A proposal to achieve intelligence through a distributed self-supervising machine learning architecture

Liu, Tao
ntobby.liu@gmail.com

Niu, Fafu
Simon Fraser University
Burnaby, BC, Canada
fafun@sfu.ca

Abstract

This article proposes a progressive definition of intelligence, which combines a causal representation model with an intention mechanism. Guided by this definition, we design a novel distributed self-supervising machine learning structure, which is a graph network of networks (**NetNet**). The components of this structure, such as self-supervised capsules, distributed memory and attention modules, and distributed reward modules, are discussed in detail. Each capsule achieves self-supervision by playing a three-player minimax game, and a detailed derivation of this mechanism is given from the perspective of information theory. Although it is a preliminary design and cannot be tested right now, the biological plausibility of it exceeds all existing models. Many viewpoints given in this article may help achieve general artificial intelligence.

1 Introduction

At present, there is no ultimate definition for intelligence, and actually, there will never be. Because human thinking and language skills are regarded as part of the intelligence or at least derivatives of it, we cannot use a natural language to define intelligence in a self-consistent way before the intelligence itself is defined. This kind of "self-reference" puzzles exists in some profound philosophical scientific problems, such as trying to prove the consistency and completeness of mathematical axiom systems by meta-mathematics [1], and the collapse of quantum systems caused by observers who themselves are quantum systems [2].

Even so, there are some preliminary definitions already. For example, Legg and Hutter [3] claimed that "Intelligence measures an agent's ability to achieve goals in a wide range of environments." In their statement, the intelligence should have three elements, namely "environment," "agent," and "goal." Naturally, a definition without those three elements is meaningless. However, this definition is far from satisfactory. A good definition should be meticulous and instructive, directing the designing of agents thinking like people. So that the definition can neither be too general, which will be meaningless, nor be too specific, which may fall into divergent paths that cannot lead to the end.

Meanwhile, artificial intelligence is now widely used in academia and industry because of the significant progress in deep learning. However, these deep learning frameworks are fundamentally different from human beings [4] in the way of thinking because they are intrinsically statistical-based pattern recognition models. Recently, many novel structures and ideas have been proposed, such as Capsule Network [5, 6], Graph Neural Network [7], Attention Mechanism [8], Generative Adversarial Nets [9]. Some researchers have also investigated in the integration of deep learning and neuroscience, like Adam H. Marblestone [10] hypothesized that "Cost functions are diverse across areas and change over development."

Preprint. Under review.

In this work, we first give a progressive definition of intelligence by using a term commonly employed in artificial intelligence, namely representation. Guided by this definition, we propose a novel distributed self-supervised machine learning architecture that integrates some ideas behind recent advances mentioned above. This article is structured as follows: In Section 2, we present a definition of intelligence as our guiding ideology. In Section 3, we introduce our architecture by elaborating on each component of it, including the self-supervised capsules, distributed memories and attentions, and distributed rewards. In Section 4 we have a brief discussion, and finally, Section 5 is the summary.

2 The Definition of Intelligence

As mentioned above that a self-consistent definition of intelligence is impossible, here we use an alternative way by describing the ability of an intelligent agent. The characteristic of this definition is that it aligns all perceptions, actions, goals, and intentions into the evolution of representations in the subsystem U, where a causal representation model of all these forms. We divide the agent into 4 stages from a low level to a high level. An agent having intelligence should meet:

1. The agent can perceive from the environment and itself, and take actions to change the environment and itself.
2. The agent has a subsystem U that is stateful, evolving, relatively independent and self-driven. The states (or sub-states) of U are representations. The agent and U are interactive, and U can generate perception representations and action representations through evolution. The agent's perceptions will cause corresponding perception representations in U, and action representations in U will trigger the agent's corresponding actions, as shown in Fig. 1.
3. (a) U can generate goal representations through evolution. The goal representation is a kind of perception representations. The agent has some specific goals (such as energy intake), each of which triggers a corresponding goal representation in U.
   (b) U runs according to its own dynamics with many trainable and adjustable factors. Through learning, U builds an evolving representation model of the external environment, the agent, and itself.
   (c) U can generate intention representations through evolution. When the agent achieves a goal, U will adjust relevant factors to have "preference" for the representation relating to the goal, consequently enabling the agent to have a preference for this goal. These preferred representations are called intentions.
   (d) Intentions are transitive. Representations that are causally related to existing intentions will be identified. Then U generates preferences for these representations to convert them to new intentions.
4. Theoretically, motivated by intentions, the agent can provide an arbitrarily extensive and detailed causal model of the environment and itself that human beings can understand.

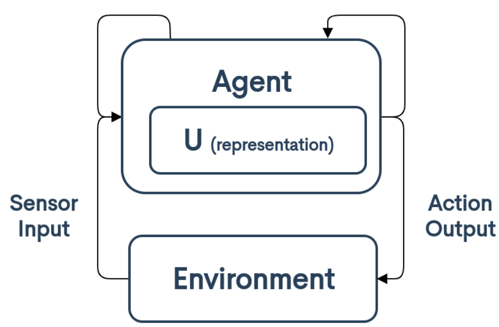

Figure 1: An agent interacts with its environment.

