philosophies

Article

Conceptualizing Machines in an Eco-Cognitive Perspective

Lorenzo Magnani

Philosophy Section and Computational Philosophy Laboratory, Department of Humanities, University of Pavia, 27100 Pavia, Italy; lmagnani@unipv.it

Abstract: Eco-cognitive computationalism explores computing in context, adhering to some of the key ideas presented by modern cognitive science perspectives on embodied, situated, and distributed cognition. First of all, when physical computation is seen from the perspective of the ecology of cognition it is possible to clearly understand the role Turing assigned to the process of "education" of the machine, paralleling it to the education of human brains, in the invention of the Logical Universal Machine. It is this Turing's emphasis on education that furnishes the justification of the conceptualization of machines as "domesticated ignorant entities", that is proposed in this article. I will show that conceptualizing machines as dynamically active in distributed physical entities of various kinds suitably transformed so that data can be encoded and decoded to obtain appropriate results sheds further light on my eco-cognitive perspective. Furthermore, it is within this intellectual framework that I will usefully analyze the recent attention in computer science devoted to the importance of the simplification of cognitive and motor tasks caused in organic entities thanks to morphological features: ignorant bodies can be computationally domesticated to make an intertwined computation simpler, relying on the "simplexity" of animal embodied cognition, which represents one of the main qualities of organic agents. Finally, eco-cognitive computationalism allows us to clearly acknowledge that the concept of computation evolves over time as a result of historical and contextual factors, and we can construct an epistemological view that depicts the "emergence" of new types of computations that exploit new substrates. This new viewpoint demonstrates how the computational domestication of ignorant entities might result in the emergence of novel unconventional cognitive embodiments.

Keywords: eco-cognitive conputationalism; physical computation; domestication of ignorant entities; ignorant entities; unconventional substrates; natural computing; Turing

check for updates

Citation: Magnani, L. Conceptualizing Machines in an Eco-Cognitive Perspective. *Philosophies* **2022**, *7*, 94. https://doi.org/10.3390/philosophies7050094

Academic Editors: Alessio Plebe and Pietro Perconti

Received: 29 May 2022
Accepted: 17 August 2022
Published: 25 August 2022

1. Introduction

I have proposed in a previous book [1] what I called "eco-cognitive computationalism", which explores computing in context, adheres to some of the key ideas presented by modern cognitive science perspectives on embodied, situated, and distributed cognition. I think this intellectual framework is especially useful to depict various conceptualizations of machines that I consider basic and important because they offer (1) a naturalization of the concept of computation and constitute (2) an antidote against dogmatic definitions or philosophical assumptions tending toward metaphysical outcomes or excessive ontological claims.

A first issue illustrated in this article regards the analysis of Alan Turing's *conceptualization of machines as unorganized brains that can be computationally organized mimicking human education*. Turing emphasizes the role of "education" of the machine, paralleling it to the education of human brains, as the way for "educating" a machine to "think". It is exactly this Turing's emphasis on education that furnishes the justification of the conceptualization of machines as "domesticated ignorant entities", that is proposed in this article.

To depict the conceptualization of machines as "domesticated ignorant entities" it is appropriate to adopt the intermediate perspective that sees *machines as physical systems*. This view shows that conceptualizing machines as dynamically active in distributed physical entities of various kinds suitably transformed so that data can be encoded and decoded to obtain appropriate results, further sheds light on a really naturalized eco-cognitive perspective. To clearly take advantage of this conceptualization of machines as physical systems it is also important to illustrate a comparison between physical computing and physics, showing that *the first is the reverse of the second.*

I will also illustrate that, at least from the classical perspective on computational machines as digital machines, *the physical entities exploited for computation are technological artefacts.* Moreover, it is within this intellectual framework that it is important to analyze the recent attention in computer science devoted to the importance of the simplification of cognitive and motor tasks caused in organic entities thanks to morphological features, as described in the original case of morphological computation. This kind of computation describes how ignorant bodies endowed with morphological features typical of living beings can be computationally domesticated to render an intertwined computation simpler, resorting to that "simplexity" of animal embodied cognition, which represents one of the main qualities of organic agents.

It is on the basis of the previous eco-cognitive considerations that the article will finally arrive to emphasize the *conceptualization of machines as the fruit of domestication of ignorant computational substrates.* In sum, through eco-cognitive computationalism, we can definitely acknowledge that the concept of computation evolves over time as a result of historical and contextual factors, and we can construct an epistemological view that depicts the "emergence" of new types of computations that exploit new substrates. This new viewpoint demonstrates how the computational domestication of ignorant entities might result in the emergence of novel unconventional cognitive embodiments (morphological computing, DNA computing, quantum computing, etc.), as illustrated in the two final sections of the article devoted to the *conceptualization of machines in terms of natural computation* and to the capacity of *biological systems to compute information units in unconventional ways.* These last sections will also provide a sketch of the recent area of research called "natural computing".

2. Alan Turing: Conceptualizing Machines as Unorganized Brains That Can Be Computationally Organized Mimicking Human Education

Eco-cognitive computationalism considers computation as something that is active in physical entities that have been adequately changed into what I called "cognitive mediators", in which data maybe encoded and decoded to generate beneficial outcomes. We must state right away that, even if cognition is viewed as computational, eco-cognitive computationalism does not equate computation to digital computation, that is the processing of digits following appropriate rules. We will see that other substrates can carry computations: it is reductive to conceptualize computation in digital terms.

