A Runtime Framework of the Mind

Fangfang Li

Galileo Mind Research

Liff1229@hotmail.com

Xiaojie Zhang

Chongqing Medical
and Pharmaceutical College
10802@cqmpc.edu.cn

Abstract

How the mind works is the ultimate mystery for human beings. This paper proposed a framework to solve it. We call it the self-programming system. The self-programming system can learn, store and apply the functions of bodies, external tools, and even the mind itself uniformly. However, due to the generality of the mind, traditional scientific methods are not suitable for validating a theory of mind. Therefore, we appeal to show the explanatory power of the self-programming system. Due to this reason, we applied this framework to solve the problem of consciousness. By comparing our solution to the popular theories of consciousness, we found that these theories only captured different aspects of our solution to consciousness. Beyond this, our solution also solved the hard problem of consciousness by showing that the existence of objectively unexplainable qualia is a matter of course even in a purely physical world. Our theory of consciousness manifests a new comprehensive perspective to understand what reality and knowledge are. And our framework provides a new approach to building human-level artificial intelligence.

Keywords: mechanism of mind, cognitive system, human-level AI, hard problem of consciousness, consciousness, ontology

1. Introduction

How the mind works is one of the most fascinating unsolved mysteries for human beings. In our previous article, we pointed out that the mind should be viewed as a self-programming system. (Li, 2022) Then, in order to build this self-programming system, we proposed a storage system for storing knowledge. Based on this storage system, we explained the relationship between inductive and deductive reasoning and also concluded the nature of causality.

In this article, we will further discuss how this storage system can support the runtime of the self-programming system, specifically, how the self-programming system can run based on the storage system and how to learn new knowledge. Through an analysis of the runtime and learning mechanisms of the self-programming system, we will show how the system can fully interpret consciousness from both functional and phenomenal aspects.

But before proceeding in detail, it is necessary for us to discuss the idea and purpose of our paper from the perspective of the methodology of validating scientific theories. The reason we do this seemingly superfluous step is that there is an irreconcilable contradiction between the nature of the mind and the popular scientific research paradigm. Therefore, we have to choose a way to validate our theory that does not quite fit this paradigm. Due to this situation, if we don't clarify these contradictions, it may lead readers to misinterpret the relationship between the puzzles we solved and our proposed framework.

First, let's look at why the popular paradigm is not applicable to validate the theory of mind. Under the popular scientific paradigm, a theory should be justified either through formal methods or experimental data. Let's begin with the justification by formal methods.

Specifically, to formally validate a theory of mind, what we actually do is to assume that the behavior of the agent conforms to certain rules, and then assume that the world in which the agent exists has certain laws that are independent of the behavior of the

agent. Then we can justify the theory by showing that the agent's behavior is reasonable/optimal/rational under the assumption about the world.

But, such reasoning is not reliable for validating a theory of mind. The key reason is that assuming the laws of the world is equivalent to admitting these laws are the absolute truths of the world. So the effectiveness of this proof must appeal to there exists an absolute truth and it has been attained and properly expressed in the assumption.

However, both from our experience of daily lives and scientific explorations, we know a key characteristic of our mind is that we can only progressively find more effective and general relative truths rather than relying on assumptions of absolute ones. Thus, we argue that once a theory of mind assumes some laws of the world independent of the observer will inevitably lose its generality, since it is only applicable to a special world following some particular laws. However, if we don't make assumptions about the world, but only assumptions about the mechanism of the mind, then formal methods will not lead to any meaningful results.

Now, let's look at verifying theories through empirical data. For a theory of mind or human-level intelligence, such methods can be further divided into two cases. The first is to only verify in a specific domain. The second is to verify by building a real concrete agent and placing it in the real world. And then examining its performance by the standard for a normal human.

The first is actually the current method of verifying weak artificial intelligence. However, this validation is meaningless for human-level intelligence.

Regardless of whether an artificial agent performs below or above humans in a special domain, it cannot represent the correctness or even the incorrectness of the theory. This is because, if an agent is inferior to humans, it can be attributed to the knowledge that humans have learned in other domains being used in this specific domain. The reason why humans are smarter is not because of the incorrectness of the theory. The reason could be the knowledge from this specific domain is not self-contain. In other words, to result that the theory is wrong from the fact that the agent underperforms humans must rely on the epistemic independence of each domain.

However, such independence is false even from the everyday experience of ordinary lives. Even for mathematics, which is usually considered domain-independent, this is still not the case. Because it is actually independent only in knowledge, but not in the way of what and how mathematical knowledge is inspired and accumulated.

On the other hand, outperforming humans in a specific domain does not imply the correctness of a theory of mind. For example, in doing arithmetic, computer programs are far superior to humans, but we do not think that arithmetic programs are a program of the mind.

The second way of validation empirically is to put an agent completely into human society, train it like a child, and then verify its effectiveness. This method is theoretically feasible, but its main drawback is that there will be no stage results to help further exploration. Therefore, this approach, even if feasible, does not fit the path of the scientific community to explore the world.

Such dilemma for validating theories of human-like intelligence has been identified in past literature but is seldom carefully studied. For example, the reason Turing invented the subjective-oriented Turing test as a way of validating artificial general intelligence could result from the similar reasons that we discussed above (Turing, 1950).

So is there any other way to validate a theory of mind? The answer is yes. In fact, the way of validating a theory has never been restricted to the quantitative approaches by competing for accuracy or precision in the longer history. A theory's generality and the ability to change "surprising facts" to "a matter of course" are also strong evidence to validate a theory (Woodward & Ross, 2021). Moreover, such evidence is especially important for theories at the foundational level. For example, Newton's theory is the unification of the laws of motion of celestial bodies and objects on the ground, and Maxwell's theory is the unification of electricity and magnetism. What's more, this approach also applies to the theory of evolution, which encountered the same problems of validation as a theory of mind. Due to its nature, the theory of evolution obviously cannot be verified quantitatively until it develops enough details. This even led to Popper (1974) refusing to acknowledge it as a scientific theory and commenting on it

as “a metaphysical research program”. However, the theory of evolution treats all living things under a unified framework and makes the existence of amazingly advanced living beings and the diversity of living things themselves an inevitable result. And these two points are the key reasons that the theory of evolution is being taken seriously.

