Vector-based navigation using grid-like representations in artificial agents

Banino et al., Nature, 2018
DeepMind & University College London

2021년 8월 30일 월요일
발표: 김성신

Artificial General Intelligence (AGI)

- Learn automatically from raw inputs-not pre-programmed
- General-same system can operate across a wide range of tasks
- General-purpose learning algorithms – flexible, adaptive, creative

Brain is the only example of the AGI

Modified from Hassabis's talk (2016)

Brain as a model of AI

Is the brain a good model for machine intelligence?

To celebrate the centenary of the year of Alan Turing's birth, four scientists and entrepreneurs assess the divide between neuroscience and computing.

equivalent to Turing's finite-state machine with an infinite tape and a finite symbol set, and that does computation.

In 1943, Warren McCulloch and Walter Pitts[2] noted the "all-or-none" nature of the firing of neurons in a nervous system, and suggested that networks of neurons could be modelled as logical propositions. They modelled a network of neurons as circuits of logic gates, noting that these may "compute only such numbers as can a Turing machine". But more, they proposed that everything at a psychological level happens in these networks. Over the decades, such ideas begat more studies in neural networks, which in turn begat computational neuroscience. Now those metaphors and models pervade explanations of how the brain 'computes'. But these binary abstractions do not capture all the complexities inherent in the brain.

So now I see circles before my eyes. The brain has become a digital computer; yet we are still trying to make our machines intelligent. Should those machines be modelled on the brain, given that our models of the brain are performed on such machines? That will probably not be enough.

When you are stuck, you are stuck. We will get out of this cul-de-sac, but it will take some brave and bright souls to break out of our circular confusions of models.

ILLUSTRATION BY ANDY POTTS

DEMIS HASSABIS

Model the brain's algorithms

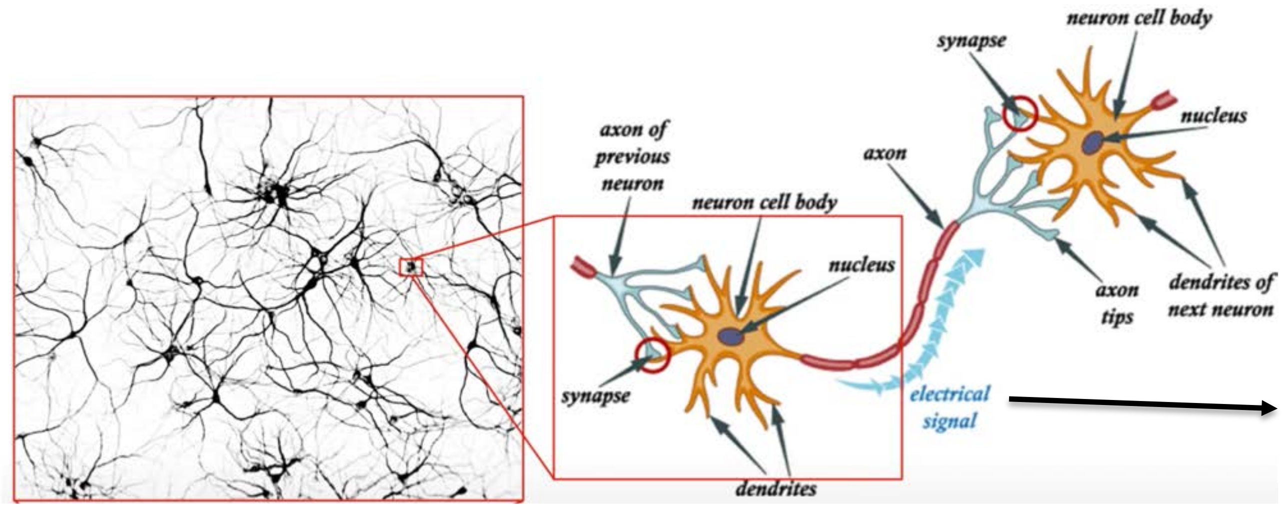
Neuroscientist, computer-game producer and chess master, University College London

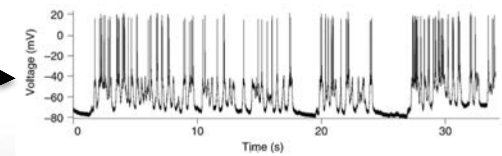
Alan Turing looked to the human brain as the prototype for intelligence. If he were alive today, he would surely be working at the inter-

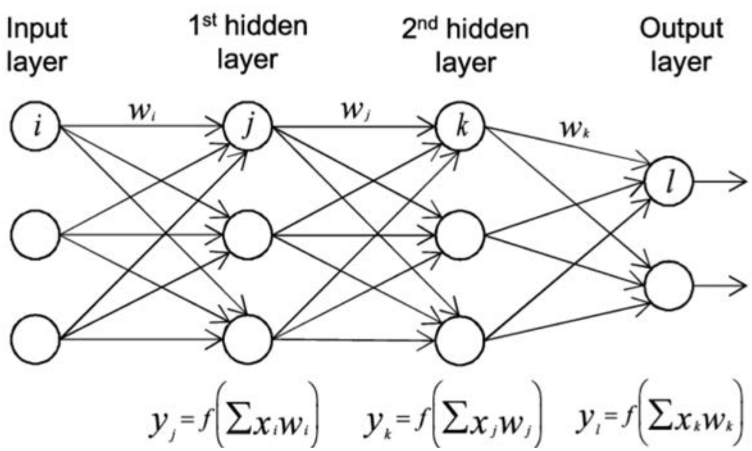
"To advance AI, we need to Better understand the brain's working at the algorithmic level"

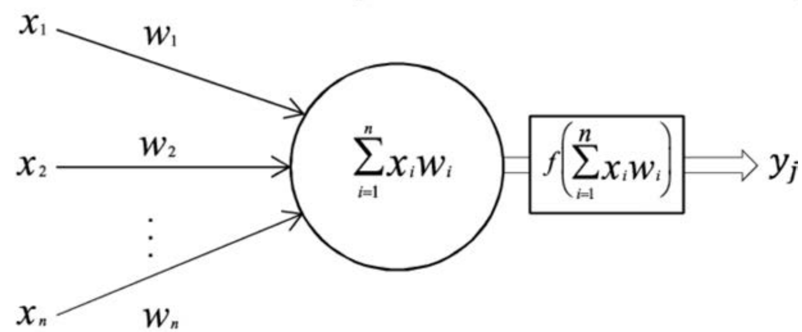
Demis Hassabis, *Nature*, 2012

Real vs artificial neural networks

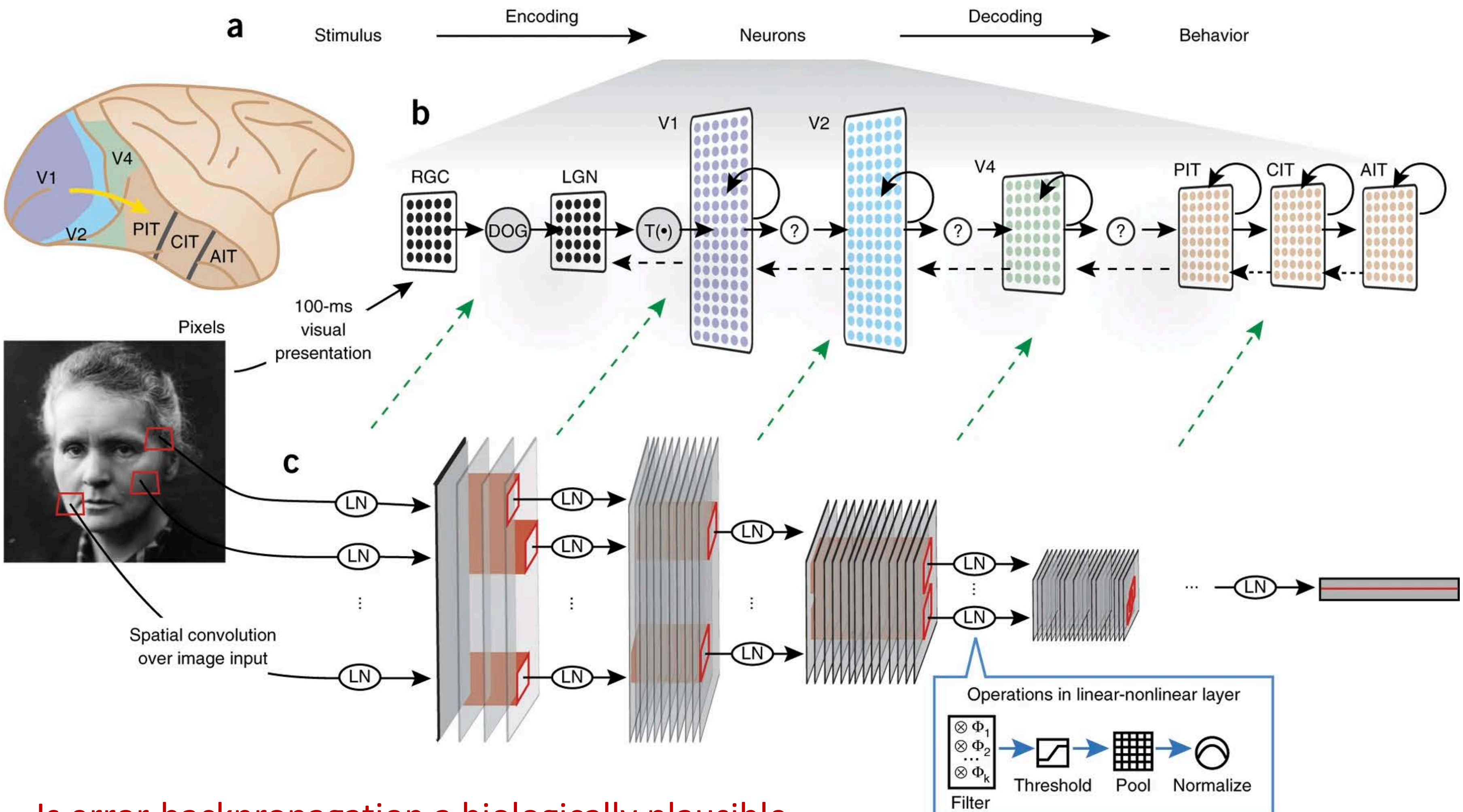








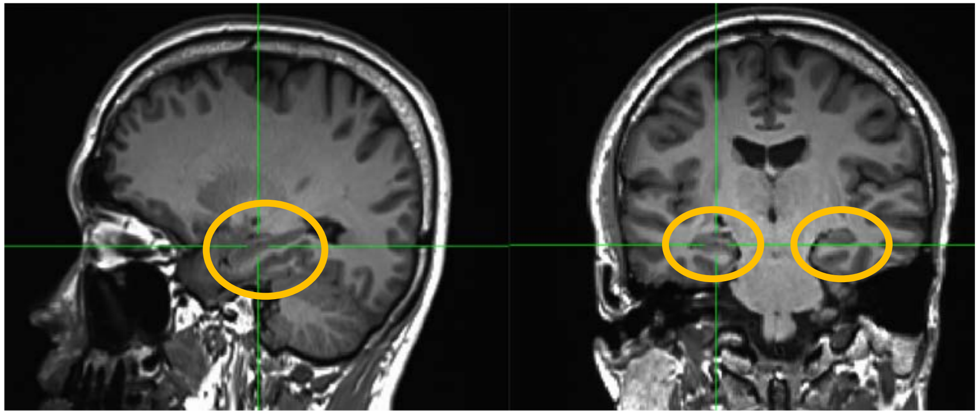
Convolutional Neural Networks



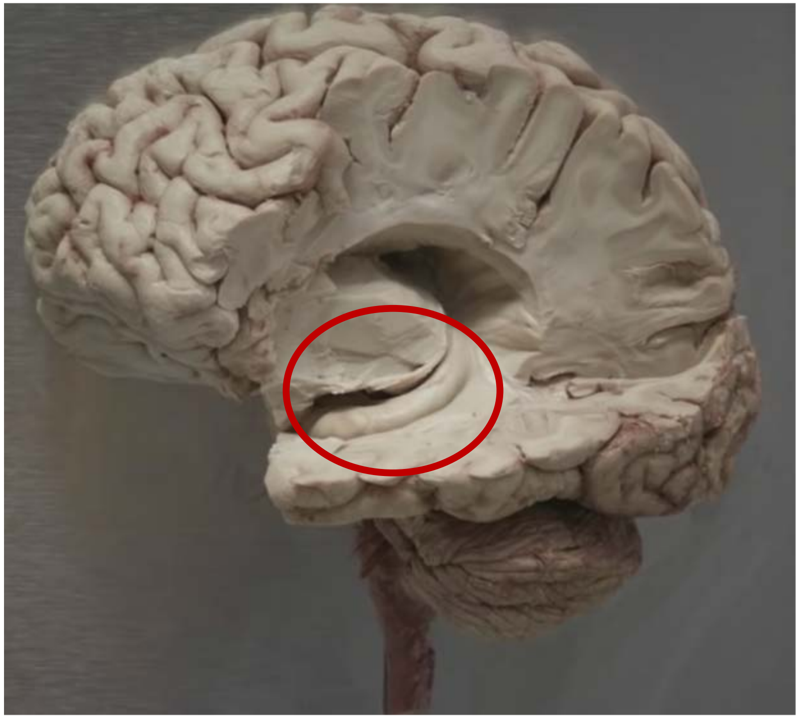
Is error-backpropagation a biologically plausible learning algorithm?

