A Characterization of Processing Loops in AI and Biological Systems and its Implications for Understanding Consciousness

Malcolm J. Lett1

1No affiliation

**\* Correspondence:**Malcolm Lett  
malcolm.lett@gmail.com

Keywords: Access Consciousness, Phenomenal Consciousness, Biosemiotic Process, Visceral Loop, Monitoring and Control, Recurrent Process, Non-physical Action, Conscious Content

Abstract

todo

# Introduction

Computational and connectionist theories of consciousness are inherently built upon the idea of a state machine and loops, but they fail to draw specific reference to this dependency.....GWT, etc. etc. (citations). Thus, there is an avenue for further insights.

The field of meta-cognition has begun to make inroads. Originally focused on the most outward behavioural aspects of the fact that people who are more aware of their own learning strategies, strengths, and weakneses, do better. Now, meta-cognition research investigates how the brain performs those behaviours. Furthermore, many have suggetsed that meta-cognition may be the basis for consciousness itself (citations). Meta-cognition has been implicated in ............(behaviours, with citations)....

Like theories of consciousness, much of the meta-cognition theorising is at the level of behaviours or whole of brain processes. Some attempt to draw references to specific brain regions (..citations..) but that work is still very speculative. Only a few ...(citations)... have attempted to simulate such processes in connectionist models. Those simulations are usually very simple. For example, they (...citations...) simulate the construction of higher-order representations about certainty, but don't use that as a feedback signal for the system to incorporate into its processing.

The present paper attempts to strengthen the meta-cognition research in two ways. First, it attempts to bridge the gap between existing connectionist mechanisms and meta-cognitive theories by highlighting specific low-level connectionist mechanisms that might form the basis for meta-cognition. Secondly, it examines different connectionist architectures, and shows how those architectures lead to different observable results.

In order to focus on practicality, a "design stance" is taken.... (citation and explanation).... This leads us to focus on the bottom-up design, which serves two purposes. i) It provides a stronger proof of the value in the arguments, and ii) it offers direction for using the knowledge to build systems with these capabilities.

# Meta-management in connectionist architectures

Here the focus shifts from biology to artificial neural networks, so that we may examine the problem from a "design stance" point of view (citation and explanation, if not covered above).

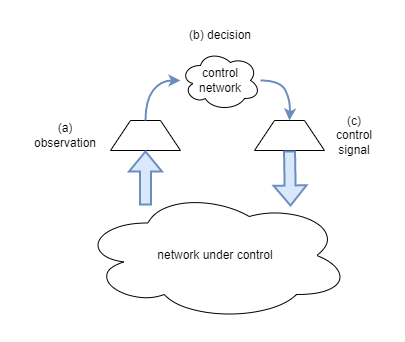
Why might we want to add meta-management processes to connectionist architectures? Deep AI techniques have had many successes of late (citation). However, these networks still lack some of the most basic adaptive capabilities that we see in many biological organisms (citations). Here some specific meta-management features are discussed that could benefit existing deep AI architectures.

Todo: discuss each of the following in terms of:

* + benefits
  + relationship to meta-representation
  + relationship to observation/logical reasoning/control components of meta-management
  + theoretical mechanisms
  + empirical evidence, if any
  + give citations of AI examples showing the benefits of each of these

## Dimensionality reduction

Meta-management is management over reduced dimensionality.

Figure 1: Dimensionality reduction. A large network under control (NUC) can be observed and controlled efficiently by a much smaller network. Dimensionality reduction benefits the process in three ways. (a) a reduced dimensionality observation of the state of the NUC enables the control network to interpret that state without an exponential increase in the total number of neurons in the system. (b) When operating over a reduced dimensionality, the control network can learn with fewer training iterations, and apply more advanced decision rules with less resources. (c) the output of the control network can also be in a dimensionally reduced space, further simplifying its computations, as hierarchical models provide a mechanism for low-dimensionality signals to control higher-dimensionality networks.