We believe that a theory of mind is at the same foundational level as the theory of evolution. As philosopher Dreyfus (1972) has pointed out, a correct mental model should be able to bring insightful answers to many questions in epistemology and philosophy of mind. By following this way, it is reasonable to validate a theory of mind by validating its generality and the ability to change "surprising facts" to "a matter of course". In other words, to test the effectiveness of a theory of mind, one should test whether it is a biologically attainable unified explanatory framework (BAUEF for short) for all mental-related phenomena.

Along the way of showing BAUEF, our goal becomes to explain and solve puzzles of epistemology and cognitive tasks through the self-programming system. However, since every epistemological puzzle is far more complex than normal problems, our goal can only be achieved step by step. This means we have to show generality through a series of articles. In other words, if we consider articles in this series separately, they show how self-programming systems can turn "surprising facts" into "a matter of course" on a special problem. For example, Hume's induction problem and causality problem has been solved in our previous article and the consciousness problem is to be solved in this article. If we consider all articles in this series together, it demonstrates the generality of self-programming systems as a theory of mind.

For the above reasons, in this paper, *we do not propose a self-programming system to address consciousness, but rather we explain consciousness as one of the justifications underpinning the self-programming system*.

In addition, the problem of consciousness addressed in this paper has another important role -- it is the basis for solving other cognitive tasks. Even if we only collect experience from common lives, we can easily find that there are both conscious and unconscious processes in recognition tasks, problem-solving, decision-making, and the use of language. This means that for a theory of mind can solve these cognitive tasks,

it must first address the problem of conscious and non-conscious processes. For our self-programming system which is based on symbolic manipulation, one of the key issues is to explain how to solve the problem of perception. However, we cannot directly deal with it before understanding consciousness.

Next, we will first introduce the research background of consciousness and briefly demonstrate our solution. Then, in the second chapter, we will detail the runtime framework of the self-programming system, including its basic setting, runtime environment, and learning mechanism. Moreover, we will detail the relationship between consciousness and the self-programming system.

1.1 Functional aspect of Consciousness

How does consciousness exist? This question, like how intelligence works, has haunted all intellectuals since ancient history. However, due to its difficulty, it had been excluded from the field of science for a long time. Even worse, it was excluded from discussion by advocates of positivism, along with metaphysics. Until recent decades, attributed to the accumulation of empirical conclusions about how the brain works from neuroscience, scholars began to try to reason consciousness and proposed various theories (Seth & Bayne, 2002). Among these theories, one major category is constructed from the perspective of brain function. For example, the Global Workspace Theory (GWT) regards consciousness as a global space for information interaction. (Baars, 1988, 1997, 2002; Dehaene & Changeux, 2011; Mashour, Roelfsema, Changeux & Dehaene, 2020) The information in it will be broadcast to various subsystems, thus these subsystems can be combined to determine the optimal behavior globally.

Another class of influential theory from the functional perspective is the higher-order theory (HOT). The core idea of these theories is that if some information is conscious, then it must be the information for meta-representation. (Brown, Lau, & LeDoux, 2019; Rosenthal, 2005) The meta-representation here refers to a description that is not a direct description of the world but a higher-level description that goes beyond objective facts. For example, "yesterday, the vase was broken and seriously

affected my mood." In this case, the broken vase is a description of the objective world, and the whole sentence is a meta-representation beyond the objective.

Another new claim to consciousness that has developed in recent ten years is called Attention Schema Theory. This theory argues that the mind possesses an attention schema just like it possesses a body schema. And it is this attention schema that constructs consciousness (Graziano, 2013, 2019; Graziano, Guterstam, Bio & Wilterson, 2019).

In addition to these theories that view consciousness as a function of the brain, there is also a class of theories that argues that consciousness arises from some structures, and as long as these structures exist, it is conscious. For example, the well-known Integrated Information Theory (IIT) falls into this category. (Tononi, 2004; Tononi & Koch, 2015; Tononi et al., 2016) It claims that consciousness research should start from the phenomena of consciousness, and then infer the corresponding structures from these phenomena. Then it concluded that any system with these specific structures has consciousness.

However, if we start from a framework of mind, there is another way to think about the source of consciousness--why consciousness is indispensable in this framework.

Specifically, to claim a component of a framework can play the role of consciousness functionally should satisfy the following criteria: First, the existence of consciousness is because it provides some internal meaningful functions. And these functions are essential to this framework. This means that once these functions are missing, the framework cannot work from its beginning, and thus subsequent functions cannot be completed. As a counterexample, a model of a very simple organism that needs only to respond to external conditions has no need for consciousness.

Second, this function is exclusive, namely, the system is inoperable if it is treated in the same way as the unconscious function.

Third, the information expressed by its function is extractable, and this extractability is inherent meaningful for the framework.

In the framework of self-programming systems, consciousness is part of the runtime state space. The information in it will be compared with the information in the

storage system. This will extract the abstract relationships based on the storage system. Then these abstract relationships will spontaneously trigger operations that can manipulate the storage system such as retrieval of corresponding information.

In the self-programming system, abstracting based on the storage system is the core of realizing self-programing. And also all learning in this system depends on such abstraction. So its indispensability is obvious.

Second, because the process of abstraction is automatic and repeatable, If there is no distinction between the unconscious state and the conscious state, the abstract process will repeat infinitely. In other words, it will fall into an endless loop. Consciousness is therefore exclusive.

Third, since the operation of the self-programming system is similar to the invocation of functions in computer programming, the new operation often needs to rely on the past operation results. Therefore, the extraction of conscious information from the past is inherent significant to the system.

1.2 **Phenomenal aspect of consciousness**

In the previous section, we briefly described the background and our solution to consciousness from the functional perspective. Consciousness, however, can be viewed not only from a functional perspective, it can also be viewed from the phenomenal perspective, that is, the subjective experience of consciousness. Moreover, it is generally believed that to interpret such subject experience is indeed the hard part of explaining consciousness (Chamlers, 1996; Nagel, 1974; Levine, 1983, 1993, 2001). This question can be formulated more precisely as why there exists the subjective experience of consciousness that seems unexplainable by the usual scientific methods.

To this issue, past literature can be divided into four categories:

The first category holds the view that there is no subjective conscious experience (Rey, 1986; Dennett, 1991). However, this view is inconsistent with our experience.