It needs to point out that the real situation is not as clear and organized as the above paradigm suggests. Representations are distributed, vague and evolving, so that it may not be able to point out what representations really are, nor to tell whether a representation is an intention or not. As Brooks[11] pointed out, "Out of the local chaos of their interactions there emerges, in the eye of an observer, a coherent pattern of behavior." We use this term only because we regard it as a disposable scaffold in the process of building intelligence.

We think that the agent in stage 2 has the ability of unconditioned reflex and the agent in stage 3 has the ability of conditioned reflex and limited causal reasoning. Human beings ourselves are in stage 4. The essential difference between human beings and other creatures is the ability to compress and transfer knowledge using a symbolic system. The general artificial intelligence(GAI) that we are pursuing is an agent based on electronic computers working in stage 4. Because electronic computers work on switches of transistors, we can understand the operation logic of any systems running in it as long as we are patient enough. Any extensive and accurate models the agent built are within the range that human beings can understand. Intrinsically the tool we used in defining intelligence is to draw an analogy between the agent and human.

3 The Distributed ML Architecture

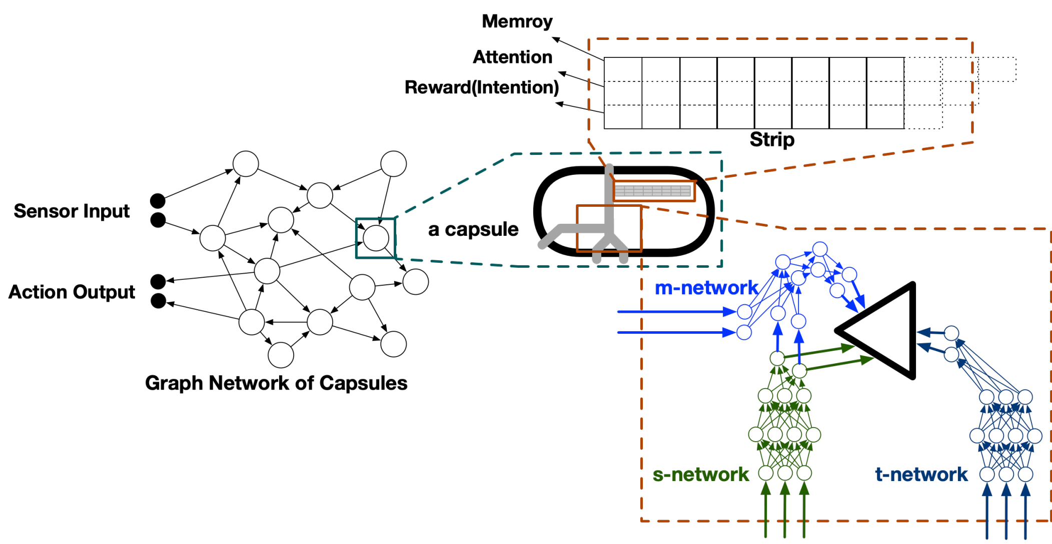

Figure 2: Distributed self-supervising network architecture

Guided by the definition of intelligence above, we propose a distributed machine learning architecture, which is a network of networks (**NetNet**). As shown in Fig. 2.

Note that there are three points totally different from most traditional deep learning networks. First, the inputs from sensors are raw, interactive and continuous, rather than samples selected from a prepared data set. Second, the topology type of the graph is a directed cyclic graph. There is no final "exit" for the flow of information in it and it will run endlessly. Third, the action output nodes do not output data like vocabularies or values but control actions that can transmit information, such as controlling manipulators to write words.

Although **NetNet** can be regarded as a graph neural network, the nodes in the graph (except for the input and output nodes) are self-supervising capsules. Each capsule contains three inner networks and a strip. The strip in the capsule is responsible for memory, attention, and intention. The three inner networks are our highlights, namely **s**-network, **m**-network and **t**-network. They three are combined to form a **Predictive Adversarial Network** (PAN) to be introduced below. The output of the capsule, which represents information encoding representations, is from the output of s-network or memories. The capsules are connected with each other to form a directed graph and also connected with sensor input nodes and action output nodes. We will elaborate on these in the following subsections.

3.1 Networks inside Capsules

3.1.1 Theory

As disucssed in Section 2, **NetNet** is to be is an extensive and detailed causal representation model of the environment and itself.

Firstly, it is a representation model, which means it maps inputs to representations, as shown in Eq. 1:

$$R(inputs(t'))(t)|_{0\le t'\le t} \tag{1}$$

where $R(t)$ is representations and is the functional of $inputs(t)$.