I think that to clarify the concept of computation it is necessary to provide a new analysis of the related concepts of information and cognition. No static and unchangeable definitions of the concepts of information, cognition, and computation will be provided because eco-cognitive computationalism takes into account the importance of historical and variable aspects, which lead to constant amendments of the status of their meanings.

As the title "Alan Turing: conceptualizing machines as unorganized brains that can be computationally organized mimicking human education" of the present section illustrates, the foundational conceptualization that leads to the birth of the modern computer science with the final establishment of both the Universal Logical Computing Machine and the Universal Practical Computing Machine (this last concept obviously refers, for example, to the digital laptop that we use every day) is due to some Turing's speculations. Turing himself describes a kind of proto-eco-cognitive computationalism that sees the evolutionary emergence of information, meaning, and the primitive forms of cognition as the outcome of a complicated eco-cognitive intertwining and simultaneous *coevolution*, temporally oriented, of the states of brain, body, an external surroundings. Indeed Turing explicitly states—adopting a philosophical approach—that the brain of a child is an "unorganized machinery”, which may be organized by adequate “interference training”. The organization might lead to the machine being modified into a universal machine or something similar.

Turing adds that “From the standpoint of evolution and genetics, this depiction of the cortex as an unstructured apparatus is highly satisfying” [2] (p. 16), a statement that actually is endowed with a speculative value: indeed I should add that Turing’s depiction of an infant’s brain as an unorganized machine is mainly theoretical, and it primarily serves as a useful heuristic in his creative cognitive processes. Indeed, we know from fundamental neuroscientific studies that the brain is instead a highly ordered system. Turing further observes that babies’ cortices are “socially” constructed through language. The proposed random building of the baby brain, in this view, does not address neurobiological issues but rather the idea of an unorganized brain as fundamentally characterized by a lack of training, i.e., a lack of indications received through the senses from the external (including social) environment, such as language. As a result, the term “unorganized machinery” has a distinct meaning: indeed Turing says “It is conceivable that the same machine might be regarded by one man as organized and by another as unorganized” [2] (p. 9) (further details are illustrated in Magnani [1] (chapter one)).

Turing provides the conceptual framework necessary to demonstrate how, by replicating the aforementioned procedure, the Universal Logical Computing Machine and the Universal Practical Computing Machine may be created. Computationalizing a machine is a matter of *education,* exactly as in the case of the human brain. In this respect, the dual innovation is realized as the externalization (a kind of domestication through education) of computational faculties in those artefactual physical entities that compute for some human or artefactual agents: those computers that I dubbed “mimetic minds” [3], taking advantage of the viewpoint supplied by Turing[1].

It is spontaneous for Turing to use the notion of education as it is occurring in the case of the growth of babies’ brains to develop the new concept of computing: physical machines may also be *educated* to generate specific types of behavior. In this perspective, education and programming coincide. Yes, programming is similar to human education in that it allows us to computationally domesticate machines in order to get suitable results. The new notion of the (Universal) Logical Computing Machine (LCM) belongs to this intellectual framework together with the companion one of the (Universal) Practical Computing Machines (PCM), which is the implementation of the first in a digital computer.

3. Conceptualizing Computational Machines as Physical Systems

In the first chapter of my book [1] I have illustrated some aspects of the relationships between the concepts of information, cognition, and computation, attempting to answer the question “Is cognition computation?” We may say yes, it is, but it is not merely computing, thus cognition and computation are not interchangeable. We can affirm that both information processing and computation are intertwined with cognition, and a lot of studies have been realized to clarify the various roles and types of computation and information processing implicated in cognition, even if such research is doomed to become obsolete, at least from an eco-cognitive point of view. I prefer to assert, from a naturalistic epistemological perspective, that, given the fact that the concept of computation changes when new research results are ready, and so it is sensible to semantic variations, the other two concepts, which are already illustrated in themselves by precarious definitions, that are linked to changes of knowledge, technologies, and cultural ideas, also change.

A recent perspective I especially favor, that I have illustrated in detail in the part “Information processing, cognition, and computation” of chapter one of my book [1], try to elucidate the notion of digital computation by clearly demonstrating and emphasizing that digital computation is implemented in *physical systems*[2].

My eco-cognitive computationalism does not try to provide a firm and unchangeable description of the concepts of information, cognition, and computing. Instead, I will propose an epistemological view that depicts how we can grasp modifications and expansions of their meanings through the example (and the illustration of the “emergence”) of what can be called the *new* "computational domestication" of physical entities, thanks for example to the new exploitation for computation of new substrates, such as in the case of morphological computing. To that end, I shall take the viewpoint on physical computation presented in [4] (p. 14), which is especially suited to my concerns[3].