The second category's view is that there exist conscious experiences and they can be explained by a normal mechanism (Churchland, 1986; Crick, 1994; Koch, 2004; Hurley, 1998; Noë, 2005, 2009). The main problem with such a view is that they fail to

explain that we seem to be perfectly capable of producing a mechanism with the same function but without consciousness.

Research in the third category acknowledges that conscious experience exists and it is not scientifically explainable. However, they believe such inexplicability is not so significant. The only important work is to know how to connect consciousness experience to physical stimuli (Block, 2002; Block and Stalnaker, 1999; Hill, 1997; Loar, 1997, 1999; Papineau, 1993, 2002; Perry, 2001).

The fourth category is dualism, that is, the world has both physical and consciousness. So it is not surprising that consciousness cannot be explained physically. This view can be traced back to Descartes. But this view is generally not accepted because it is divergent from the current scientific paradigm (Collins, 2011). Another alternative view is that although there are both physical and phenomenal objects, phenomenal experience does not have an impact on the physical world (Campbell, 1970; Jackson, 1982; Robinson, 2004). The natural question of this viewpoint is why there is such a non-necessary phenomenal experience.

None of the above four answers are satisfactory enough if we count their power to turn a surprising fact into a matter of course. However, based on the self-programming system, we can show that, even if the world we live in is purely physical, the epistemic world constructed by the mind will inevitably lead to subjective experience that cannot be explained by the physical laws of the world. The reason why scholars are puzzled in the past is because of the confusion between the ontology of the world and that of the epistemic world.

Next, let's take a look at how the self-programming system works.

2. The Runtime Framework of the Self-programming System

In our previous article, we introduced a storage system. In this article, we will further introduce how to use this storage system to implement the runtime framework of the self-programming system. Specifically, we will divide the following content into three

parts: 1) Define the components in this framework. 2) Explain how the self-programming system runs. 3) Introduce its learning mechanism.

2.1 Basic operations and basic elements

The components we first introduce are Basic Operations and Basic Elements. In the general-purpose computer, basic operations and basic elements refer to some basic symbols that are preset in the computer language. For example, logical operations like NAND or mathematical operations like addition and subtraction. The basic elements generally refer to symbols that can be manipulated such as numbers and identifiers. In other words, these computers are essentially defined on the basis of logic and mathematics. But in our framework, Basic Elements and Basic Operations have completely different meanings from that of traditional computers.

Specifically, both Basic Operations and Basic Elements refer to certain identifiable basic operations or signals provided by peripherals. These peripherals can refer to a certain part of the body, or they can refer to a module in the brain, such as a module that generates emotions.

So what are the Basic Operations and Basic Elements that peripherals provide? Generally speaking, since the functions of each peripheral are different, the basic operations and basic elements provided by each peripheral are also different. For the eyes, the basic operations can be rotation, positioning, focusing, and so on. The basic elements of the eye can be certain color blocks or a specific shape. For limbs, the basic operation can be some kind of rotation or movement. The basic elements can be moving to a certain angle or some tactile signal and so on.

In the self-programming system, although basic operations and basic elements are the basis for the thinking process, they are not so critical. This non-criticality is reflected in the fact that the system can display the same level of intelligence even without certain peripherals. There is only one category of basic operations that is indispensable here, which is the operations provided by the storage system.

Specifically, it only needs to have a storage system and any one way to interact with the external world, then it can produce an effective representation. In the next sub-

section, we will formally define components that are related to building these representations. Before diving into these technical details, it is crucial to illustrate the philosophical implication of these Basic Elements and Basic Operations.

Since the self-programming system is established upon Basic Elements and Basic Operations and the body's Basic Elements and Basic Operations are from its sensorimotor, the self-programming system seems to appear similar to Merleau-Ponty's phenomenology of perception and the embodied cognition developed from it. However, there are substantial distinctions between our way of achieving the goal and their mainstream method.

First, in the self-programming system, both Basic elements and Basic operations can be viewed as symbols. It's just that these symbols accompany by a look-up table to indicate their meanings. The advantage of this setting is that the schema composed of these basic symbols is independent of the specific existence of the components that provide these symbols. Thus, it enables functions from various components can be expressed uniformly. In this sense, the self-programming system indeed establishes a schema composed of basic symbols that depict all external and internal objects by exploring the relationship between these symbols.

More importantly, applications of this schema don't need knowledge about the lookup table. One may doubt this conclusion by arguing: if you don't interpret the internal representations by virtue of the lookup table, how can you know the true phenomenon happened in the objective world? In fact, the reason for this question is that it is presupposed to seek objective truth from the perspective of a third party. But, in fact, the mind does not need such conversion, because phenomena and the relationships between these phenomena already have been expressed internally. Thus the mind can carry out various thinking activities directly through internal expressions, such as planning, judgment, etc. In this case, objective reality is not a necessary factor for the functioning of the mind. This feature further implies the robustness of the self-programming system is not suscept to the disturbance of the look-up table, since changes of the look-up table will lead to corresponding modifications of the schema.

This view is different from the current mainstream methods of schema based on perception phenomenology. Specifically, the mainstream representations of schemas are relying on the form of the existence of these components. For example, body schemas are encoded in 3D space (Morasso et al., 2015; Macaluso & Maravita, 2010).

Second, a basic symbol does not necessarily correspond to a unique external object. It may correspond to a set of objects. A particular external object is determined by a set of multiple basic symbols. For example, one basic symbol might represent a circular area that appears on the retina, while another basic symbol represents the size of the area on the retina. Neither of these two symbols, respectively, can identify any unique retinal stimulus. But the combination of the two symbols can correspond to a certain stimulus.

Finally, since we defined the self-programming systems in virtue of basic symbols, this contradicts the mainstream view of embodied cognition. They believe that embodied cognition and symbolic manipulation are not compatible with each other (Varela, Rosch, Thompson, 1991; Shapiro and Spaulding, 2021). However, we believe it is feasible to realize embodied cognition based on symbols by adopting self-programming systems. Thus, let's continue how we can achieve this purpose technically.

2.2 Operations, Properties, Property set and Storage system

We first give the definitions of the following four concepts, and then make further analysis on this basis:

Operations: a sequence of other operations or basic operations that can be executed under specific conditions; these specific conditions here refer to the object that can be operated must have certain properties.