We use $H(X)$ to denote the information entropy of X, and so the information entropy of representations $R(t)$ is denoted as:

$$H(R(t)) \tag{2}$$

H_t is used to denote the joint entropy from 0 to t, as shown:

$$H_t = H(R(0), R(1), \cdots, R(t)) \tag{3}$$

The model is built to be arbitrarily **extensive** and **detailed**, so that **NetNet**'s goal is to optimize:

$$\underset{R}{max}\, H_\infty \tag{4}$$

The analytical form of H is not available, and even these notations are inappropriate because R is not in probability distributions. A simple strategy can be adopted to evaluate the probability distribution of R, which is to build two networks with the first one producing R and the second one producing $\mathcal{R}$. Then we use $\|R-\mathcal{R}\|^2/2$ to estimate the variance of R. But different from the conventional understanding, we argue that $H(R(t))$ contains zero useful information. **NetNet** does not use end-to-end supervised learning, so that R's exact numeric value is meaningless. In the following, we use $\mathcal{H}$ to denote the "renormalized" information entropy: $\mathcal{H}(R(t)) = 0$. And so Eq. 4 is rewritten to:

$$\underset{R}{max}\, \mathcal{H}_\infty \tag{5}$$

In the following, $\mathcal{H}_{t_0,t_1}$ is used to denote:

$$\mathcal{H}_{t_0,t_1} = \mathcal{H}_{t_1} - \mathcal{H}_{t_0} \tag{6}$$

, so that $\mathcal{H}_\infty = \frac{1}{\Delta t}\sum_{t=0}^{\infty} \mathcal{H}_{t,t+\Delta t}$ where Δt is small time intervals.

We assume that Eq. 5 can be approximated by the following equation:

$$\underset{R}{max}\, \mathcal{H}_\infty \doteq \frac{1}{\Delta t}\sum_{t=0}^{\infty} \underset{R}{max}\, \mathcal{H}_{t,t+\Delta t} \tag{7}$$

Because $R(t)$ is continuous and Δt is small, we assume that $\mathcal{H}_{t,t+\Delta t} \le 1(bit)$ at any time t. More specifically, if $R(t+\Delta t)$ is predictable from $\{R(0), R(1), \cdots, R(t)\}$, $\mathcal{H}_{t_1,t+\Delta t} = 0(bit)$, or else $\mathcal{H}_{t,t+\Delta t} = 1(bit)$. This is derived from information theory according to which data without uncertainty brings zero information. We would build a new network producing $R'(t)$ to predict $R(t)$ from $\{R(0), R(1), \cdots, R(t-\Delta t)\}$:

$$R'(R(t'))(t)|_{0\le t'\le t-\Delta t} \tag{8}$$

Without losing generality, we choose an analytical form for $\mathcal{H}_{t,t+\Delta t}$:

$$\mathcal{H}_{t-\Delta t,t} = 1 - e^{-\frac{\|R(t)-R'(t)\|^2}{\|R(t)-\mathcal{R}(t)\|^2}} \tag{9}$$

Secondly, **NetNet** is a casual model, which means that it can predict its representations before inputs come.

Eq. 1 can also be written as:

$$R(t) = R(inputs(t'), inputs(t''))(t)|_{0 \leq t' \leq t-\Delta t', t-\Delta t' < t'' \leq t} \tag{10}$$

We would build a new network producing $R''(t)$ to predict $R(t)$ in advance of $\Delta t'$:

$$R''(inputs(t'))(t)|_{0 \leq t' \leq t-\Delta t'} \tag{11}$$

$I(X;Y)$ is used to denote the mutual information of X and Y. Because of the information bottleneck in $R(t)$, if $\Delta t' \approx \Delta t$, we can get:

$$I(R(t); R''(t)) > I(R(t); R'(t)) \tag{12}$$

, which means that $R''(t)$ can predict newly formed information in $R(t)$. The network producing R'' can substitute the one producing $\mathcal{R}$. So that Eq. 9 can be rewritten as:

$$\mathcal{H}_{t-\Delta t,t} = 1 - e^{-\frac{\|R(t)-R'(t)\|^2}{\|R(t)-R''(t)\|^2}} \tag{13}$$

This substitution is necessary to avoid a trivial solution, i.e. the two networks producing R and $\mathcal{R}$ become the very image of each other.

Both $R'(t)$ and $R''(t)$ are predictions of $R(t)$, but they have opposite impact on $\mathcal{H}_{t-\Delta t,t}$, so that **NetNet**'s goal in Eq. 7 is parsed into:

$$\underset{R,R''}{min}\ \underset{R'(R)}{max}\ e^{-\frac{\|R(t)-R'(t)\|^2}{\|R(t)-R''(t)\|^2}} \tag{14}$$

We name this as **Predictive Adversarial Value Function** and it is the core idea of this article.

R is a high-dimensional vector, and we assume that most of its components are unrelated with each other and so can encode information independently, like

$$\begin{aligned} R &= R^{(1)} \otimes R^{(2)} \otimes \cdots \otimes R^{(n)} \\ R' &= R'^{(1)} \otimes R'^{(2)} \otimes \cdots \otimes R'^{(n)} \\ R'' &= R''^{(1)} \otimes R''^{(2)} \otimes \cdots \otimes R''^{(n)} \end{aligned} \tag{15}$$

, and we can get:

$$\mathcal{H} = \mathcal{H}^{(1)} + \mathcal{H}^{(2)} + \cdots + \mathcal{H}^{(n)} \tag{16}$$

, where $(1), (2), \cdots, (n)$ can denote different component subsets.