1. A computer is a physical system having real constituent elements and internal interactions that allow it to transition from one physical state to another one.
2. As a result, I agree with Horsman et al., who claim that physical computing is "the use of a physical system to predict the outcome of an abstract evolution" [4] (p. 14). I have to add that it is worth noting that in computations, we are not dealing with a physical system that has to be explained (as in physics), but rather with an abstract object that we wish to evolve using the physical system itself. When we recognize that a physical entity is being exploited in such a way, physical evolution executes a computation that is then interpreted as such by a human or another artificial agent, (together with its cognitive content).
3. Standardly a computer is a technology created by applying scientific concepts to precision-engineer physical systems to get certain expected results (see Section 3.2 below for further information).
4. When a special link between abstract mathematical/logical entities and the physical ones is realized, we can conclude that a physical system (for example a digital computer) is computing.
5. Contrarily to the previous observations concerning computation as physical, and beyond their use in conventional (digital) and eccentric cases (see the following sections of this article), the concept of *computation,* and its related system property, *information,* have been imported into other scientific areas to the aim of making intelligible and "explain" various natural processes such as photosynthesis and the consciousness, and a considerable quantity of contemporary cross-disciplines has proposed the conclusion that "everything is information", "the universe is a quantum computer", or "everything computes" [4] (p. 2).

   Obviously, many researchers— and I agree with them—plausibly contend that defining the universe and everything in it as a computer, the notion of *physical computation* becomes *empty.* Cf. also the section "Pancomputationalism 'naturalized" of chapter two of Magnani [1][4].

3.1. The Illuminating Comparison between Physical Computing and Physics

Following the perspective suggested by the already cited Horsman et al. and as quickly indicated in the previous subsection, physical computing may be advantageously seen as the opposite of mathematical physics: *a physical system is used to predict the result of an abstract dynamics,* whereas in physics, *an abstract model is used to predict physical dynamics.* Indeed, physics is engaged in abstractly modeling physical systems thanks to the exploitation of abstract mathematical theories to predict the outcome of physical evolution.

Again: if physical computing is the opposite of mathematical science, and is related to the exploitation of *a physical system to predict the outcome of an abstract dynamics,* then a physical system that lacks this predictive aspect is not a computer; similarly, a cluster of mathematical equations is a bad physical model if it lacks predictive power. As a result, in physical computation, we wish to take an abstract thing, a computation and represent it into a physical entity. We face a *representational* process, which allows for a "comparison between physical processes and mathematical described computations" [4] (p. 2). On the contrary, in physics, just to make an example, an electron is represented as a wave function, which serves as a map between physical and abstract spaces: in a computation, "the initial impetus is not a physical system that needs to be described, but rather an abstract object that we wish to evolve. An abstract problem is the reason why a physical computer is used" (cit., p. 10). As a result, we begin with the problem of a reversed representation right away.

3.2. Computational Physical Entities Are Technological Artefacts

The creation of what Turing referred to as Universal "Practical" Computing Machines is absolutely linked to the role of suitable technologies capable of "domesticating"[5] physical artefacts to act in expected ways: after all, in order to possess appropriate physical computations, we need technical help, to build physical artefactual entities capable of completing the work.

It is surely an astonishing result the fact that Turing's abstract machine, as a Discrete-State Machine (DSM) [8], has been implemented in a physical machine, thanks to the building of artificial technological processes that can evolve through discrete states, taking advantage of the well-known silicon electronic devices. We also know that, at least starting from Turing, computer science explained that physical artefactual digital entities are created to process information and knowledge reorganizing them in little boxes, bits, and pixels, that is in discrete packages, so avoiding fuzziness and contingency and chaotic deterministic aspects, in which a fluctuation/variation below the measuring interval produces multiple and different evolutions for a system.

Turing emphasizes in his concept of Discrete State Machine (DSM), already in 1950, that this effect is potentially avoidable: the Turing machine is an abstract (logical, as he put it) computer with a Laplacian behavior; as a DSM, it manages values that are invariably discrete. The notion of a program, as well as its mathematical properties connected to its implementation, are deterministic in the Laplace sense: the application of rules (or, in the case of Laplacian physics, equations) implies predictability. The aim of a scientific theory is to reach success when confirmed practically exempt from huge anomalies, explanatorily satisfactory, and standardly capable to predict the behavior of a lot of processes: a situation that represents the best example of the prototype of the success of a progressive scientific research program, as illustrated by Lakatos [9]. These kinds of theories are used in engineering and so are fruitful epistemic tools capable to engineer physical entities, that are in such a way "domesticated" to obtain a desired reliable behavior.

4. Conceptualizing Machines as the Fruit of Domestication of Ignorant Computational Substrates

It is well-known that today's conventional computing substrates are highly engineered silicon devices whose processes are based on the stable knowledge of those scientific theories that also contributed to originating them. A current trend in computer science and AI is related to the aim of exploiting more and more physical substrates capable to support computational performances and considerably reliable in granting good outcomes of the related processing of data.

We have said in the first section of this article that physical computing is the opposite of mathematical science, the most important task is to "educate" a physical entity, beyond digital computation, to *predict* the outcome of an abstract dynamics and not the contrary (that is the case of an abstract tool capable to predict a physical behavior, an equation for example). As a result, understanding the physical system's predictive capabilities is critical. There are cases that do not fit this requirement: a stone or a piece of marble will not be suitable to perform in time modifications of themselves capable to match a reliable and repeatable computational passage from initial states represented in them to final states that should be picked up later on, but there are other unsuspected entities that are not so "unusable" to computational aims. . .