Adapted from Yamins and DiCarlo, 2016

Memory in the brain: Hippocampus

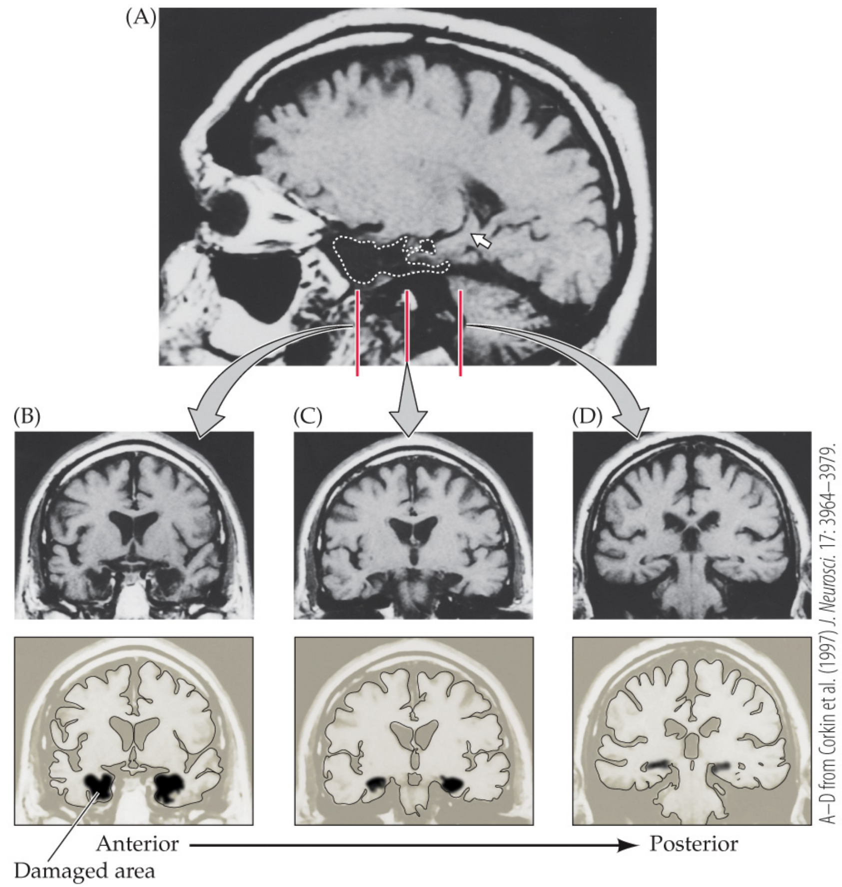


Left: Hippocampus + fornix
Right: Seahorse



Patient H.M.

Henry Molarison, then 60 at MIT in 1986
(1926-2008)



NEUROSCIENCE 6e, Clinical Applications 30 (Part 1)

Navigation in the brain: Hippocampus

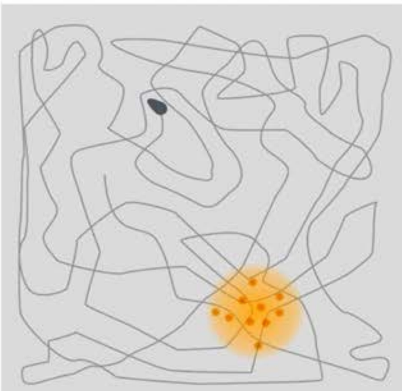


Fig. 1

John O'Keefe

John O'Keefe discovered, in 1971, that certain nerve cells in the brain were activated when a rat assumed a particular place in the environment. Other nerve cells were activated at other places. He proposed that these "place cells" build up an inner map of the environment. Place cells are located in a part of the brain called the hippocampus.

May-Britt Moser and Edvard I. Moser

May-Britt och Edvard I. Moser discovered in 2005 that other nerve cells in a nearby part of the brain, the entorhinal cortex, were activated when the rat passed certain locations. Together, these locations formed a hexagonal grid, each "grid cell" reacting in a unique spatial pattern. Collectively, these grid cells form a coordinate system that allows for spatial navigation.

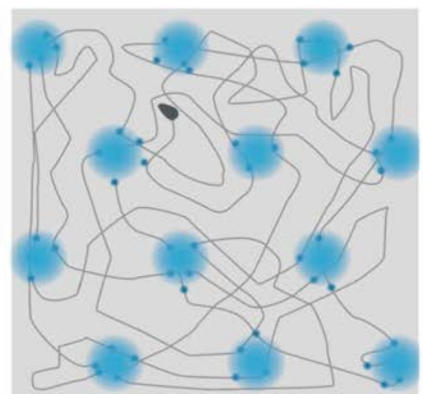
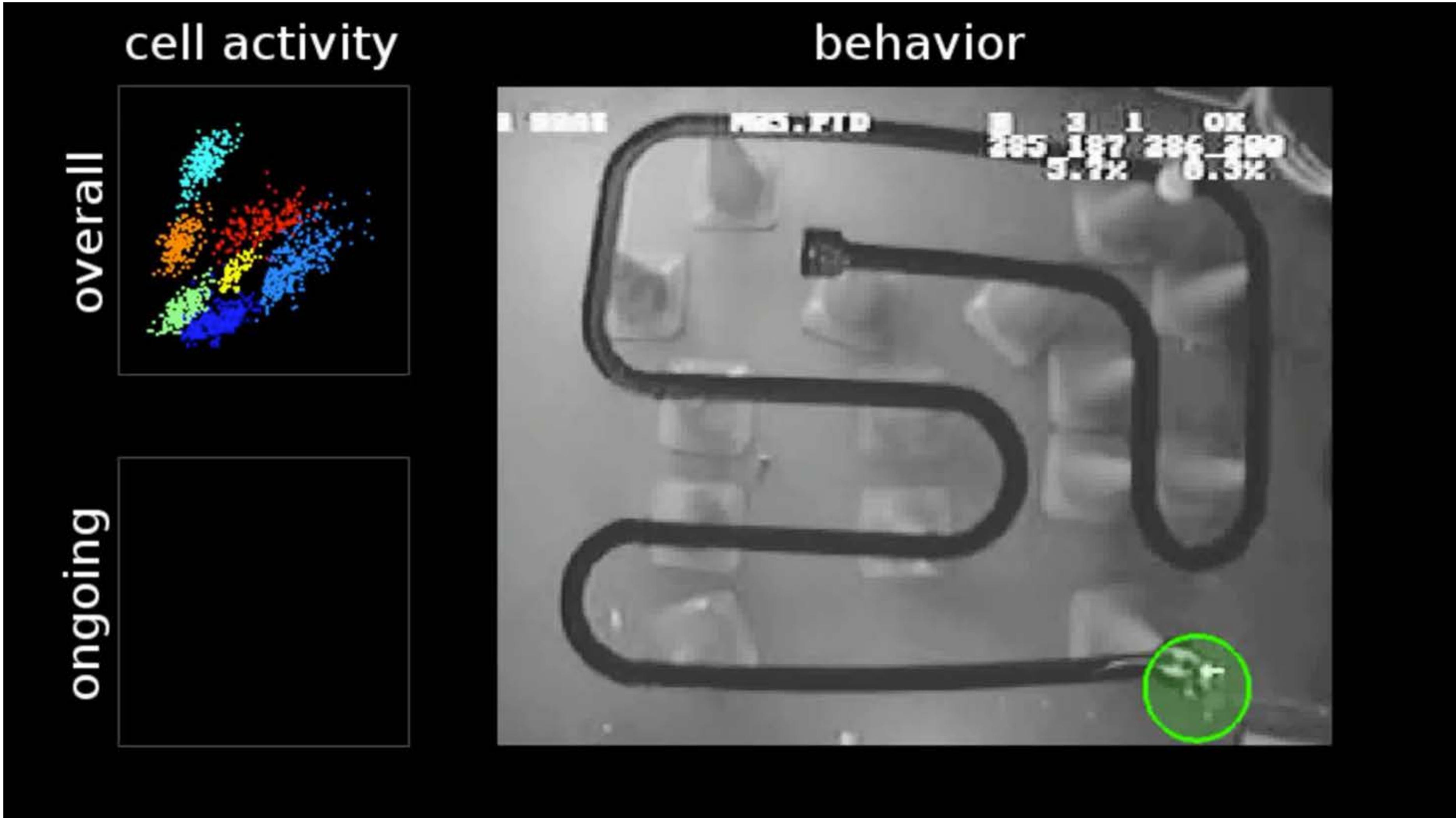


Fig. 2

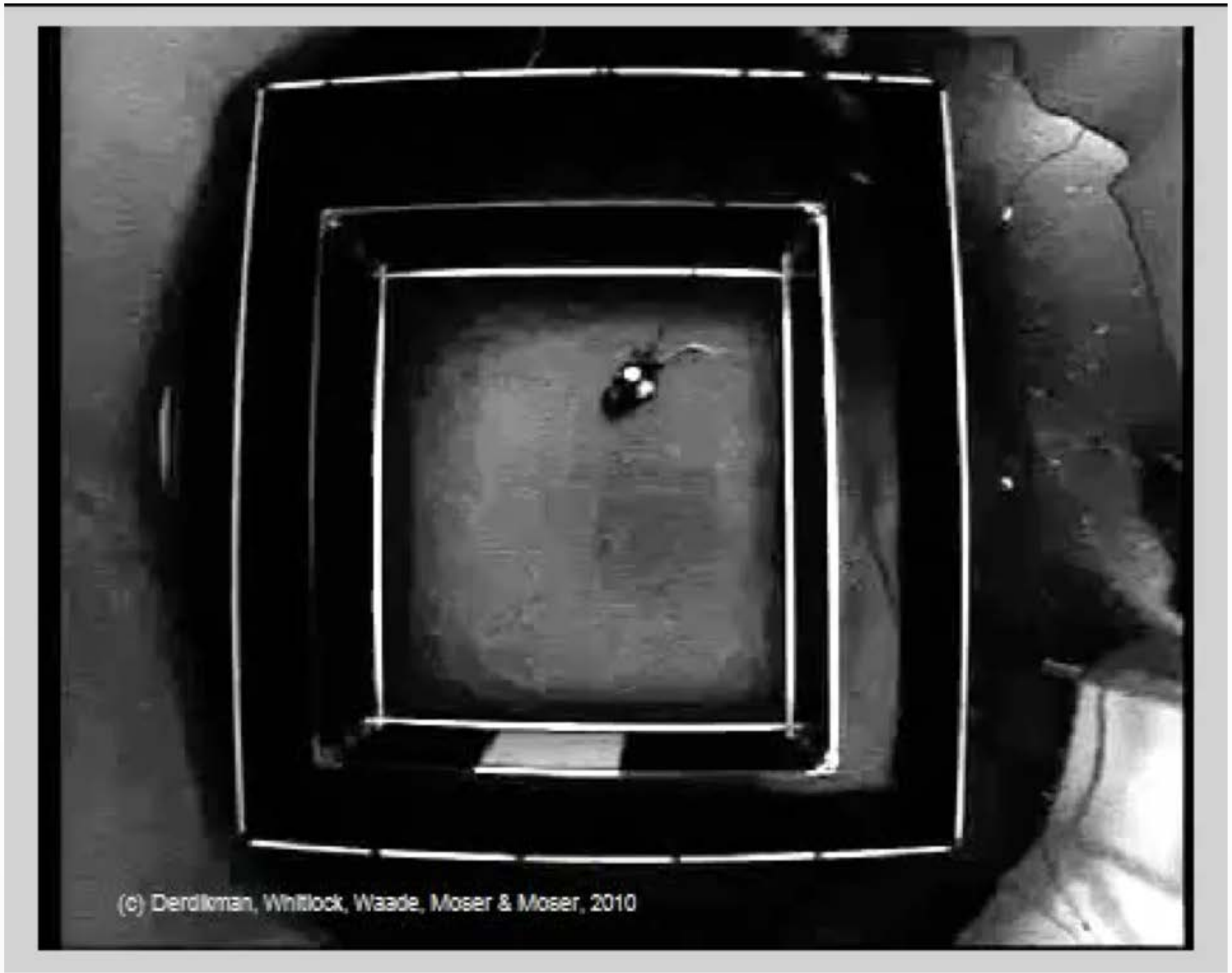
The 2014 Nobel Prize in Physiology or Medicine with one half to **John O'Keefe** and the other half jointly to **May-Britt Moser and Edvard I. Moser for their discoveries of cells that constitute a positioning system in the brain**

Place cells

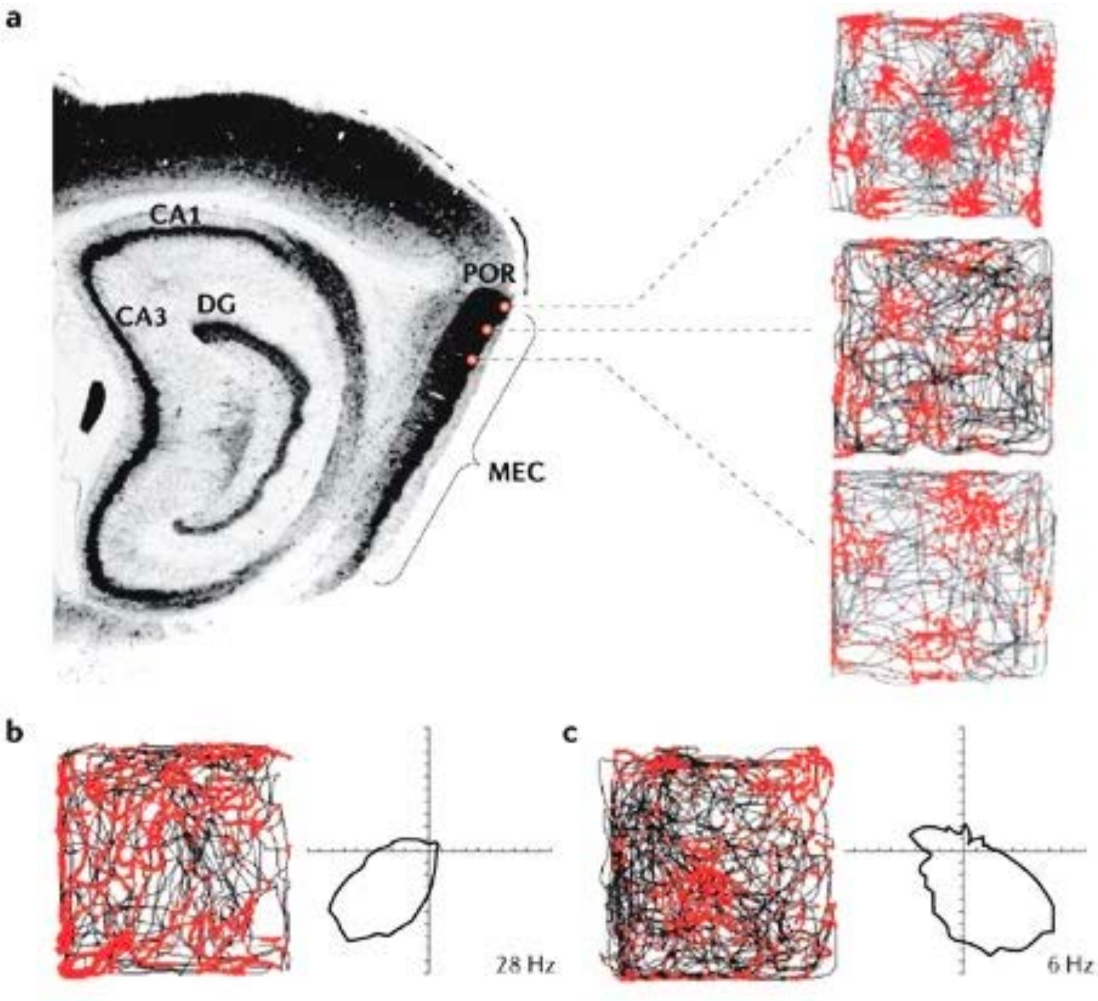


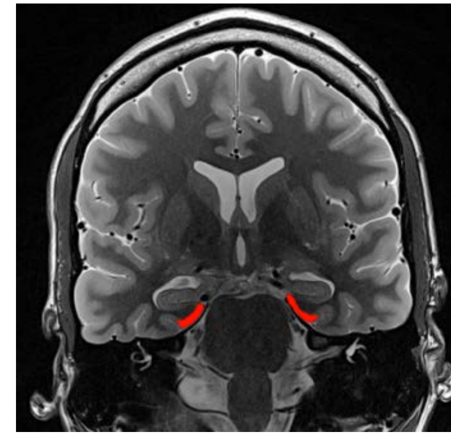
Prof. Wilson Lab (M.I.T.)