3.1.2 Predictive Adversarial Network

The main body of **NetNet** is a directed graph of capsules. Capsules output representations and each of them outputs a different component subset indicated by Eq. 15. As seen in the bottom-right inset graph of Fig. 2, each capsule contains three sub-networks, namely **s**-network, **m**-network and **t**-network. The **s**-network accepts inputs from outputs of other nodes, and outputs one subset of R indicated by Eq. 15. The **m**-network accepts inputs from outputs of the **s**-network and other capsules, and outputs the corresponding subset of R'. The **t**-network accepts inputs from outputs of other nodes and outputs the corresponding subset of R''. In a capsule, we use S_t, M_t, and T_t to donate the vectors outputted by them three respectively at every clock-cycle[1] t.

The **s**-network, **m**-network and **t**-network can be neural networks of any structure, including feed-forward neural networks, recurrent neural networks, etc. There will be a preset time lag, for example

[1]A clock-cycle is the calculation time from input to output for each internal network, no matter how deep it is. In the eyes of the external network, all capsules complete one round of calculation in one clock-cycle.

16 clock-cycle, in **m**-network and **t**-network, which is required by Eqs. 8 and 11. Meanwhile the structure of **s**-network is restricted to prevent **s**-network being the very image of **t**-network.

Derived from Eq. 14, the capsule will play the following three-player minimax game with value function $V_t(S_t, M_t, T_t)$:

$$\min_{S,T} \max_{M(S)} V_t(S_t, M_t, T_t) = e^{-\frac{\|S_t - M_t\|^2}{\|S_t - T_t\|^2}} \tag{17}$$

3.1.3 Capsule Activation Function

In traditional predictive coding [12] approach, neural networks learn the statistical regularities of the natural world, signaling deviations from such regularities to higher processing centers. This reduces redundancy by removing the predictable, and hence redundant, components of the input signal. **NetNet** adopts an variation of this approach. The value function V_t is a measure of whether the capsule is working "productively". If it is, it will output the output of **s**-network. We name this activation. We recommend that the activation function is a multiplication of S_t and a value of binomial distribution, as shown:

$$activation(t) = S_t * x, and\ x \in \{0, 1\},\ Pr(x = 1, t) = 1 - sigmoid(\alpha V_t) \tag{18}$$

where α is a scale factor. The activation possibility depends negatively on V_t. When $x = 1$, the capsule is activated. If the capsule is activated, its output is S_t, or else the output is 0 (the output may be other values from memories, as shown in Eq. 20). $Pr(x = 1) \leq 0.5$ is chosen for dropout.

Note that this activation function implicitly prevents the capsule from outputting easily predictable values, such as constant or periodic values.

3.1.4 Distributed Optimization

Because each capsule has its own value function, in the view of the external network, the optimization is distributed and independent. But we suppose the optimization of one capsule may affect its upstream neighbors. Therefore, a hyper-parameter called $coverage$ (CV) is introduced to measure how far a capsule can influence its neighbors, as illustrated in Fig. 3.

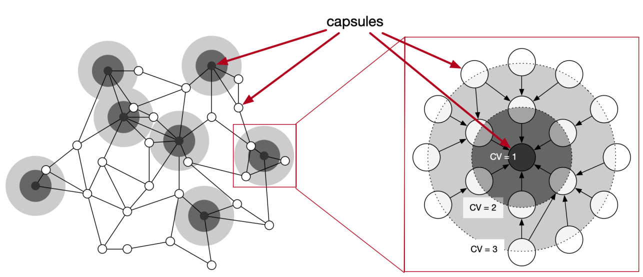

Figure 3: Distributed optimization (left), optimization of one capsule with different $coverage$(CV) (right)

3.2 Distributed memory and attention

Attention "can be used to describe the processes involved in selecting the information to be processed and stored in memory" [13]. The performance of attention-based networks[8] has surpassed that of recurrent networks for the former maintains representations that scale with the size of the source.

An attention function is to map a query and a set of key-value pairs to an output. In **NetNet**, memories are stored as records on the strip in each capsule, where each memory record contains a value and a key, as shown in Fig. 4[A]. Values are historical outputs of the capsule when activated, keys are historical outputs (attention vectors) of a global attention network, and the query is the attention vector at that clock cycle.