An object has a certain property: if places a certain operation on this object, it will inevitably obtain another object that definitely possesses all properties of a certain property set or have a property represented by a basic element.

Property set: A collection of one or more specific properties. It is the basic storage object in the storage system.

Storage system: It consists of two parts, one is a collection of all property sets, and the other is some specific operations that can retrieve and compare information stored in this storage system.

At first glance, the above definition seems to have a circular definition problem. For example, the definition of an operation depends on a condition, and a condition is a property, but on the other hand, the definition of a property depends on the definition of operation. In addition, the definition of a property set depends on the property, and the definition of the property itself depends on other property sets. However, if we think in terms of construction, the above definition is logically clear.

The reason is that these definitions can be built up step by step starting from basic elements and basic operations. Specifically, the combination of basic elements and basic operations is sufficient to construct a sequence of operations and their results. Thereby, properties are constructed. And multiple properties actually form a set of conditions, which can be combined with a sequence of other basic operations to form a new operation. In other words, the conditions of an operation are actually constructed gradually in order, that is, the properties constructed first become the conditions under which the new operation can be created. The same method can also be used to construct property sets, that is, starting from the property set represented by a single property, and gradually defining more complex property sets. (See Figure 1)

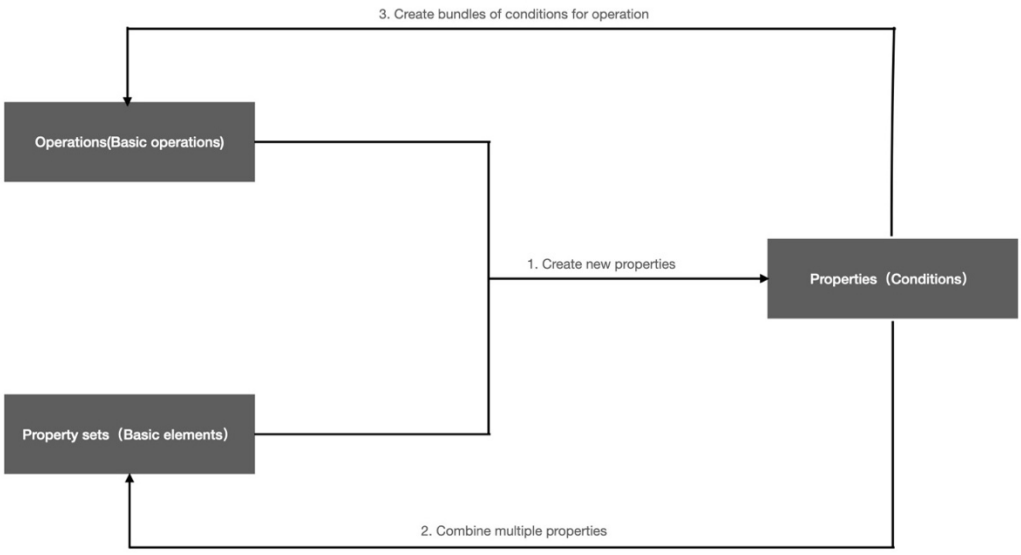

Figure 1 The relationship between operations, properties and property set

2.3 The runtime of the self-programming system

Next, let's take a look at how the self-programming system utilizes its storage system to run. The running of a self-programming system can be summed up in one sentence: it is a mapping from a runtime state to an operation. We have already talked about the definition of operation, but what is the runtime state?

First, at any given moment, the runtime states can be divided into two parts, the explicit state and the implicit state. An explicit state refers to a set of states that express the external world and internal states that are currently perceived through observation, perception, or computation. For example, if someone saw a plate on the table with an apple in it, his/her explicit state will include these property sets that represent the apple, the plate, and the table, and the network that represents the positional relationship between these three. In this way, the explicit state represented the observed state of the external world. The internal state express, for example, the current mood or the feeling like hunger in the body. At the same time, in the explicit state, there is also a goal. For example, when you are hungry, the goal can be to find a way to eliminate hunger. In a similar way, the explicit state can also represent relationships that have a time span, such as, the "self" just tried to solve a problem in a certain way but failed.

Then what is the implicit state? Simply speaking, the implicit state is the relationship between the explicit state and the storage objects in the current storage system. For example, let's say the current explicit state is that there is an apple on the table as described above, and the goal is to eliminate hunger. Then the implicit state may be: all storage objects that represent apples in the storage system can eliminate hunger by "eating it" (state 1); it could also be: there are some storage objects that represent apples indicate that apples can eliminate hunger, but others indicated not, such as existing a storage object representing a toy apple. (state 2).

On the surface, there seems no essential difference between implicit and explicit state, since both of them are established through some kind of mental operation. But in fact, there are two differences between them. First, the transition from explicit state to implicit state is spontaneous. Without distinguishing between implicit and explicit, the

runtime state will grow indefinitely. This is because the current explicit state generates the implicit state, the implicit state can continue to be compared with the objects in the storage system, resulting in a second-order implicit state. If there is no controlling, this process will continue endlessly. Once these two states are separated, the process of generating implicit states runs only when the information is deliberately put into the explicit state. In other words, this process can be called if and only if it is necessary.

On the other hand, the process of generating an implicit state from an explicit state is parallel and imperceptible. It is like the inherent function of the storage system. This means that the process and the outcoming implicit states are not recorded inside the self-programming system. Thus, there has no way to recall what the implicit state once existed.

Based on the relationship between explicit state and implicit state, we suggest that explicit states are the space of consciousness. Also, there should be a peripheral in the brain that automatically records the history of explicit states. The history of these explicit states can be traced back by using some specific operations. Therefore, we can and can only be aware of the information that once appeared in the explicit state, but cannot perceive the information in the implicit state.

After discussing the runtime states, we can go back and understand the runtime itself. As we said earlier the runtime itself is an ongoing process of mapping from the current runtime state to a specific operation. Now we can discuss what this mapping exactly is.