The Original Case of Morphological Computation

Recent research in cognitive science paid a lot of attention to the simplification of cognitive and motor processes that can be performed by living entities thanks to their morphological features, a simplification of course exploited in robotics. Can morphological features be used to compute or at least accompany classical digital computation to reinforce and simplify it? So following in this case the cognitive importance of the so-called *simplexity*, that characterize embodied capacities of various organic agents, expressing a beneficial mixture between complexity and simplicity [10,11].

The literature has indicated three primary functions that morphology plays in cognition[6]

1. morphology that aids control in the absence of a control system, motors, or sensors;
2. morphology that makes perceiving easier;
3. morphological computation in the precise sense, such as "reservoir" computing, where embodiment and computation are inextricably linked and integrated.

The passive dynamic walker is an example of the first case (in which the behavior is purely mechanical); in the second case, the Gecko feet are a classical example, which can be thought of as active expansions of the passive walker, employing actuators and sensors (the Gecko's special ability is the fruit of its morphology in its intertwining with specific surroundings—that is an ability that is not related to a higher-level central control). Another good example is the case of the fly's eye, which involves not only movement but also cognitive capacities related to visual perception. The third case is the so-called Reservoir Computing, which was first advanced by the neural network researchers and later extended to the role of physical apparatuses that can work as the so-called reservoirs (Physical RC [Reservoir Computing]), in which the body is de facto domesticated to the aims of computation. Reservoir computing is not restricted to morphological calculations; as I just said, it is derived from neural network research and takes into account the aggregation of neurons, as Müller and Hoffmann summarize,

> [. . .] with nonlinear activation functions and with recurrent connections that have a random but bounded strength; this is referred to as a dynamic reservoir. These neurons are randomly connected to input streams, and the dynamics of the input is then spread around and transformed in the reservoir, where it resonates (or "echoes"—hence the term "echo-state networks") for some time. It turns out that tapping into the reservoir with simple output connections is often sufficient to obtain complex mappings of input stream to output stream that can approximate the input–output behavior of highly complex nonlinear dynamical systems. During training, the weights from the input streams and between the reservoir neurons are left intact; only the output weights—from the reservoir to the output layer-are modified by a learning algorithm (e.g., linear regression). The complexity of the training task has been greatly reduced (as opposed to training all the connections) by exploiting the reservoir to perform a spatiotemporal transformation of the input stream (the temporal aspect of the input sequence has basically been unfolded by the reservoir and can be retrieved directly at any instant). Furthermore, if feedback loops from the output back to the reservoir are introduced and subject to training, the network can be trained to generate desired output streams autonomously [12] (p. 6).

The exceptional innovativeness is that these organic entities can exactly also act as a reservoir, giving rise to the already quoted Physical RC, which can also consist of the body of an actor: organic bodies interfaced with their surroundings can indeed display the required features— high dimensionality, nonlinearity, and fading memory, as clearly explained in [12] (p. 3) and [13,14].

We need to discuss certain cognitive and epistemological features of the so-called RC to delve deeper into the ways of domesticating non-standard items for computing beyond the typical common use of silicon devices. In this situation, a traditional "digital" computational controller instrument (such as a robot, to make and example) is strengthened by the fact that some computational "merits" are relocated to the morphology of physical entities with the goal of obtaining the benefits of an "analog" computation. Where should the distinction between morphological and pure digital computation be drawn? Hauser et al. [14] (p. 227) provide the following description of the RC mechanisms: several physical entities, while not specifically designed for computing, might possibly function as reservoirs. Hauser, H. et al. [13–15] show instances of real implemented reservoirs, such as the amazing case of a soft silicone-based octopus limb, which was used to imitate desirable nonlinear dynamical systems and to perform reliable calculations and create a feedback controller.

> At the core of RC lays the so-called *reservoir*, a randomly initialized high-dimensional, nonlinear dynamical system, which maps the typically low-dimensional input (stream) onto its high-dimensional state space in a nonlinear fashion. In that sense, the reservoir takes the role of a kernel (in the machine learning sense, i.e., the nonlinear projection of a low dimensional input into a high-dimensional space [. . .]). In addition, the reservoir, being a dynamical system, has the inherent property to integrate input information over time, which is obviously beneficial for any computation that needs information on the history of its input values. It is important to note that the reservoir is not altered during the learning process. Although it is randomly initialized, its dynamic parameters are fixed afterward. In order to learn to emulate a desired input output behavior (to be more precise, the desired mapping from input streams to output streams), one has to add a linear output layer, which simply calculates a linearly weighted sum of the signals of the high-dimensional state space of the reservoir. These output weights are the only parameters that are adapted during the learning process [14] (p. 227).

Real physical bodies are utilized as reservoirs and computational devices in the implementation of the so-called Physical RC. Of course neither the entities engaged in traditional highly developed silicon devices nor the entities participating in Physical RC body exploitation are aware that they are part of a computational process. They just follow the principles of physics and react in a natural way. Because the body is a stable physical entity, we must notice that this also means that the body does not over— or under—compensate, in the case of Physical RC. To finish the computational process, the suggested arrangement merely adds some linear readouts to the body. If there were no readouts at all, the body would behave in the same way. If this output/result is utilized, for example, as a control signal belonging to a robot in a feedback loop, the robot's performance should be another one if we end the loop by adding the readout. It is worth noting that a physical computer for morphological computing does not have to be deliberately and intelligently designed: it can grow organically (and so computationally). This means that living creatures or portions of organisms (as well as their artefact replicas) might possibly do information processing and be utilized to perform our computation.