Observation: A network cannot micro-manage itself. In order to observe the full state of every neuron would require at least just as many neurons again, or probably many times more. Thus, the dimensionality of the observation of system state must be significantly reduced for the practical purpose of avoiding an exponential scaling out in the number of neurons of the total system. Predictive mechanisms are well suited to this. Typically predictive mechanisms are used to infer the hidden *latent* state of a system, based on observations obtained about that system. A side effect is that the inferred latent state is only an estimated representation of the true system latent state, and consequently it usually has significantly less dimensionality than the true latent state. Thus, the predictive mechanism can also be seen as a dimensionality reduction mechanism that produces a self-stabilising (auto-convergent) simpler representation of the state of the system under observation.

Decision: A reduced dimension state space is beneficial for the control logic. Learning good control methods/parameters is more efficient and more stable in a lower dimensional state space. Additionally, the control system can apply more complex rules with less resources than it would otherwise need.

Control signals: Lastly, a reduced dimensionality is also good for the final output of the control system, for all the same reasons as above. However, that reduced dimensionality may need to be subsequently up-scaled if it is to control at the low-level scales. Thankfully there is well-established precedent for that in the form of U-Nets (citation) and in hierarchical predictive models (citations).

Most of the other "why"s mentioned here benefit hugely from using higher-order representations because it reduces the dimensionality of state spaces for: monitoring current internal state, monitoring external feedback, learning associations. Additionally, where those higher-order representations are inferences over the latent states, then they unify multiple sources of information (different sensory modalities, information presented over time).

## Objective learning

(TODO, introduce this in an abstract agent way first, then use biology as an example)

Learning higher-order objectives from sparse RL feedback and associating them to hard-wired basic needs. eg: how does a meaningless inedible coloured token translate to basic life preservation objectives?

Biological organisms clearly are born pre-wired with some basic evolutionarily hard-wired seeking of basic needs, eg: basic life preservation, and seeking of food. They can even be hard-wired external behaviours that force certain sequences of muscle contractions (rooting behaviour in infants). But how does that translate into complex social interactions that change more rapidly than evolution can adapt to? The associations must be learned through experience. The dimensionality of the search space would be too vast if learning at the level of muscles. And the RL feedback is often sparse. Thus higher-order representations are necessary to drastically reduce the dimensionality of both the control space and the environment space.

## Mode selection

strategy / goal / context / module / attention selection

Exploration vs exploitation

Priors or other tuning mechanism. Even priors on meta mgtmt layers.

eg: adjusting priors, inhibition, excitation

## State trajectory control

Observing performance over time

Predicting future outcomes from current trajectory

Predicting expected future utility of current trajectory

Applying tuning control where current trajectory is sub-optimal.

eg: in my own first simulations I ran into a problem of stagnant state cycles (infinite loops)

## Distributed cooperation

Managing competition and cooperation between many sub-processes.

In humans, probably a first-order network concern because we don't consciously experience and control that process. But it is in principle possible to be done at either level.

eg: adjusting priors, inhibition, excitation

## Certainty measurement / reaction

Eg: low level simulations linking certainty encoding to attention.

Not sure how used for meta mgtmt, but has a plausible low level mechanism.

# Agents as Regulators

# Meta-management Architectures

# Conclusions

Bottom-up design stance has been taken. This offers opportunities to build systems based on these architectures, and to measure empirically their relative benefits for different problem domains.

# Regulation

An autonomous embodied agent, depending on its purpose, may need to control either its environment, itself, or both, towards some static or dynamically determined target. That agent can be described as a *regulator* of its target system.

For example, an agent that regulates its environment operates within a system containing environment state *senv* which changes with some ambient dynamics *denv(t)*. The agent must perform an action, *aenv*, against the environment in order to regulate itself towards some target. After an action has been executed the environment state outcome *oenv* is influenced by both *denv(t)* and *a*env. This can be summarized as the following equation:

According to the *good regulator theorem*, if the agent is to regulate the environment state it must be a "model of the system" (Conant & Ashby, 1970). Furthermore, we can say that the efficiency of the agent to regulate its environment depends on its accuracy in modeling the system. Errors in the accuracy of the model result in errors in the regulation of the system. In learning agents, those errors are used for subsequent training of the model.