As we discussed earlier, at a given moment, there are runtime states which contain explicit and implicit state. Then, how does the self-programming system use these runtime states? From a perspective of the runtime procedure, the runtime states will be first mapped to an implicit operation. The role of this implicit operation is to find the appropriate operations that were recorded in the storage system, namely explicit operation, for the current runtime state. And also the implicit operation will determine how to use these explicit operations, such as direct execution or sending to the explicit state, etc. Since explicit operations are recorded by the storage object, an implicit

operation that extracts an explicit operation is an operation that acts on the storage system itself. (see Figure 2)

For example, if the implicit operation corresponding to the implicit state happens to find that there is only one explicit operation that can achieve the goal in the explicit state (as in the case of state 1 in the previous example). Then the implicit operation can choose to run this explicit operation directly.

What if the implicit operation find not a single appropriate explicit operation? In some situations, there may exist multiple ways to achieve the goal? For example, if you want to calculate 324x99, you can directly use the general multiplication method, but you can also use 324x100-324 to calculate; Similarly, there may not exist any known operations in the storage system that can achieve the goal, for example, the goals like how a light-speed spacecraft can be built. There may also exist some way that can only achieve the goal with uncertainty, such as state 2 in the previous example.

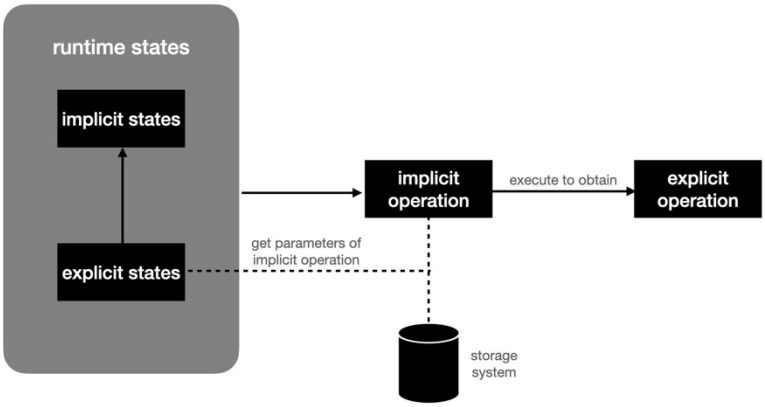

Figure 2 The procedure of runtime

In each of the above situations, there are further subdivisions. For example, in the case of State 2 mentioned above, the implicit operation may choose the explicit operation based on whether there are properties that can be easily collected and helpful for making further decisions. If such a property exists it can execute the explicit

operation that can collect this property at first. Corresponding to State 2 of the previous case, it is possible to touch the apple first and decide whether to eat it.

In some cases, the state of the explicit operations discovered by the implicit operation can also be put into the explicit state for further calculations of what should be done. For example, if no possible solution is found, some attempts may be made by using the functions provided by other peripherals, such as a search that allows combining two operations together.

In cases where there are multiple explicit operations, it is also possible to put all these explicit operations into an explicit state to determine which one is more appropriate.

We can see that the process of putting the result found by the implicit operation corresponding to the implicit state into the explicit state is a process that can go on in an infinite loop. So is it possible to get stuck in an endless loop of thinking without actually solving any problem? In theory, this possibility does exist. But in fact, the self-programming system will try to avoid this problem in some ways.

For example, if the same content is written to an explicit state multiple times, a state will be generated in the implicit state, that is the multiple writes in the explicit state are the same. Then, relying on this new state, system can jump out of the infinite loop by give up. For example, reasoning the problem of chicken and eggs which comes first can be followed into this category.

So can this way guarantee that the infinite loop will not happen? The answer is no, for example: in some extreme cases, the system may not be able to find that it has entered an infinite loop, or even if it finds that it has entered an infinite loop, its implicit operation chooses to not change, then the system may enter an infinite loop.

However, we believe that this kind of infinite loop is not a design error of the self-programming system, because such an infinite loop also occurs in the thinking process of human. For example, a person who has symptoms of anxiety or depression is dropped into such a situation. Taking anxiety disorders as an example, it is that the anxiety state will guide the implicit operation to find the anxiety source and try to solve it, but when

the anxiety source cannot be solved, it will further cause the anxiety state. Thus entering an infinite loop.

In summary, the runtime of a self-programming system provides a function that maps to the execution of specific operations based on conditions and goals. This function is obtained by comparing the current runtime state with the information in the storage system. Therefore, the whole process of locating and executing a specific operation from the runtime state can be regarded as a basic operation (BO) provided by the storage system. Since an operation in a storage system is a composition of basic operations, this means that the operation that invokes the runtime can actually also be a possible component of the operation that compose properties.

If we compare this point to computer programming, the storage system is equivalent to providing a dynamic mapping from function names to function implementations. This mapping will be automatically updated as the storage objects in the storage system are updated.

More importantly, we can see that when an operation of call runtime procedure is a component of another operation of call runtime procedure, the calling of the parent operation will cause the child operation to be called. Combining this mechanism with the nature of causality introduced in our previous article, and seeing the "self" as a storage object(we will introduce how a storage object that represents "self" can be created in section 2.6), then a motivation-driven chain of causality can be formed, that is, the reason that executes the child operations is because of the call of the parent operation. For example, why someone is booking a flight might be because he wants to travel to relax. In this case, booking a flight is a child operation determined by the operation of travel.

Combining the content of our previous article, we can see that no matter the causality of the objective world or the causality of human behavior, all of them are consistent with our theory of causality, that is, the causality is a description of the modification of default operation.

Through the study of the self-programming system, we can discover some important properties. First, a self-programming system is by no means a combination

of multiple domain-specified systems. The reason is that the key to realizing a self-programming system is the relationship between the storage system and external observations, and how to operate the data in the storage system under these relationships. This is a completely abstract domain that is independent of any specific domain. No matter what domain a problem belongs to, it ultimately lies in how to manipulate the data in the storage system. This means that, for any information, as long as it can be stored, it can be processed in the same way.

On the other hand, we can see that when the runtime state triggers an operation, the operation could consist of a sequence of sub-operations that may trigger new mappings. This is a process similar to fractal problems in complex science. Therefore, solving one part of a problem is no easier than the whole problem. In other words, without a proper understanding of the storage system, even trying to solve some seemingly simple problems will lead to clueless.

Third, sample-based methods like current machine learning cannot learn the essence of the self-programming systems. Because the sample-based method can only discover specialized mapping of implicit operations. These mappings are just products of storage-system-related general implicit mappings. (We will discuss how this could be done in the part of the learning mechanism). The failure of learning the general mappings will lead to machine learning exhibiting poor flexibility in a transient environment.