As a result, the body is "ignorant", in that it acts the same whether readouts are added or not (exactly as in the case of a digital computer), and may be utilized in several computations characterized by the same input at the same time adopting the needed readout. A note has to be added concerning my use of the term ignorance in this case. Identifying the body as ignorant does not imply that we may locate a more knowledgeable component of it. Rather, given that no part of the system is aware that it is a part of the system, I believe it is important to highlight the body's "ignorant" aspect as a fundamental part of the system, not in relation to other possible "knowledgeable" parts, but to highlight a specific condition of ignorance that may be linked to it, This is not simply a lack of information's permanence (or contamination), but rather a kind of apathy toward the computation that occurs via it.

Furthermore, traditional physical computational tools (hardware) such as silicon devices or robots characterized by an abstract controller are founded on the discreteness of their *digital* features and, when seen from the standpoint of their general aspects, are steady, unperturbable, and resistant in the face of external effects distinct from the processes of encoding and decoding that are situated at the beginning and at the end of the computational process. Unconventional computational devices, such as in morphological computation, in the case of Physical RC, are characterized by a computation that happens in the continuous reality as a patent sort of a very rapid *analog* computation, in which the exploitation of real elements obviously presents explicit restrictions and constraints.

We should also observe that a kind of *morphosis* (for example, repositioning the body in a new position) [14] (p. 235) is required to compensate the physical adopted reservoir certain lack of cognitive/computational plasticity (this means that, obviously, it presents, so speak, less "universality" than the classical Turing machines): the morphology might be changed online to get many computational advantages, also taking use of the environment potential influence on the entire system[7].

To conclude, in the context of morphological computation, bodies that are "ignorant", as I demonstrated above, become *mimetic bodies,* that is, bodies that may imitate different cognitive mediating activities. I use the term "mimetic" in the same way that Turing uses it, as I have illustrated in the first section of this article: "mimicking education, we should hope to modify the machine until it could be relied on to produce definite reactions to certain commands" [2] (p. 14) Turing adds. In an analogous way, we may acquire computational results mimicking morphological features: a classic example I already cited is provided [15,17], who "purely" mimicked part of an octopus morphology using an artificial model of a soft robotic arm motivated by it. Considering recent morphological computation breakthroughs, we can safely infer that we are confronted with new epistemological and technological dimensions of what has been named *distributed computation* [1]: to achieve novel cognitive outputs and outcomes, computing is more and more being dispersed over a wide range of entities, props, instruments, bodies, and gadgets. The potential of morphological computation principles in robot design might lead to a new generation of robots with more adaptability and fewer control parameters. The possible suggestions derived from morphological computation research in robot design might result in a new variety of robots with improved flexibility and a smaller set of control parameters.

5. Conceptualizing Machines in Terms of Natural Computation: Nature-Inspired Computing and Computing that Occurs in Nature

The use of novel substrates for computation is not limited to morphological computing: a recent guidebook [18] on the so-called *Natural Computing* depicts both

1. human-designed computing that is inspired by nature and
2. natural computing (and information processing) occurring in nature.

Computing that is inspired by nature refers to neural computation that takes advantage of analogies with the mechanisms of the brain; evolutionary computation that takes advantage of analogies with Darwinian and post-Darwinian scientific frameworks; cellular automata that take advantage of analogies with intercellular communication; swarm intelligence that takes advantage of analogies with the behavior of groups of organisms; artificial immune systems that take advantage of analogies with the natural immune system; artificial life systems that take advantage of analogies with the properties of natural life in general; membrane computing that takes advantage of analogies with the compartmentalized ways in which cells process information; and amorphous computing that takes advantage of analogies with morphogenesis.

The handbook cites molecular computing and quantum computing among the studies aimed at replacing standard silicon devices with novel substrates (which I was showing in the previous section of this article using the example of morphological computing). Information is encoded as biomolecules in molecular computing, and subsequently, molecular biology entities are utilized to process the data, so generating computations. Quantum mechanics is employed in quantum computing to perform computations and create communications more effectively than in the case of standard hardware. It is important to emphasize that morphological computing is an essential facet of this field of study since it concerns entities that become part of computation due to their structural properties.

The second kind of study, which refers to natural computing (and information processing) occurring in nature [so adopting an ontological commitment, so to speak], concerns research on the computational character of self-assembly, which is typical of nanoscience, the computational character of developmental processes, biochemical reactions, bacterial communication, and brain functioning, and the systems biology approach to bionetworks, where cellular processes are analyzed using communication and interaction, and thus computation.

Finally, we can see that certain detractors of natural computing think that this approach generates a loss of generality (universality) of the notion of computation. We can say that their worry is exaggerated: we can calmly conclude that we are facing a variety of computational entities, some of which are general and others that are specialized for certain computations. For example, quantum computers will be highly specialized computers, vastly superior to our current computers in certain applications but lacking in others. After all, computers are similar to other tools and instruments in this regard. We would never consider building a device that could function as both a microscope and a telescope. Similarly, we will have computers that will be excellent quantum system simulators but useless for playing computer games.