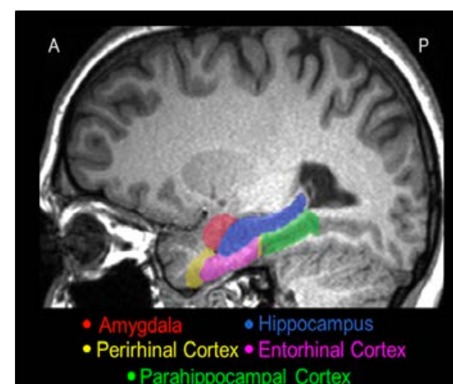
Grid cells



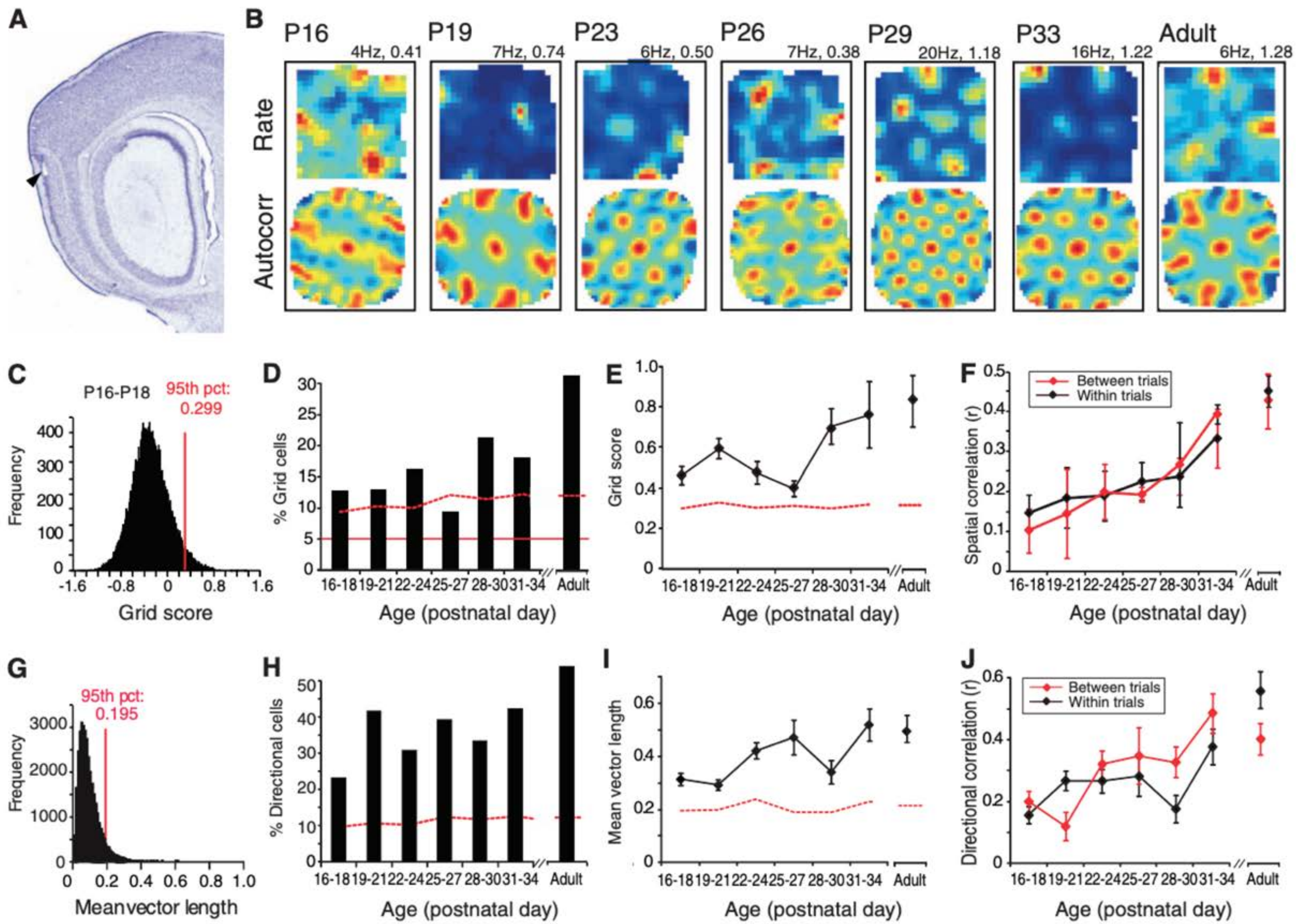
Multiple spatial scale in the medial entorhinal cortex







Development of the spatial representation system in the rat



Models of place cell and head-direction cell

Place cell activations

$$c_i = \frac{e^{-\frac{\|\vec{x} - \vec{\mu}_i^{(c)}\|_2^2}{2(\sigma^{(c)})^2}}}{\sum_{j=1}^{N} e^{-\frac{\|\vec{x} - \vec{\mu}_j^{(c)}\|_2^2}{2(\sigma^{(c)})^2}}}$$

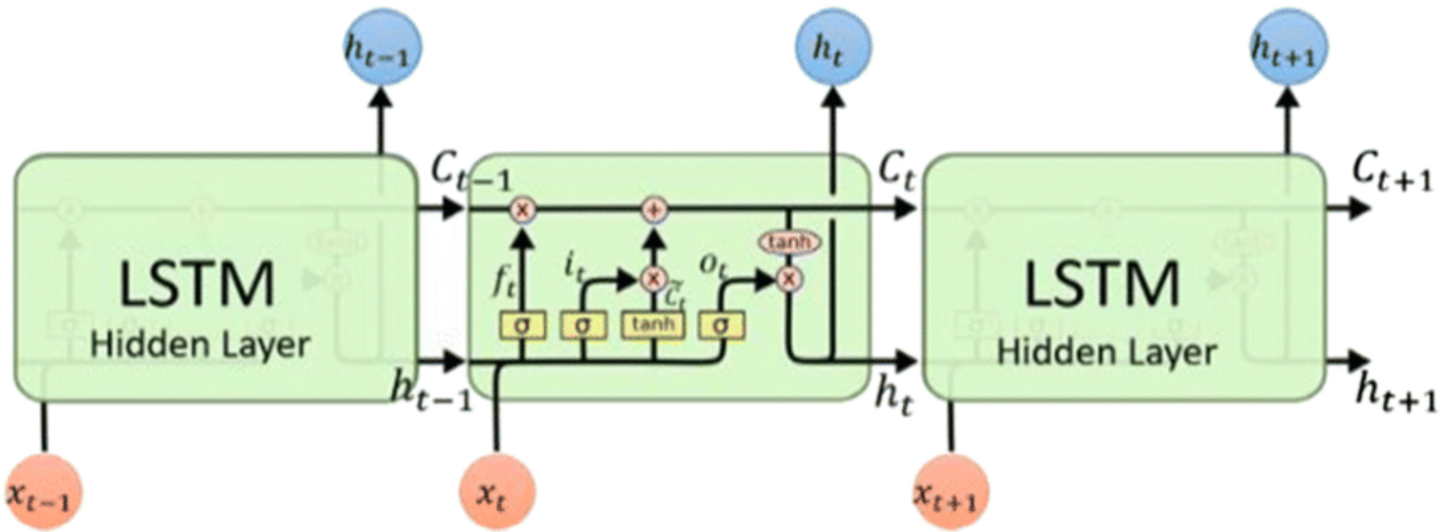
$$\vec{c} \in [0,1]^N$$

Head-direction cell activations

$$h_i = \frac{e^{k^{(h)} \cos(\varphi - \mu_i^{(h)})}}{\sum_{j=1}^{M} e^{k^{(h)} \cos(\varphi - \mu_j^{(h)})}}$$

$$\vec{h} \in [0,1]^M$$

LSTM network



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

$$\vec{l}_t = W^{cp}\,\vec{c}_t + W^{(cd)}\,\vec{h}_t$$

$$\vec{m}_t = W^{hp}\,\vec{c}_t + W^{(hd)}\,\vec{h}_t$$

LSTM input

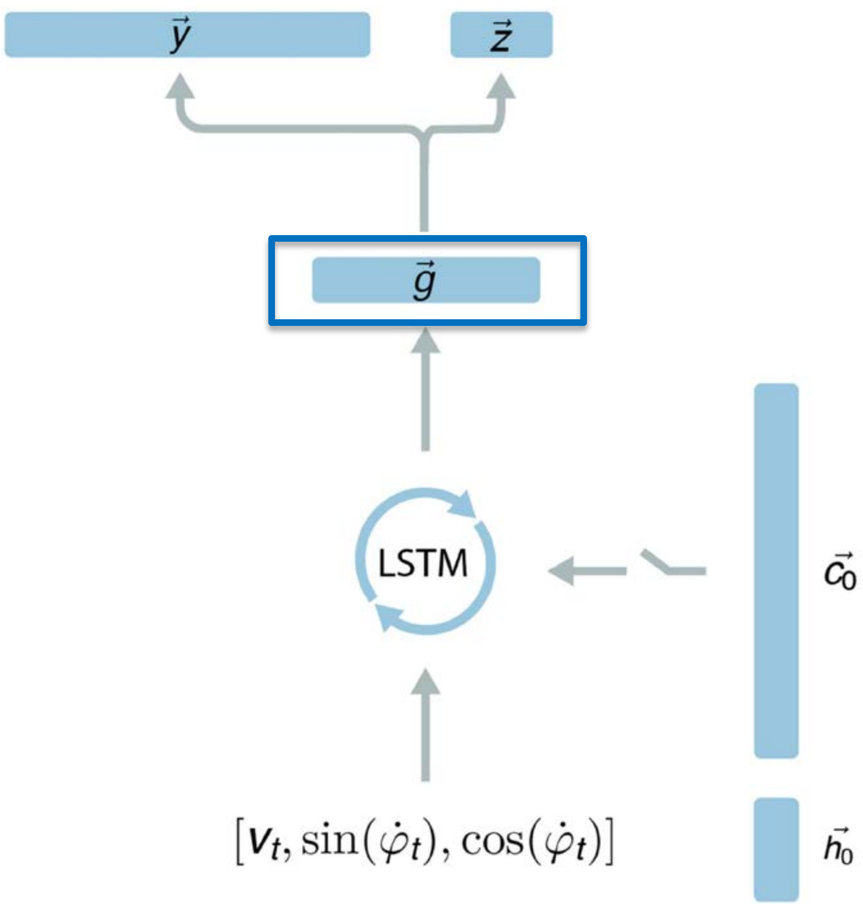
$$\vec{x}_t = [v_t, sin(\varphi_t)\,, cos(\varphi_t)]$$

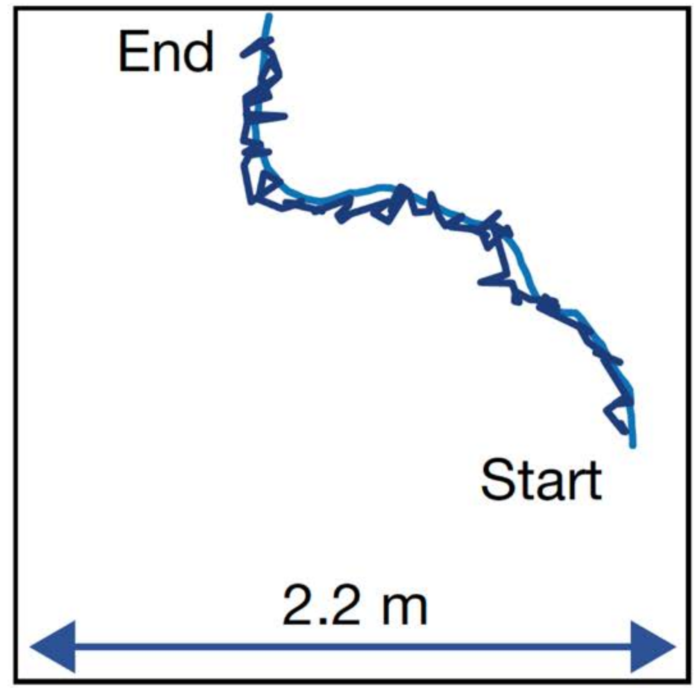
$\vec{c}_t$: vector of place cell activations
$\vec{h}_t$: vector of head-direction cell activations
$\vec{l}_t$: vector of cell state values
$\vec{m}_t$: vector of hidden state values

Supervised learning for path integration

$$\mathcal{L}(\vec{y}, \vec{z}, \vec{c}, \vec{h}) = -\sum_{i=1}^{N} c_i \log(y_i) - \sum_{j=1}^{M} h_j \log(z_j)$$ Loss function

