry generally.

Few philosophers nowadays accept this radical conclusion. The most popular response to Putnam and Searle's argument rejects unlimited pancomputationalism. This is usually achieved by imposing a global, context-insensitive constraint on implementation. There are quite a few proposals about how this might go. The most prominent suggestions appeal to counterfactual,[1] causal,[2] mechanistic,[3] representational,[4] information-theoretic,[5] naturalness,[6] and pragmatic[7] constraints in an effort to block unlimited pancomputationalism. Others are surely possible. Perhaps unsurprisingly, however, none has emerged as a clear victor, and Putnam and Searle's challenge remains with us today.

In light of this situation, I want to explore an alternative response. This alternative is designed to incorporate the insights of these others while going beyond them in certain respects. In particular, I will suggest that under certain plausible assumptions about the nature of computational ascription, the Putnam-Searle argument loses much of its force. Roughly put: physical systems implement computational models only relative to a contextually salient way of regarding a system computationally. But, relative to such a way of regarding, there is little reason to think that computation is trivial. Or so I will argue.

I start in Section 2 by reconstructing a generic triviality argument. Then, I argue in Section 3 that when systems implement computations, they do so relative to specific ways of regarding those systems computationally — what I will call labelling schemes. Section 4 argues that there can be non-arbitrary grounds for describing physical systems in terms of specific labelling schemes. Section 5 contrasts my response to others in the literature. Section 6 concludes.

[1](Copeland, 1996; Rescorla, 2014).

[2](Chalmers, 1996; Chrisley, 1994; Melnyk, 1996; Scheutz, 1999).

[3](Milkowski, 2013; Piccinini, 2015, 2020).

[4](Rescorla, 2014; Shagrir, 2022; Sprevak, 2010)

[5](Millhouse, 2019; Anderson & Piccinini, 2024).

[6](Godfrey-Smith, 2009).

[7](Dewhurst, 2018; Matthews & Dresner, 2017; Schweizer, 2019).

2 Triviality arguments

Different presentations of the triviality argument may be found in the literature, varying in terms of their exact scope and formulation.[8] My aim in this section is to sketch a generic version of the argument. On my reconstruction, the argument proceeds in two main steps. The first establishes unlimited pancomputationalism. The second moves from unlimited pancomputationalism to the claim that physical computation is trivial. I will take each in turn.

2.1 Unlimited pancomputationalism

I will begin with some orienting remarks about the notion of computational implementation. Computer and cognitive science seek to understand the behavior of certain physical systems in terms of computations those systems perform. These endeavours are routinely framed in terms of abstract, mathematically characterized computational models drawn from theoretical computer science. Typical examples include machine models like finite automata, pushdown automata, or Turing machines, and program models written in any of a variety of languages like Lisp, Python, or Haskell. When a physical system is accurately described by such a model, it is said to 'implement' or 'realize' that model.[9]

Philosophical theories of implementation attempt to specify in general terms the conditions under which a physical system implements a computational model. Perhaps the most straightforward is the simple mapping account (Godfrey-Smith, 2009), which holds that a physical system P implements a computational model M just in case the physical state transitions carried out by P approximately mirror the formal state transitions specified by M. This is usually cashed out by saying that the computational model and the physical system (at a certain level of description) are *isomorphic*:

> **The Simple Mapping Account (SMA)**
> P implements M if and only if P is isomorphic to M; that is, just in case:
>
> 1. There is a grouping of microphysical states of P into state-types, and a function f mapping state-types of P to states of M, such that

[8]See (Sprevak, 2019) for a survey.

[9]Computational models in the this sense are scientific models and their targets are physical systems which perform computations. According to a broader usage, computational models are scientific models which rely on computational techniques, but which may not ascribe computations to their targets (Weisberg, 2013). Triviality arguments concern computational models in the former, narrower sense (Piccinini, 2007).

2. Under f, the state-transitions of P mirror those of M; i.e., whenever P is in state p_1, where $f(p_1) = m_1$, and $m_1 \to m_2$ is a formal state transition, then P goes into state p_2, where $f(p_2) = m_2$.

SMA is prima facie attractive. Not only does it characterize implementation in terms of the antecedently well-understood notion of a structure-preserving map, but it also smoothly accommodates the implementation conditions for a wide range of computational models (Sprevak, 2019; Schweizer, 2019). Yet, despite these virtues, **SMA** is widely taken to be untenable on the grounds that it entails unlimited pancomputationalism:

> **Unlimited Pancomputationalism (UP)**
> Every sufficiently sophisticated physical system simultaneously implements every computational model.[10]

UP follows from **SMA** if every sufficiently sophisticated physical system is isomorphic to every computational model; that is, if for any physical system and any computational model, there is a mapping from a system's state-types to the states of a computational model, such that, under that mapping, the two are isomorphic. And this follows under two quite modest additional assumptions.

The first is the assumption that there is an isomorphism between a physical system and a computational model whenever the former's state-types are equinumerous with the latter's formal states. This assumption is backed by a basic model-theoretic technique known as Push-Through, which provides a generic method for defining an isomorphism between any set-theoretic structures whose carrier sets are equinumerous. Intuitively, given any computational model, Push-Through shows us how to define a transition relation on physical state-types that exactly mirrors the formal state transitions specified by that model. [11]

The second is the empirical assumption that for any give time interval, any physical system has (at least) denumerably infinitely many microphysical states.

[10]Sometimes, unlimited pancomputationalism is characterized as the claim that every sufficiently sophisticated physical system computes every Turing-computable function. These characterizations are for all intents and purposes equivalent, since the model or models implemented by a system determine the function(s) it computes.

[11]A little more precisely, let P be a physical system and let S_P be a set of state-types defined over P's microphysical states. And let $M = \langle S_M, T_M \rangle$ be a computational model, where S_M is a set of formal computational states and $T_M : S_M \to S_M$ is M's transition function. And suppose that S_P and S_M are equinumerous, so that there is a bijection $f : S_P \to S_M$. Now we can define a physical state-transition function $T_P : S_P \to S_P$ as follows: let $T_P(s_1) = s_2$ iff $T_M(f(s_1)) = f(s_2)$. Then the structure $P = \langle S_P, T_P \rangle$ is isomorphic to M. For more on Push-Through, see, e.g., (Button, Walsh, & Hodges, 2018).

