

Symbolic Neural Dynamics Allow for Modeling Retrograde Amnesia as Well as False Memories



Pierre Bonzon

Abstract Symbolic neural dynamics abstracting the functionalities of synaptic plasticity has been proposed as a new approach to modeling brain cognitive capabilities and used to define the basic mechanisms of an associative memory. This formalism is extended here to reproduce optogenetic manipulations, thus defining a computational model of memory engrams. It is illustrated through simulations of reversible retrograde amnesia and false memories of contextual fear conditioning that reproduce the behavioral schedules of actual experiments. These results support the hypothesis that separate processes are involved in long-term memory, i.e., the retention of specific patterns of connectivity between engram cells required for the storage of information, on the one hand, and the synaptic strengthening needed for its consolidation and retrieval, on the other. Defined by a logic program, this simulation platform could be used to design and predict the results of experiments involving inhibitory/excitatory loops formed between various brain regions.

1 Introduction

The advent of optogenetic technology (Deisseroth et al., 2006) has opened new doors toward the investigation of the brain. It is now possible to trace down simple cognitive processes down to the activation of a group of neuronal cells. Following the seminal work of Tonegawa and colleagues (Liu et al., 2012; Ramirez et al., 2013; Ryan et al., 2015; Roy et al., 2017), new ideas about the formation and use of long-term memory are emerging (Tonegawa, 2015; Trettenbrein, 2016). Briefly, the hypothesis is that memory storage and retrieval involve two different circuits and mechanisms, i.e., the retention of specific patterns of connectivity between engram cells required for the storage of information, on the one hand, and the synaptic strengthening needed for its consolidation and retrieval, on the other. These hypotheses are supported by the observation that various expressions of memory can be obtained by leaving

P. Bonzon (✉)

Department of Information Systems, Faculty of HEC, University of Lausanne, Lausanne, Switzerland

e-mail: pierre.bonzon@unil.ch

© Springer Nature Singapore Pte Ltd. 2022

A. Lintas et al. (eds.), *Advances in Cognitive Neurodynamics (VII)*, Advances in Cognitive Neurodynamics,

https://doi.org/10.1007/978-981-16-0317-4_5

connectivity patterns untouched and by acting on synaptic strengths only. These various expressions can be controlled, observed and measured through optogenetic manipulations, which in turn allow for the experimental induction of *retrospective amnesia*, the direct activation of *memory engrams* and the creation of false memories. Silent memory engrams (Roy et al., 2017), defined as memory traces whose access can be temporarily blocked and then restored at will, stand as the key concept of this new theory.

These findings are addressed here from a computational point of view, i.e., toward the goal of defining a model of such a dual memory that could lead to simulated experiments. Brain simulations using either artificial neural networks (Hopfield, 1982) or analytical methods (Izhikevich, 2006; Markram et al., 2015) (i.e., mainly differential equations modeling electrical currents (Hodgkin & Huxley, 1952)), as customarily performed today in computational neuroscience, have not been used so far to model such memories. Symbolic neural dynamic (Bonzon, 2017) abstracting the mechanisms of synaptic plasticity has been proposed as a new approach to modeling brain cognitive capabilities and used to define the basic mechanisms of an associative memory with dual store and retrieval processes. This formalism is extended here to reproduce optogenetic manipulations, thus defining a computational model of memory engrams.

2 Materials and Methods

This section introduces a formalism that has been previously published (Bonzon, 2017, 2019).

2.1 *A New Approach to Modeling Brain Cognitive Functionalities*

In this new formalism, brain processes representing synaptic plasticity are abstracted through asynchronous communication protocols and implemented as virtual *micro-circuits*. The basic units of these microcircuits are constituted by *threads*, which correspond either to a single or to a cluster of connected neurons. Contrary to traditional neuron models in which incoming signals are summed in some integrated value, thread inputs can be processed individually, thus allowing for threads to maintain parallel asynchronous communications. Threads can be grouped into disjoint sets, or *fibers* to model neural assemblies, and discrete *weights* (e.g., integer numbers) can be attached to pairs of threads that communicate within the same fiber. A fiber containing at least one active thread constitutes a *stream*. Mesoscale *virtual circuits* linking perceptions and actions are built out of microcircuits. Circuits can be represented either graphically or by sets of *symbolic* expressions. These expressions can

```
...-P->=>-Q-...
```

Fig. 1 Circuit fragment implementing a synaptic transmission. Reproduced from Bonzon (2019)

be compiled into *virtual code implications* that are used just in time to deduce instructions to be finally interpreted by a *virtual machine* performing contextual deductions (Bonzon, 1997).