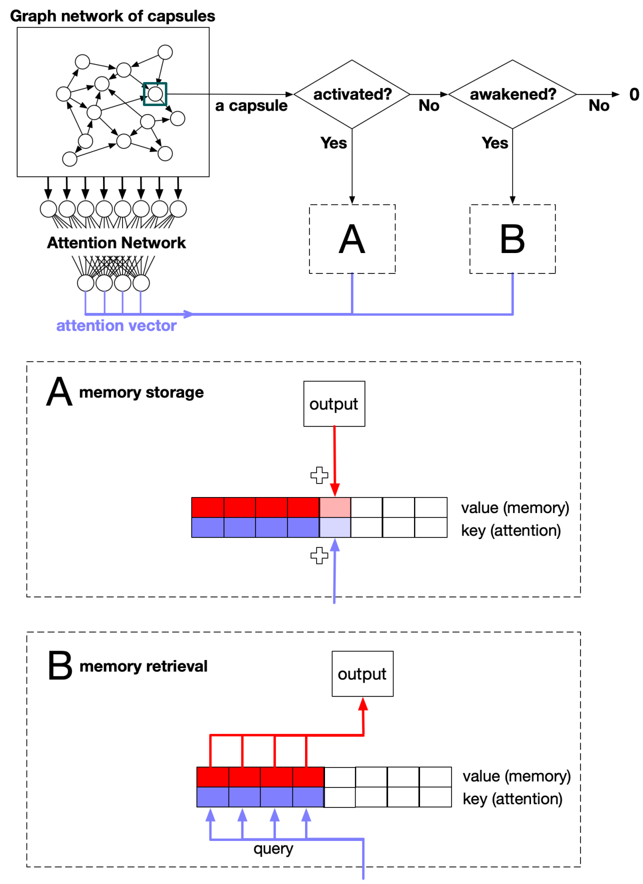

Figure 4: Memory storage and retrieval

3.2.1 Memory Storage and Retrieval

The global attention network is to embed highest-dimensional inputs, i.e. outputs of all capsules, into a low-dimensional attention vector. In **NetNet**, the calculation is real-time, which means that the attention network generates a global attention vector in every clock cycle.

As shown in Fig. 4, if a capsule is activated in any clock cycle, its output (as the value) and the global attention vector at that time (as the key) will be recorded by appending them to the strip inside the capsule. If the capsule is not activated at that clock cycle, controlled by the awakening function described in the next section, it will try to retrieve memories from its inside strip.

Capsule Awakening Function The **Awakening Function** determines when and how to output the retrieved memory. At clock cycle t, we use v_{it} to denote the i-th value on the strip, W_{it} to denote the weight (before applying softmax) assigned to it, $W\_sum_t = \sum_i W_{it}$ to denote the sum of weights assigned to each value, $v_t = \frac{\sum_i (e^{W_{it}} * v_{it})}{\sum_i e^{W_{it}}}$ to denote the weighted sum of the values, and $awaken(t)$ to denote the final outcome of memory retrieval.

$$awaken(t) = v_t * y, and\ y \in \{0, 1\}, Pr(y = 1, t) = sigmoid(\beta * W\_sum_t) - 1/2 \tag{19}$$

where β is a scale factor. If $y = 1$, it means that this capsule is awakened.

The final output of a capsule is either from activation (see Eq. 18) or from memory (see Eq. 19).

$$output(t) = activation(t) + (1 - x) * awaken(t) \tag{20}$$

where x was defined in Eq. 18 indicating whether the capsule is activated.

3.2.2 Optimization

The attention network would be optimized in the same process as introduced in Section 3.1.4 with a $coverage > 1$.

3.2.3 Flow of Attention

Attention vector can be regarded as a context where all outputs of all capsules are compressed. By recording the attention vector, each memory record retains its temporal context. Ideally, a complete context could be evoked by several clue memories recording it.

Considering that human's attention can flow in a direction, we need to give an "impetus" in the attention mechanism. Here, we provide two feasible methods.

Periodic Space-time Encoding A periodic space-time encoding information can be integrated into the attention vector, inspired by the theta phase precession in hippocampal neuronal populations [14].

Anti-attention As the opposite of attention, anti-attention [2] is to decide which values should receive **less** attention. Historical attentions will be used as anti-attentions to push forward the attention by time, as shown in Fig. 5. Its side effect is to make attention more focused.

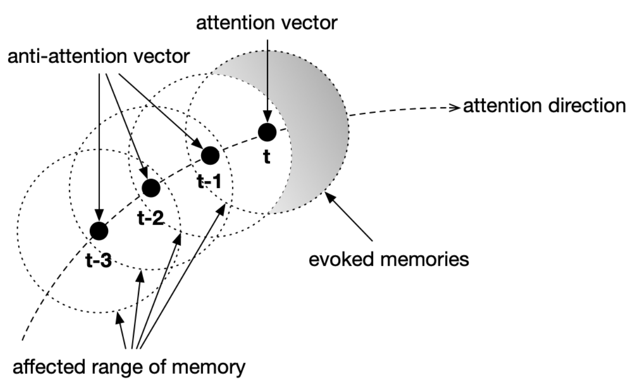

Figure 5: Anti-attention. The figure represents an attention space. Black solid dots are attention vectors at different clock cycles. The range of a dashed circle represents the attention vectors corresponding to the memories that may be evoked by the attention vector at the center of the circle. Only new (crescent) memories can be evoked due to anti-attention.