This ensures that it is possible to find a set of physical state types equinumerous with those of any computational model. Different philosophers motivate this step in different ways. For instance, Putnam argues that under certain physical assumptions, every 'ordinary open system'—such as a rock lounging in the sun—proceeds through uncountably many microphysical states in any given time interval. Shenker and Hemmo (2022), by contrast, argue for this assumption on general statistical mechanical grounds. However we establish the existence of these states, the important point is that by judiciously grouping them, we can identify a set of physical state types equinumerous with the formal states of any computational model.

Thus **UP** follows from **SMA** given basic model theory and a modest empirical assumption. Briefly: let P be any physical system and M be any computational model. By the empirical assumption, P has state-types equinumerous with the states of M. By Push-Through, the two are isomorphic. Hence, by **SMA**, P implements M. **UP** follows, since P and M were arbitrary.

2.2 From pancomputationalism to triviality

Why is **UP** a problem? The usual charge is that it renders computation 'trivial'. This charge is perhaps best understood as a complaint about the theoretical status of computation, in that any account which entails **UP** suffers from a variety of distinct but related theoretical failures. These failures concern an account's descriptive and explanatory ambitions, respectively.

Descriptive failures

To begin, notice that the argument for pancomputationalism relies on extraordinarily weak empirical assumptions. The only empirical facts cited in the argument are facts about number of states or parts of physical systems. The argument does not rely on facts about the character of these states or parts, nor does it rely on facts about their arrangement, interactions, or activities. Nor, for that matter, does it rely on facts about the gross structure of physical reality, concerning the distribution of matter, the structure of spacetime, or which natural laws obtain. Given all this, it is unsurprising that any theory that entails pancomputationalism suffers from a variety of descriptive failures:

- *Lack of empirical content.* The empirical demands required to establish **UP** are extremely modest, concerning only the cardinalities of a system's state-types. Insofar as these demands would be met in almost every physically possible situation, it would thus appear that theories of implementation lack empirical content almost entirely.

- *No empirical discoveries.* While one might have thought that substantial empirical investigation would be required to determine whether a system computes, under **UP** such questions are answerable largely from the armchair, by reflecting on the number of microphysical states of the system in question.

- *Widespread predictive failure.* Information about what a system implements ought to be predictively useful. The fact that a robot implements this pathfinding algorithm should presumably allow one to predict that it will take that path through an obstacle course. Yet it is hard to see how to use such knowledge to predict anything given **UP**. Under **UP**, the robot simultaneously implements a wide variety of different pathfinding algorithms, some of which may find quite different paths through the course. It is thus hard to see what would justify the prediction that the robot will take one path instead of another.

- *Unwarranted retrodictive success.* While one might have thought that retrodiction of past pathfinding behavior is the sort of thing that we could in principle get wrong, under **UP** we cannot help but retrodict successfully. This is because, no matter what path the robot in fact takes, there will be some pathfinding algorithm it implements under which it would have taken that path.

- *Extensional inadequacy.* Whereas the computational sciences ascribe computational properties to comparably few physical systems — chiefly engineered devices and cognitive systems — **UP** entails that these notions apply to every physical system. Thus while the account correctly classifies paradigmatic cases, it incorrectly classifies anti-paradigms more or less across the board (Piccinini, 2015; Sprevak, 2019).

Explanatory failures

A second concern is that **UP** undermines **SMA**'s explanatory power. Computer and cognitive science attempt to explain the capacities and behavior of certain physical systems in terms of the specific computational models they implement. For instance, the reason why a laptop takes $O(n^2)$ time to sort a list of n digits might be that it implements SELECTION-SORT rather than some other sorting algorithm. Had it implemented a more efficient sorting algorithm, like MERGESORT, it would have sorted a list of n digits in $O(n \log n)$ time. Yet if **UP** holds, the laptop does implement MERGESORT— indeed, it implements every sorting algorithm at once.

One way of framing this problem foregrounds the contrastive character of many scientific explanations. This is sometimes expressed by the idea that scientific explanations answer "what if things had been different?" questions (Woodward, 2003). In the computational case, such questions concern how physical systems would have been different if those systems had been different computationally. Yet under **UP** there is no clear sense in which things might have been different computationally, for in every possible world even remotely similar to ours every physical system is computationally indistinguishable from every other.

This overall diagnosis can be refined by more sharply delineating two distinct explanatory scenarios. In *intrasystemic* scenarios, computations explain contrasts between the actual versus merely possible properties or behavior of a system. The laptop has a certain performance profile because it uses one sorting algorithm rather than another. Similarly, a robot takes one path through a maze (rather than another) because it implements this path-finding algorithm (rather than that one). In *intersystemic* scenarios, by contrast, computations explain differences between systems. The fact that the visual system computes depth from binocular disparity partly explains why healthy individuals exhibit competent grasping behavior while those with various cortical injuries — not to mention manifestly non-cognitive systems such as rocks, walls, and pails of water — do not.

The trouble is that successful explanation in either case requires a contrast (Hitchcock, 2013). The fact that healthy visual systems compute some depth-extraction function explains differences grasping behavior only if those same depth-extraction computations are not also implemented by injured visual systems (or rocks, walls, etc.). Similarly, the fact that the robot implements some pathfinding algorithm explains why it takes a certain path only if it doesn't also implement another algorithm which charts a different course. Yet under **UP** no such contrasts obtain. Rocks and injured visual systems implement the same computational models as healthy visual systems, and the robot implements every pathfinding algorithm, no matter what path it in fact takes. **UP** threatens computational explanation, then, because it violates certain basic requirements on contrastive explanation.

Stepping back, while I do not claim to have identified all the problems that might be raised by **UP**, these failures are arguably the most serious. In light of this, I will take the principle task in responding to triviality to be to ameliorate all or at any rate most of the descriptive defects discussed above, and to vindicate computational explanation in both intersystemic and intrasystemic scenarios. I consider next how this might be done.

3 Implementation and the applicability of mathematics

Given my reconstruction, we can resist the triviality argument either by rejecting **UP**, or by rejecting the inference from **UP** to triviality. Most responses take the first route—we will come back to some examples later on. Here, I want to explore the possibility of addressing the triviality argument *without* showing that **UP** is false. My starting point is the idea that describing physical systems computationally is an instance of the more general practice of describing systems mathematically. Given this orientation, it is natural to consider whether an analogue of the triviality argument applies to other applications of mathematics as well. This section considers such an argument and uses it to offer an intuitive diagnosis of what's gone wrong in the original triviality argument.

