Coordinating Control and Planning for Navigation on Simplicial Complex Attractors

Mien Brabeeba Wang, Nancy Lynch, MIT CSAIL, USA
Michael Halassa, Tufts University School of Medicine, USA

Tufts University School of Medicine

MIT CSAIL

Poster No. 1-037

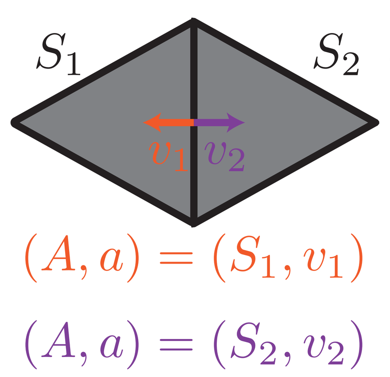
Introduction

Animals and humans navigate complex spaces—both physical and conceptual—using internal cognitive maps, thought to be encoded as neural manifolds [1]. However, **how the brain represents and navigates manifolds with complex geometry** remains unclear.

We propose that neural circuits approximate these manifolds by **stitching together piecewise linear patches**, known as **simplicial complexes (SCs)**. A local feedback **control mechanism enables navigation within a single patch**, while a global **hierarchical planning system computes efficient paths between patches**, supporting flexible navigation.

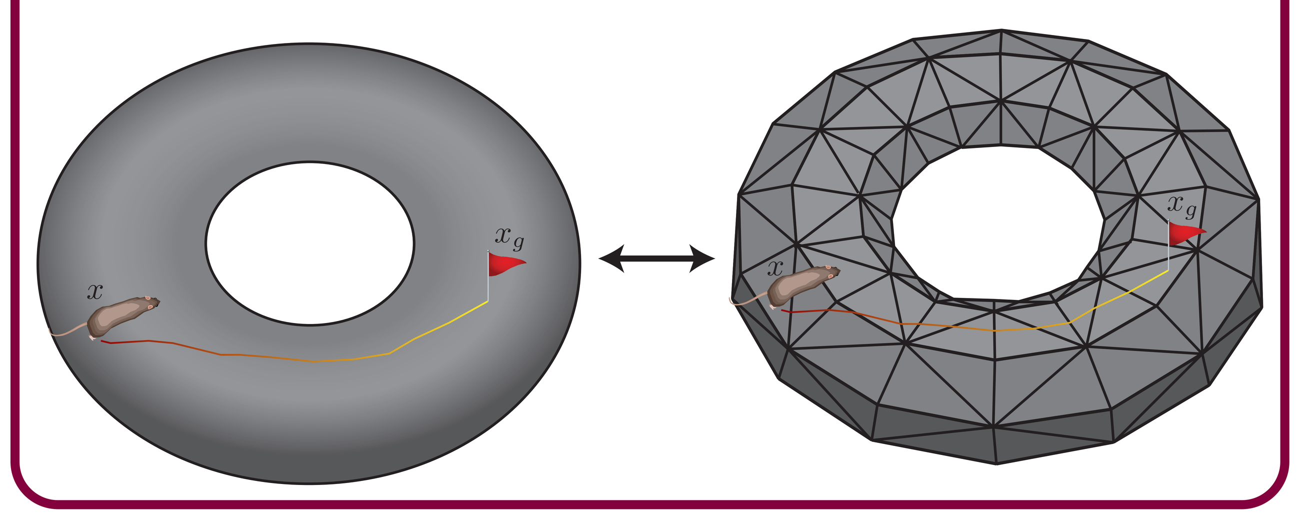
Navigation on simplicial complexes

An n-**simplicial complex (SC)** S is a collection Σ of m-simplexes, where $m \le n$, **glued together** in such a way that **the intersection of any two simplexes is also a simplex**.

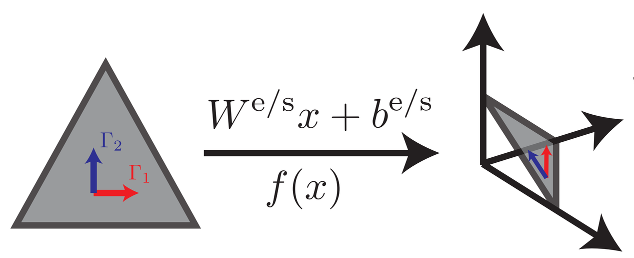


Given a basis $\Gamma^\sigma \in \mathbb{R}^{\dim(\sigma)\times\dim(\sigma)}$ for each simplex $\sigma \in \Sigma$, the action dynamics on the SC are described as: $\tau_e \dot{x} = \sum_{\sigma\in\Sigma} \mathbf{1}_{x\in\sigma,\sigma=A}\Gamma^{(\sigma)} a_{1:\dim(\sigma)}$, where $a : \mathbb{R}^n \times \mathbb{R}$ and $A : \Sigma \times \mathbb{R}$ represent the time-varying actions and the simplex to act on.