2.4 Learning mechanism

As can be seen from the previous analysis of runtime, if the mapping of runtime states to implicit operations and the information in the storage system are given, the run of the self-programming system will be determined. In other words, how the self-programming system works depends on the information in the storage system and the implicit mapping. There is a naturally following question that is how the storage objects and implicit mapping are established? Or what is the learning mechanism behind them?

The problem is both simple and complex. First of all, we believe that the mechanism of establishing storage objects and implicit mapping is no different from

the establishment of conditioned reflex in biology. Simply put, a new property can be formed if conditions, operations, and result objects are triggered repeatedly. Since the properties are the content of the storage object, creating properties is equivalent to creating new storage objects.

However, an answer like this can only be regarded as a basic functional explanation of the learning mechanism. The more important question is, what are the application conditions of the above mechanism or when will this mechanism be used? Unfortunately, facing this question, we can only answer part of it. The other part cannot be summed up by the nature of the self-programming system.

In the self-programming system, the application of any function has two different levels, namely the spontaneous level and the purposeful level. This rule is also applicable to the learning mechanism. Its spontaneous level refers to the fact that this learning mechanism is automatically triggered during the operation of the system. The role of the learning mechanism at this spontaneous level is relatively simple and can be described. It works on at least the following three aspects.

First, the most immediate aspect is to work with explicit state at runtime. Specifically, if a certain storage object happens to be triggered at some point, its properties are loaded into the explicit state. At this time, if the same result that generated by an operation happened repeatedly, then a new property that contains the new operation and the result will be created. And this new property combines with the properties from the original object to generate a new storage object.

Second, since the runtime state not only has explicit state and explicit operations, but also has corresponding implicit states and implicit operations, the learning mechanism works should also work on the implicit aspect. In the implicit aspect, learning refers to building mappings from implicit states to appropriate implicit operations. Taking the previous calculation 324x99= as an example, the implicit state is that there are multiple ways to calculate this result, and the implicit operation is to list this method into the explicit state and consider it further.

Third, specializing implicit mappings. We introduce this aspect by an example. Assume there is a problem, and both operations A and B known in the system can

solve it. We know that in this case both operations A and B shall be put into the explicit states to be evaluated by a more general implicit operation. Here, we further assume that the result of the evaluation is that Operation A executes faster so Operation A is always called in more urgent situations; Operation B has a higher success rate, thus it is always called in situations with spare time. Then if these operations are called repeatedly, two new implicit mappings will be created: Calls Operation A under emergency situation. Call Operation B when there is spare time. In this way, the process of loading the implicit state into the explicit state is avoided by forming a specialized mapping, thereby reducing the computational cost.

After talking about spontaneous learning, let's talk about purposely learning. As we said before, if certain states, operations, and results occur repeatedly, then a new storage object will be generated. This newly created storage object expresses a specific function by its properties. The learning mechanism can still be viewed as a function, thus it can also be expressed by a storage object which is created by the repeat of the spontaneous learning process. The result is that a storage object that expresses the learning mechanism will exist in the storage system.

Once the above storage object is created, the self-programming system can use the learning mechanism to create new storage objects purposefully like other peripherals. In this case of purposely learning, the question of when to apply the learning mechanism becomes a non-summerizable question, since its application conditions are completely determined by the self-programming system itself. As we said earlier, the problem of self-programming is a fractal problem. So in this sense, summarizing it is equivalent to resummarizing the whole self-programming system.

2.5 Evidence from neuroscience

As pointed out at the beginning, we have found a necessary reason for the existence of consciousness from the perspective of creating the mind, so as to find a computational explanation of consciousness within the self-programming system. If this explanation is correct, it should be able to integrate with existing theories of consciousness, complement what is missing, and help analyze where they are

complementary and where they are conflicting with each other. Next, we will look at the relationship between our theory of consciousness and existing theories of consciousness.

First and foremost, we believe that the four types of consciousness theories described in the introduction do not have any essential conflict in their starting points. In our interpretation, the role of the conscious space is to compare with the storage system for abstracting the relationship in the storage system. This abstracted information will further trigger operations on the storage system. Consciousness, therefore, is both a product of structure and at the same time intended to achieve some kind of meta-representation. This is consistent with both these theories that hold that consciousness is based on a structure, as well as with higher-order theories.

Moreover, because the storage system stores functional information aggregated from the brain、body and even other tools, it is able to determine appropriate behavior globally. From this point of view, it is consistent with the Global Workspace Theory. So the starting points of these three theories just describe three different perspectives of consciousness.

The situation of the attention schema theory is slightly different. We believe it accurately captured the correlation between attention and consciousness. However, it reversely concluded the causal relationship. In other words, consciousness should be the basis to define attention rather than the reverse.

If we use the notion of schema to correspond to the storage system of the self-programming system, then the storage system is an integrated schema of all known objects. This is because all functions of known objects are stored there. From this perspective, the attention schema actually refers to the part of implicit operations that will send new information to the explicit state. In this sense, attention is defined by implicit operations and explicit state space. Based on the equivalence of explicit state space and consciousness space, we can conclude that consciousness defines attention rather than attention constructs consciousness.

Second, we can also make the understanding of higher-order theory and global space theory clearer. Specifically, what is the meta-representation in the HOTs (Brown,

2015; Cleeremans, 2011; Cleeremans et al. 2020; Fleming, 2020; Lau & Rosenthal, 2011; LeDoux & Brown, 2017) and what does the "global" in the GWT refer to? (Bayne, 2010; Carruthers, 2019) These two problems have always been one of the core problems they face respectively.

In our theory of consciousness, both these points can be clearly explained. The core of the meta-representation is the relationship of the conscious content to the storage system. The global space in the global space theory is actually the space used to extract abstract relationships based on the storage system.

Finally, our theory can also directly explain the unity of consciousness (Bayne, 2010; Bayne & Chamlers, 2003)—that is, why the information that is realized is always integrated rather than separated aspects. In our theory, the role of information in consciousness is to correspond to the storage objects in the storage system, and each object is a complex of multiple properties. Therefore, what consciousness must perceive is not the individual properties, but the integration of them.