3.1 Harmless abundance

One widely noted feature of the application of mathematics to physical systems is that physical systems do not wear their mathematical characteristics on their sleeves. Regarded one way, a physical system has a certain mathematical character; regarded a different way, it has another. Perhaps the most well-known example of this involves the application of finite cardinals in ordinary counting practice. Here's Frege in the *Grundlagen*:

> If I give someone a stone with the words: Find the weight of this, I have given him precisely the object he is to investigate. But if I place a pile of playing cards in his hands with the words: Find the Number of these, this does not tell him whether I wish to know the number of cards, or of complete packs of cards, or even say of points in the game of skat ... Number, cannot be said to belong to the pile of playing cards in its own right, but at most to belong to it in view of the way in which we have chosen to regard it ... What we choose to call a complete pack is obviously an arbitrary decision, in which the pile of playing cards has no say. (Frege, 1884, §21)

Frege observes that different 'ways of regarding' physical stuff may yield different counts.[12] As we now say, counting proceeds under an appropriate sortal concept, which individuates stuff into objects. Relative to the concept CARD this stuff is fifty-two; relative to DECK, it is one.

[12]Stuff-talk is awkward but ineliminable if we wish to avoid begging questions. There is typically no deep question about the size of a *collection*, for instance, because collections come preindividuated. That is, collections are collections of *objects*, and it is plausible to think that we cannot meaningfully talk of objects without presupposing some way of individuating them. Frege's point is that undifferentiated physical matter, by itself, does not determine a particular count.

Frege uses this observation to motivate the controversial view that number belongs to concepts, not physical objects. I take no stand on that further claim here. What's important for my purposes is Frege's observation that multiple distinct numerical descriptions apply simultaneously to a given hunk of stuff. This bears more than a passing resemblance to Putnam and Searle's claim that multiple *computational* descriptions apply to a fixed physical system. It is therefore instructive to consider whether an analogue of the triviality argument might be developed for applications of finite cardinals as well.[13]

The first step of such an argument would be to establish, not pancomputationalism, but *pancardinalism*:

> **Pancardinalism**
> Every sufficiently large expanse of physical stuff simultaneously falls under every sortal concept.

To establish **Pancardinalism**, we begin with an analogue of the simple mapping account, which holds that a mathematical structure applies to a system (at a certain level of description) just in case the structure and that system, so described, are isomorphic.[14] In the case of finite cardinals, this amounts to the claim that a collection is n just in case the objects in that collection are equinumerous with the first n cardinals.

The next step is to show that for *any* finite cardinal and any sufficiently large expanse of physical stuff, there is a concept relative to which that matter falls under that cardinal. And although it would be tedious to argue for this point in detail, it is not too hard to see that with a little ingenuity we will be able to identify such concepts. For instance, although we might count cards or decks, we might also count faces or suits or even more recondite entities like half-cards, quarter-cards, thirds of cards, and so forth. While many of these will be highly artificial, there is no reason to think that we cannot proceed in this way and thereby assign extraordinarily many distinct cardinals to a given hunk of physical stuff. This would establish **Pancardinalism**. Should we therefore conclude that whole number counting is trivial?

Clearly not. We cite finite cardinals to address numerosity questions of the form "how many Fs are there?". Such questions are raised in particular conversational contexts, which fix the relevant sortal F. It is of course true that had we

[13]See (Matthews & Dresner, 2017) for a related thought concerning the representational theory of measurement.

[14]This way of understanding the applicability of mathematics traces back at least to (Quine, 1960); (Pincock, 2011, ch. 2) is a more recent statement. Even philosophers who deny that the existence of an appropriate isomorphism is sufficient for a mathematical structure to apply to a given system typically accept that such isomorhpisms are necessary (see, e.g., Batterman, 2010; Bueno & French, 2018; Bueno & Colyvan, 2011).

varied F, we may have arrived at a different count — focus on CARD rather than DECK and the count is fifty-two, not one. But as long as a given count is generated in light of an appropriate sortal, it is hard to see how the existence of different counts, generated by different Fs, trivializes anything. The reason is straightforward: in that conversational context, those alternative counts answer questions that weren't being asked.

Moreover, although we might regard them as bizarre, unprincipled, or otherwise useless, it seems clear that we nevertheless *can* count with gruesome gerrymandered sortals (even if in some cases we have little reason to). This is a manifestation of the more basic fact that arithmetic (and indeed mathematics more generally) furnishes an extremely flexible set of domain-neutral tools for reasoning about physical systems and their properties. It would be a mistake to limit our ability to deploy this tool by imposing additional constraints over and above the existence of an appropriate isomorphism. And this suggests that it would be a mistake to respond by attempting to *constrain* our account of numerical 'implementation' and thereby block **Pancardinalism**. Rather, it is enough to note that, relative to a specific conversational context, certain ways of counting are simply irrelevant.

The provisional lesson is that ordinary counting practice is not in any obvious sense trivialized by the fact that multiple cardinals simultaneously apply to a given hunk of physical stuff. Although there is an abundance of ways we *could* count Frege's cards, this abundance is harmless because counts proceed under specific, contextually salient ways of regarding— that is, specific, contextually salient concepts. Focus attention on a specific concept, and applications of finite cardinals are as non-trivial as one could reasonably want. With this in mind, let's turn back to the original triviality argument.

3.2 Diagnosis

To begin, observe that the triviality argument does not show that a physical system implements every computational model *tout court*. Rather, when a physical system implements every computational model, it does so relative to distinct 'way of regarding' that system computationally. This is perhaps easiest to see with Putnam's original argument. Putnam argues that a rock implements every deterministic finite automaton (DFA). The crux of this argument is a recipe that takes a DFA and shows us how to label microphysical states of the rock so that, under that labelling, the rock implements that DFA. Importantly, however, the recipe yields a distinct labelling for each DFA: it uses one for the automaton ABA, another for the automaton ABABA, and so on. Thus, although the rock simultaneously implements every DFA, it does so only relative to distinct labellings—distinct 'ways of regarding' the rock computationally in each case.