To introduce this formalism, let us consider a simple case of synaptic transmission between any two threads **P** and **Q** (NB throughout this text, identifiers starting with a capital letter stand for variable parameters). This can be represented by the circuit fragment (or wiring diagram) contained in the simple stream given in Fig. 1, where the symbol **->=>-** represents a synapse.

This circuit fragment can be represented by two symbolic expressions involving a pair of **send/receive** processes as shown in Fig. 2.

In Fig. 2, the thread **P** (e.g., a sensor thread **sense(us)** with **us** representing an external stimulus as in Fig. 4) will fire in reaction to the capture of an external stimulus, with the **send** process corresponding to the signal, or spike train, carried by a presynaptic neuron’s axon. In the thread **Q** (e.g., an effector thread **motor(X)**, where the variable **X** becomes instantiated as the result of the stimulus), the **receive** process represents the possible reception of this signal by a postsynaptic neuron. The compilation of these expressions will give rise to the execution of virtual code instructions implementing the communication protocol given in Fig. 3.

This protocol corresponds to an *asynchronous* blocking communication subject to a threshold. It involves a predefined weight between the sender **P** and the receiver **Q** that can be either incremented or decremented. On one side, thread **P** fires thread **Q** if necessary and sends it a signal. On the other side, thread **Q** waits for the reception of a signal from thread **P** and proceeds only if the weight between **P** and **Q** stands above a given threshold. The overall process amounts to opening a temporary *pathway*

```
thread (P, [..., send(Q) ] )
thread (Q, [receive(P) ,...])
```

Fig. 2 Thread patterns for a synaptic transmission. Reproduced from Bonzon (2019)

```
P:      ...
        send(Q)           activate Q if Q is not active and post a signal for Q

Q:      receive(P)        wait for a signal from P and proceed if weight(P,Q)>0
        ...
```

Fig. 3 Communication protocol for an asynchronous communication. Reproduced from Bonzon (2019)

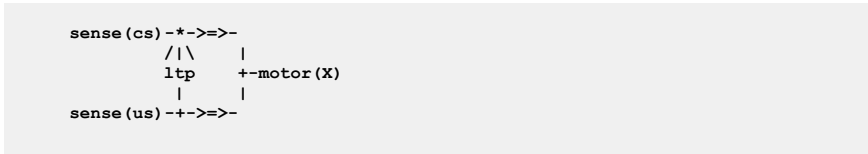


Fig. 4 A circuit implementing classical conditioning. Reproduced from Bonzon (2019)

between **P** and **Q** and allows for passing data by instantiating variable parameters (see example below).

Example

As a simple example, let us consider the classical conditioning of *Aplysia californica* (Kandel & Tauc, 1965). In this experiment, a light tactile conditioned stimulus **cs** elicits a weak defensive reflex, and a strong noxious unconditioned stimulus **us** produces a massive withdrawal reflex. After a few pairings of **cs** and **us**, where **cs** slightly precedes **us**, **cs** alone triggers a significantly enhanced withdrawal reflex. The corresponding circuit, adapted from a previous similar schema (Carew et al., 1981), is represented in Fig. 4. In this circuit, the symbol $/|\backslash$ represents the modulation of a synaptic transmission, the sign $*$ used in the upper branch indicates the conjunction of converging signals, and the sign $+$ indicates either the splitting of a diverging signal, as used in the lower branch, or a choice between converging signals, as used in the right branch instantiating the thread **motor(X)**, where **X** is a variable parameter to be instantiated into either **cs** or **us**.

In Fig. 4, the thread **ltp** (standing for *long-term potentiation*) acts as a facilitatory interneuron reinforcing the pathway between **sense(cs)** and **motor(cs)**. Classical conditioning then follows from the application of Hebbian learning (Hebb, 1949), i.e., “neurons that fire together wire together.” Though it is admitted today that classical conditioning in *Aplysia* is mediated by multiple neuronal mechanisms including a postsynaptic retroaction on a presynaptic site (Antonov et al., 2003), the important issue is that this activity depends on the temporal pairing of the conditioned and unconditioned stimuli, which leads to implement the thread **ltp** as a *detector of coincidence* as done in the protocol given in Fig. 5.