Biological Systems Compute Information Units in Unconventional Ways

In the perspective of my eco-cognitive paradigm, an intriguing instance of exploitation of biological organisms to conduct computation must be highlighted. Biological beings may compute unique forms of informational units, units that can be found in unexpected and unusual computing systems, that are not comparable to the ones I discussed in the preceding section on morphological computation. I am referring to the contribution of the so-called DNA computers, which are a relatively well-known type of non silicon-oriented computation. Molecular computing is of course an area of study situated at the intersection of several approaches that developed techniques and methods to "domesticate" molecules in order to obtain the desired computation, or manufacture the desired product, or govern the functioning of a given molecular system. Rozenberg [18] (pp. viii–ix) usefully observes that the core notion underlying molecular computing is that data may be encoded as (bio)molecules, such as DNA strands, and then transformed using molecular science methods. So to speak, a molecular program is a collection of molecules that will play a particular role, that is, the one of running the program represented by this collection when placed in a suitable substrate. It is a tradition of studies that also promoted and enhanced the possibility of making intelligible some central aspects of nanosciences, such as for example self-assembly.

To summarize, DNA computing is based on the notion that molecular biological processes may be used to execute arithmetic and logical operations on data recorded as DNA strands [19]. DNA bio-operations can be employed for computations, resulting in a biocomputation consisting of a series of bio-operations: it is obvious to highlight the fact that DNA-encoded information differs significantly from silicon-based electronically encoded information, and bio-operations clearly vary from electronic computer procedures[8]. Sensors, processors, and actuators, all formed of molecular devices, make up a molecular robot, which reacts autonomously to its surroundings by checking it, making choices with its computers, and performing actions on it. The intelligent controllers of such molecular robots should thus be molecular computers. To create our "DNA computers", as Kary [21] calls them, the most crucial comparison that had to be established to unveil the secret of DNA computing, was the one between two procedures, one biological and the other mathematical: (a) the result of applying simple operations (copying, splicing, etc.) to initial information encoded in a DNA sequence produces the complicated structure of an organic being, and (b) the result $f(w)$ of applying a computable function to an argument w can be discovered using a mixture of fundamental simple functions to w.

The prosperity of research on computational domestication of new substrates, as demonstrated by the morphological computation, DNA, and quantum computing, among other examples, also adumbrates some philosophico/epistemological concerns intended to show some skepticism regarding the consistency of ultimate viewpoints such as pancognitivism, paniformationalism, and pancomputationalism, which are relatively popular in the current debate about computational philosophy. After all, it seems plausible to state that nature does not compute in itself, and information is not an intrinsic characteristic of it. Also, cognition cannot be seen as a constitutive attribute, neither of organic nor of inorganic entities. I think, following a naturalistic approach, it is only the huge "semiotic" human activities that create a world of computational domestications of ignorant entities, a world of meanings carried by cognitive representations in which the life of informational aspects is rich and fundamental[9].

At this point the readers would like to hear some views on the following question: why is it only the huge "semiotic" human activities, I have just quoted, that create a world of computational domestications of ignorant entities? I think the answer could be achieved in future work taking advantage of the suggestions that arrive from some cognitive paleoanthropological results on the birth of material culture and evolutionary research I already sketched in [24]. It will be necessary to further deepen three sets of considerations mainly concerning human cognition: (1) how semiotic brains built (and build) the so-called *cognitive niches*[10]; (2) the role of *abduction* (that is of reasoning to hypotheses, more or less creative) in building a semiotic artificial world; (3) the biosemiotics of the so-called *disembodiment of the mind.* Human semiotic brains are engaged in a continuous process of disembodiment of the mind as a delegation and distribution of cognitive functions to the environment to lessen cognitive limitations, also and especially taking advantage of what I have called "manipulative abduction"[11]. These design semiotic activities are closely related to the process of "cognitive niche construction", which should be regarded as the second major participant, after natural selection, in evolution. In sum, an important effect of this semiotic brain activity is a continuous process of disembodiment of the mind that exhibits a new cognitive perspective on the mechanisms underlying the semiotic emergence of meaning processes: the computational domestication of ignorant entities could be seen as the final and more complicated effect of this activity of disembodiment. In sum, we can guess that the birth of material culture, which created a huge process of disembodiment of thoughts that otherwise would have soon disappeared, without being transmitted to other human beings, realized a systematic process of semiotic delegation to the external environment: it is in this perspective that we can certainly see it as the first condition of possibility of the recent enormous process of computational domestication of ignorant entities.

6. Conclusions

In this article, taking advantage of what I called eco-cognitive computationalism, I have illustrated various ways of conceptualizing machines, starting from the fundamental role Turing assigned to the process of "education" of a machine, paralleling it to the education of human brains, that wonderfully favored the invention of the Logical Universal Machine. It is this Turing's emphasis on education that furnishes the justification of the conceptualization of machines as "domesticated ignorant entities", that I advanced in this article. I have described that conceptualizing machines as dynamically active in distributed physical entities of various kinds suitably transformed so that data can be encoded and decoded to obtain appropriate results sheds further light on my eco-cognitive perspective. I also analyzed the "morphological computation", in which the importance of the simplification of cognitive and motor tasks caused in organic entities by the morphological features is computationally exploited: I have said that ignorant bodies can be computationally domesticated to render an intertwined computation simpler, resorting to that "simplexity" of animal embodied cognition, which represents one of the main quality of organic agents. Finally, I stressed that we cannot reach a stable and reliable conceptualization of machines because the concept of computation changes, depending on historical and contextual causes, and I have illustrated the "emergence" of new kinds of computations that take advantage of the exploitation of new substrates, also sketching some aspects of the so-called "natural computing".