Putnam concludes that the claim that the rock implements *any* specific DFA is trivial. But this is too quick. Relative to a *fixed* labelling, claims about the computational properties of the rock are no more trivial than are claims about the numerical properties of Frege's cards, when the latter are subsumed under a specific concept. Indeed, once we focus attention on a specific labelling of the rock, claims about its computational properties and behavior do not obviously suffer from the descriptive or explanatory defects identified earlier. For instance, given the ABA-labelling, we can accurately predict that it will be in state A at the next time interval if it is currently in state B and we can explain that it is currently in state B because at the previous interval it was in state A. To be sure, these claims are hardly surprising. But this shows only that it is scientific unilluminating to describe the rock in terms of a two-state DFA in the manner suggested by Putnam's recipe. It does not on its own show that such descriptions are *trivial*.

This is perhaps easy to overlook if we focus only on examples like Putnam's rock. Because the rock exhibits little interesting macroscopic behavior, it would be admittedly arbitrary to prefer one of Putnam's schemes to any other when describing it computationally. It is thus unsurprising that attempts to describe the rock computationally strike us as trivial. But this shows, at most, that the rock is not fruitfully understood in computational terms. It would be a mistake to conclude that there cannot be non-arbitrary grounds for working with specific labellings when we wish to understand other kinds of systems computationally, just as there can be non-arbitrary grounds for working with specific concepts in our ordinary counting practice.

When we consider the kinds of systems characteristically investigated by the computational sciences, we must therefore ask whether there are grounds for describing these systems in terms of specific labellings. If there are, then we would be in a position to make substantive judgments about the descriptive and explanatory virtues of the models implemented under those labellings. This would be so despite the fact that these systems might implement other computational models—indeed, perhaps even all of them—under other labellings. Thus, to defuse the triviality argument, it would be enough to show that there *are* or at any rate *could be* non-arbitrary grounds for describing these systems in terms of specific kinds of labelling schemes. Of course, to be absolutely clear, I have not yet argued that there *are* such grounds. But I think there are. That's coming up next.

4 Putting implementation in context

At the center of my proposal is the idea that when a physical system implements a computational model, it does so relative to a specific way of regarding that system computationally. Following Copeland (1996), I call these ways *labelling schemes.* I will regiment this idea with the following slight modification of **SMA**:

> **The Relativized Simple Mapping Account (RSMA)**
> P implements M relative to a labelling scheme L just in case, under L, P is isomorphic to M.

According to **RSMA**, implementation is a three-place relation between a physical system, computational model, and a labelling scheme. This section starts by examining labelling schemes in more detail. Then, I argue that in certain engineering and scientific contexts there can be principled reasons for describing systems in terms of specific kinds of labelling schemes. I round off the section by revisiting the descriptive and explanatory failures identified in Section 2.

4.1 Labelling schemes

RSMA is designed to accommodate the observation that that physical systems do not wear their computational characteristics on their sleeves. Rather, physical systems implement computational models only under specific 'ways of regarding' them computationally—only under an appropriate labelling scheme. Perhaps the simplest illustration of this concerns a bistable digital circuit which transforms input voltages into output voltages according to a certain pattern. Perhaps it outputs 5V just in case both input voltages are 5V; otherwise it outputs 0V. By varying the logical interpretation of these voltage levels, we can vary the function computed by the gate. If we interpret the 5V state as logical 1 and the 0V state as logical 0, the gate computes logical AND; under the reverse assignment, it computes logical OR. Under different ways of regarding the circuit computationally—under different labelling schemes—it computes a different logical function.[15]

Labelling schemes are the computational analogue of concepts. They are part of the representational machinery *used* by practitioners to describe physical systems computationally.[16] Labelling schemes accomplish two main tasks: they group the microphysical states of a system into a set of state types, and they assign specific mathematical states or values to the state types. Both of these tasks can be

[15]It is also possible to vary the logical function implemented by employing different groupings of the circuit's microphysical states. For examples and recent discussion of such cases, see (Papayannopoulos, Fresco, & Shagrir, 2022a, 2022b; Fresco, Copeland, & Wolf, 2021; Shagrir, 2022; Piccinini, 2020; Shagrir, 2022; Piccinini, 2020).

[16]It is important to distinguish this aspect of labelling schemes from the idea that a physical system implements a computational model (only) if it has certain representational properties (e.g., Shagrir, 2022). Even those who reject the claim that computation involves representation should accept that labelling schemes are representational in the sense that they are part of our representational machinery.

rendered formally as functions: a grouping function which takes microphysical states to state-types, and an assignment function which takes state-types to labels. A labelling scheme can then be treated as the composition of these two functions.

How states are grouped depends on their properties, and different schemes will group states in different ways. Intuitively, the idea is to collect together microphysical states which are similar in some antecedently specified respect. For instance, in the circuit we grouped states according to their electromagnetic properties: a microphysical state falls under one state type if its voltage is within some delta of 5V, while it falls under another if it is within some delta of 0V. Of course these are not the only groupings possible. Another grouping might collect together states that fall in some circumscribed range; perhaps states in a 0-6V range fall into one type,and states in a 6-8V range fall into another. Yet other groupings might collect states according to their causal or representational features, so that microphysical states are grouped together only when they occupy approximately similar positions in a network of causal relations or they carry approximately similar representational content—I return to these alternatives later on, in Section 5.

The domain of the grouping function is a set of microphysical states. This set is determined by a prior, *non-computational* characterization of a system's microphysical states, furnished by some background theory appropriate for systems of that sort (cf. Scheutz, 1999; Anderson & Piccinini, 2024). We must assume that the background theory yields a determinate answer to the question whether a given microphysical state falls under a given state-type to ensures that any set of state-types can support a structure-preserving map. However, we do not need to assume that there is a single background theory appropriate for every kind of physical system, although it is convenient to do so. We should also not assume that the salient background theory will yield a *unique* set of state-types for a given physical system—ordinarily it will not. For instance, nothing about electromagnetic theory per se (if that is the salient background theory) forces us to work with the groupings identified in the last paragraph.

The assignment function takes these state-types and assigns them mathematical values. Some of these values will be internal states of a model, like states of a Turing machine's read-write head. Others will be inputs, outputs, or intermediate values of a data structure like entries on the Turing machine's tape. For instance, in the circuit, we assigned logical 1 and logical 0, although these are assigned in different ways in the two alternatives.

For any system with even a modest number of microphysical states, there will typically be an abundance of schemes to choose from when we wish to describe it computationally. This can be because there are multiple distinct possible groupings, or multiple distinct possible assignments, or both. As a result, it is possible for a single system to implement the same model under different schemes. It is also possible for a single system to implement different models under different schemes. Importantly, however, this abundance will be harmless if there are principled grounds for preferring some schemes over others when we wish to understand phenomena computationally. I will argue next that there are.