3.3 Distributed reward and reinforcement learning

Reinforcement learning plays an indispensable role in **NetNet**. In our plan, reinforcement learning is responsible for converting representations to intentions. Without intentions, the system would do nothing actively.

Deep-Q learning is the most popular method nowadays and has reached the human level in some computer games [15]. But our strategy is totally different from it and instead inspired by its preceding biological experiments [16]. We believe that memories shape habits and character, so that if we make some memories (and attentions) to be more easily evoked, we are converting those relative representations to new intentions.

The reward is a scalar[3] to measure the "happiness" of **NetNet**. As shown in the top-right inset graph of Fig .2, the memory strip has 3 sub-strips. The first two are responsible for memory and attention at each clock cycle as discussed above, and the third strip is responsible for recording the reward at

[2]We suggest that anti-attention may improve the performance of other architectures based on attention mechanism, while this is not the subject of this article.

[3]Vector is also a choice. Its direction indicates different types of rewards, and its norm indicates the degree of happiness.

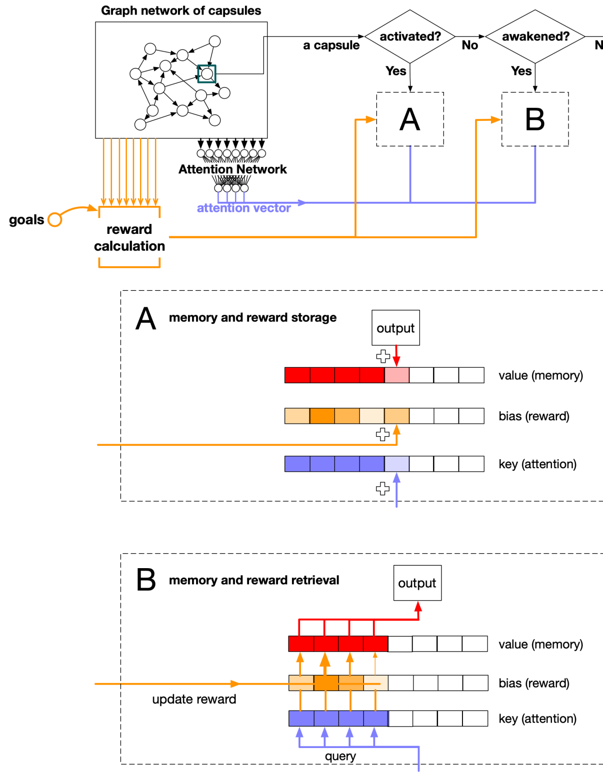

Figure 6: Storage and retrieval of memory and reward. For convenience of demonstration, the third sub-strip recording the reward was drawn in the middle of the other two sub-strips.

each clock cycle. Rewards accompany memories in storage and retrieval as shown in Fig. 6. In the memory retrieval process, rewards will bias the weights assigned to different memories. Actually, memories with higher rewards would receive higher weights, thus attracting more attentions. At clock cycle t, we use R_{it} to denote the i-th reward on the strip, the meaning of other symbols is the same as that of the symbols in Section 3.2.1. Now the weighted sum of the values is:

$$v_t = \frac{\sum_i (e^{W_{it}+R_{it}} * v_{it})}{\sum_i e^{W_{it}+R_{it}}} \tag{21}$$

The remaining problem is how to calculate the reward. Most of the time, the reward is the weighted average of rewards of all evoked memories from all capsules. But when the system achieved or missed some goals, there might be a fluctuation in the reward level.

3.3.1 Goals

It is essential to figure out what goals are. Traditionally, motivations can be divided into intrinsic motivation and extrinsic motivation [17], so we assume that goals can be divided in the same way.

Extrinsic Goals Such as energy intake, successful mating, and also to achieve manually set goals for artificial intelligence.

Intrinsic Goal Intrinsic goal is that things work as expected. This can be quantified by the statistics of the population of activated capsules. Considering activation possibility is controlled by value function in Eq. 17, for the system, more activation means more efficient perceptions of the environment. From the perspective of capsules, more activation means better predictions by **t**-networks and poorer predictions by **m**-networks.

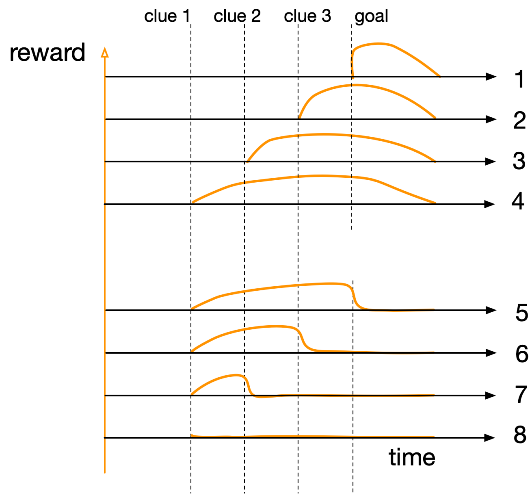

Figure 7: Reward propagation. The curves on 1,2,3,4 axes show how a rising edge of the reward propagates to leading memories when frequently achieving a goal, and the curves on 5,6,7,8 axes show how a falling edge propagates to leading memories when frequently missing a goal.

