Working memory facilitates reward-modulated Hebbian learning in recurrent neural networks

Brief explanation of using working memory in recurrent networks

Roman Pogodin, Dane Corneil, Alexander Seeholzer, Joseph Heng, Wulfram Gerstner

Gatsby Computational Neuroscience Unit, University College London, London, UK

School of Computer and Communication Sciences and School of Life Sciences,

Brain Mind Institute, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Supervisor: Dr. Hamid Aghajan

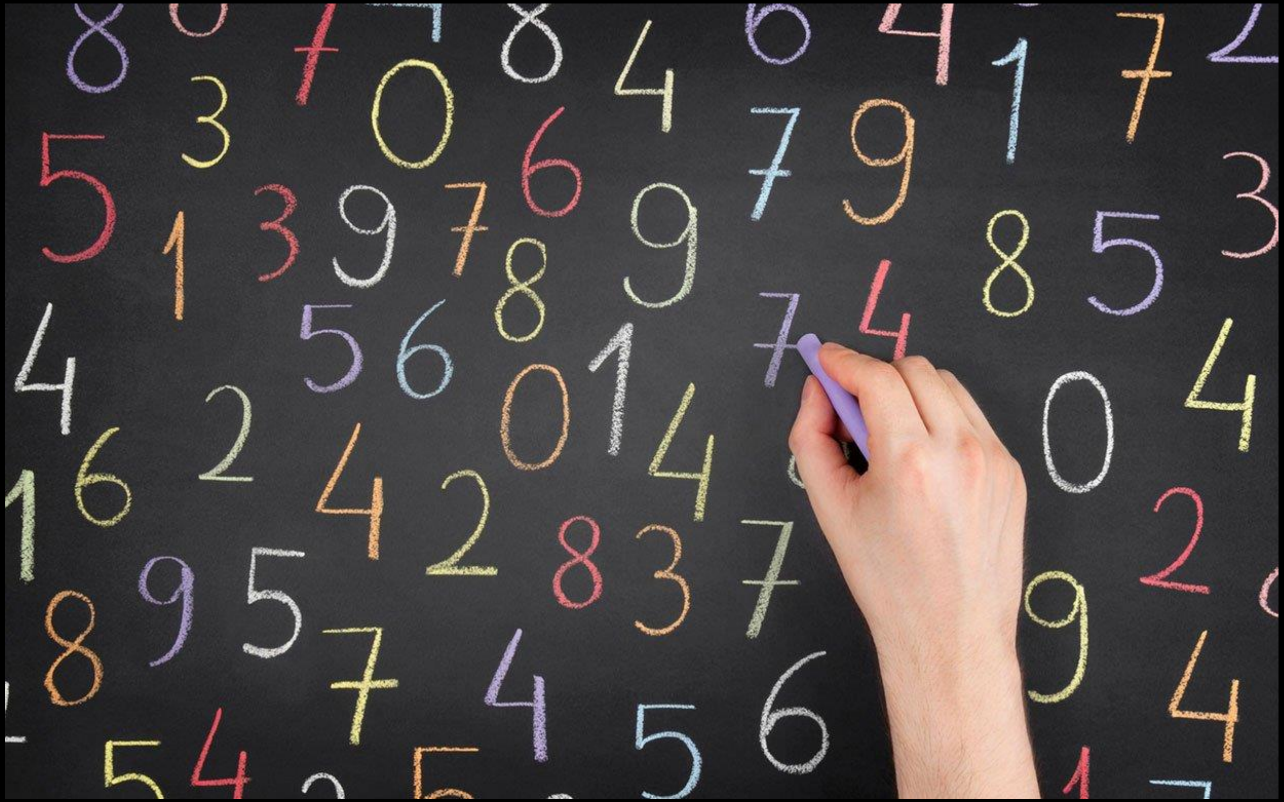
Presenters: Armin Panjehpour - Aliasghar Pourghani

- Terminologies
- Introduction
- Model
- Model`s advantages
- Conclusion

Some required terminologies

- FORCE learning
- Reservoir computing
- Hebbian learning
- Attractor networks

- Acts like writing a number on the blackboard looks easy to us but is challenging for computational brain models.