The generic microcircuit abstracting the mechanism of long-term potentiation is reproduced in Fig. 5 with its communication protocol. In order to detect the coincidence of **P** and **Q**, thread **P** fires an **ltp** thread that in turn calls on process **join** to wait for a signal from thread **Q**. In parallel, thread **Q** calls on process **merge** to post a signal for **ltp** and then executes a **send(R)** command to establish a link with thread **R**. After its synchronization with thread **Q**, thread **ltp** increments the weight between **Q** and **R**.

The circuit in Fig. 4 can be represented by the fiber, or set of symbolic expressions, given in Fig. 6.

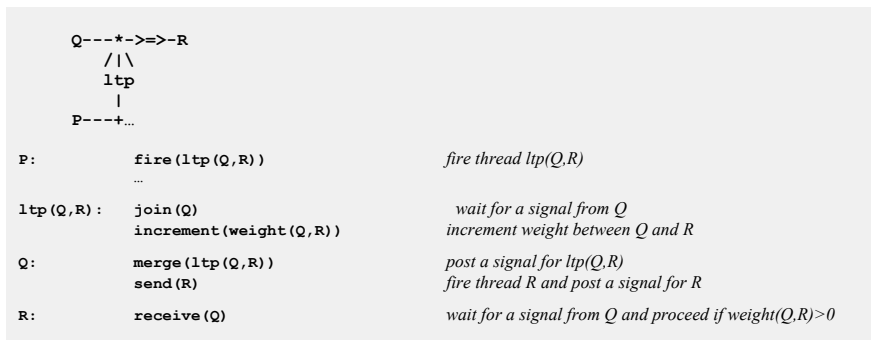


Fig. 5 Microcircuit and communication protocol for *ltp*. Reproduced from Bonzon (2019)

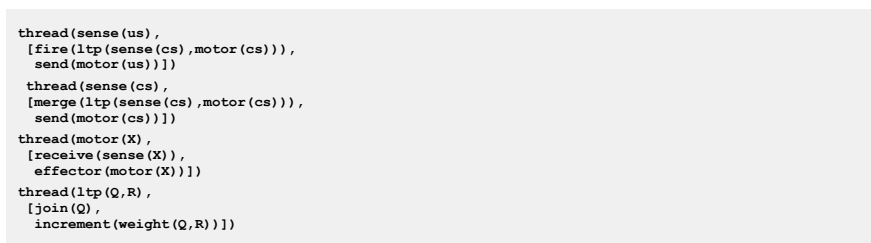


Fig. 6 Fiber corresponding to a circuit of classical conditioning

2.2 A Computational Model of an Associative Long-Term Memory

The concept of an associative memory has been studied from various perspectives (Palm, 1980). In our framework, an associative memory extends the mechanism of long-term potentiation by allowing for two threads **P** and **Q** attached to separate streams (and thus also possibly active at different times) to be associated in order to trigger a *recall* thread **R**. These two streams are linked together through a double communication protocol applied to a long-term memory **ltm(P)** thread, this construct being depicted by the symbol **-{P}**— meaning that **P** is both stored and retrievable through the thread **ltm(P)**. This new protocol involves two complementary *long-term storage/retrieval* (**lts/ltr**) threads that allow for the building of a storage trace and later retrieval of a previously active thread. This is in line with results by Rubin and Fusi (Rubin & Fusi, 2007) demonstrating that if the initial memory trace in neurons is below a certain threshold, then it cannot be retrieved immediately after the occurrence of the experience that created the memory. The corresponding microcircuit is given in Fig. 7 together with its communication protocol.

As a distinctive difference from an **ltp(Q,R)** thread (which gets fired by **P** and waits for a signal from **Q** in order to relate **Q** and **R**), an **ltr(P,Q,R)** thread is fired