The objective of navigation is to find actions $(A(x,x_g), a(x,x_g))$ such that, for a given goal $x_g \in S$, x converges to x_g efficiently.



Learning to represent a simplex attractor



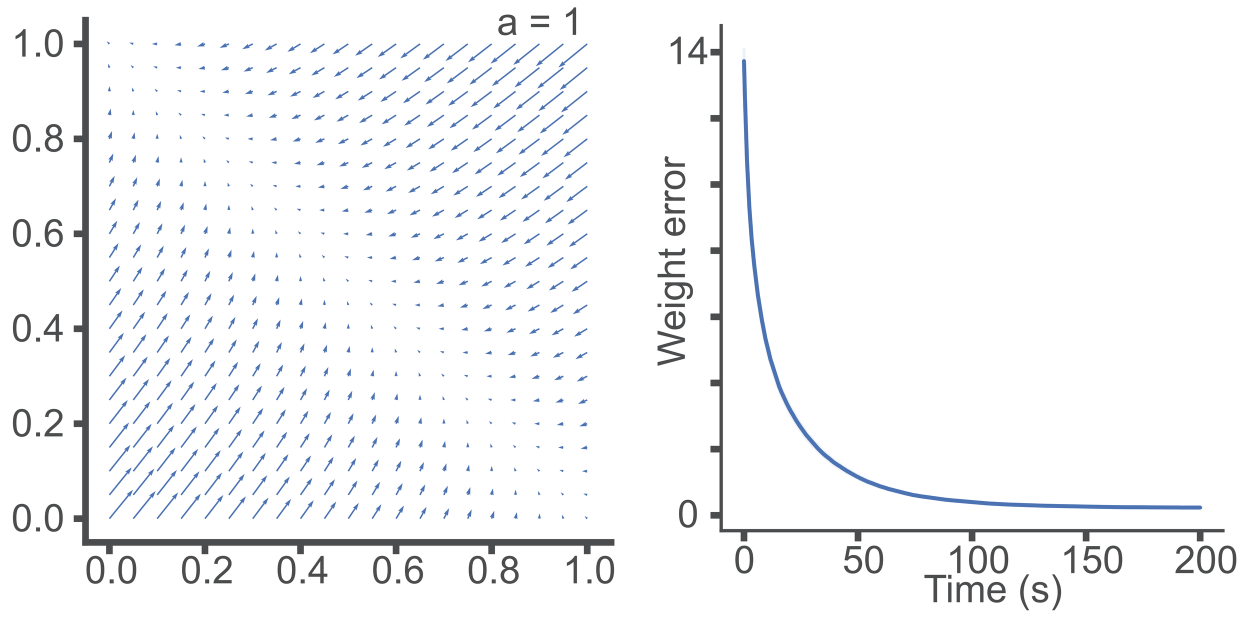
Consider an n-simplex Δ_n with action basis Γ and a linear input function $f(x) = W^{e/s}x + b^{e/s}$.

State neurons y^s represent the simplex. It receives inputs from motor neurons y^m and its dynamic is:

$$\tau_n \dot{y}^s = -y^s + \left[W^s y^s + W^{m/s} y^m + b^s\right]_+ .$$

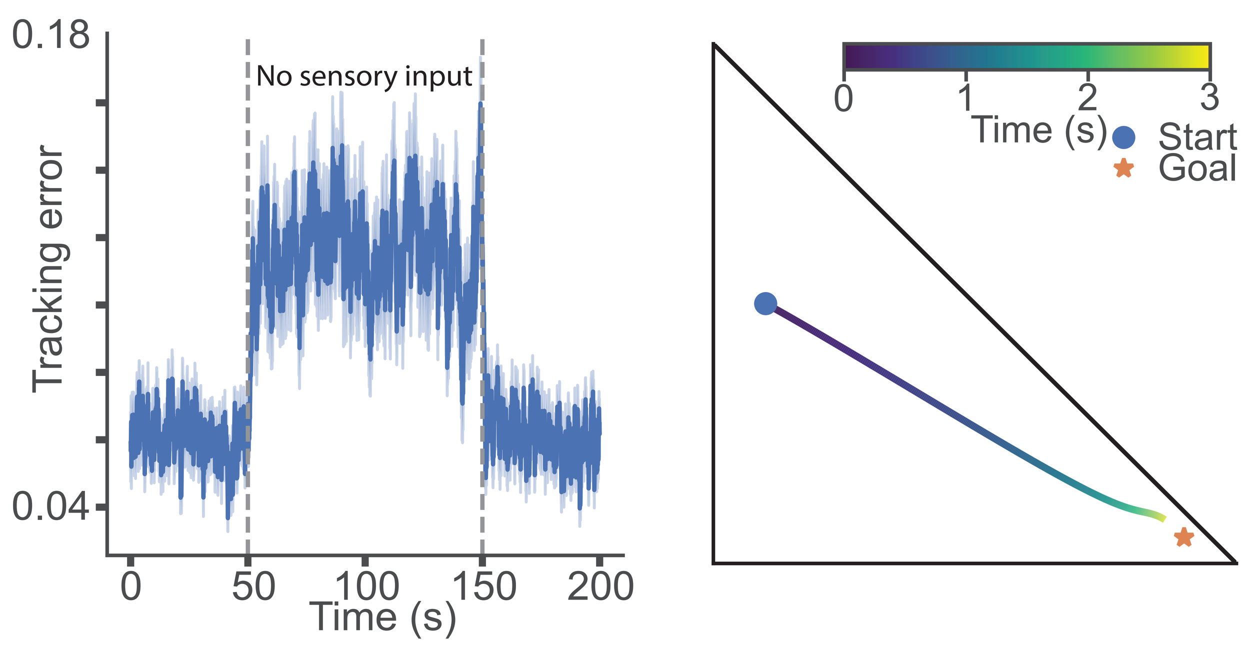
We can prove the following theorem:

Let $W^s = -(b^s - \hat{b}^{e/s})(\hat{b}^{e/s})^\dagger + W^{e/s}(W^{e/s})^\dagger$, $W^{m/s} = \frac{\tau_n}{\tau_e} W^{e/s}\Gamma$, and $a = y^m$. Here, $\hat{b}^{e/s} = (I - W^{e/s}(W^{e/s})^\dagger)b^{e/s}$. Then, the **dynamics of y^s corresponds to the pushforward of dynamics of the simplex** $f(\Delta_n)$. When $y^m = 0$, y^s in particular forms an **attractor of the simplex** $f(\Delta_n)$.



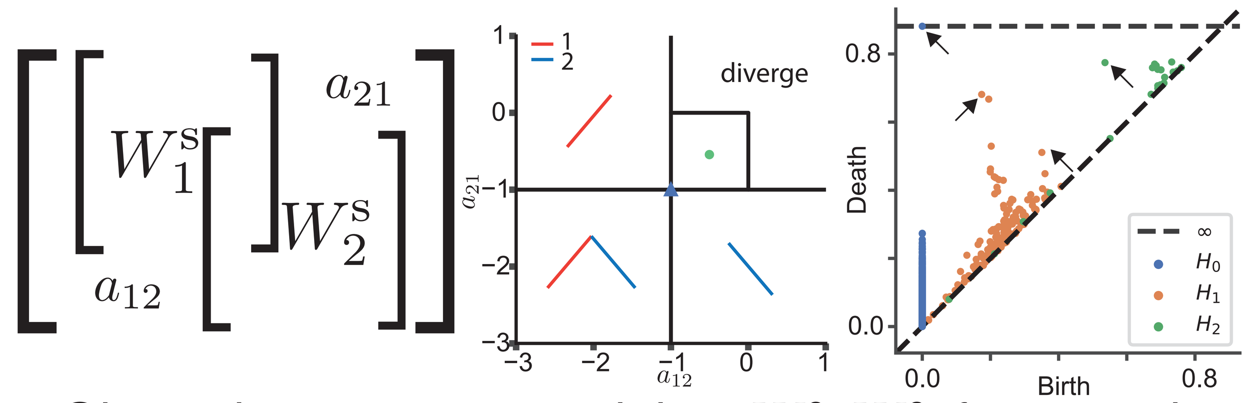
To learn W^s, we show that the **following learning rule can exponentially converge to the desired attractor weight**: $\tau_w \dot{W}^s = -e^s(e^{e/s})^\top + e^{e/s}(e^{e/s})^\top - \kappa W^s$ where $e^s, e^{e/s}$ is the eligibility trace of $y^s, f(x)$.