Place cell $\vec{c}$ $\vec{h}$ Head-direction cell





Implementation: Grid Network

```
class GridTorch(nn.Module):

    def __init__(self,
                 target_ensembles,
                 init_conds_size=268,
                 nh_lstm=128,
                 nh_bottleneck=256,
                 dropoutrates_bottleneck=0.5,
                 bottleneck_has_bias=False):

        super(GridTorch, self).__init__()
        self.target_ensembles = target_ensembles
        self.rnn = nn.LSTMCell(input_size=3, hidden_size=nh_lstm)
        self.state_embed = nn.Linear(init_conds_size, nh_lstm)
        self.cell_embed = nn.Linear(init_conds_size, nh_lstm)
        self.dropout = nn.Dropout(dropoutrates_bottleneck)
        self.bottleneck = nn.Linear(nh_lstm, nh_bottleneck, bias=bottleneck_has_bias)
        self.pc_logits = nn.Linear(nh_bottleneck, target_ensembles[0].n_cells)
        self.hd_logits = nn.Linear(nh_bottleneck, target_ensembles[1].n_cells)
```

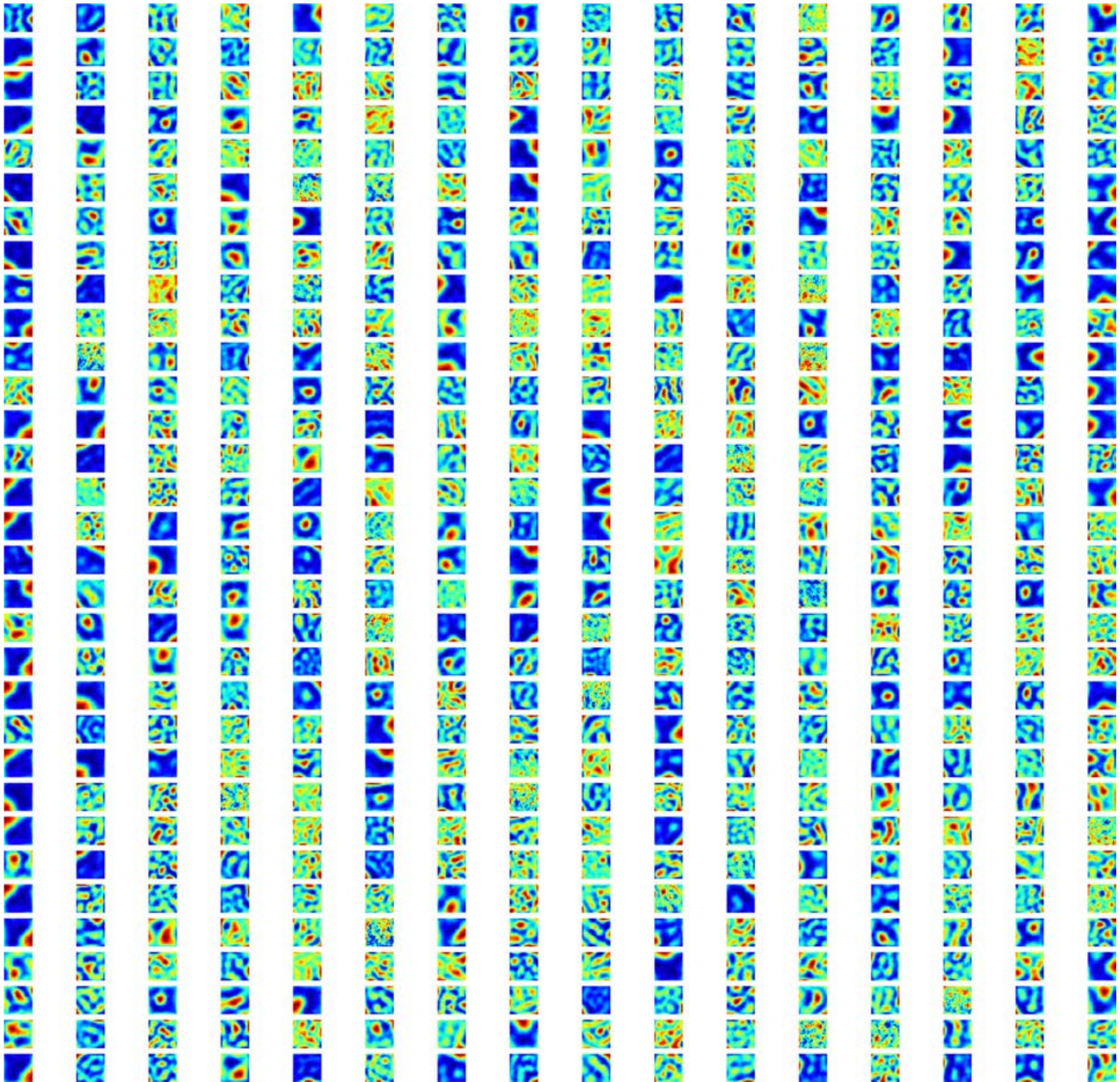
$$\vec{m}_t = W^{hp}\,\vec{c}_t + W^{(hd)}\,\vec{h}_t$$

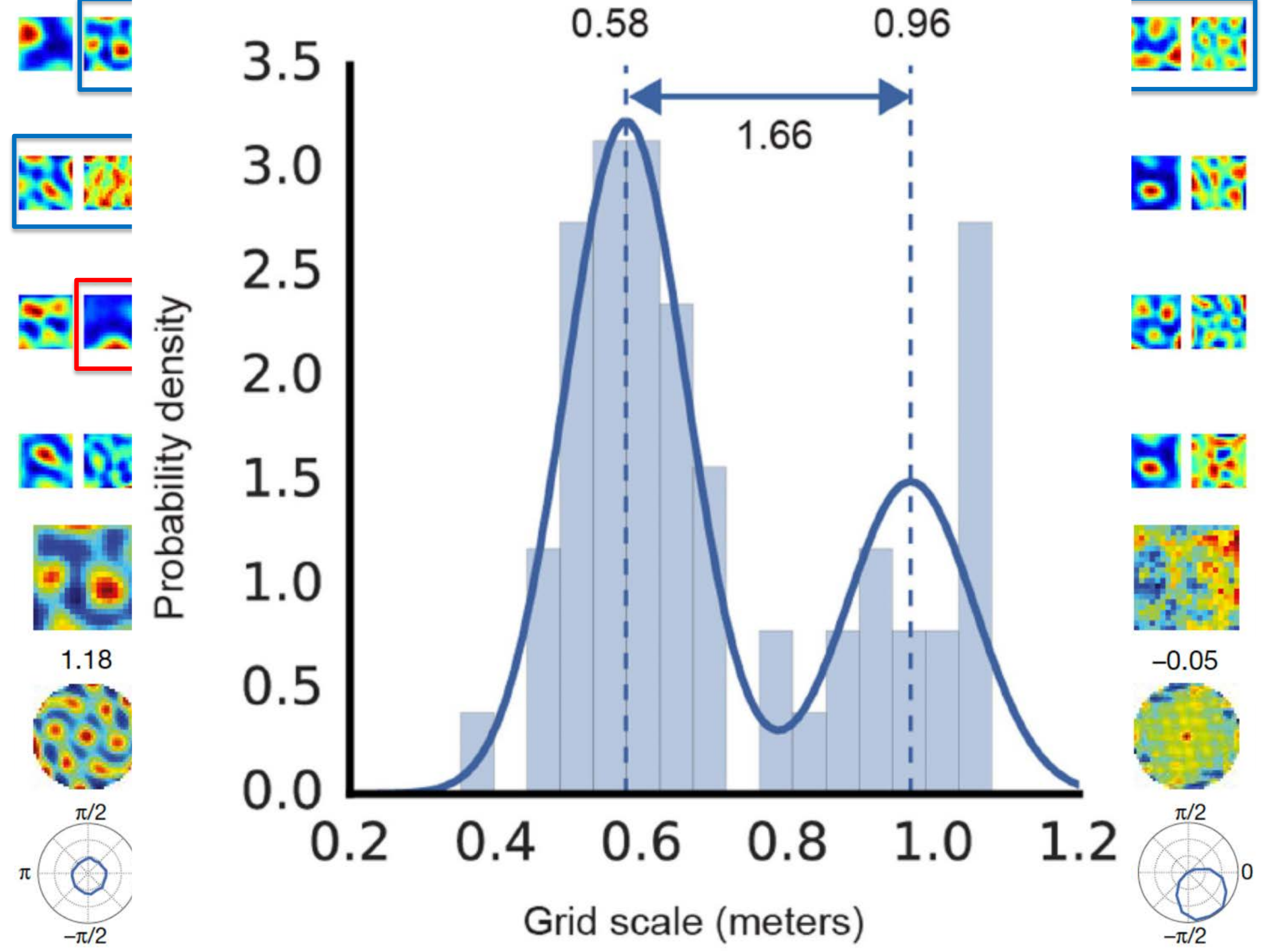
$$\vec{l}_t = W^{cp}\,\vec{c}_t + W^{(cd)}\,\vec{h}_t$$

```python
def forward(self, x, initial_conds):
    init = torch.cat(initial_conds, dim=1)

    init_state = self.state_embed(init)
    init_cell = self.cell_embed(init)

    h_t, c_t = init_state , init_cell
    logits_hd = []
    logits_pc = []
    bottleneck_acts = []
    rnn_states = []
    cell_states = []
    for t in x: # get rnn output predictions
        h_t, c_t = self.rnn(t, (h_t, c_t))

        bottleneck_activations = self.dropout(self.bottleneck(h_t))

        pc_preds = self.pc_logits(bottleneck_activations)
        hd_preds = self.hd_logits(bottleneck_activations)

        logits_hd += [hd_preds]
        logits_pc += [pc_preds]
        bottleneck_acts += [bottleneck_activations]
        rnn_states += [h_t]
        cell_states += [c_t]

    final_state = h_t
    outs = (torch.stack(logits_hd),
            torch.stack(logits_pc),
            torch.stack(bottleneck_acts),
            torch.stack(rnn_states),
            torch.stack(cell_states))
    return outs
```

$$\vec{m}_t = W^{hp}\,\vec{c}_t + W^{(hd)}\,\vec{h}_t$$
$$\vec{l}_t = W^{cp}\,\vec{c}_t + W^{(cd)}\,\vec{h}_t$$

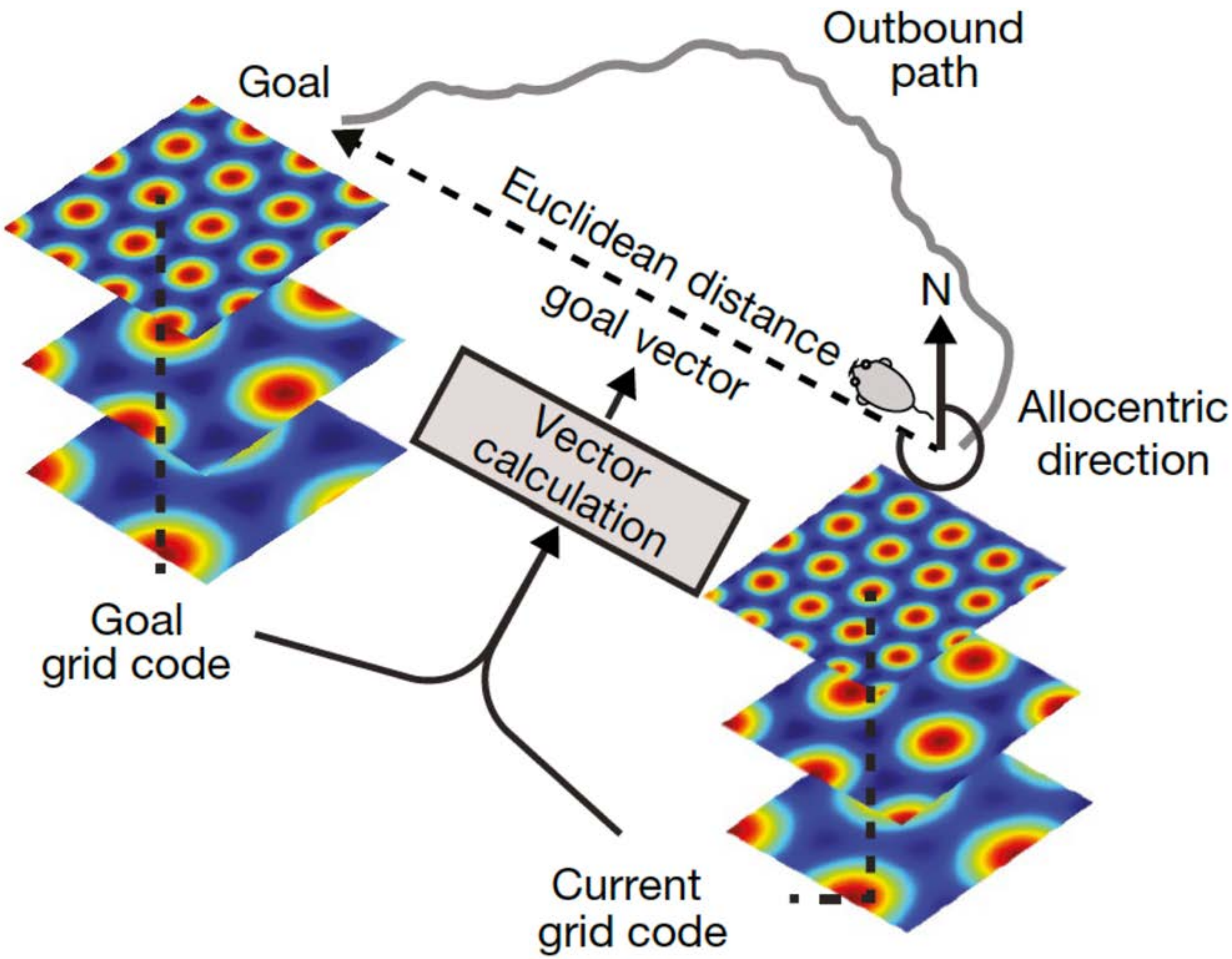
Emergence of grid cell activation pattern in the linear layer