4.2 Scheme selection

Let me start by managing expectations. I somewhat doubt there is an informative general story told about when a given labelling scheme should be preferred over others. This is because the question of what makes a given computational description relevant is an instance of the more general question of what makes a given *scientific* description (e.g., model or theory) relevant. And there is, to put it lightly, no firm agreement amongst philosophers of science about how to answer the latter question, at least in general terms. Of course, scientists ought to favour descriptions which save the relevant phenomena, are simple, explanatory, improve understanding, and so forth. But this tells us little about which kinds of schemes will exhibit these virtues in practice.

Matters are more promising, however, once we bracket general considerations and attend to specific descriptive or explanatory projects. Computational models are applied by researchers in specific investigative contexts, to fashion devices with certain desirable properties or to understand the properties or behavior of certain kinds of systems. These applications are guided and constrained by a variety of local considerations including (i) a prior, non-computational characterization of the phenomena or systems of interest, (ii) the available investigative or engineering techniques and practices (e.g., measurement, modeling, or fabrication techniques), and (iii) the evaluative standards and aims accepted by the researchers in question. Although I cannot hope to do full justice to computational practice in what follows, I will suggest that these considerations can motivate the decision to work with a specific labelling scheme, or at any rate a principled family of such schemes, in the most important engineering and scientific contexts in which computational descriptions are typically deployed.

Engineering practice and computing artefacts

To begin, consider a simple artificial computing system, like the bistable digital circuit considered earlier. These devices are explicitly engineered to compute specific logical functions while meeting various design criteria (concerning, e.g., reliability, modularity, power consumption, and so forth). Thus, it is reasonable to describe them in terms of a labelling scheme that allows us to use them to compute these functions while satisfying these criteria. That's to say, given this goal, it is hardly arbitrary to describe the circuit in terms of a specific scheme.

Concerning the grouping of states into state-types, contemporary CMOS digital circuits are explicitly designed to distinguish two stable states: a low state (typically around 0V) and a high state (nowadays, around 2V or 3V) (Harris & Harris, 2013; Weste & Harris, 2011). Thus, it is eminently reasonable to work with a labelling scheme that groups states according to these specific electromagnetic properties: a microphysical state falls under the 'low' state type if it falls within some delta of 0V; it falls under the 'high' state type if it falls within some delta of 2V or 3V; and it is outside the scope of the labelling scheme otherwise. Of course, as recently noted, there are many other ways of grouping a circuit's microphysical states, many of which will cut across this particular grouping. Interpreting a device in term of these alternatives will ordinarily not yield a description that is suitably modular, reliable, or useful for solving problems of interest, and so are safely ignored.

Once microphysical states are grouped, they are assigned mathematical values. Conventionally, states in the ground range are assigned logical 0, while states in the high range are assigned logical 1. This assignment is not mandatory. Some early electronic computers use the reverse assignment (Burks & Burks, 1988). Indeed, for very simple systems such as atomic digital circuits, nothing about their physical structure mandates the assignment of '0' to the ground state and '1' to the high state. There is thus a degree of slack between groupings and the assignment of logical values that may taken up by conventional or pragmatic considerations (Dewhurst, 2018).

Although proponents of **UP** sometimes suggest that the decision to describe a system computationally one way rather than another is a matter of 'free interpretation' (e.g., Searle, 1992), it is evident that this decision is guided by a variety of considerations about how to best build devices that meet our needs. And even if these considerations sometimes involve pragmatic or conventional aspects, as when assigning logical values, that does not render descriptions in terms of that scheme arbitrary. Moreover, there are sometimes principled reasons for departing from a conventionally established scheme. Virtual machines are a case in point. In such cases, we employ a new labelling scheme to see a system as performing a different kind of computation (e.g., as running a different operating system).

Stepping back, the point is that our engineering goal of building a system that computes a certain logical or mathematical function guides how artefacts like contemporary circuits or digital computers are best described computationally. What matters is that one is able to use a device to solve a problem of interest, under some fixed characterization of the device's inputs and outputs. It is hardly a trivial question whether the device in fact solves *this* problem, under *this* characterization, even if it could in principle be used to solve other problems under *other* characterizations.

Scientific modeling and natural computing systems

Matters are more complicated for natural computing systems. This is because which scheme or schemes it is appropriate to use when theorizing about some system is itself an open empirical question. Minimally, an adequate scheme should be consistent with the available behavioral data and known structural and functional features of a target system. In practice, there may be a range of schemes consistent with what is known about a target system.

Regarding behavior, computational models are often developed to explain some non-computationally described behavioral phenomenon. Adequate labelling schemes must ensure that hypothesized computations smoothly interface with behavioral inputs and outputs described non-computationally. Not just any scheme will adequately capture how patterns of retinal irradiation produce competent grasping behavior, for instance (Egan, 2012; Godfrey-Smith, 2009).

However behavioral data alone will typically not fix a unique computational description of a system. A given pattern of input/output behavior can be produced by many different computations. Information about the non-computational structural and/or functional features of a system helps to winnow down the range of promising schemes. This information bears primarily on the grouping function. For natural computing systems, a grouping function will typically cite the specific structural and functional features of the system under investigation. For instance, for computations in the brain, groupings will ordinarily be 'neurobiologically plausible' in the sense that states are grouped, at least in part, according to the known structural or functional features of the relevant brain structures. Of course, there is nothing in principle that would prevent researchers from investigating the computational properties of brains under more recondite groupings. If models implemented under such groupings yield novel predictions, provide satisfying explanations of otherwise puzzling phenomena, and so forth, then there is reason to take such groupings seriously.

As with artificial computing systems, there may be flexibility when assigning mathematical labels to grouped states. Here too slack may be picked up by conventional or pragmatic considerations. This possiblity looms largest for highly complex natural computing systems, such as the brain. Given our limited understanding, the choice to model a specific brain region in terms of a given computational model may come down in part to the fact that the model is familiar, straightforward to work with, mathematically elegant, and so forth. This should come as no surprise. Pragmatic considerations such as these guide the application of mathematics throughout science, and it would be remarkable if the application of computational notions were an exception.