Navigation on a simplex via control



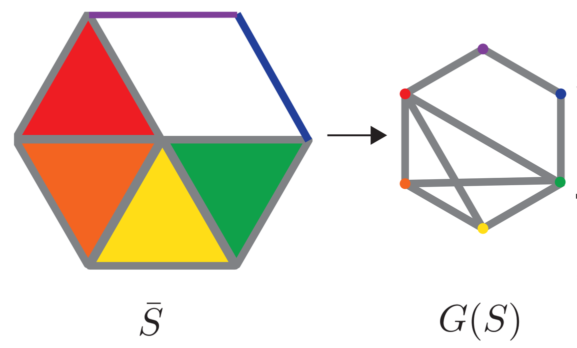
Motor neurons y^m receive inputs from both state neurons y^s and goal neurons y^g and their connections can be learned such that $y^m = \tau_e(W^{e/s}\Gamma)^\dagger(y^g - y^s)$. In this case, the **dynamics of simplex follow** $\tau_e \dot{x} = \tau_e(x_g - x)$, **driving** x **exponentially fast toward** x_g.

Stitching simplex together to form SC



- Given the recurrent weights W_1^s, W_2^s for two simplexes, if they **share neurons with coincident connections**, we can form an union weight $W_{1\cup 2}^s$.
- If the **mutual inhibition between simplexes are large, then one can stitch them together**
- We represent a torus by stitching triangles together. The persistent topology of the neural data reveals **one connected component** (H_0), **two loops** (H_1), and **one void** (H_2), **capturing toroidal topology**.