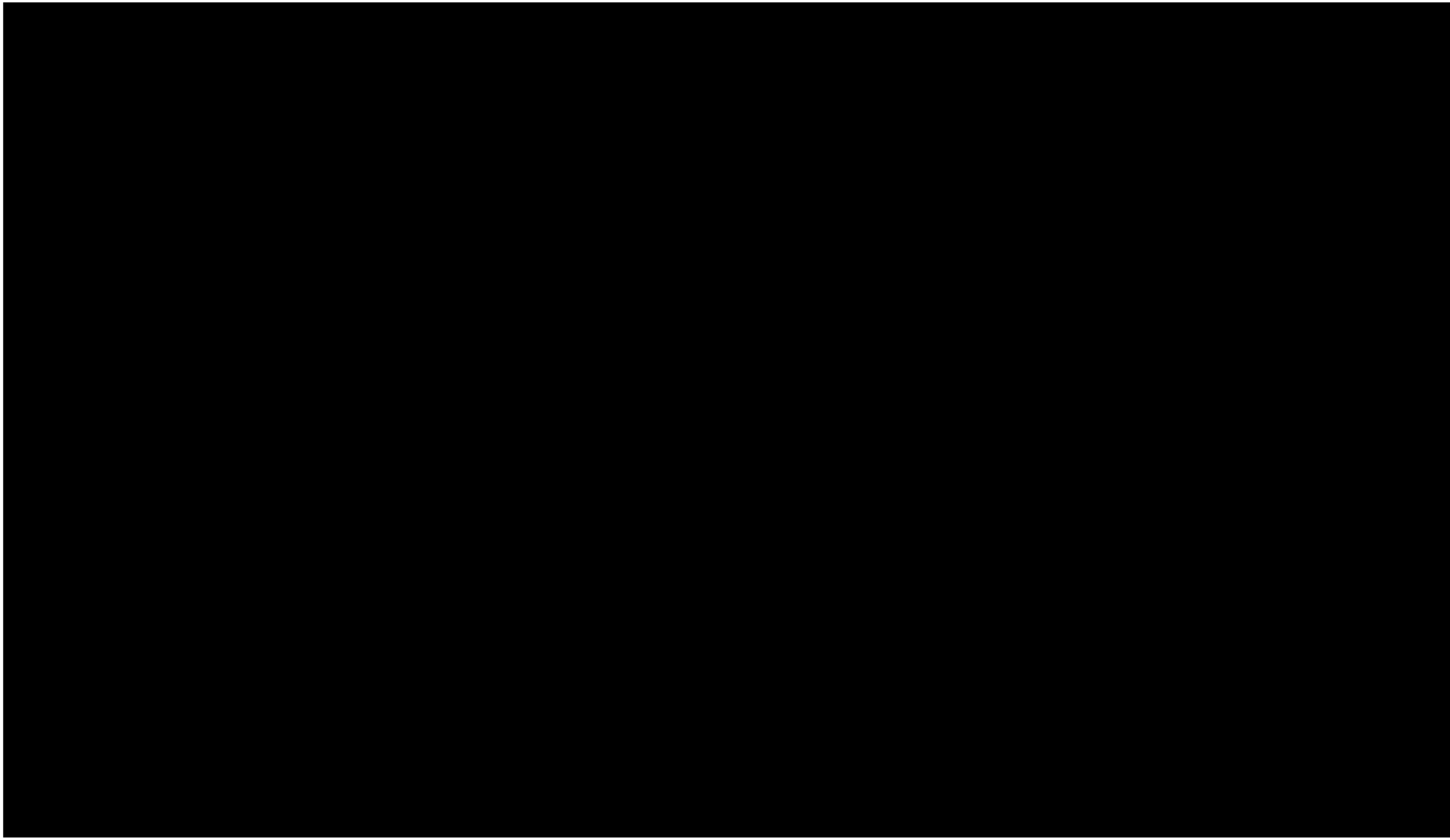


0.58
0.96
1.66
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Probability density
0.2
0.4
0.6
0.8
1.0
1.2
Grid scale (meters)
1.18
−0.05
π/2
π
−π/2
0

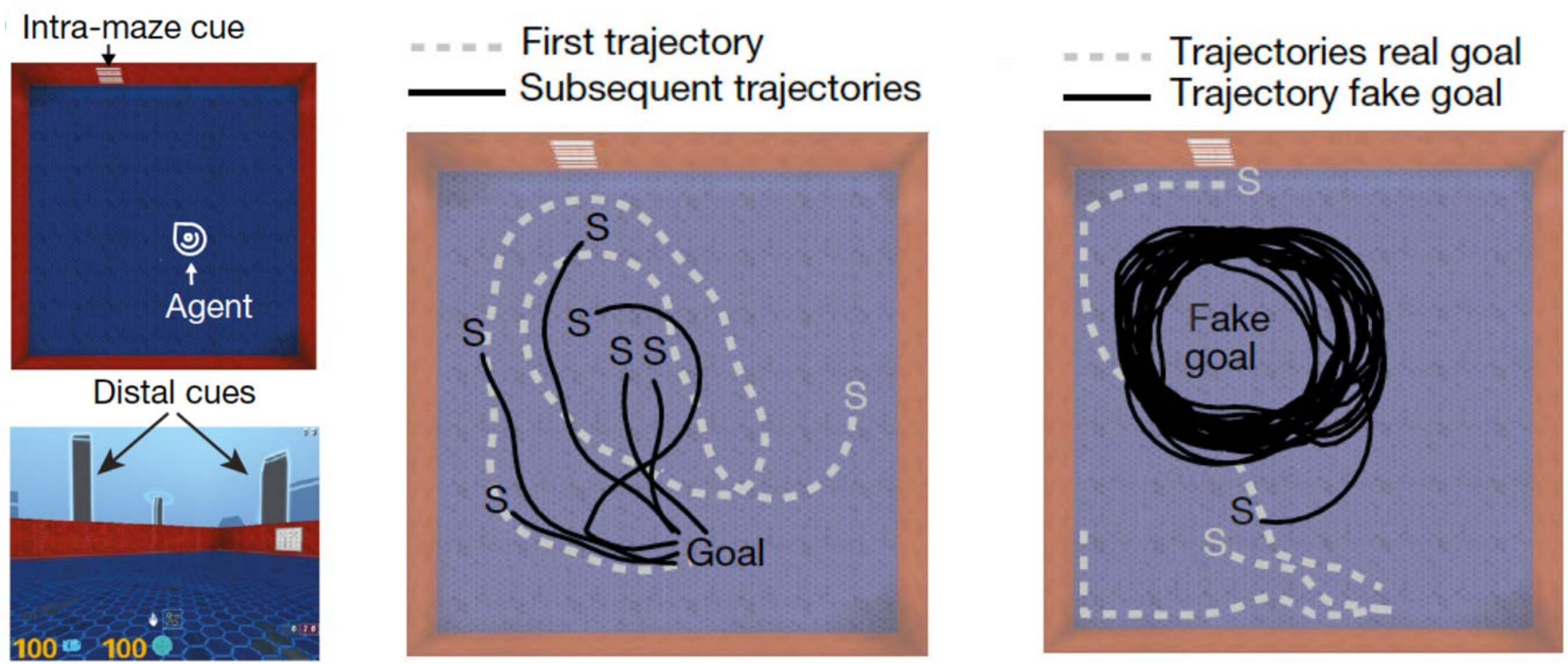
Vector-based navigation



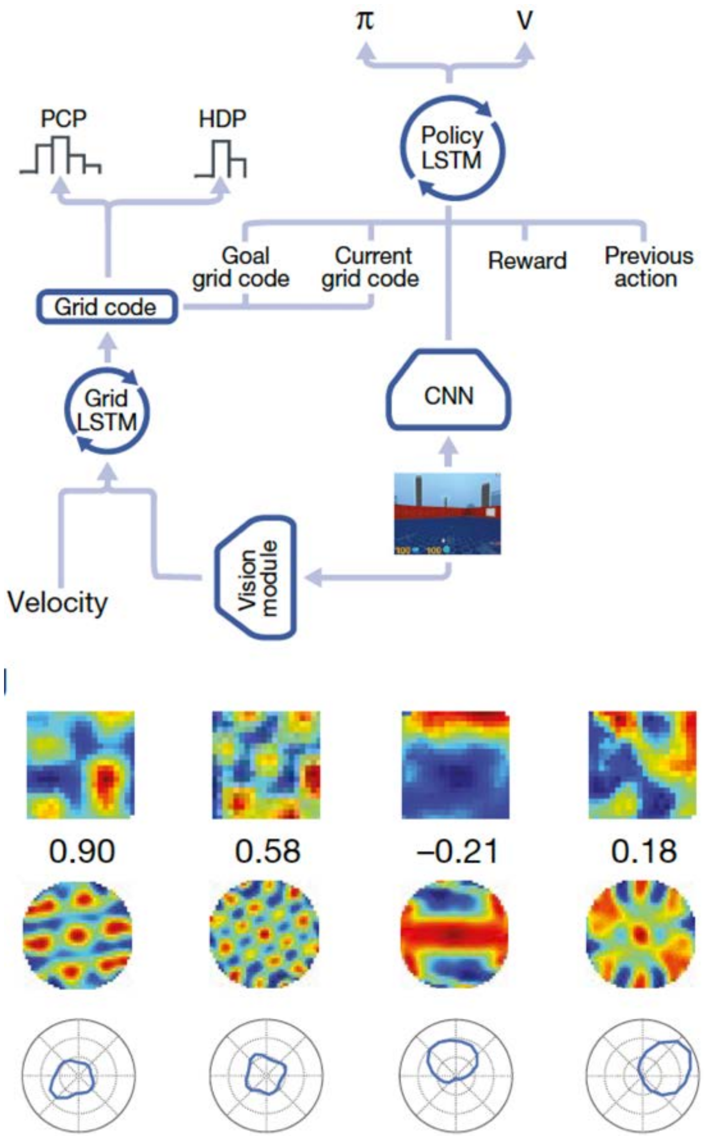
Morris 수중 미로 (water maze) 과제

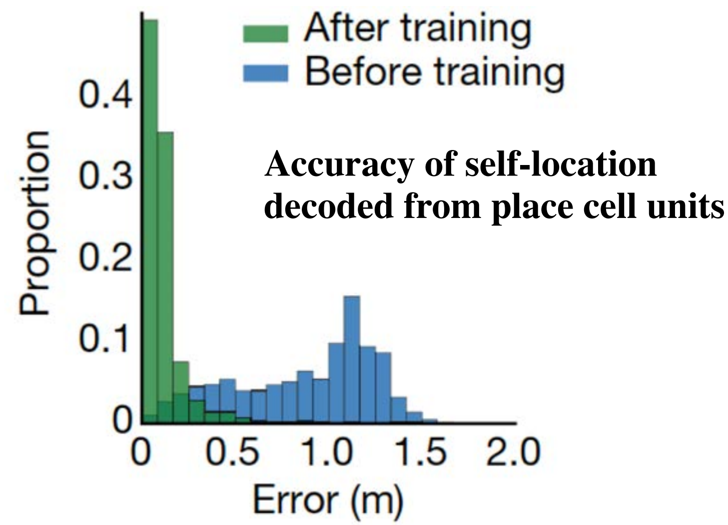


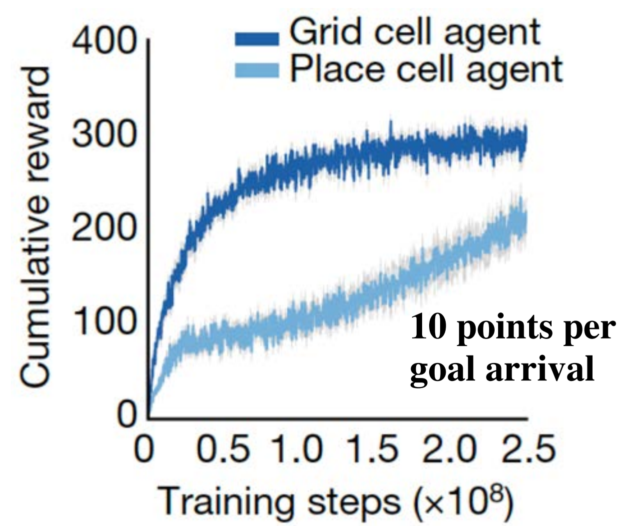
Learning to navigate in a simulated water maze

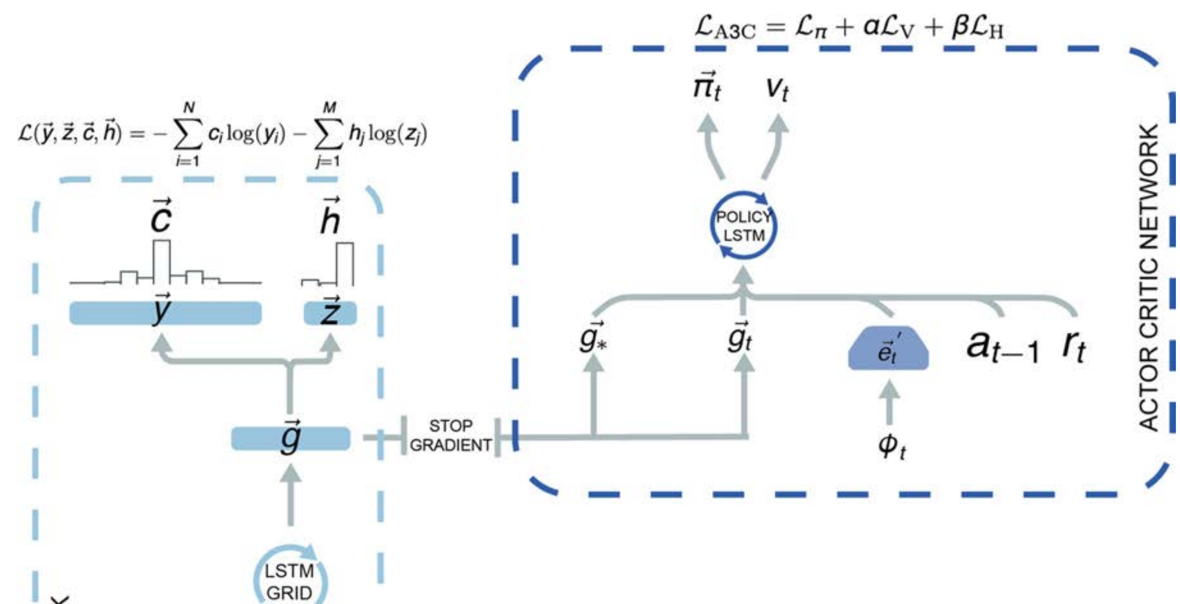


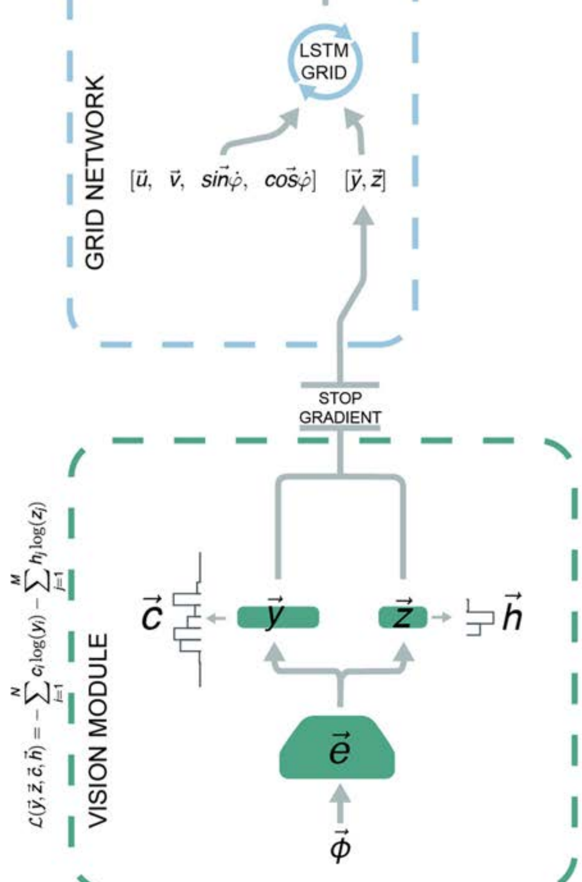
Reinforcement learning using grid-cell network