An embodied agent with complex actions may require an additional level of regulation. For example, an animal must not only regulate its external environment, but also regulate its own physical state. This includes both maintaining homeostasis and controlling the efficiency and effectiveness of its actions against the environment. The agent performs action *abody* against its body with the intent to regulate the body towards efficiently achieving environment action *aenv* while satisfying its requirement for body homeostasis. Such an agent thus operates in a system that additionally has body state *sbody* with ambient dynamics *dbody(t)*. The agent performs action *abody* against its body, producing outcome *obody*, summarized as follows:

Agents that incorporate multi-step processing have a third kind of action: one that changes its internal data state without affecting its physical state. This system requires regulation for the same reasons as for environment and body, but such *non-physical* actions do not elicit any change to *sbody* or *senv*. Thus the agent must regulate its non-physical state *smind*, having ambient dynamics *dmind(t)*. The target state in this case is dynamically inferred based on its requirement for environment action *aenv*, body action *abody*, and possibly for some form of non-physical homeostasis of *smind*. In order to regulate towards that target it performs action *amind* producing outcome *omind*, summarized as follows:

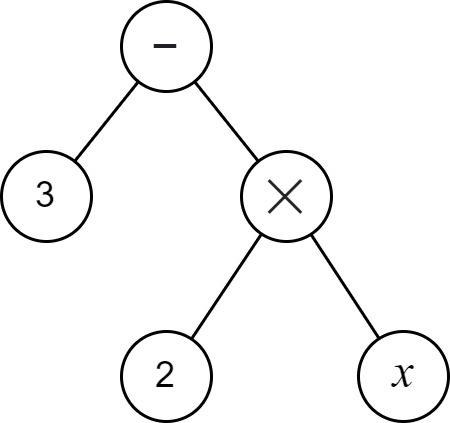
By way of example of the importance of such mind regulation, consider the case of fluent aphasia caused by damage to the Wernicke's area[[1]](#footnote-2) of the brain. Individuals with fluent aphasia can easily produce speech, but it is typically full of many meaningless words and often unnecessarily long winded. Wernicke's area is associated with language comprehension. In a neurotypical individual, the comprehension of their own vocalizations provides a corrective mechanism. This illustrates the importance of feedback in the regulation of one's own actions, and by way of analogy extends to the regulation of non-physical actions.

# Models

All of the systems described above are of the form . The production of the optimal action *a* for a given situation can be computed by a function, *f*, such that . In this way, function *f* becomes a *model* of the system in exactly the way meant by Conant and Ashbey. There are many different ways of constructing such a function, with implications on how much its inherent model can be introspected for purposes other than merely computing the next action.