Funding: Research for this article was supported by the PRIN 2017 Research 20173YP4N3—MIUR, Ministry of University and Research, Rome, Italy.

Acknowledgments: Some themes of this article were already discussed in my book *Eco-Cognitive Computationalism. Cognitive Domestication of Ignorant Entities*, Springer, Cham, Switzerland, 2022. For the informative critiques and interesting exchanges that assisted me to enrich my analysis of the issues treated in this article, I am indebted and grateful to Gordana Dodig-Crnkovic, Joseph Brenner, Marcin J. Schroeder, Giuseppe Longo, Vincent Müller, Jordi Vallverdú, John Woods, Atocha Aliseda, Woosuk Park, Luís Moniz Pereira, Ping Li, to the really helpful and constructive observations of the two reviewers who have allowed to expand and enrich the content of the article, and to my collaborators Selene Arfini and Alger Sans Pinillos.

Conflicts of Interest: The author declares no conflict of interest.

Notes

[1] In chapter one of Magnani [1] the idea of mimetic mind is discussed in detail.

[2] These investigations provide the advantage of avoiding the so-called pancomputationalism, that is the philosophical view that attributes computational capacities to all natural entities, that is, the belief that every physical system is a digital computing system that can be defined in computational terms, which I consider metaphysically biased; cf. especially the part "Pancomputationalism naturalized" of chapter two of [1].

[3] The reader interested in deepening Hosman et al.'s approach to computation can refer to my recent article [5].

[4] The Actor model [6] is an example of an alternative unorthodox view: computation is conceptualized as scattered in space, where computational devices communicate asynchronously and the entire computation is not in any well-defined state. Reacting to a message an Actor may conduct local choices, produce other Actors, send new messages, and determine how to reply to the next message received. Turing's model is a subset of the Actor Model. According to this viewpoint, the Internet conducts unconventional computations and appears to be governed by super-recursive algorithms [7].

[5] The so-called "domestication of physical entities" will be described in the following Section 4.

[6] See, for example, the rich illustration furnished by Müller and Hoffmann [12].

[7] The reader interested in deepening this new kind of research related to reservoir computing can refer to my recent article [16].

[8] Further details concerning this difference can be usefully found in Kari and Rozenberg [20]. On further taking computational advantage of several artificial molecular "machines", encompassing some composed of DNA or RNA to construct robotic systems cf. [19] (p. 4).

[9] More details on these last aspects that regard computational philosophy can be found in [22,23], and chapter four of [1].

[10] Recent scientific results in the area of evolutionary theories provided a new fundamental concept: humans manufacture the so-called *cognitive niches* (cf. [25–28]): "Niche construction should be regarded, after natural selection, as a second major participant in evolution. [...] Niche construction is a potent evolutionary agent because it introduces feedback into the evolutionary dynamics" [29] (p. 2). Representational delegations to the external environment that are configured as parts of cognitive niches are those cognitive human actions that transform the natural environment into a cognitive one. Humans have built huge cognitive niches, characterized by informational, cognitive, and, finally, computational processes, as described by the studies in the field of biosciences of evolution by Odling-Smee, Laland, Sterelny, and Feldman [29–31].

[11] In the perspective of an eco-cognitive and semiotic approach to cognition, we can see manipulative abduction as a kind of reasoning hybridized with externalities and also "multimodal" (cf. [24] (chapter four), that is related to a variety of representations, linguistic, symbolic, iconic, etc. In the case of manipulative abduction the assessment/acceptation of a hypothesis is realized taking advantage of the progressive obtainment of sequential external information with respect to future interrogation and control, and not with the help of an ultimate actual experimental test, in the classical sense of empirical science.

References

1. Magnani, L. *Eco-Cognitive Computationalism. Cognitive Domestication of Ignorant Entities*; Springer: Cham, Switzerland, 2022.
2. Turing, A.M. Intelligent Machinery [1948]. In *Machine Intelligence*; Meltzer, B., Michie, D., Eds.; Edinburgh University Press: Edinburgh, UK, 1969; Volume 5, pp. 3–23.
3. Magnani, L. Mimetic minds. Meaning formation through epistemic mediators and external representations. In *Artificial Cognition Systems*; Loula, A., Gudwin, R., Queiroz, J., Eds.; Idea Group Publishers: Hershey, PA, USA, 2006; pp. 327–357.
4. Horsman, C.; Stepney, S.; Wagner, R.C.; Kendon, V. When does a physical system compute? *Proc. R. Soc. A* **2014**, *470*, 1–25. https://doi.org.10.1098/rspa.2014.0182. [CrossRef] [PubMed]
5. Magnani, L. Computational domestication of ignorant entities. Unconventional cognitive embodiments. *Synthese* **2021**, *198*, 7503–7532. Special Issue on "Knowing the Unknown" (guest editors L. Magnani and S. Arfini). [CrossRef]
6. Hewitt, C. What is computation? Actor model versus Turing's Model. In *A Computable Universe. Understanding and Exploring Nature as Computation*; Zenil, H., Ed.; World Scientific: Singapore, 2012; pp. 159–185.