3.3.2 Reward Propagation

As we mentioned above, most of the time, the reward is determined by the evoked memories. Let's assume that the reward is 0 then. When the system achieved a goal, the reward jumps to 1. In this case, the difference was +1. This difference would update and raise the rewards of memories evoked, so that next time before the goal was achieved, some pre-evoked memories would raise the reward in advance.

Assuming that next time before the goal was achieved, the reward had been raised to 1 in advance by some pre-evoked memories, but after a while the goal was missed. Then when the attention changed, there was no goal to protect the reward from falling to 0. In this case, the difference was -1. This difference would update and reduce the rewards of memories evoked before.

The rising and falling of reward propagating back to memories are illustrated in Fig. 7. Note that only abrupt rises and falls of reward are allowed for propagation.

3.4 Connection between capsules

This section will cover the topology of **NetNet**, which is a graph network. Different from the conventional graph networks which are intended to represent structure information [7], **NetNet** is intended to represent unstructured information. The graph in **NetNet** is designed as a directed cyclic graph. In **NetNet**, new information is continuously inputted to the net through sensor input nodes, and then it is processed endlessly within the net like waves trapped in a swimming pool.

Each capsule has a limited number of input edges because each input edge should be connected to one input node of one of the three inner networks. Although the inner network may be substituted through distillation, the count is still limited. So that a fully connected network is not acceptable in most cases. We assume that, for small-scale systems, a randomly connected sparse network is competent.

Corticalization For large-scale systems, there are so many capsules that the connection between capsules would become too sparse. In this case, inspired by the brain cortex, we assume that all capsules are placed on one plane like cortex, and closer capsules will have a denser connection density. In order to transmit information over a cross-domain distance, we envisage a fiber transmission network inspired by white matter of the brain, as shown in Fig. 8.

Hebbian Theory Hebbian theory[18] is a neuroscientific theory claiming that an increase in synaptic efficacy arises from a presynaptic cell's repeated and persistent stimulation of a postsynaptic

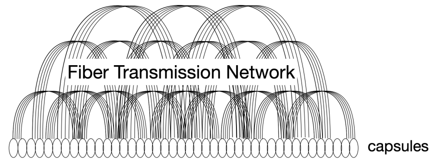

Figure 8: A 2D slice of 3D distribution of the fiber transmission network and capsules.

cell. Inspired by that, we assume that two capsules may create a new connection if their time patterns of activation or awakening are compatible.

In **NetNet**, the inner networks are enclosed and isolated in capsules. But this limitation can be broken. The outputs of hidden nodes of the network inside one capsule may be sent to some nodes of adjacent capsules' inner networks. However, this mechanism will increase the computational complexity.

3.5 Tasks to be solved

For now, several crucial tasks remain unsolved for building a true general artificial intelligence (GAI) system.

Attention Control Considering us human beings, we can turn our attention to a completely irrelevant direction, but **NetNet** can not handle its attention freely. In other words, it lacks the ability to control attention and anti-attention. We must say that this is a key skill. If **NetNet** has it, complex intentions can be formed, experience can be used efficiently through repeated recollections, and causality in the model can prevail over relevance. In our opinion, it is also helpful for long-term memory formation. This ability should be from a combination of nature and nurture. We suspect that the attention vectors generated and stored under some special conditions will attract future attention to jump here like anchors and magnets.

Long-term Memory As is described above, the capsule's strip is responsible for memory. There is a need for a mechanism[4] to convert short-term memories into long-term memories. Long-term memory is from the extraction and compression of short-term memories, but this compression process should not be isolated in each capsule. We speculate that long-term memories are formed with frequent or specific retrieval of short-term memories, and long-term memories are stored and retrieved in a similar way as short-term memories, as shown in Fig. 9.

Action Control In natural creatures, rate coding is used to control muscle contraction. In **NetNet**, how to process the output information of the action output nodes has not been solved. A subsystem responsible for action control might be needed. Action control is also from a combination of nature and nurture. **NetNet** would form a casual representation model on its actions and outcome of its actions, so it needs to perceive the outcome of its actions to form a closed loop involving the environment. Also, additional information, such as observing the behavior of peers, can help form this causal model.

Hyper-control Some hyper-parameters such as the scale factor α and β in Eqs. 18 and 19 may be controlled by the system itself. These kinds of control can be classified as hyper-control. Even the control of attention may be regarded as a kind of hyper-control. Some data, such as the population of activated and awakened capsules, the reward, and the attention vector are the outcome of hyper-control and will be sent back to specific input nodes. The system controls its hyper-parameters in a similar way as controls action.