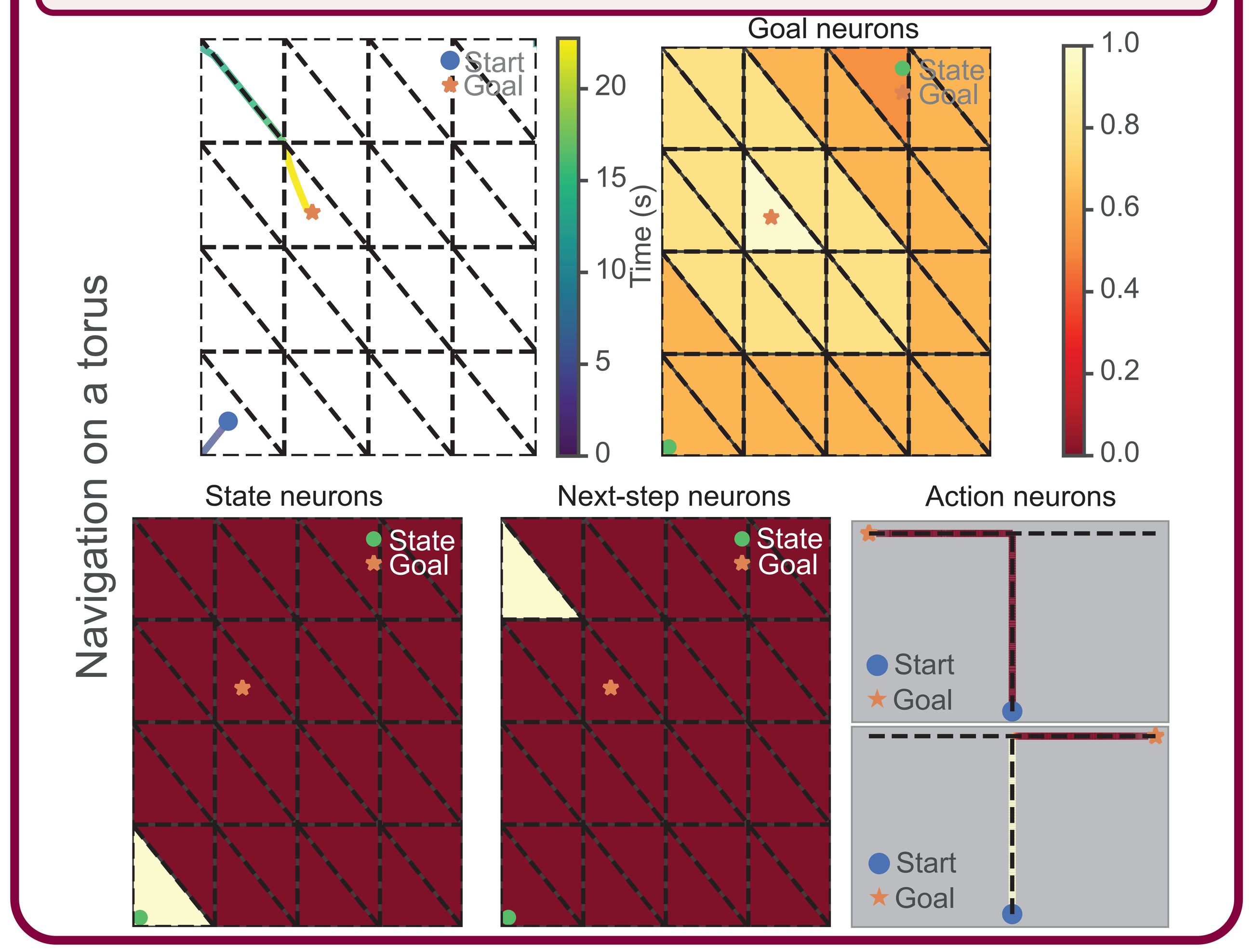
Planning to navigate across simplex



For planning, we treat the **simplicial complex as a hypergraph, where each vertex represents a simplex**, and an edge exists if two simplexes share a vertex

High-level logic of the circuit:

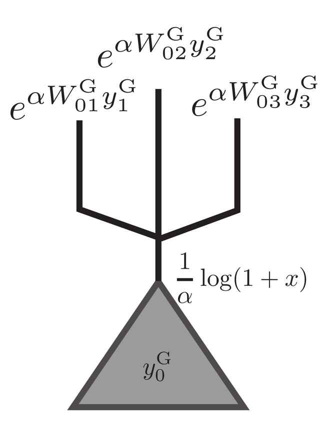
- **State neurons** y^S: receive input from SC attractors and indicate the current vertex.
- **Goal neurons** y^G: computes the shortest path so that $y_i^G = \gamma^{\text{dist}(i,G)}$ via recurrent dynamics.
- **Next-step neurons** y^N: receive inputs from y^G and parent vertexes in y^S and select the closest neighboring vertex to the goal via WTA.
- **Action neurons** y^A: integrate y^N, y^S to select the action leading to the next vertex in y^N.



Dynamics to compute the shortest path

To compute the shortest path, the goal neurons learn the adjacency matrix of $G(S)$ as its weight W^G, receive one-hot goal inputs I^G and evolve as follows:

$$\forall i \in G(S),\ \tau_n \dot{y}_i^G = -y_i^G + \gamma \max\left\{W_{ij}^G y_j^G, I_i^G\right\}$$



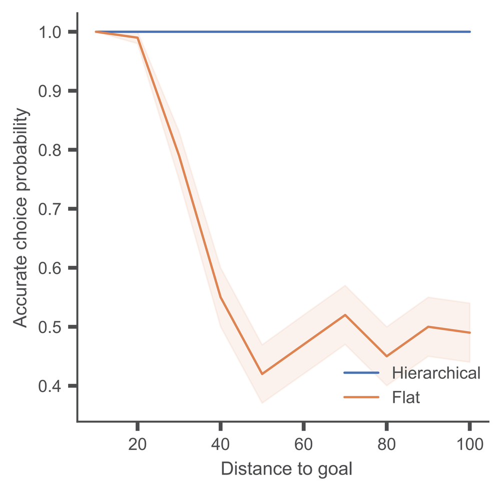
It can be proved that the **dynamic converges to** $y_i^G = \gamma^{\text{dist}(i,G)}$, **analogous to Dijkstra's algorithm**. However, it is unclear if a neuron can compute WTA on synaptic inputs. Therefore, we consider the following approximation where $u_{ij} = \exp\left(\alpha W_{ij}^G y_j^G\right), u_i^I = \exp(\alpha I_i)$:

$$\forall i \in G(S),\ \tau_n \dot{y}_i^G = -y_i^G + \frac{\gamma}{\alpha}\log\left(\sum u_{ij} + u_i^I + 1\right)$$

Here, u_{ij} is dendritic supralinear activation and $\log(1+x)/\alpha$ is somatic nonlinearity. **When α is large, the neuron approximates WTA on synaptic inputs**.