2.6 Self and the hard problem of consciousness

In the previous section, we explained what consciousness is in terms of its functional aspect. However, for a theory that aim to fully interpret consciousness, this solves at best part of the problem. And this part is called the easy problem of consciousness by David Chalmers (1995). The hard problem of consciousness is how the subjective experience of consciousness can be interpreted. This question is still open nowadays.

So what is the nature of the hard problem of consciousness? To understand it, we need to start with what the self is. In fact, we already discussed in the learning mechanism section that the reason a storage object is formed is to pack the properties of the object being perceived. Thus a storage object expresses the observed object. If the observed object is a body part, then there will be a storage object representing the body part; if the observed object is an external being, then there will be a storage object expressing the external being. So what if the object being observed is the runtime of the self-programming system itself? Then the storage object formed will express all the content that appears continuously in the explicit state space. Since we already know the

content of the explicit state space is actually a result of both implicit manipulation and external stimuli based on the body. Therefore, from an external point of view, this storage object represents the whole experience of the mind. Therefore, it expresses the subjective self.

From this, we can conclude that both the representation of the external world and the self are combinations of basic elements and basic operations. And they all exist in the storage system in the form of storage objects associated with each other. Based on this conclusion, we can further infer from the definition of "objective" that the objective representation of the world is the remaining part after all the properties connected to the self in the storage system are removed.

Then, let's look at what the nature of interpretation is. The so-called interpretation is actually that some observed properties can be deduced from other properties. These properties that are used to deduce are called basic laws. Because the objective representation of the world is what remains after removing properties associated with the self. Therefore, the basic laws in the so-called objective interpretation must be the set of properties contained in the part without any property related to the self.

However, we also know that the self is the collection of all subjective experiences. Therefore, any basic laws that can explain subjective experience necessarily require the inclusion of the subjective experience of the basic elements of the cognitive system which must be related to the self, so they cannot be contained in the basic laws of the objective part. This means that objective laws cannot be used to explain subjective experience. So the inexplicability of phenomenal consciousness by the analysis from the functional perspective is the inevitable result of consciousness generated by self-programming systems.

From the above analysis, we can also see that if there is an objective world, and a subject in the world who can cognize the world and itself. Then it is entirely possible to produce subjective experiences that cannot be explained by the laws of this objective world. This conclusion also implies that if we hope to explain the subjective experience, we must first define the subjective experience of the basic elements of the cognitive system as axioms.

3. Summary

Combining our last article's discussion of induction/deduction, causality and the discussion of consciousness in this article, we have shown the power of the self-programming system theory on three domains, thus indirectly demonstrating its rationality.

In fact, our theory can also explain important phenomena pointed out by other theories. For example, one of our key arguments is that various functions of the body and brain are recorded in storage systems in abstract form. In this way, when a problem needs to be solved, it can be planned through the storage system. These recordings can also be extended to tools outside the body. In this sense, we collectively call these tools that can be recorded by the storage system as peripherals. For example, the learning mechanism described in this article is a key peripheral in the brain. This viewpoint is consistent with enactive and extended cognition beliefs, which claim that there is no clear separation between mental and non-mental processes and no essential difference between external tools and bodies (Varela, Thompson and Rosch, 1991; Clark and Chamlers, 1998; Menary 2010). This means that we achieved some of the main goals of embodied cognition with the approach of symbolic manipulation without appealing to their anti-symbolic convention, such as autopoiesis, etc.

Reference

1. Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press.

2. Baars, B. J. (1997). In the theatre of consciousness: Global workspace theory, a rigorous scientific theory of consciousness. *Journal of Consciousness Studies, 4*(4), 292–309.

3. Baars, B. J. (2002). The conscious access hypothesis: Origins and recent evidence. *Trends in Cognitive Sciences, 6*(1), 47–52. https://doi.org/10.1016/S1364-6613(00)01819-2

4. Bayne, T. J. (2010). *The Unity of Consciousness.* Oxford University Press UK.

5. Bayne, T. J. & Chalmers, D. J. (2003). What is the unity of consciousness? In Axel Cleeremans (ed.), *The Unity of Consciousness*. Oxford University Press.

6. Block, Ned (2002). The Harder Problem of Consciousness. *Journal of Philosophy* 99 (8):391.

7. Block, Ned & Stalnaker, Robert (1999). Conceptual analysis, dualism, and the explanatory gap. *Philosophical Review* 108 (1):1-46.

8. Brown, R. (2015). The HOROR theory of phenomenal consciousness. *Philos Stud* **172,** 1783–1794. https://doi.org/10.1007/s11098-014-0388-7

9. Brown, R., Lau, H., & LeDoux, J. E. (2019). Understanding the higher-order approach to consciousness. *Trends in Cognitive Sciences, 23*(9), 754–768. https://doi.org/10.1016/j.tics.2019.06.009

10. Campbell, Karlyn K. (1970). *Body and Mind*. Doubleday.

11. Carruthers, P. (2019). *Human and animal minds: The consciousness questions laid to rest.* Oxford University Press. https://doi.org/10.1093/oso/9780198843702.001.0001

12. Chalmers, D. J. (1996). *The conscious mind: In search of a fundamental theory.* Oxford University Press.

13. Churchland, Patricia Smith (1986). *Neurophilosophy: Toward A Unified Science of the Mind-Brain*. MIT Press.

14. Clark, Andy & Chalmers, David J. (1998). The extended mind. *Analysis* 58 (1):7-19.

15. Cleeremans A. (2011). The Radical Plasticity Thesis: How the Brain Learns to be Conscious. *Frontiers in psychology*, *2*, 86. https://doi.org/10.3389/fpsyg.2011.00086

16. Cleeremans, A., Achoui, D., Beauny, A., Keuninckx, L., Martin, J. R., Muñoz-Moldes, S., Vuillaume, L., & de Heering, A. (2020). Learning to Be Conscious. *Trends in cognitive sciences*, *24*(2), 112–123. https://doi.org/10.1016/j.tics.2019.11.011

17. Collins, Robin (2011). The Energy of the Soul. In Mark C. Baker & Stewart Goetz (eds.), *The Soul Hypothesis: Investigations Into the Existence of the Soul*. Continuum Press. pp. 123-133.