[4] A forgetting mechanism is also needed, but there are many easy solutions

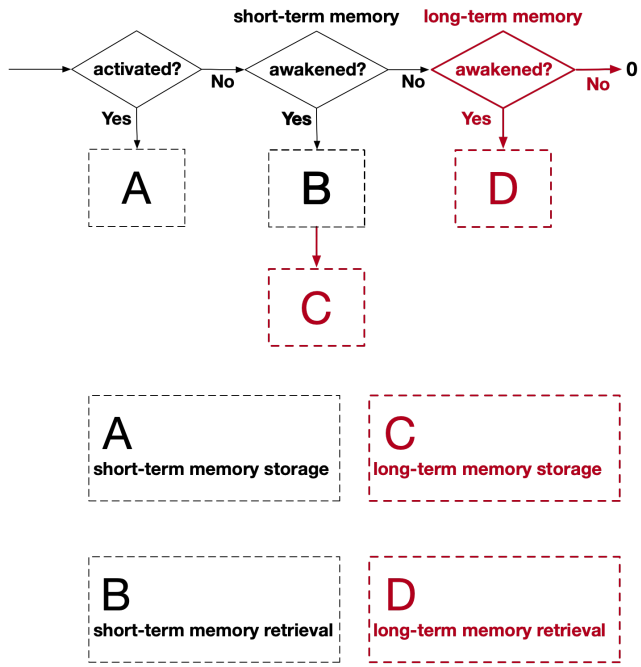

Figure 9: Long-term memory storage and retrieval.

4 Discussion

4.1 Discussion on biological plausibility

Biological plausibility is a guideline to achieve general artificial intelligence. Many counterparts of **NetNet** structures can be found in natural creatures. The inner networks are the counterpart of the neuronal circuits of the neocortex [19]. The fiber transmission network is the counterpart of white matter. The attention network is the counterpart of hippocampus, because both of them map information from a high-dimension form to a low-dimension form. The reward calculation is the counterpart of the function of dopamine neurons.

Some researchers believe that the brain is operating at a critical point [20, 21]. In **NetNet**, the activation or awakening of a capsule would provide new information into the network, so that it may trigger other capsules' activation or awakening and this is a proliferation mechanism. When activated, the capsule's activation function implicitly prevents it from outputting easily predictable values, and so this is an inhibition mechanism. These two mechanisms are conditions for maintaining the critical state.

Although CNN with pooling has been hugely successful in the field of deep learning, Geoffrey Hinton claimed that "The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster". But what we want to add is that the convolutional operation is also a mistake. Convolution is a strong inductive bias that strongly conflicts with the flexibility of the brain cortex. CNN receives static pictures while we make saccades across a picture. Brain areas like V1 are heavily affected by feedback from higher levels. There are many conflicts between CNN and biological plausibility. However, **NetNet** can provide perfect explanations for those phenomenons. In **NetNet**, the feedback from a higher level is to provide predictive information to the **t**- and **m**-networks of capsules. Only with a saccade can a capsule receive efficient predictive information from other capsules detecting the same features on a different receptive field.

5 Summary

This article proposes a self-supervising machine learning architecture which is actually a two-step model. The first step is to construct a causal representation model, and the second step is to promote intentions for it to complete tasks. Because outputs of the agent are directed by its own intentions, it is unknown whether a wrong result comes from a wrong model or a wrong intention. At the same time, the output is not clean data that can be supervised directly but is action control signals sparsely coding information. In these situations, the end-to-end supervised learning is not applicable anymore. As an alternative, this article adopts a strategy of combining distributed optimization and reinforcement learning. Compared with the huge potential benefits, the increased complexity in training is worthwhile. What we keep in mind is that an intelligent agent should be obtained in a way of education rather than training.

Distributed self-supervision is the biggest innovation of this article, which is not found in previous works. This is the first hypothesis that can support many phenomena found in the biological brain. We also give an explanation from the perspective of information theory. With this mechanism as the core, we have designed the modules of attention, memory, and reward. Although there are still some key tasks to complete, the great advantage of this structure in biological plausibility encourages us to believe that a real general artificial intelligence would be built on it or on its variants.

References

[1] Kurt Gödel. *On formally undecidable propositions of Principia Mathematica and related systems*. Courier Corporation, 1992.

[2] John Neumann. *Mathematical foundations of quantum mechanics*. Princeton university press, 1955.

[3] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and machines*, 17(4):391–444, 2007.

[4] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[5] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[6] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[7] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Adam H Marblestone, Greg Wayne, and Konrad P Kording. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:94, 2016.

[11] Rodney A Brooks. Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159, 1991.

[12] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79, 1999.

[13] Peter Robinson. Attention, memory, and the "noticing" hypothesis. *Language learning*, 45(2):283–331, 1995.

[14] György Buzsáki and Edvard I Moser. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, 16(2):130, 2013.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[16] Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[17] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[18] DO Hebb. Organization of behavior. new york: Wiley. 1949.

[19] Rodney J Douglas and Kevan AC Martin. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27:419–451, 2004.

[20] Per Bak. *How nature works: the science of self-organized criticality*. Springer Science & Business Media, 2013.

[21] John M Beggs and Dietmar Plenz. Neuronal avalanches in neocortical circuits. *Journal of neuroscience*, 23(35):11167–11177, 2003.