Pragmatic considerations may enter the picture by another route. As with any scientific modelling practice, the choice of scheme, and a model implemented thereunder, may be underdetermined by the available behavioral, structural, or functional evidence. In such cases, we may again fall back on pragmatic or conventional considerations. Of course, once a model, under a specific scheme, is adopted as a working hypothesis, we may in light of further investigations be led to refine or even abandon it and the scheme that supports it. But underdetermination is a fact of scientific life, and there is no reason to think that the existence of underdetermination makes the decision to work with a given scheme (or family of schemes) arbitrary.

Clarifications

Two comments before moving on. First, I remain agnostic about the question whether these considerations will recommend schemes which are metaphysically privileged in some deeper sense (joint-carving, natural, or what have you). My claim is that despite **UP**, in specific investigative or engineering contexts, there can be good reasons for choosing to work with a specific (kind of) labelling scheme, just as in specific *conversational* contexts there can be good reasons for choosing to work with a specific concept. This, I will argue next, is enough to address the descriptive and explanatory failures usually thought to attend **UP**.

Second, however, I am *not* merely suggesting that philosophers simply defer to scientific or engineering practice when considering how to describe systems computationally. Although any adequate response to the triviality problem should be grounded in computer and cognitive scientific practice, the ultimate tribunal is whether these schemes meet our broader descriptive and explanatory demands. They might not—there are no a priori guarantees. Progress on this question will require a closer look at the computational sciences than I have space for here.

4.3 Theoretical failures revisited

With these points in mind it is time to revisit the theoretical failures thought to be incurred by **UP**. In what follows I assume that there are non-arbitrary grounds for describing a system in terms of a specific labelling scheme, or at least a principled family of schemes.

Descriptive failures

Lack of empirical content. Relative to a fixed labelling scheme, it is an empirical question whether a given system implements a given computational model.

Consider a scheme under which a digital circuit computes AND. This scheme maps voltage levels to logical values. We can determine empirically whether a given physical system implements an AND-gate under this scheme by determining whether it has the right electromagnetic properties and transforms them in the right way. And although the empirical demands imposed by this scheme are rather minimal, more complicated schemes, necessary for more complicated models, will impose more exacting demands. And in both cases the empirical demands require far more than that a system simply has a certain number of microphysical states.

No empirical discoveries. Relative to a fixed scheme, we may well discover that a physical system implements one computation rather than another, or indeed no computation at all. While this criterion is less important for artificial computing systems, it is more important for natural systems. For instance, relative to the scheme for AND, by examining how a configuration of neurons transforms voltage levels we may discover that they compute AND rather than OR. We may also discover that there is *no* good way to characterize a give system computationally, for we may find that there is no labelling scheme under which computational descriptions help us understand its properties or behavior. This, I submit, is the lesson to be learned from Putnam's rock.

Widespread predictive failure. Relative to a specific scheme, a robot will implement a specific pathfinding algorithm, hence will choose a determinate path through the obstacle course. We can thus make substantive predictions about the robot's behavior, at least some of which may succeed. Of course, for any prediction there will always be *some* scheme under which that prediction comes out correct. But there is no reason to think that these alternatives will be indistinguishable. For instance, some will yield accurate predictions in a wider range of scenarios, including novel scenarios, than others, and this will be prima facie reason to prefer them.

Unwarranted retrodictive success. Similarly, given the path actually taken, once we fix upon a particular scheme there is a genuine question whether the pathfinding computation implemented relative to that scheme could have produced that path. Under a salient scheme, we can make substantive retrodictions about the robot's past pathfinding behavior, at least some of which may fail. Of course, we must strike a balance between schemes that balance predictive and retrodictive success. This is no different in principle from curve-fitting exercises elsewhere in science. We might fit a set of data points perfectly by introducing additional terms into our formula, at the cost of simplicity and and perhaps predictive accuracy. All else equal, we should prefer curves that satisfactorily balance simplicity, retrodictive and predictive accuracy, and much else. Ditto for labelling schemes and the models they underwrite.

Extensional adequacy. This one is trickier. The trouble is that labelling schemes appropriate for one class of systems might not apply to another class of systems. Suppose, for the sake of argument, that (i) schemes most appropriate for contemporary digital computers group states according to their electromagnetic properties, while schemes most appropriate for brains group states at least partly according to their chemical properties, and (ii) schemes appropriate for digital computers do not apply to brains, and vice versa. Then, relative to the former, brains don't compute, whereas relative to the latter digital computers don't compute. Isn't this a problem?

In response, I suggest that we reject a context-independent conception of extensional adequacy. What matters is that we can correctly categorize systems according to the standards operative in a given scientific context. It is enough if, for instance, we can distinguish digital computers from rocks, walls, etc. in those contexts where digital computers (and their properties, behavior, etc.) are the phenomena of interest. Admittedly, this does leave open the possibility that scientists in different contexts might talk past each other if they rely on different labelling schemes. In such circumstances, one can try to disambiguate by flagging the schemes at issue, being clear about one's theoretical goals and interests in describing systems computationally, and so forth. I hazard that in many cases this will be enough to avoid serious misunderstanding. However, while I think such scenarios will be rare, I see no reason to think that they will be completely unavoidable. But that's life.

Explanatory failures

With respect to computational explanation, the main point is that there can be genuine computational contrasts relative to a fixed labelling scheme. Start with the intrasystemic case. Relative to a specific labelling scheme, there is a fact of the matter about what model a system implements. We can thus advert to that specific model to explain the properties and behavior of that system. For instance, we can explain that the robot takes this path because it implements that pathfinding algorithm, relative to a scheme appropriate for explaining how the robot moves about its environment.

Regarding the intersystemic case, the point to notice is that schemes appropriate for explaining one kind of system will typically not apply to other kinds of system. For instance, schemes appropriate for explaining computations in the visual system will presumably key in to specific features of such systems: neuronal organization and activation patterns, voltage levels, spike rates, and so on. Relative to a scheme framed in terms of these features, healthy visual systems will quite likely perform different computations than injured visual systems. This is because injured systems display patterns of organization and activation, spike rates, and so forth. Thus, relative to such a scheme we can explain why individuals with healthy visual systems display more competent grasping behavior (etc.) than their injured counterparts.

Moreover, relative to a scheme appropriate for healthy visual systems, it is highly unlikely that manifestly non-cognitive systems such as rocks, walls, or pails of water compute anything, much less complex depth-from-disparity computations. The reason is straightforward: these systems will simply do not have the kinds of features cited by schemes appropriate for explaining visual systems: they aren't composed of neurons, don't have spike rates, and so on. Accordingly, in cognitive scientific contexts we can appeal to the computations carried out by the visual system to explain why individuals with healthy visual systems, but not rocks, walls, etc. display certain grasping behavior.