18. Crick, Francis (1994). *The Astonishing Hypothesis: The Scientific Search for the Soul*. Scribners.

19. Dehaene, S., & Changeux, J. P. (2011). Experimental and theoretical approaches to conscious processing. *Neuron*, *70*(2), 200–227. https://doi.org/10.1016/j.neuron.2011.03.018

20. Dennett, Daniel C. (1991). *Consciousness Explained*. Penguin Books.

21. Dreyfus, H. L. (1972). *What computers can't do: A critique of artificial reason.* New York: Harper & Row.

22. Fleming S. M. (2020). Awareness as inference in a higher-order state space. *Neuroscience of consciousness*, *2020*(1), niz020. https://doi.org/10.1093/nc/niz020

23. Graziano, M. S. A. (2013). *Consciousness and the social brain.* Oxford University Press.

24. Graziano, M. S. A. (2019). Attributing awareness to others: The attention schema theory and its relationship to behavioural prediction. *Journal of Consciousness Studies, 26*(3-4), 17–37.

25. Graziano, M. S. A., Guterstam, A., Bio, B. J., & Wilterson, A. I. (2020). Toward a standard model of consciousness: Reconciling the attention schema, global workspace, higher-order thought, and illusionist theories. *Cognitive Neuropsychology, 37*(3-4), 155–172. https://doi.org/10.1080/02643294.2019.1670630

26. Hill, Christopher S. (1997). Imaginability, conceivability, possibility and the mind-body problem. *Philosophical Studies* 87 (1):61-85.

27. Hurley, Susan L. (1998). *Consciousness in Action*. Harvard University Press.

28. Jackson, Frank (1982). Epiphenomenal qualia. *Philosophical Quarterly* 32 (April):127-136.

29. Koch, Christof (2004). *The Quest for Consciousness a Neurobiological Approach*. Roberts & Co.

30. Lau, H., & Rosenthal, D. (2011). Empirical support for higher-order theories of conscious awareness. *Trends in Cognitive Sciences, 15*(8), 365–373. https://doi.org/10.1016/j.tics.2011.05.009

31. LeDoux, J. E., & Brown, R. (2017). A higher-order theory of emotional consciousness. *PNAS Proceedings of the National Academy of Sciences of the United States of America, 114*(10), e2016–e2025. https://doi.org/10.1073/pnas.1619316114

32. Levine, Joseph (1983). Materialism and qualia: The explanatory gap. *Pacific Philosophical Quarterly* 64 (October):354-61.

33. Levine, Joseph (1993). On Leaving Out What It's Like. In Martin Davies & Glyn W. Humphreys (eds.), *Consciousness: Psychological an Philosophical Essays*. MIT Press. pp. 543--557.

34. Levine, Joseph (2001). *Purple Haze: The Puzzle of Consciousness*. Oxford University Press USA.

35. Li, F. (2022). Why is the mind a self-programming system?. https://doi.org/10.31234/osf.io/jvs58

36. Loar, Brian (1997). Phenomenal states II. In Ned Block, Owen Flanagan & Güven Güzeldere (eds.), *The Nature of Consciousness: Philosophical Debates*. MIT Press.

37. Loar, Brian (1999). David Chalmers's The Conscious Mind. *Philosophy and Phenomenological Research* 59 (2):465 - 472.

38. Macaluso, E., & Maravita, A. (2010). The representation of space near the body through touch and vision. *Neuropsychologia, 48*(3), 782–795. https://doi.org/10.1016/j.neuropsychologia.2009.10.010

39. Mashour, G. A., Roelfsema, P., Changeux, J. P., & Dehaene, S. (2020). Conscious Processing and the Global Neuronal Workspace Hypothesis. *Neuron*, *105*(5), 776–798. https://doi.org/10.1016/j.neuron.2020.01.026

40. Menary, R. (Ed.). (2010). *The extended mind.* MIT Press. https://doi.org/10.7551/mitpress/9780262014038.001.0001

41. Morasso, P., Casadio, M., Mohan, V., Rea, F., & Zenzeri, J. (2015). Revisiting the body-schema concept in the context of whole-body postural-focal dynamics. *Frontiers in human neuroscience*, *9*, 83. https://doi.org/10.3389/fnhum.2015.00083

42. Nagel, Thomas (1974). What is it like to be a bat? *Philosophical Review* 83 (October):435-50.

43. Noë, Alva (2005). *Action in Perception*. MIT Press.

44. Noë, Alva (2009). *Out of Our Heads: Why You Are Not Your Brain, and Other Lessons From the Biology of Consciousness*. Hill & Wang.

45. Papineau, David (1993). Physicalism, consciousness and the antipathetic fallacy. *Australasian Journal of Philosophy* 71 (2):169-83.

46. Papineau, David (2002). *Thinking About Consciousness*. Oxford University Press UK.

47. Perry, John (2001). *Knowledge, Possibility, and Consciousness*. MIT Press.

48. Popper, K. (1974) Darwinism as a metaphysical research programme. In: Schillp, P.A., Ed., The Philosophy of Karl Popper. Open Court, LaSalle, 133-143.

49. Rey, Georges (1986). A question about consciousness. In Herbert R. Otto & James A. Tuedio (eds.), *Perspectives on Mind*. Kluwer Academic Publishers.

50. Robinson, William S. (2004). *Understanding Phenomenal Consciousness*. Cambridge University Press.

51. Rosenthal, David M. (2005). *Consciousness and Mind*. Oxford University Press UK.

52. Shapiro, Lawrence and Shannon Spaulding, "Embodied Cognition", *The Stanford Encyclopedia of Philosophy* (Winter 2021 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/win2021/entries/embodied-cognition/>.

53. Seth, A.K., Bayne, T. (2002) Theories of consciousness. *Nat Rev Neurosci* **23,** 439–452. https://doi.org/10.1038/s41583-022-00587-4

54. Tononi, G. (2004). An information integration theory of consciousness. *BMC Neurosci* **5,** 42 . https://doi.org/10.1186/1471-2202-5-42

55. Tononi, G., & Koch, C. (2015). Consciousness: here, there and everywhere?. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, *370*(1668), 20140167. https://doi.org/10.1098/rstb.2014.0167

56. Tononi, G., Boly, M., Massimini, M. *et al.* Integrated information theory: from consciousness to its physical substrate. *Nat Rev Neurosci* **17,** 450–461 (2016). https://doi.org/10.1038/nrn.2016.44

57. Varela, F. J., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, Mass: MIT Press.