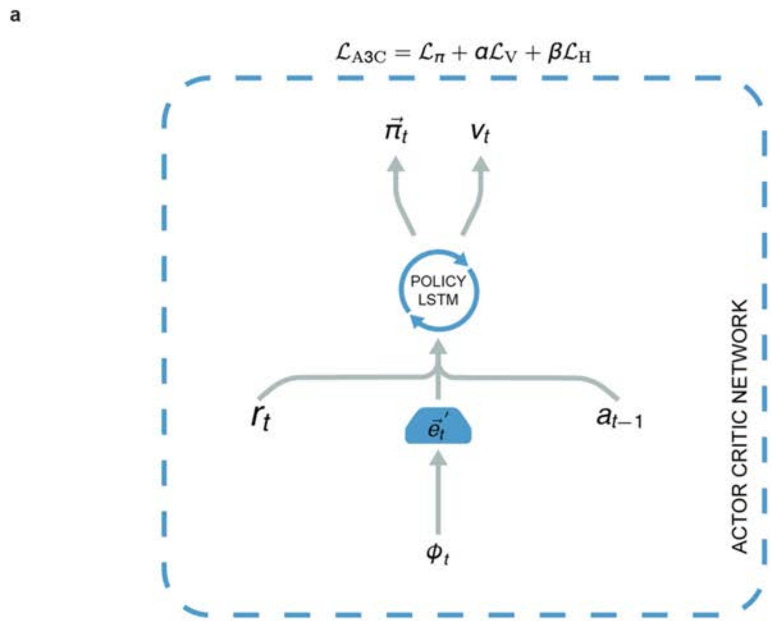




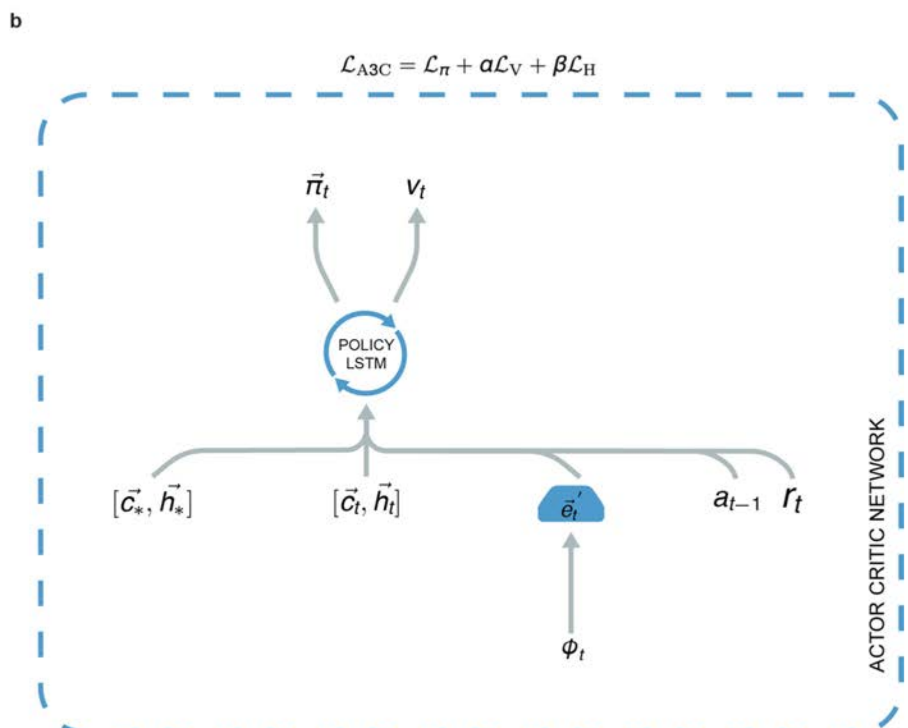




Training algorithms. We used the A3C algorithm[41], which implements a policy $\pi(a|s,\theta)$ and an approximation to its value function $V(s,\theta)$ using a neural network parameterised by θ. A3C adjusts the network parameters using n-step lookahead values, $\hat{R}_t = \sum_{i=0\ldots n-1} \gamma^i r_{t+1} + \gamma^n V(s_{t+n}, \theta)$, to minimize: $\mathcal{L}_{A3C} = \mathcal{L}_\pi + \alpha\mathcal{L}_V + \beta\mathcal{L}_H$, where $\mathcal{L}_\pi = -\mathbb{E}_{s_t\sim\pi}[\hat{R}_t]$, $\mathcal{L}_V = -\mathbb{E}_{s_t\sim\pi}[(\hat{R}_t - V(s_t,\theta))^2]$, $\mathcal{L}_\pi = -\mathbb{E}_{s_t\sim\pi}[H(\pi(\cdot|\, s_t,\theta))]$ where $\mathcal{L}_H$ is a policy entropy regularization term (see Supplementary Methods for details of the reinforcement learning approach). The grid cell network and the vision module were trained with the same loss reported for the supervised learning: $\mathcal{L}(\vec{y}, \vec{z}, \vec{c}, \vec{h}) = -\sum_{i=1}^{N} c_i \log(y_i) - \sum_{j=1}^{N} h_j \log(z_j)$.

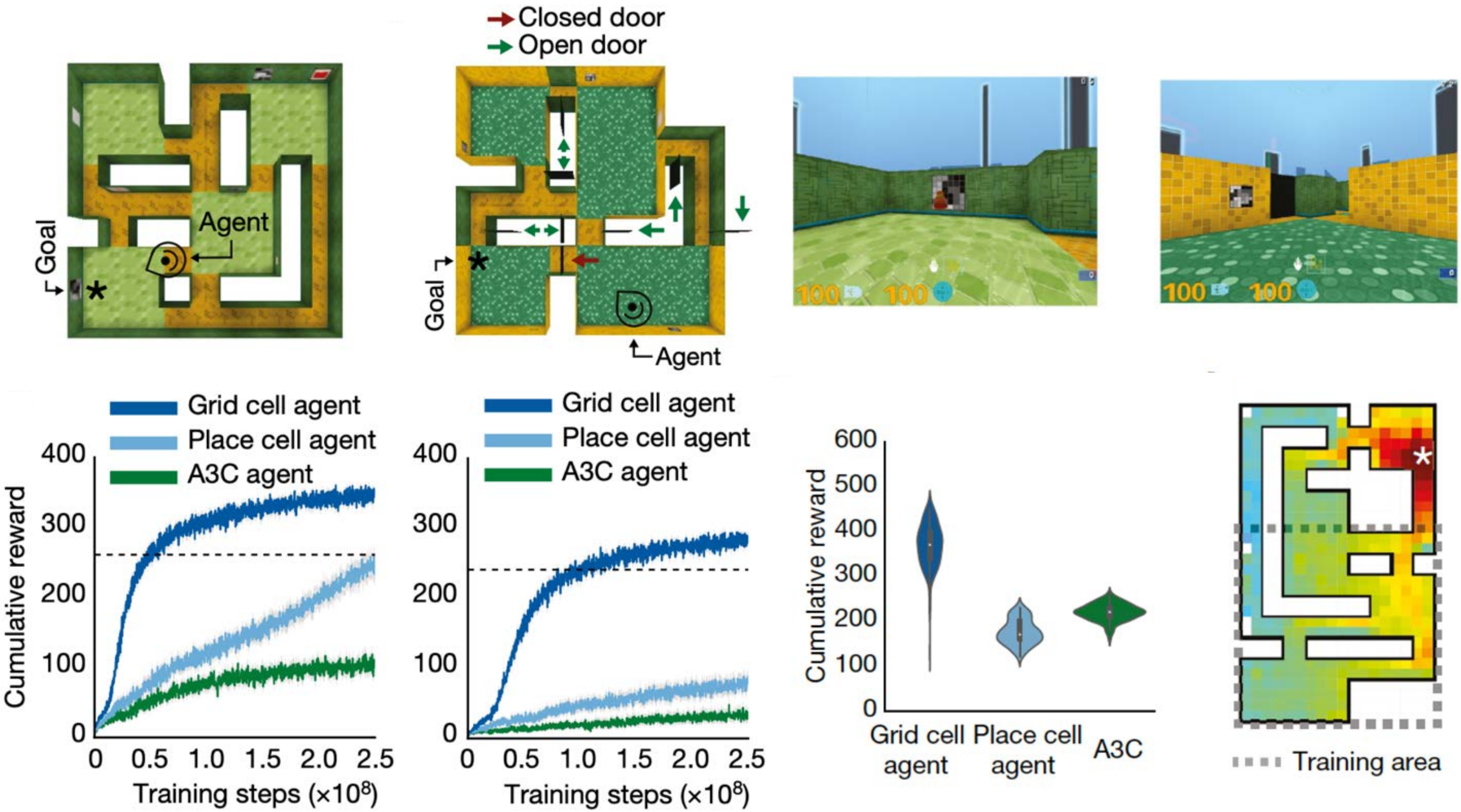


A3C agent

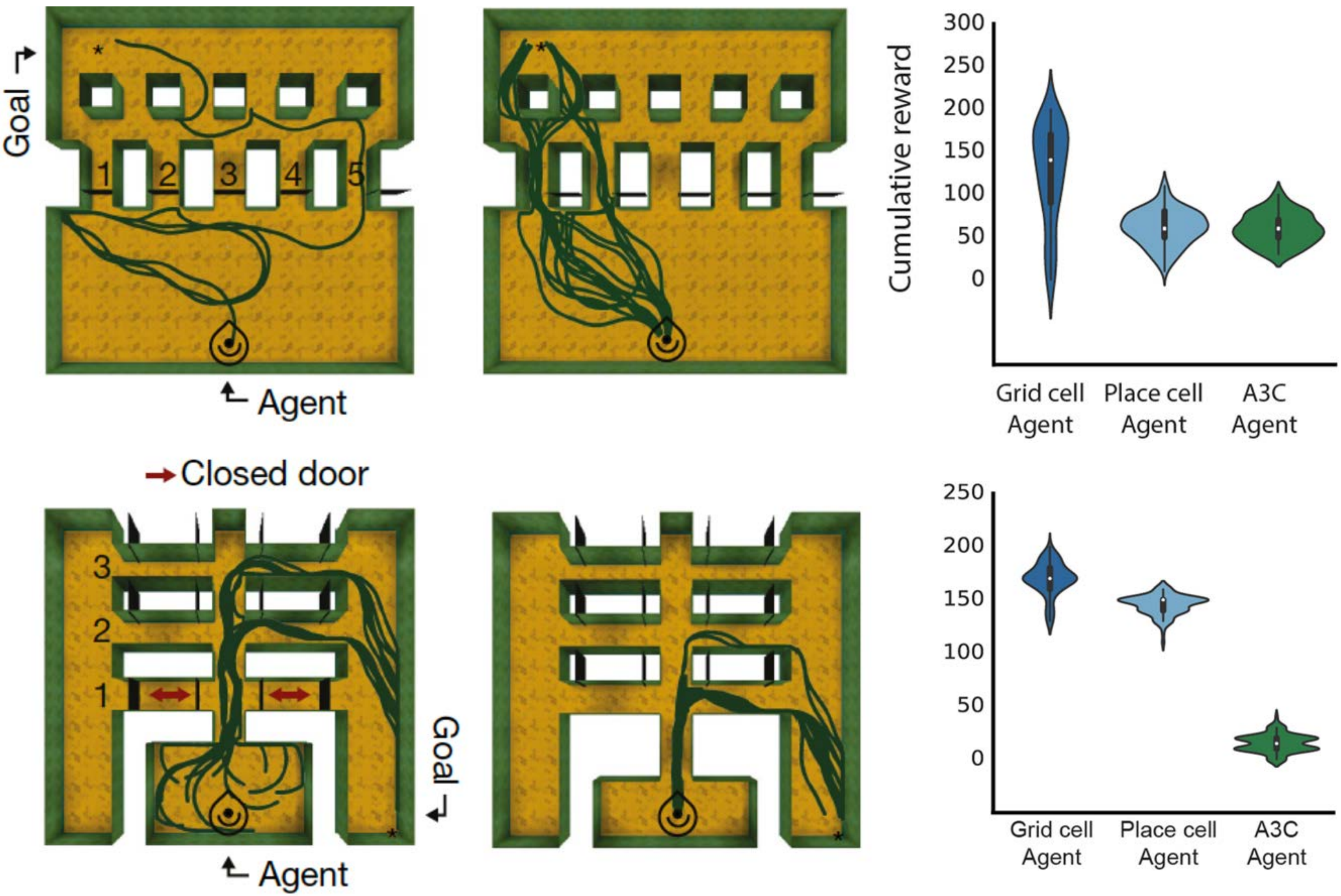


Place-cell agent

Navigation in challenging environments



Finding a shortcut with changed environment



Review comments by Edvard Moser, Nobel prize winner

In this incredible study, Banino and colleagues found that an artificial agent with access to only a few sensory-motor cues spontaneously formed grid cells with properties that were remarkably similar to those in brain. Once the artificial agent formed mature grid cells, it could solve spatial tasks and generate novel routes and shortcuts, as seen in animals. This truly special paper highlights the power of studying artificial intelligence for understanding our own brains.