Of course, rocks presumably implement these computations somehow, relative to some (perhaps highly gerrymandered) labelling schemes. But relative to the kinds of labelling schemes operative in the contexts in which cognitive scientists usually operate, these schemes will be utterly inappropriate for explaining physical systems. Cognitive science countenances schemes appropriate for explaining the kinds of systems it investigates, namely brains. Claims about computations in, say, the visual system, must be assessed relative to such schemes. Insofar as rock-computations rely on schemes utterly inappropriate for describing brains, there is little reason to think that **UP** threatens the use of computational notions in the cognitive scientific contexts in which those notions are ordinarily deployed.

5 The usual suspects

Next I would like to contrast my response to the other main strategy in the literature, which targets **UP** directly. That strategy, recall, attempts to block **UP** by supplementing **SMA** with an in-principle, global constraint on admissible groupings of states into state-types. In the language of labelling schemes, the idea is that only certain *kinds* of labelling schemes may feature in genuine implementation relations. For instance, causal accounts privilege causal labelling schemes, according to which microphysical states may be grouped together only when they occupy approximately similar positions in a network of abstract causal relations, while representational accounts privilege representational labelling schemes, which hold that microphysical states may be grouped together only when they instantiate approximately similar representational properties.[17] And others are possible, as I

[17]Bear in mind that 'representational' labelling schemes in this sense apply *only* to systems which instantiate the relevant representational properties. This is distinct from the claim that

noted in the introduction. However, for simplicity's sake I will focus on these two proposal in what follows.

This is not the place to argue that my account is all-things-considered preferable to these alternatives. Instead, I would like to sketch how things look from the vantage of **RSMA**. Understood as competing theories of the nature of computational implementation, causal and representational accounts are in tension. But from the perspective of **RSMA**, causal and representational labelling schemes reflect distinct and potentially complementary ways of *applying* computational notions for certain descriptive or explanatory ends. From this perspective, the principle question is *not* whether causal or representational (or whatever) schemes yield a satisfactory account of *implementation*, but rather whether the kinds of computational models sanctioned by these schemes allow us to adequately account for phenomena of interest.

To help bring this out, observe that these schemes are well-suited to different kinds of explanatory tasks. Suppose we want to explain performance differences between two chips, measured in the number of primitive mathematical or logical operations required to compute some function. Perhaps the chips differ because they implement different instruction set architectures (ISAs). An ISA describes the basic logical and mathematical operations supported by the machine. Accordingly, facts about a system's ISA can explain certain performance differences: perhaps one device uses a reduced instruction set while another uses a complex instruction set, with more primitive instructions for complex operations. Because an ISA does not specify how these instructions are executed in hardware, they are naturally understood as a kind of abstract causal description. Consequently, we can use ISA descriptions to explain this kind of performance difference while ignoring finer-grained details concerning, e.g., their datapath, memory hierarchy, physical constitution, and so forth.

At the same time, our explanatory interests can encourage us to look beyond the abstract causal structure of a system. Some such cases are naturally addressed in terms of representational schemes. For instance, in some cases, physical systems which are indiscernible with respect to their causal structure (at a fixed level of description) implement distinct computational models. This can happen when the systems are embedded in distinct linguistic communities, relative to which the systems compute distinct numerical functions (Rescorla, 2013). In other cases, systems with distinct causal structures implement the same computational model: for certain explanatory purposes, computer scientists treat different kinds of microprocessors as register machines, despite substantial differences at the level of

labelling schemes are part of the theoretical machinery used for describing and reasoning about physical systems.

their internal dynamics (Curtis-Trudel, 2022). To be sure, opponents of representational accounts of implementation can make various maneuvers in responses to these cases (e.g., Piccinini, 2020). My point, however, is that such maneuvers needn't be couched as defences of an alternative *theory* of implementation. Rather, it is enough to treat these as differences about the sorts of labelling schemes that are required meet our explanatory interests and demands.

As a final illustration, consider longstanding disputes about whether cognitive computations should be understood representationally or not. One might attempt to settle the issue by pairing a computational theory of cognition with a non-representational theory of computation. The inference, to put it extremely crudely, runs as follows: cognition is computational, but computation is non-representational, so cognitive computations are non-representational.[18] Observe, however, this is fundamentally a dispute about cognition, not computation. Accordingly, it should be answered in light of the explanatory interests and standards of cognitive science — whether we should explain cognitive processes in representational terms or not. An account of implementation should be flexible enough to accommodate both of these alternatives without prejudging the issue.

Stepping back, while I have focused here only on causal and representational schemes, I submit that similar remarks apply to others in the literature. To reiterate, the important point is that from the point of view of **RSMA**, these proposals need not compete *as accounts of implementation*. Rather, they are better understood as proposals about how to describe systems as implementing computational models, under certain kinds of labelling schemes, for certain theoretical purposes. If such proposals disagree, it is because they disagree about how best to describe or explain some theoretical target, *not* whether they capture some deeper fact about computation as such.

6 Limitations and outlook

I have argued that in certain scientific or engineering contexts, there can be non-arbitrary grounds for describing physical systems computationally. However, this conclusion is in certain respects limited. First, although I have argued that my response avoids the most serious descriptive and explanatory failures incurred by unlimited pancomputationalism, I have not argued that my response is preferable all things considered. One critical avenue for future work will be to assess my response against a fuller range of adequacy criteria for theories of implementation (for more on which, see Piccinini, 2015; Shagrir, 2022).

[18]I doubt anyone nowadays would accept such a crude inference. But (Stich, 1983) perhaps comes close.

Second, I have not argued that computation is non-trivial in every theoretical context. Indeed, on my view, it will typically be an empirical question whether computational notions can be applied non-arbitrarily. But the discovery that computational notions cannot be applied non-arbitrarily to a given system need not undermine our confidence in computation wholesale. Rather, it would be a philosophically important insight, for it would illuminate which kinds of physical phenomena are fruitfully understood in computational terms and which ones are not. Elaboration on this point would benefit from closer investigation of the particular schemes employed in computer and cognitive science. This too remains on the agenda for future work.[19]

These limitations notwithstanding, I have argued that computation faces no *special* triviality worry. The triviality argument was supposed to show that computational descriptions are unlike physical (chemical, biological, etc.) descriptions, in that the former but not the latter applied to physical systems indiscriminately. However, if computational descriptions are on a par with other kinds of mathematized scientific descriptions, then the former are no more trivial than the latter. Indeed, the question of what makes a given computational description relevant turns out to be an instance of the more general question of what makes a given scientific description relevant. This is, of course, a familiar general problem, among the hardest in the philosophy of science. It is no less a problem for being general (or familiar). But it is no objection to computational descriptions per se that we lack a uniform account of what makes them relevant for various scientific purposes.