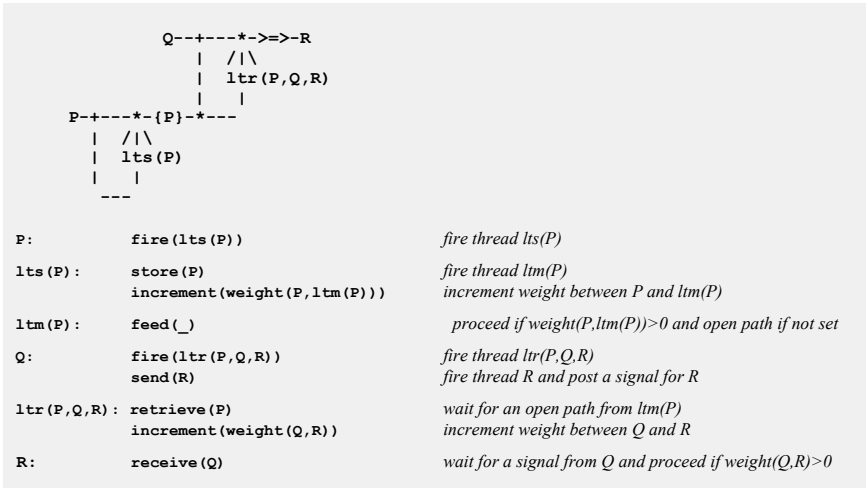


Fig. 7 Microcircuit and communication protocol for a long-term associative memory

by Q and waits for a path from $ltm(P)$ in order to relate Q and R , thus defining the basic mechanisms of an associative memory.

2.3 Simulation Platform

A simulation platform has been designed to implement the formalism described above. Defined by a logic program of about 300 lines, this platform can be run on any PC equipped with a Prolog compiler, which thus allows for an easy reproduction of results. It does rely on three fundamental concepts, i.e., the formal notions of

- an *object in context* represented by symbolic expressions in a logical language
- *communicating processes* between *concurrent threads* that is used to model a network of interactive objects
- a *virtual machine* interpreting virtual code that differs from a processor's native code and thus constitutes the key mechanism allowing for interfacing high-level abstract objects, e.g., software, with their low-level physical support, e.g., hardware.

At its top level, this virtual machine executes a “sense-act” cycle of embodied cognition as defined in Fig. 8 (see the *Supplementary information* for its complete operational specifications).

As a key point, let us point out the *ist* predicate standing for “is true” and implementing *contextual deduction* (Bonzon, 1997). Clock register values T are used to deduce, for each active thread, a possible next instruction. As postulated independently (Zeki, 2015), there is no central clock, leading thus to the modeling of the brain

```

Let Model represent the state of the virtual machine holding virtual code implications compiled from fibers.
A virtual machine run is constituted by a loop comprising a sense-react cycle:

run(Model)
  loop sense(Model)
        react(Model)

Sense corresponds to the capture and input of an interrupt triggered by a spike train directed to a stream:

sense(Model)
  if interrupt(Stream(Input))
  then input(Model(Stream), Input)

React loops on each thread within each stream to deduce a virtual instruction at its clock time T and execute it:

react(Model)
  for each Stream(Thread), T:Instruction,
  such that ist(Model(Stream)(Thread), (clock(T), T:Instruction))
  do execute(Model(Stream)(Thread), T:Instruction)

```

Fig. 8 High-level definition of a virtual machine run. Reproduced from Bonzon (2019)

as a massively asynchronous, parallel organ. Whenever an instruction is executed successfully, the thread clock is advanced and the next instruction is deduced and executed, and whenever it fails, the current instruction is attempted again until it eventually succeeds. Before being executed, virtual machine instructions are deduced “just in time” from *circuits* which have been compiled into virtual code implications. The execution of virtual instructions leads to a wiring/unwiring process that produces model configurations that are akin to plastic brain states. This procedure matches a fundamental principle in circuit neuroscience according to which *inhibition* in neuronal networks during baseline conditions allows in turn for *disinhibition*, which then stands as a key mechanism for circuit plasticity, learning and memory retrieval (Letzkus et al., 2015). This framework thus represents a computing device that greatly departs from a traditional von Neumann computer architecture, and could prove to be close to that of a real brain.

3 Results

3.1 A Mesoscale Circuit Representing a Memory Engram of Contextual Fear Conditioning

Contextual fear conditioning can be viewed as a case of classical conditioning and modeled in our framework as represented in Fig. 9 with two parameters, i.e., a first parameter designating a context (e.g., **a** or **b**) that recruits cells for a second parameter designating a percept such as **fear**, where **_** stands for the absence of a perception.

Memory engrams have been defined as connected neuronal ensembles that allow for the recall of information through various types of activations (Roy et al., 2017).