Hierarchical graph navigation via BTSP

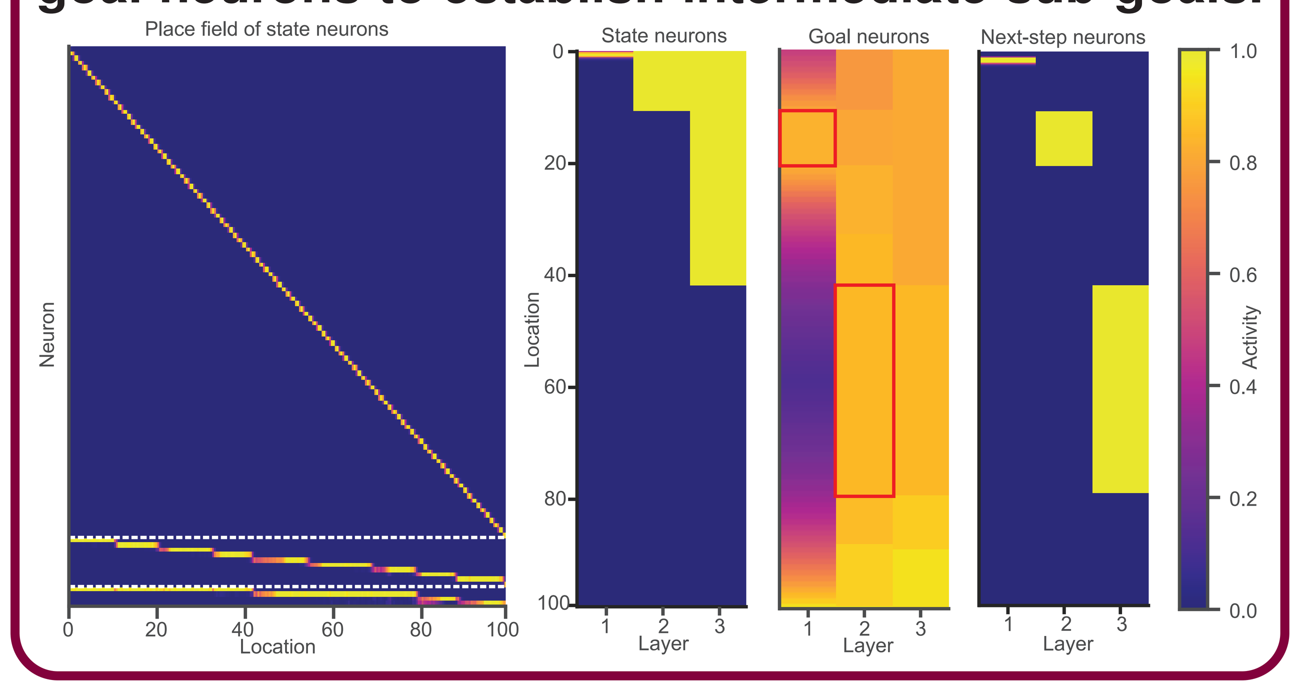
In large graphs, $\gamma^{\text{dist}(i,G)} \approx 0$ for distant nodes, making next-step WTA selection unreliable.



To combat this, we introduce **behavioral timescale plasticity (BTSP)** [2] to learn **hierarchical graphs and sub-goals**, where W^{SS_i} is feedforward weights from layer i, u_{i+1} the synaptic input to layer $i+1$, and e_i the eligibility trace of layer i.

$$\tau_w W^{SS_i} = (2 - W^{SS_i}) \cdot u_{i+1} e_i^\top - W^{SS_i} \cdot u_{i+1} e_i^\top$$

The circuit then functions as follows: the higher-level goal neurons help select **the higher-level next-step**, which **serves as an input to lower-level goal neurons to establish intermediate sub-goals.**



KEY REFERENCES

[1] Niel, E. H. et al. Nature. 2021
[2] Bittner, K. C. et al. Science. 2017

CONTACTS

Email: brabeeba@mit.edu
Open to collaboration and discussion!