7. Dodig-Crnkovic, G. Significance of models of computation, from Turing model to natural computation. *Minds Mach.* **2011**, *21*, 301–322. [CrossRef]
8. Longo, G. Critique of computational reason in the natural sciences. In *Fundamental Concepts in Computer Science*; Gelenbe, E., Kahane, J.P., Eds.; Imperial College Press/World Scientific: London, UK, 2009.
9. Lakatos, I. Falsification and the methodology of scientific research programs. In *Criticism and the Growth of Knowledge*; Lakatos, I., Musgrave, A., Eds.; The MIT Press: Cambridge, MA, USA, 1970; pp. 365–395.
10. Kluger, J. *Simplexity. Why Simple Things Become Complex (and How Complex Things Can Be Made Simple)*; Hyperion Books: New York, NY, USA, 2008.
11. *Complexité-Simplexité*; Berthoz, A.; Petit, J.L., Eds.; Collège de France: Paris, France, 2014.
12. Müller, V.C.; Hoffmann, M. What is morphological computation? On how the body contributes to cognition and control. *Artif. Life* **2017**, *23*, 1–24. [CrossRef] [PubMed]
13. Hauser, H.; Ijspeert, A.J.; Füchslin, R.M.; Pfeifer, R.; Maass, W. Towards a theoretical foundation for morphological computation with compliant bodies. *Biol. Cybern.* **2011**, *105*, 355–370. [CrossRef] [PubMed]
14. Hauser, H.; Füchslin, R.M.; Nakajima, K. Morphological computation—The physical body as a computational resource. In *Morphological Computation: The Body as a Computational Resource*; Hauser, H., Füchslin, R.M., Pfeifer, R., Eds.; Self-published: Bristol, UK, 2014; pp. 226–244.
15. Nakajima, K.; Hauser, H.; Li, T.; Pfeifer, R. Information processing via physical soft body. *Sci. Rep.* **2015**, *5*, 10487. [CrossRef] [PubMed]
16. Magnani, L. *Discoverability. The Urgent Need of an Ecology of Human Creativity*; Springer: Cham, Switzerland, 2022.
17. Nakajima, K.; Hauser, H.; Kang, R.; Guglielmino, E.; Caldwell, D.; Pfeifer, R. A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm. *Front. Comput. Neurosci.* **2013**, *7*, 91. [CrossRef] [PubMed]
18. *Handbook of Natural Computing*; Rozenberg, G.; Bäck, T.; Kok, J., Eds.; Springer: Cham, Switzerland, 2012.
19. Hagiya, M.; Aubert-Kato, N.; Wang, S.; Kobayashi, S. Molecular computers for molecular robots as hybrid systems. *Theor. Comput. Sci.* **2016**, *632*, 4–20. [CrossRef]
20. Kari, L.; Seki, S.; Sosík, P. DNA computing: Foundations and implications. In *Handbook of Natural Computing*; Rozenberg, G., Bäck, T., Kok, J., Eds.; Springer: Cham, Switzerland, 2012; Volume 3, pp. 1073–1128;
21. Kari, L. *DNA Computing Based on Insertions and Deletions*; University of Western Ontario: London, ON, Canada, 2013.
22. Magnani, L. Computationalism in a dynamic and distributed eco-cognitive perspective. In *Philosophy and Methodology of Information*; Dodig-Crnkovic, G., Burgin, M., Eds.; World Scientific: Singapore, 2018; Volume 1, pp. 365–288.
23. Magnani, L. Eco-cognitive Computationalism: From mimetic minds to morphology-based enhancement of mimetic bodies. *Entropy* **2018**, *20*, 430. [CrossRef] [PubMed]
24. Magnani, L. *Abductive Cognition. The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*; Springer: Berlin/Heidelberg, Germany, 2009.
25. Lewontin, R. Organism and environment. In *Learning, Development and Culture*; Plotkin, H., Ed.; Wiley: New York, NY, USA, 1982; pp. 151–170.
26. Tooby, J.; DeVore, I. The reconstruction of hominid behavioral evolution through strategic modeling. In *Primate Models of Hominid Behavior*; Kinzey, W.G., Ed.; Suny Press: Albany, NY, USA, 1987; pp. 183–237.
27. Pinker, S. *How the Mind Works*; W. W. Norton: New York, NY, USA, 1997.
28. Pinker, S. Language as an adaptation to the cognitive niche. In *Language Evolution*; Christiansen, M.H., Kirby, S., Eds.; Oxford University Press: Oxford, UK, 2003; pp. 16–37.
29. Odling-Smee, F.J.; Laland, K.N.; Feldman, M.W. *Niche Construction. The Neglected Process in Evolution*; Princeton University Press: Princeton, NJ, USA, 2003.
30. Laland, K.N.; Sterelny, K. Perspective: Seven reasons (not) to neglect niche construction. *Evol. Int. J. Org. Evol.* **2006**, *60*, 4757–4779. [CrossRef]
31. Laland, K.N.; Brown, G.R. Niche construction, human behavior, and the adaptive-lag hypothesis. *Evol. Anthropol.* **2006**, *15*, 95–104. [CrossRef]