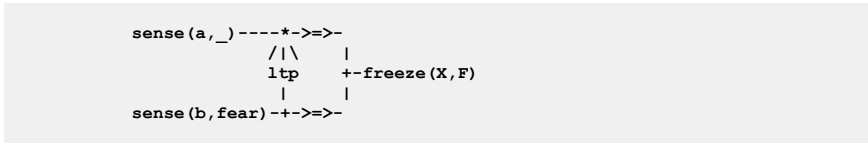


Fig. 9 Microcircuit implementing contextual fear conditioning

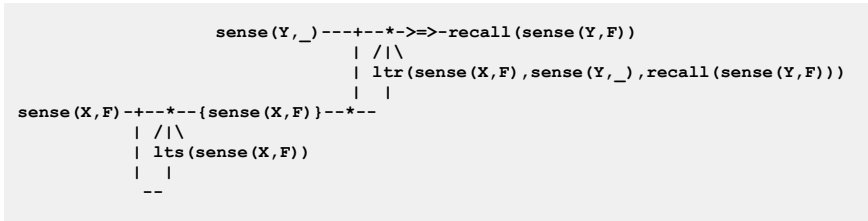


Fig. 10 Microcircuit representing a memory engram of contextual fear conditioning

In our framework, this can be achieved by extending the circuit of Fig. 9 into an associative memory as represented in Fig. 7. Replacing threads **P**, **Q** and **R** in Fig. 7 with, respectively, **sense(X,F)**, **sense(Y,_)** and **recall(sense(Y,F))** gives rise to the circuit given in Fig. 10. Depending on the application, **X** and **Y** can represent either the same (e.g., in the case of retrograde amnesia) or different (e.g., in the case of conditioning false memories) contexts. As for the possible projection of this circuit into actual brain regions, it is suggested that the left part of the circuit, i.e., the formation of **{sense(X,F)}** via **lts**, be identified with the upstream connections between the medial entorhinal cortex (MEC) and the dentate gyrus (DG) engram cells, on one side, and the right part, i.e., the possible **recall** via **ltr**, with the downstream connections of the DG with the hippocampal CA3 and basolateral amygdala (BLA) engram cells, on the other side.

3.2 Experimental Schedule Implementation

Using optogenetic technology (Liu et al., 2012; Ramirez et al., 2013; Ryan et al., 2015; Roy et al., 2017), memory engram cells can be first *tagged* by injection and *labeled* during a training session. At will, they can be then *blocked*, *unblocked* and *lighted* through various injections according to specific experimental schedules. Simulations performed with the circuit of Fig. 10 have reproduced some of the original behavioral schedules of actual experiments (Liu et al., 2012; Ramirez et al., 2013; Ryan et al., 2015; Roy et al., 2017). These schedules have been assembled from the set of fibers given in Fig. 11 (for the interpretation of instructions such as *tag*, *label*, *block*, *unblock*, *light*, *fire*, *send*, *receive*, see the Supplementary information section).


```

Targeting fiber
thread(inject(tag,X,F), [tag(sense(X,Y),recall(sense(X,F)))]) .

Training fiber
thread(sense(X,F), [fire(lts(sense(X,F))),
label(sense(X,F),recall(sense(X,F))),
effector(sense(X,F))]) .

Testing fiber
thread(sense(X,F), [fire(ltr(sense(X,F),sense(X,_),recall(sense(X,F))))),
send(recall(sense(X,F)))] .
thread(recall(sense(X,F)), [receive(sense(X,F)),
effector(sense(X,F))]) .

Closing fiber
thread(inject(block,X,F), [block(sense(X,F),recall(sense(X,F)))] .

Opening fiber
thread(inject(unblock,X,F), [unblock(sense(X,F),recall(sense(X,F)))] .

Activating fiber
thread(inject(light,X,F), [light(sense(X,F),recall(sense(X,F))),
effector(sense(X,F))]) .

```

Fig. 11 Fibers for the implementation of behavioral schedules

Each schedule starts with a **targeting** phase in which the dentate gyrus of transgenic mice is *tagged* with the injection of viruses and implanted with optic fibers. Active cells are then *labeled* through a contextual fear conditioning **training** session that results in the retention of a specific pattern of connectivity between engram cells required for the storage of information. A **testing** phase allows for its retrieval through natural cues. Memory consolidation can be *blocked* by an injection that inhibits protein synthesis and thus closes the pathway to memory consolidation, and subsequently *unblocked*. Finally, **activating** *lighted* cells demonstrate *silent engrams* (Ramirez et al. 2013) and the creation of *false memories* (Roy et al., 2017).

3.3 Simulation Results

Simulations based on the circuit of Fig. 10 and assembled from the fibers of Fig. 11 have reproduced the results of actual experiments (Liu et al., 2012; Ramirez et al., 2013; Ryan et al., 2015; Roy et al., 2017). Inputs to sensors and injectors are preceded by a prompt | : and outputs from effectors by >>. Additional outputs report about the successive states of the engram.