Banino and colleagues began by training their artificial agent to path integrate. The agent was equipped with a simplistic neural network where information about linear and angular velocity fed into a recurrent layer, which in turn projected to a linear layer and finally onto separate units that represented place and head direction. This architecture took several very basic cues from biology and behavior, but importantly was not a faithful reproduction of the hippocampal-entorhinal circuit of mammals. Instead, the goal of the artificial agent was simply to learn the path integration task, and the neural network was free to converge on the optimal solution, whatever that happened to be. Remarkably, many units in the linear layer became like the grid cells found in the medial entorhinal cortex (MEC) of mammals -- these cells had clearly defined firing fields that were arranged in a hexagonal pattern in both square and circular enclosures.

The findings above would alone make for a landmark study, but hexagonality itself is a stable state of many systems (for example, the famous Turing patterns {2,3,4}) and likewise border- and directional- tuning are not so surprising. What is truly astonishing in the Banino study is how closely their properties match those found in the brain in non-trivial ways. First, the percentage of grid cells (25%), head direction cells (10%) and border cells (8.7%) in the artificial network is very close to what is seen in the MEC. Second, the artificial grid cells followed a modular organization, such that the spatial scale of the grid pattern clustered in the population with a fixed ratio between modules. This finding confirms previous theoretical work that the modular organization of grid cells maximizes the spatial resolution at the population level while using the least number of units {5,6}. Third, the directionality of the head direction units was not random but showed a six-fold symmetry, echoing the six-fold symmetry found in the human brain during active navigation {7}.

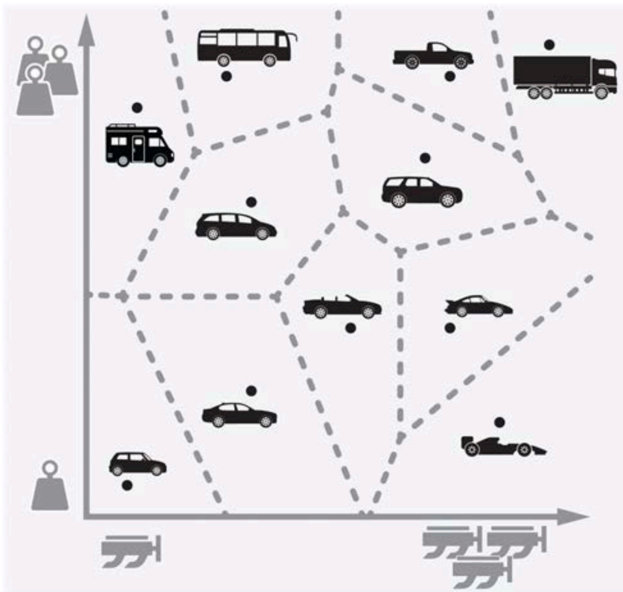
The authors next looked at the behavior of the artificial agent – and this is perhaps the most astonishing result. The agent was able to efficiently move to a hidden goal, navigate through challenging environments and take shortcuts. The latter feat is a notoriously difficult task in robotics and suggests that the grid code, which was trained for the simple purpose of performing path integration, can be exploited to perform efficient vector-based navigation (5). Together, these results strongly suggest that the brain’s grid code of space (at both the single cell and population level) is not merely an artifact but rather an optimal solution for how to navigate through an environment.

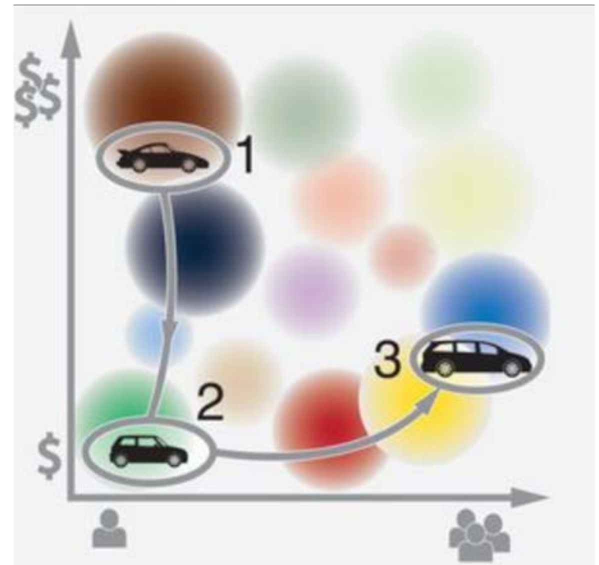
The paper highlights how the fields of neuroscience and artificial intelligence can be mutually reinforcing. In this case, the artificial agent demonstrated the power of the grid code for facilitating spatial navigation. In the future, though, such artificial agents might actually create completely new predictions for us to look for in nature.