References

Anderson, N. G., & Piccinini, G. (2024). *The Physical Signature of Computation.* Oxford University Press.

Batterman, R. W. (2010). On the Explanatory Role of Mathematics in Empirical Science. *The British Journal for the Philosophy of Science*, *61*(1), 1–25.

Bueno, O., & Colyvan, M. (2011). An Inferential Conception of the Applicability of Mathematics. *Noûs*, *45*(2), 345–374.

Bueno, O., & French, S. (2018). *Applying Mathematics: Immersion, Inference, Interpretation.* Oxford University Press.

Burks, A. R., & Burks, A. W. (1988). *The first electronic computer: The Atanasoff story.* Ann Arbor: University of Michigan Press.

Button, T., Walsh, S., & Hodges, W. (2018). *Philosophy and model theory.* Oxford: Oxford University Press.

[19]See (Papayannopoulos et al., 2022b) and (Richmond, 2024) for promising steps in this direction.

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

Chrisley, R. L. (1994). Why everything doesn't realize every computation. *Minds and Machines*, *4*(4), 403–420.

Copeland, B. J. (1996). What is Computation? *Synthese*, *108*(3), 335–359.

Curtis-Trudel, A. (2022). The determinacy of computation. *Synthese*, *200*(1), 1–28.

Dewhurst, J. (2018). Computing Mechanisms Without Proper Functions. *Minds and Machines*, *28*(3), 569–588.

Egan, F. (2012). Metaphysics and Computational Cognitive Science: Lets Not Let the Tail Wag the Dog. *Journal of Cognitive Science*, *13*(1), 39–49.

Frege, G. (1884). *The Foundations of Arithmetic.* Evanston, Illinois: Northwestern University Press.

Fresco, N., Copeland, B. J., & Wolf, M. J. (2021). The Indeterminacy of Computation. *Synthese*, *199*, 12753–12775.

Godfrey-Smith, P. (2009). Triviality arguments against functionalism. *Philosophical Studies*, *145*(2), 273 – 295.

Harris, D. M., & Harris, S. L. (2013). *Digital Design and Computer Architecture* (2nd ed.). Waltham, MA: Morgan Kaufmann.

Hitchcock, C. (2013). Contrastive Explanation. In M. Blaauw (Ed.), *Contrastivism in philosophy* (pp. 11–35). New York: Routledge/Taylor & Francis Group.

Matthews, R. J., & Dresner, E. (2017). Measurement and Computational Skepticism. *Noûs*, *51*(4), 832–854.

Melnyk, A. (1996, September). Searle's abstract argument against strong AI. *Synthese*, *108*(3), 391–419.

Milkowski, M. (2013). *Explaining the Computational Mind.* Cambridge, MA: MIT Press.

Millhouse, T. (2019). A Simplicity Criterion for Physical Computation. *The British Journal for the Philosophy of Science*, *70*(1), 153–178.

Papayannopoulos, P., Fresco, N., & Shagrir, O. (2022a). Computational indeterminacy and explanations in cognitive science. *Biology & Philosophy*, *37*(6), 47.

Papayannopoulos, P., Fresco, N., & Shagrir, O. (2022b). On Two Different Kinds of Computational Indeterminacy. *The Monist*, *105*(2), 229–246.

Piccinini, G. (2007). Computational modelling vs. Computational explanation: Is everything a Turing Machine, and does it matter to the philosophy of mind? *Australasian Journal of Philosophy*, *85*(1), 93–115.

Piccinini, G. (2015). *Physical Computation: A Mechanistic Account.* Oxford, UK: Oxford University Press.

Piccinini, G. (2020). *Neurocognitive Mechanisms: Explaining Biological Cognition.* Oxford University Press.

Pincock, C. (2011). *Mathematics and scientific representation.* Oxford ; New York: Oxford University Press.

Putnam, H. (1987). *Representation and Reality.* MIT Press.

Quine, W. V. O. (1960). *Word and object.* Cambridge, Mass: MIT Press.

Rescorla, M. (2013, December). Against Structuralist Theories of Computational Implementation. *The British Journal for the Philosophy of Science*, *64*(4), 681–707.

Rescorla, M. (2014, April). A theory of computational implementation. *Synthese*, *191*(6), 1277–1307.

Richmond, A. (2024). How Computation Explains. *Mind & Language.* doi: https://doi.org/10.1111/mila.12521

Scheutz, M. (1999). When Physical Systems Realize Functions. *Minds and Machines*, *9*, 161–196.

Schweizer, P. (2019). Triviality Arguments Reconsidered. *Minds and Machines*, *29*(2), 287–308.

Searle, J. R. (1992). *The Rediscovery of the Mind.* MIT Press.

Shagrir, O. (2022). *The nature of physical computation.* New York, NY: Oxford University Press.

Shenker, O., & Hemmo, M. (2022). The multiple-computations theorem and the physics of singling out a computation. *The Monist*, *105*(1), 175–193.

Sprevak, M. (2010). Computation, individuation, and the received view on representation. *Studies in History and Philosophy of Science Part A*, *41*(3), 260–270.

Sprevak, M. (2019). Triviality arguments about computational implementation. In M. Sprevak & M. Colombo (Eds.), *Routledge Handbook of the Computational Mind* (pp. 175 – 191). London: Routledge.

Stich, S. P. (1983). *From Folk Psychology to Cognitive Science: The Case Against Belief.* MIT Press.

Weisberg, M. (2013). *Simulation and Similarity: Using Models to Understand the World.* Oxford University Press.

Weste, N. H. E., & Harris, D. M. (2011). *CMOS VLSI design: a circuits and systems perspective* (4th ed.). Boston: Addison Wesley.

Woodward, J. (2003). *Making Things Happen: A Theory of Causal Explanation.* Oxford, UK: Oxford University Press.