Contextual fear conditioning control experiment

In the run of Fig. 12, cells involved in fear conditioning are *tagged* for any context, *labeled* for context **b** though a training session ending with an upstream synaptic strength and *opened* pathway (meaning that protein synthesis for memory consolidation is active). The consolidated memory with increased downstream synaptic strength then allows for a memory recall though natural cues.

```

|:targeting([injector(tag,_,_)]) .
tagged(sense(,_),recall(sense(,_)))
|:training([sensor(b,fear)]) .
labeled(sense(b,fear),recall(sense(b,fear)))
path(opened(sense(b,fear)|_))
weight(sense(b,fear),ltm(sense(b,fear)))(1)
>>sense(b,fear)
|:testing([sensor(b,_)]) .
weight(sense(b,fear),recall(sense(b,fear)))(1)
>>sense(b,fear)

```

ChR2 injection
tagged cells
shock exposure in context b
labeled active cells
active protein synthesis
upstream synaptic strength
freezing reaction in context b
no shock exposure in context b
downstream synaptic strength
freezing reaction in context b

Fig. 12 Execution log of a control simulation run

Reversible retrograde amnesia

Under amnesia, impaired synaptic strengthening prevents the activation of engram cells by natural recall cues. Toward this end, the training session is directly followed by a **blocking** injection that causes a protein synthesis inhibition resulting in a retrograde amnesia without memory consolidation, which can be then reversed to allow for the reactivation of protein synthesis, the consolidation of memory and a freezing reaction due to memory recall though natural cues (Fig. 13).

Silent engram direct activation

Following retrograde amnesia in context **b**, the resulting silent engram for context **a** gets directly activated with *light* stimulation leading to a freezing behavior without memory recall (Fig. 14).

Creation of a false memory

In this experiment (Ramirez et al., 2013), the cells labeled in context **a** without shock exposure serve as a conditioned stimulus. They get then artificially stimulated

```

|:targeting([injector(tag,_,_)]) .
tagged(sense(,_),recall(sense(,_)))
|:training([sensor(b,fear)]) .
labeled(sense(b,fear),recall(sense(b,fear)))
path(opened(sense(b,fear)|_))
weight(sense(b,fear),ltm(sense(b,fear)))(1)
>>sense(b,fear)
|:closing([injector(block,b,fear)]) .
path(closed(sense(b,_)])
|:testing([sensor(b,_)]) .
weight(sense(b,_) ,recall(sense(b,_)))(0)
|:opening([injector(unblock,b,fear)]) .
path(opened(sense(b,fear),recall(sense(b,fear))))
|:testing([sensor(b,_)]) .
weight(sense(b,fear),recall(sense(b,fear)))(1)
>>sense(b,fear)

```

ChR2 injection
tagged cells
shock exposure in context b
labeled active cells
active protein synthesis
upstream synaptic strength
freezing reaction in context b
anisomycin injection
protein synthesis inhibited
no shock exposure
no downstream synaptic strength
PAK1 injection
active protein synthesis
no shock exposure in context b
downstream synaptic strength
freezing reaction in context b

Fig. 13 Simulation run of a reversible retrograde amnesia

```

|:targeting ([injector (tag,_,_)]) .
tagged (sense (_,_) ,recall (sense (_,_)))

|:training ([sensor (a, )]) .
labeled (sense (a, ) ,recall (sense (a, )))
path (opened (sense (a,_) |))
weight (sense (a,_) ,ltm (sense (a,_) ) (1)
>> sense (a,_)

|:training ([sensor (b, fear)]) .
labeled (sense (b, fear) ,recall (sense (b, fear)))
path (opened (sense (b, fear) |))
weight (sense (b, fear) ,ltm (sense (b, fear) ) (1)
>> sense (b, fear)

|:closing ([injector (block,b,fear)]) .
path (closed (_))

|:testing ([sensor (a,_)]) .
weight (sense (a,_) ,recall (sense (a,_) ) (0)

|:activating ([injector (light,a, )]) .
>> sense (a, fear)

|:testing ([sensor (a,_)]) .
weight (sense (a,_) ,recall (sense (a,_) ) (0)
    
```

ChR2 injection
tagged cells
habituation in context a
labeled cells
active protein synthesis
upstream synaptic strength
no freezing reaction
shock exposure in context b
labeled active cells
active protein synthesis
upstream synaptic strength
freezing reaction
anisomycin injection
protein synthesis inhibited
no shock exposure
no downstream synaptic strength
light injection with no shock in a
freezing reaction
no shock exposure
no downstream synaptic strength

Fig. 14 Simulation run of a light-induced direct silent engram activation

by *light* during the delivery of an unconditioned fear stimulus in context **b** and subsequently express a false fear memory by freezing in context **a**, but not in a novel context **c** (see the Supplementary information for a definition of the *light*-virtual instruction that allows for a displaced memory consolidation) (Fig. 15).