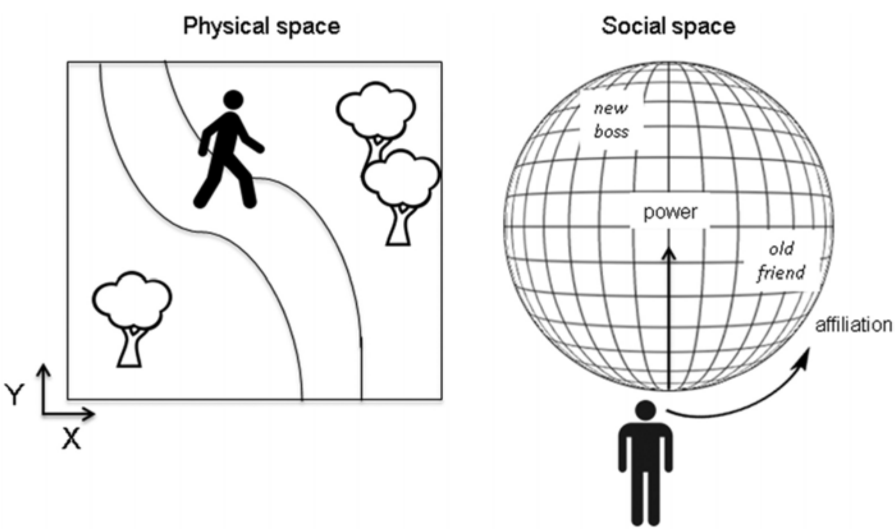
Thank you

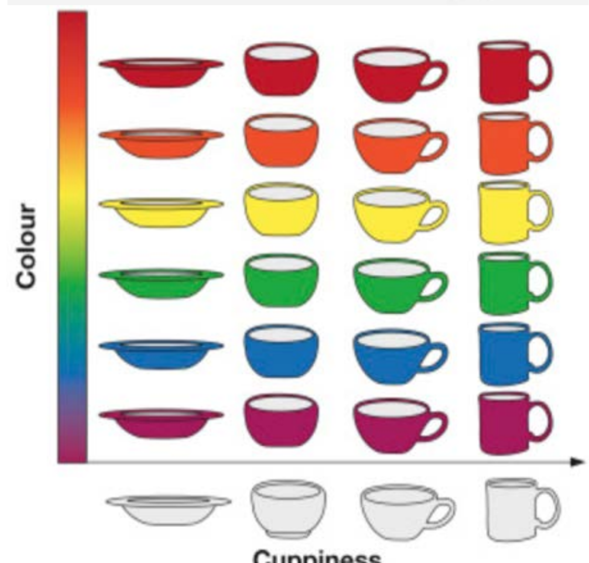
Appendix

Cognitive map in the hippocampus

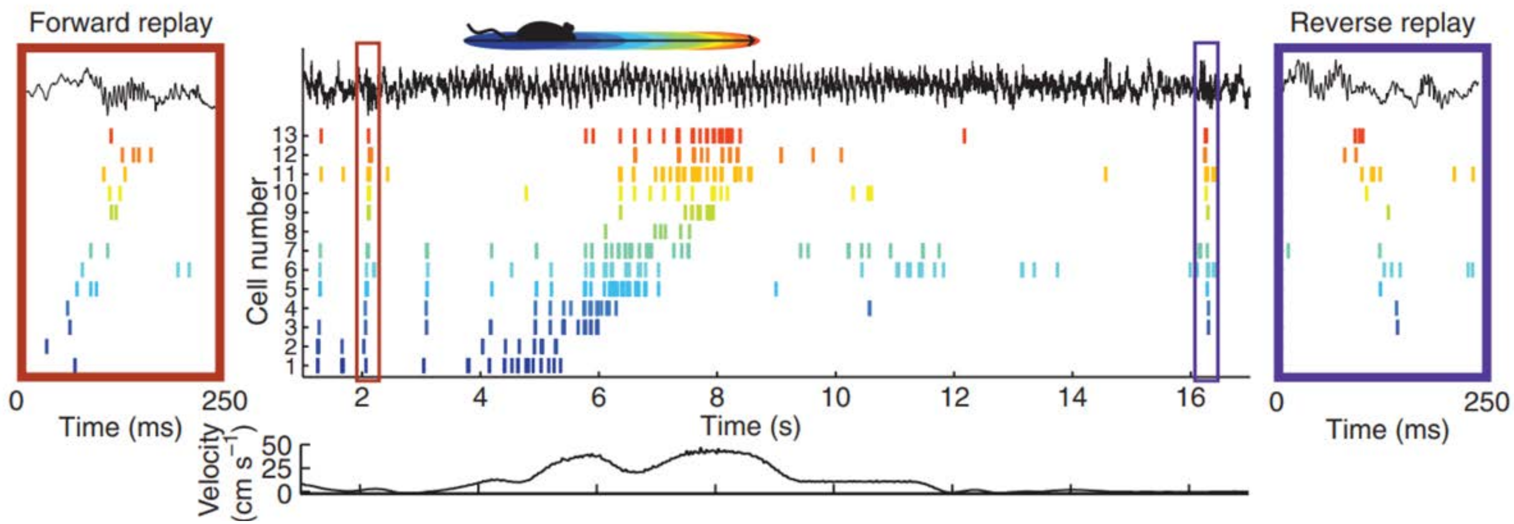


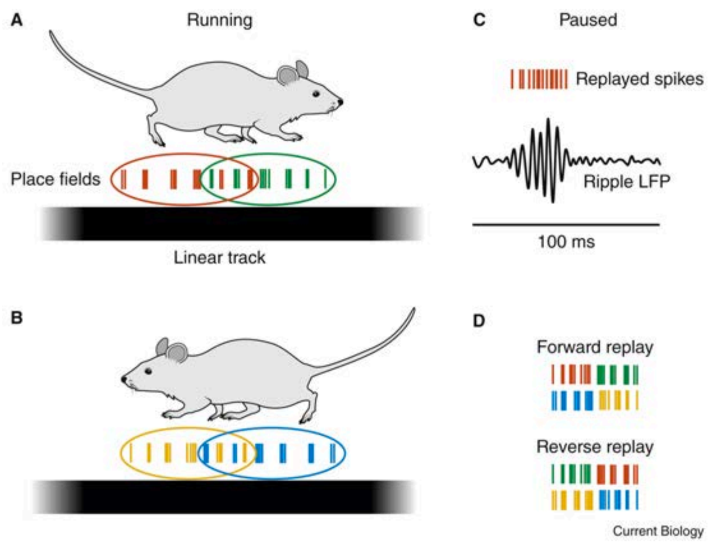


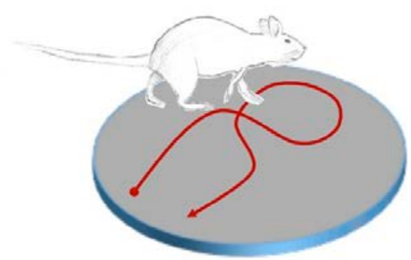


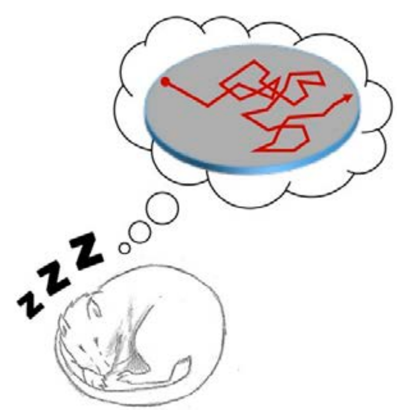


Hippocampus replay



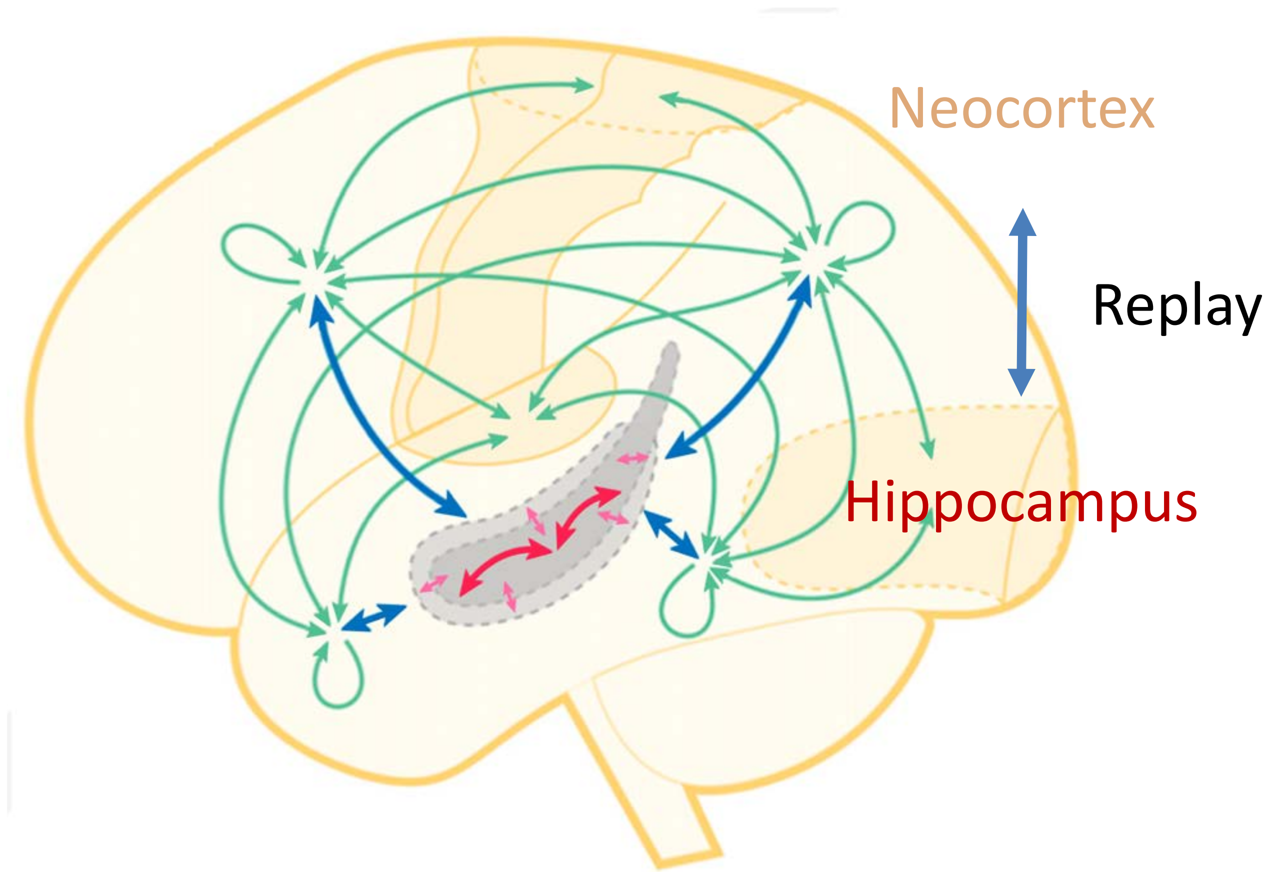






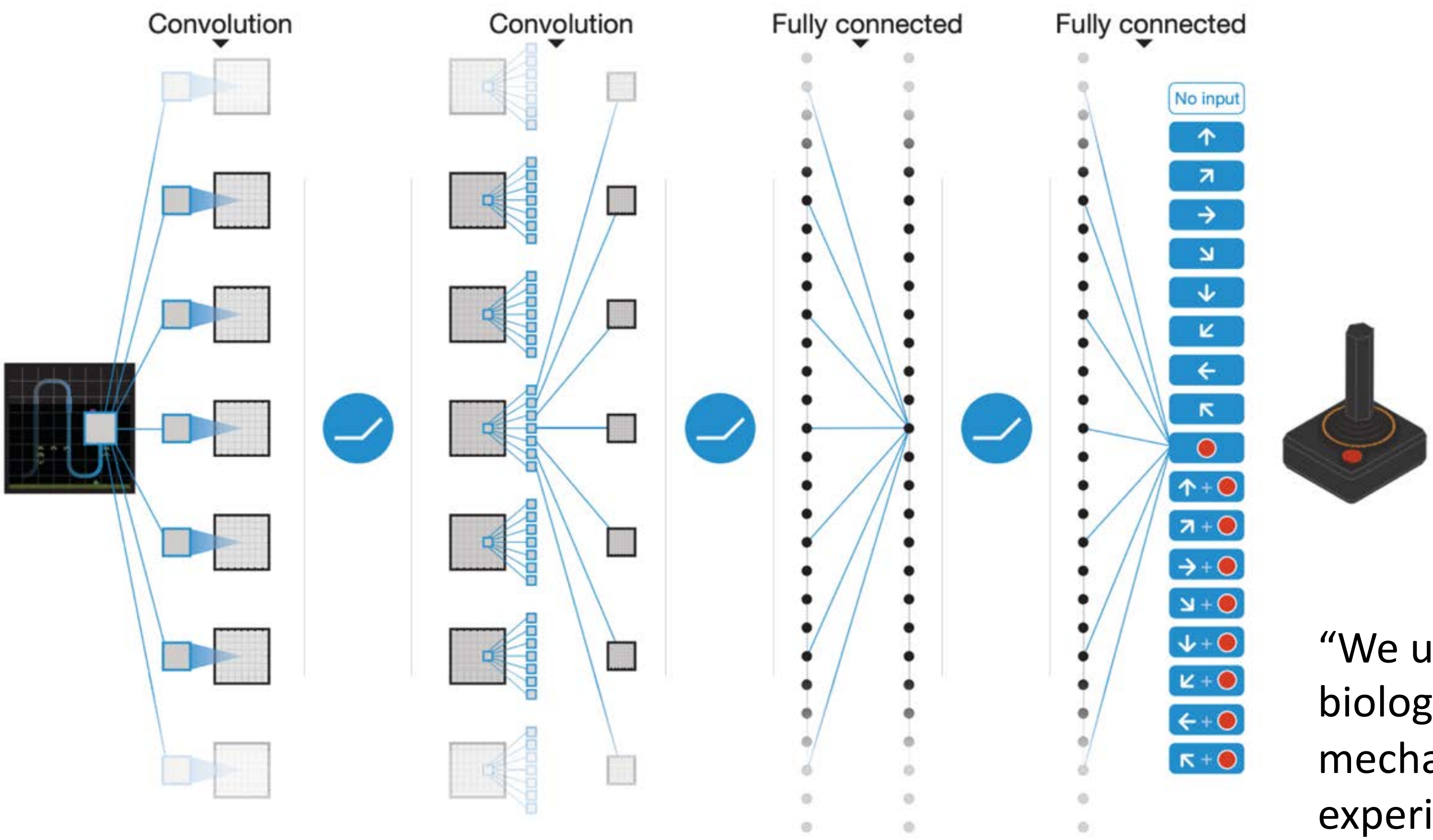
© Federico Stella

Complementary Learning System



Google DeepMind   Kumaran et al., *Trend. Cogn. Sci.*, 2016

Deep Q-learning (DQN)

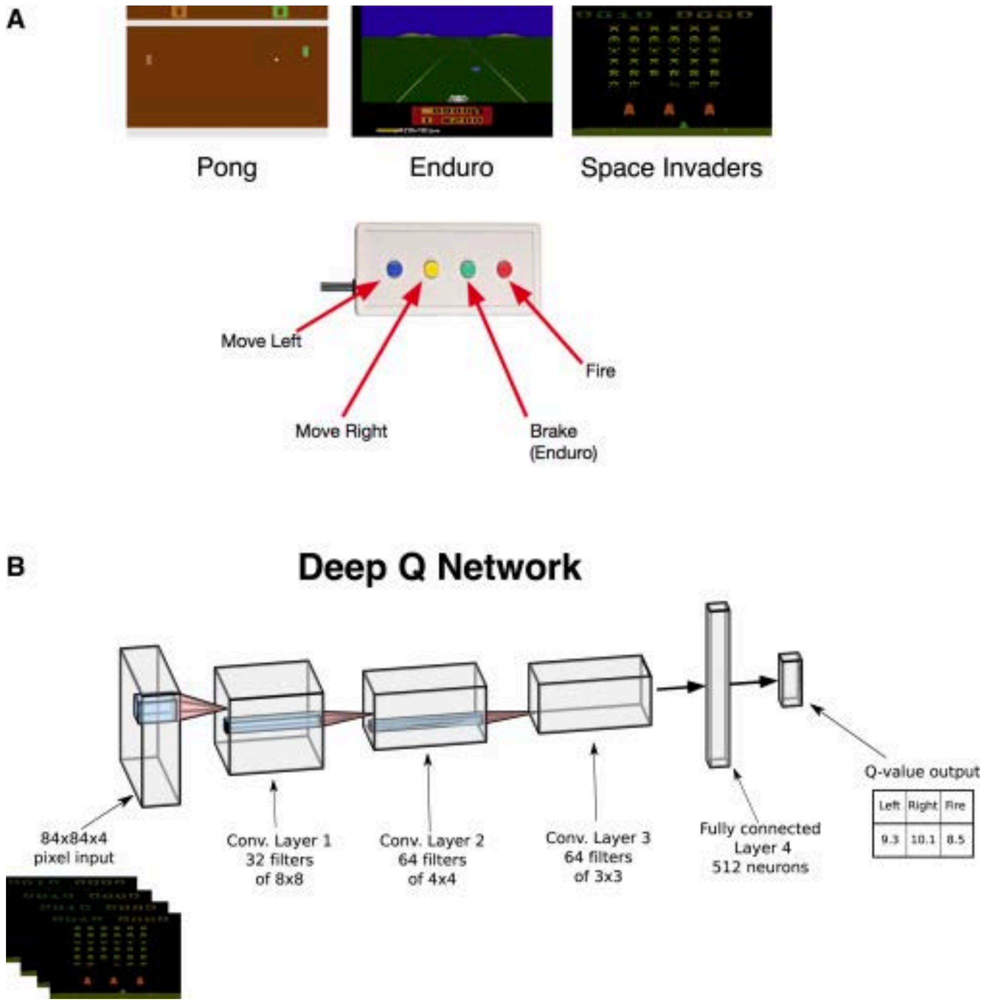


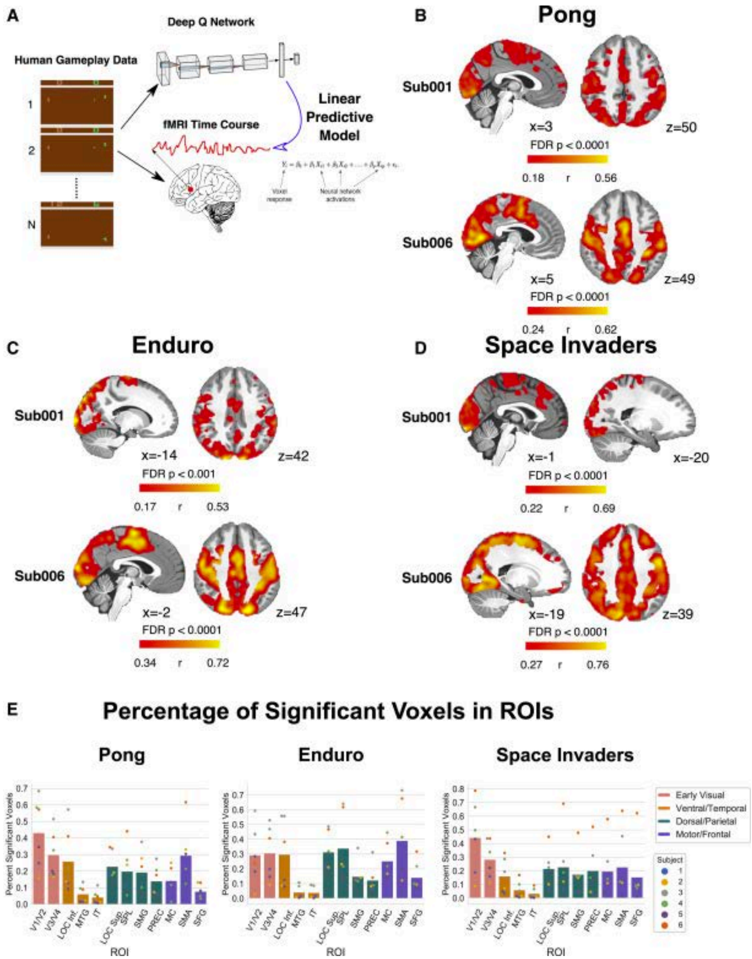
“We used a biologically inspired mechanism termed experience replay”

Google DeepMind

Mnih et al., *Nature*, 2015

DQN in the brain





Cross et al., *Neuron* 2021

Computational Learning & Memory Neuroscience Lab

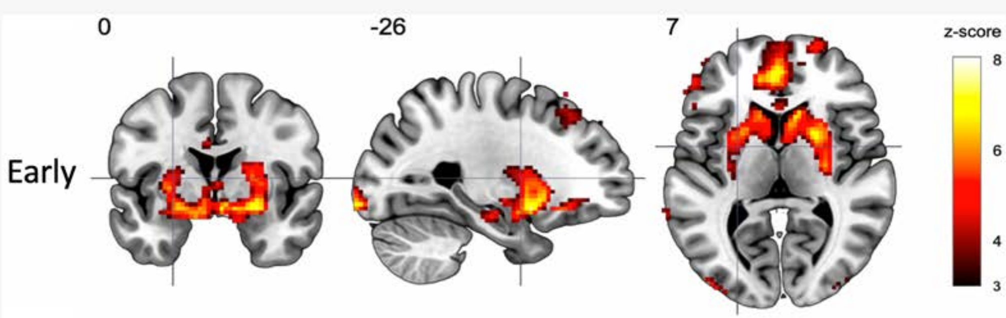
Home Research Publications Journal Club People News Media Resources Contact Us

Computational Learning & Memory Neuroscience Lab

Our lab investigates the neural mechanisms underlying human learning & memory. For this, we take a combined approach to computational modeling, behavioral experiment, fMRI, and noninvasive neuromodulation such as transcranial magnetic stimulation (TMS). Specifically, we pursue to (1) understand how functional brain network evolves as a process of learning using computational methods and (2) how learning could be modulated by TMS. Depending on learning tasks, the nature of involved memory and learning mechanisms could be differentiated, e.g., declarative memory versus procedural memory and supervised learning vs. reinforcement learning.

In the future, our team plans to introduce MR-compatible TMS and focused ultrasonic stimulation (FUS) system for deep brain structure in CNIR to facilitate synergetic collaboration with the advanced MR imaging team.
We expect the concurrent TMS-fMRI experiment would further reveal causal relationships between brain networks and their implementing functions. Building on basic scientific findings, we may develop more efficient clinical protocols for neurorehabilitation of patients with stroke and Parkinson's diseases.
For this translational research, we are collaborating with clinicians in Bundang Seoul National Hospital and Samsung Medical Center. Our laboratory located in Hanyang University is currently collaborating with the center for neuroscience imaging research (CNIR) in the Institute for Basic Sciences (IBS) funded by the Korean government.

To learn more about our research, please see our Publications and Contact Us.



http://clmnlab.com
sungshinkim@hanyang.ac.kr