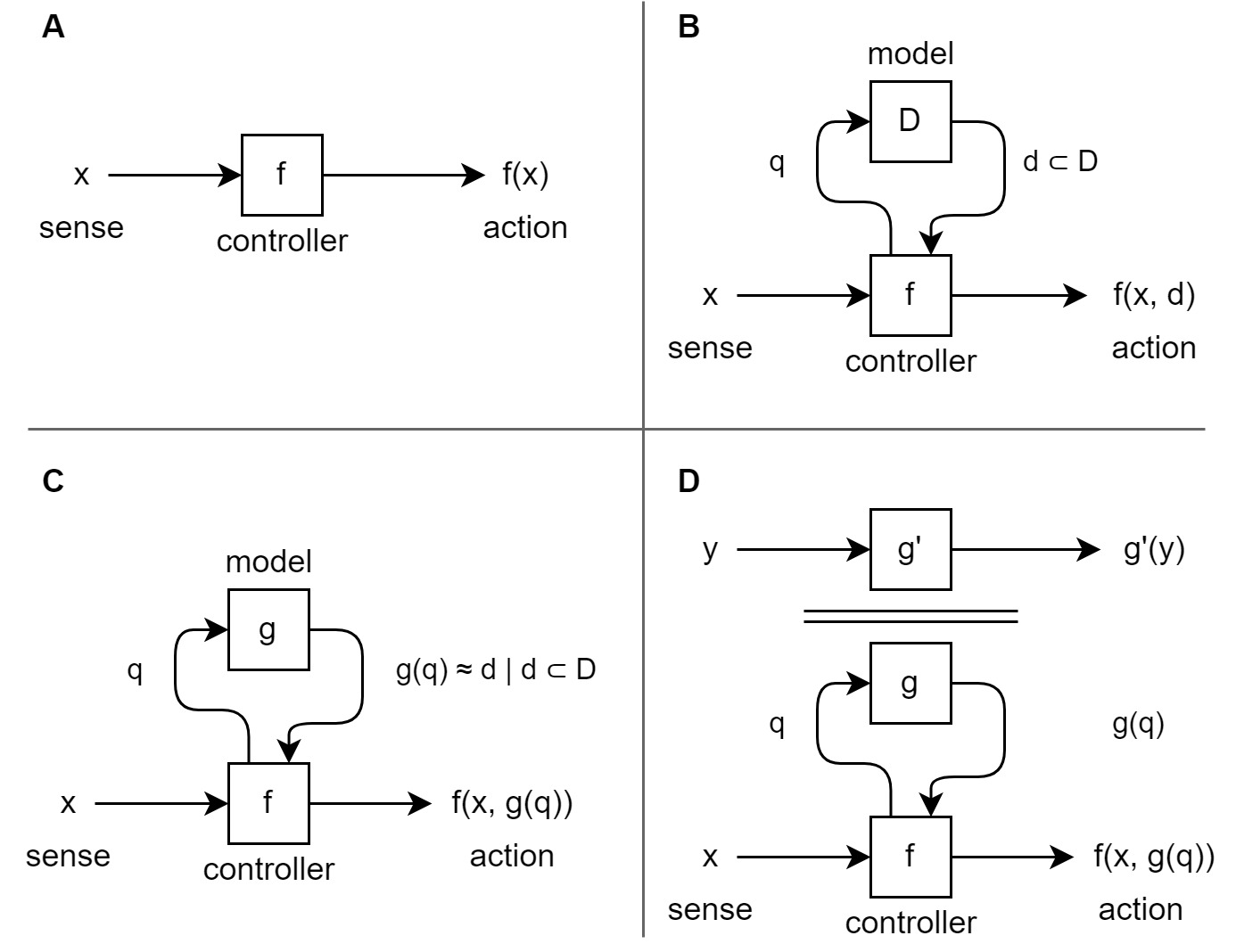
Consider the following function. This function is, for example, effective at predicting the action required to regulate towards a target state of 3 by doubling the input signal and comparing to that target state. However, an agent that merely uses this function to calculate actions cannot inspect anything about the function other than the actions it calculates for different inputs.

Alternatively consider Figure 2, which shows an abstract syntax tree[[2]](#footnote-3) (AST) of the function above, of the sort used by computer science to parse an expression within a software compiler. Instead of using the above function, a regulating agent could use this AST to calculate its next action and achieve the same outcome. However the AST is a more explicit model of the dynamics being regulated. The components of the original function are represented individually and thus they can be individually queried. So here the AST can be introspected and much more can be derived from it that may apply either to the system being modeled or to how the AST models that system.

  
Figure 2: Abstract syntax tree of 3 - 2x

To examine the introspective opportunities further, consider the task of constructing a set, *F*, that contains all beliefs that may be drawn from the model. In the case of the function, pairs of input and output action values are all that can be drawn from the model, ie: <0,3>, <2,-1>, <-3,9>, <-1,5>, etc. The AST supports the ability to draw those same pairs of input and action values. However the AST also supports that many other beliefs may be drawn from the model and added to *F*. For example that i) the target is 3, ii) input signal *x* is significant to the calculation, while *y* and *z* are not, and iii) the execution of the function depends on the operations of *subtraction* and *multiplication*.

So, it is clear that different architectures enable different levels of *introspection* of the underlying models. What about the case for neural networks? In the modern use of artificial neural networks (ANNs), it is commonplace to refer to ANNs as a *function approximator* (Goodfellow et al., 2016), and indeed many networks fall into the category of a function. For example, in *model-free* deep reinforcement learning (RL) an ANN is used to calculate either the next action or the expected value of all possible actions given the current state (Lazaridis, Fachantidis & Vlahavas, 2020). The architecture of the RL algorithm treats the ANN as a function without any introspective capabilities. See Figure 3(A) for an example. There is also *model-based* RL. One variant of model-based RL, illustrated in Figure 3(C), uses ANNs to predict the expected outcome of executing an action. The introspective ability here is the same as for model-free deep RL - the ANN is treated as a function. For the RL models mentioned so far, the set *F* of beliefs is of similar content: *F* is the set of <state,action> or <state,action,outcome> tuples. There do exist forms of model-based RL that use something more akin to the AST, usually where there is a known physics model that is represented mathematically, and which could potentially be used to introspect for more than just <state,action,outcome> tuples, such as is illustrated in Figure 3(B). However, a significant point to note here is that ANNs, and probably neural networks in general, do not lend themselves to introspection on their own.