Artificial association of independent memories

In this last example, we reproduce an experiment (Ohkawa et al., 2015) in which coincident firing of distinct neural assemblies generates an artificial link between

```

|:targeting ([injector (tag,_,_)]) .
tagged (sense (_,_) ,recall (sense (_,_)))

|:training ([sensor (a, )]) .
labeled (sense (a,_) ,recall (sense (a,_) ))
path (opened (sense (a,_) |))
weight (sense (a,_) ,ltm (sense (a,_) ) (1)
>> sense (a,_)

|:testing ([sensor (a, )]) .
weight (sense (a, ) ,recall (sense (a, ) ) (1)
>> sense (a,_)

|:activating ([injector (light,b,fear)]) .
>> sense (b, fear)

|:testing ([sensor (a,_)]) .
weight (sense (a, fear) ,recall (sense (a, fear) ) (1)
>> sense (a, fear)

|:testing ([sensor (c,_)]) .
weight (sense (c,_) ,recall (sense (c,_) ) (0)
    
```

ChR2 injection
tagged cells
no shock exposure for labeling a
labeled active cells
active protein synthesis
upstream synaptic strength
no significant freezing reaction
no shock exposure in context a
downstream synaptic strength
no significant freezing reaction
light injection with shock in b
freezing reaction in b
no shock exposure in context a
downstream synaptic strength
freezing reaction in a
no shock exposure in context b
no downstream synaptic strength

Fig. 15 Simulation run of a false memory

```

| :targeting([injector(tag,_,_)]) .
tagged(sense(,_,),recall(sense(,_,)))

| :training([sensor(a,_)]) .
labeled(sense(a,_,),recall(sense(a,_,)))
path(opened(sense(a,_) | ))
weight(sense(a,_) ,ltm(sense(a,_) )) (1)
>>sense(a,_)

| :training([sensor(b,fear)]) .
labeled(sense(b,fear) ,recall(sense(b,fear)))
path(opened(sense(b,fear)
weight(sense(b,fear) ,ltm(sense(b,fear))) (1)
>>sense(b,fear)

| :activating([injector(light,c,_)]) .
>>sense(c,_)

| :testing([sensor(a,_)]) .
weight(sense(a,fear) ,recall(sense(a,fear))) (1)
>>sense(a,fear)

```

*ChR2 injection
tagged cells*

*no shock exposure for labeling a
labeled active cells
active protein synthesis
upstream synaptic strength
no significant freezing reaction*

*shock exposure for labeling b
labeled active cells
active protein synthesis
upstream synaptic strength
freezing reaction*

*light injection in context c
no freezing reaction in c*

*no shock exposure in context a
downstream synaptic strength
freezing reaction in a*

Fig. 16 Simulation run of the artificial association of independent memories

distinct memory episodes. This looks similar to the creation of a false memory modeled above, except that in this case the conditioning occurs through the exposure to light stimulation, i.e., **activating([injector(light,c,_)])** in a third context **c** (Fig. 16).

4 Discussion

The simulations that have been presented provide an illustration of how impaired synaptic strengthening caused by the injection of a protein synthesis inhibitor immediately after contextual fear conditioning prevents the effective activation of engram cells by natural recall cues, thus leading to retrograde amnesia. The information stored in engram cell ensemble connectivity can nevertheless be retrieved by light-induced direct activation of labeled nodes. Altogether, these results support the hypothesis that separate processes are involved in long-term memory, i.e., the retention of specific patterns of connectivity between engram cells required for the storage of information, on the one hand, and the synaptic strengthening needed for its consolidation on the other (Tonegawa, 2015; Trettenbrein, 2016). In other words, synaptic connectivity could provide a substrate for memory storage whereas the potentiation of synapses would be required for its retrieval.