  
Figure 3: Different architectures for *modeling regulatory actions against the environment.* In (**A)**, the controller determines the next action by executing function *f* against sense input *x*. The function may, for example, be an ANN that is trained through many iterations of an appropriate learning algorithm. Function *f* merely models the best next action without modeling any other aspects of that environment and thus cannot be used to introspect anything other than the next regulatory action. In (**B)**, set *D* holds an explicit model of the environment which can be arbitrarily queried (*q*) to gain insight about any aspect of the environment that is encapsulated within *D*. Controller function *f* uses that to determine the next action. In (**C**), set *D* is replaced by function *g(q)* which approximates queries against *D*. This architecture is commonly used in AI where the dynamics of the environment are too complex to determine a priori, and *g(q)* is built as a second ANN that is trained through exploration. In (**D**), the secondary system *y = g'(y)* models some aspect of the environment other than the next regulatory action. For example, it may observe and predict long term trends in the environment state. Potentially further additional modeling systems are required for each additional aspect of the environment that needs to be modeled.

For that reason, a third form of modeling system exists, whereby a secondary model predicts the behaviors of the former, such as is illustrated in Figure 3(D). The secondary model may, for example, be a second ANN that captures aspects of the same underlying system but at a more macro level, and it may be more suitable for integration with other data. This macro representation is at the core of the theory of Higher Order Thought Theory (HOTT) (Rosenthal, 1997 & 2006), and of recent theories based on it such as Hierarchical Active Inference (Giovanni et al, 2018) and Integrated World Modeling Theory (IWMT) (Safron, 2020).

# Schemas

The lack of introspective ability of a simple function contrasts with the introspective ability of a human. Psychology has long identified in humans the existence of a model of the individual's body – known as the *body schema*. A good definition is given by Morasso et al (2015):

*"In summary, we view the body schema as a set of fronto-parietal networks that integrate information originating from regions of the body and external space in a way, which is functionally relevant to specific actions performed by different body parts. As such, the body schema is a representation of the body’s spatial properties, including the length of limbs and limb segments, their arrangement, the configuration of the segments in space, and the shape of the body surface".*

So the body schema is a model used in production of action control by integrating information from our main physical senses and the proprioceptive senses (Proske & Gandevia, 2012). That model can also be introspected – for example, we can know where our hands and feet are without seeing them – and those introspections can become the topic of subsequent thought. But there are aspects of the model that cannot be introspected – for example, we have no observability of the arrangement of the sense nerves used to infer the hand and feet positions, or of the effector nerves used to actuate their muscles.

This paper hypothesizes the existence of a second schema, the *mind schema*, that performs an analogous role for the regulation of the mind and non-physical actions. Anecdotally, this seems highly plausible within humans given our introspective ability towards our own mind's capabilities. For example, we can know that we are good at focusing, but struggle with math, that we are more creative when background music is present, and that we need the support of tools to help remember people's names (eg: a notebook). The underlying notion here is that the mind schema helps us to control, monitor, predict, and rationalize about our mental structure and actions in the same way that our body schema does that for our physical structure and actions. It is the regulatory model for our non-physical actions. Additionally, just as for the body schema, there is a distinct delineation between what can be introspected and subsequently thought about, and what cannot.

The suggestion of a mind schema has also been made in the form of *Attention Schema Theory* (Graziano & Kastner, 2011; Webb & Graziano, 2015; Graziano, 2017), although the meaning there is perhaps narrower than what is proposed in this paper.

# Summary

todo

# Author Contributions

The author confirms being the sole contributor of this work and has approved it for publication.

# Funding

No funding was received.

# References

1. https://en.wikipedia.org/wiki/Wernicke%27s\_area [↑](#footnote-ref-2)
2. https://en.wikipedia.org/wiki/Abstract\_syntax\_tree [↑](#footnote-ref-3)