It is acknowledged today that individual fear memories require engram cells from multiple brain regions (Tonegawa, 2015). In our simulations, non-instantiated **tagged(sense(,_,),recall(sense(,_,)))** expressions are attached to cell populations whose elements can be indifferently recruited for labeling various contexts such as **a** and **b**. As our framework readily accommodates instantiated tags that could be used for recruiting specific cells for different contexts or tasks, it can be used to design and predict the results of finer grain experiments involving multiple brain regions (Abdou et al., 2018; Oishi et al., 2019).

Competing Interests The author declares no competing interests.

References

- Abdou, K., et al. (2018). Synapse-specific representation of the identity of overlapping of memory engrams. *Science*, 360(6394), 1227–1231.
- Antonov, I., Antonova, I., Kandel, E. R., & Hawkins, R. D. (2003). Activity-dependent presynaptic facilitation and hebbian LTP are both required and interact during classical conditioning in Aplysia. *Neuron*, 37(1), 135–147.
- Bonzon, P. (1997). A reflective proof system for reasoning in contexts. In *14th National Conference on American Association Artificial Intelligence*. www.aaai.org/Papers/AAAI/1997/AAAI97-061.pdf (1997),
- Bonzon, P. (2017). Towards neuro-inspired symbolic models of cognition: linking neural dynamics to behaviors through asynchronous communications. *Cognitive Neurodynamics*, 11(4), 327–353.
- Bonzon, P. (2019). Symbolic modeling of asynchronous neural dynamics reveals potential synchronous roots for the emergence of awareness. *Frontiers in Computer Neuroscience*. <https://doi.org/10.3389/fncom.2019.00001>.
- Carew, T., Walters, E., & Kandel, E. (1981). Classical conditioning in a simple withdrawal reflex in Aplysia californica. *The Journal of neuroscience*, 1(12), 1426–1437.
- Deisseroth, K., et al. (2006). Next-generation optical technologies for illuminating genetically targeted brain circuits. *Journal of Neuroscience*, 26(41), 10380–10386.
- Hebb, D. O. (1949). *The organization of behavior. A neuropsychological theory*. J. Wiley (1949).
- Hodgkin, A., & Huxley, A. (1952). A quantitative description of membrane current & its application to conduction & excitation in nerve. *Journal of Physiology*, 17(4), 500–544.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79(8), 2554–2558.
- Izhikevich, E. (2006). Pochronization: Computation with spikes. *Neural Computation*, 18, 245–282.
- Kandel, E. R., & Tauc, L. (1965). Heterosynaptic facilitation in neurones of the abdominal ganglion of Aplysia depilans. *The Journal of Physiology*, 181(1), 1–27.
- Letzkus, J., Wolff, S., & Lüthi, A. (2015). Disinhibition, a circuit mechanism for associative learning & memory. *Neuron*, 8(2), 264–276.
- Liu, X., et al. (2012). Optogenetic stimulation of a hippocampal engram activates fear memory recall. *Nature*, 484(7394), 381–385.
- Markram, H., et al. (2015). Reconstruction & simulation of neocortical microcircuitry. *Cell*, 163, 456–492.
- Ohkawa, N., et al. (2015). Artificial association of pre-stored information to generate a qualitatively new memory. *Cell Reports*, 11, 261–269.
- Oishi, N., et al. (2019). Artificial association of memory events by optogenetic stimulation of hippocampal CA3 cell ensemble. *Molecular Brain*, 12, 2.
- Palm, G. (1980). On associative memories. *Biological Cybernetics*, 36, 19–31.
- Ramirez, S., et al. (2013). Creating a false memory in the hippocampus. *Science*, 341, 387–391.
- Roy DS, Muralidhar S, Smith LM, Tonegawa S. Silent memory engrams as the basis for retrograde amnesia. *Proc Natl Acad Sci U S A*.;114,46, (2017).
- Rubin, D., & Fusi, S. (2007). Long memory lifetimes require complex synapses & limited sparseness. *Frontiers in Computer Neuroscience*, 1, 7.
- Ryan, T. J., et al. (2015). Engram cells retain memory under retrograde amnesia. *Science*, 348, 1007–1013.
- Tonegawa, S. (2015). Memory engram storage and retrieval. *Current Opinion in Neurobiology*, 35, 109–111.

- Trettenbrein, P. (2016). The demise of the synapse as the locus of memory: A looming paradigm shift? *Frontiers in Systems Neuroscience*, *10*, 88.
- Zeki, S. (2015). A massively asynchronous, parallel brain. *Philosophical Transactions of the Royal Society B*, *370*, 20140174.