Solution?

- We know animals can learn complex tasks, so which paradigm is used in their brain?

Introduction

- Reservoir computing is a powerful tool to explain how the brain learns temporal sequences, such as movements, but existing learning schemes are either biologically implausible or too inefficient to explain animal performance.

Introduction

- **FORCE rule** :

  biologically implausible

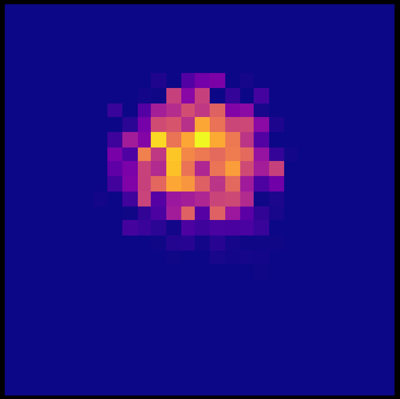
- **Reward-modulated Hebbian learning rule** :

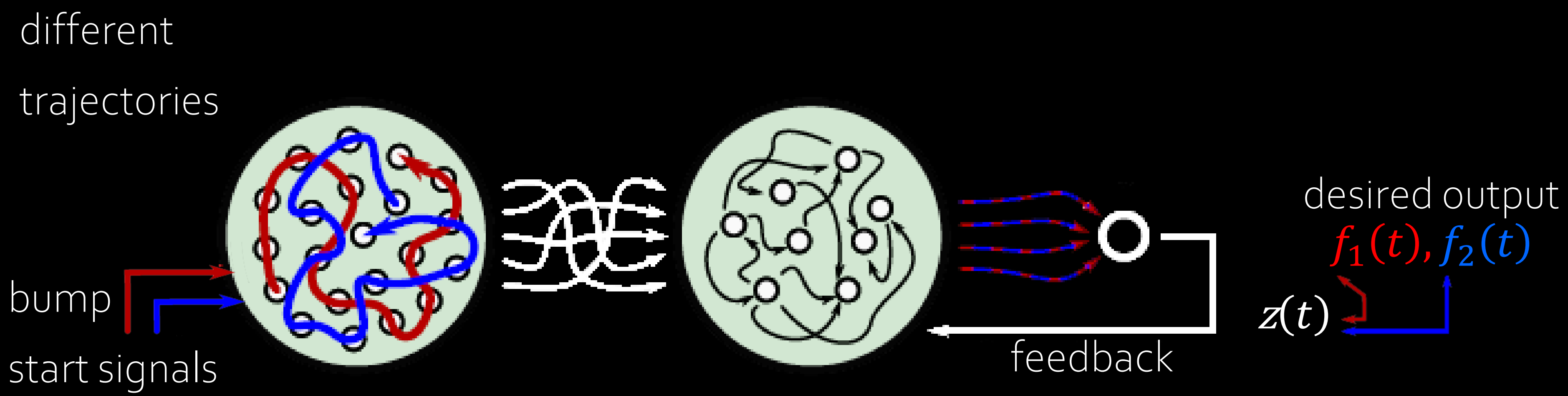
  biologically plausible but in long duration complex tasks, is not powerful enough to learn tasks.

- Using a working memory can solve both of the problems.

Model

1. Attractor network (memory): Creates long–lasting reliable activity patterns which also depend on the input e.

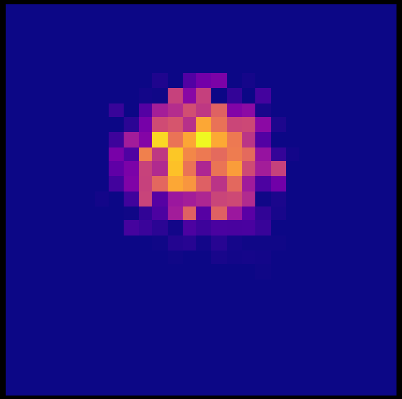


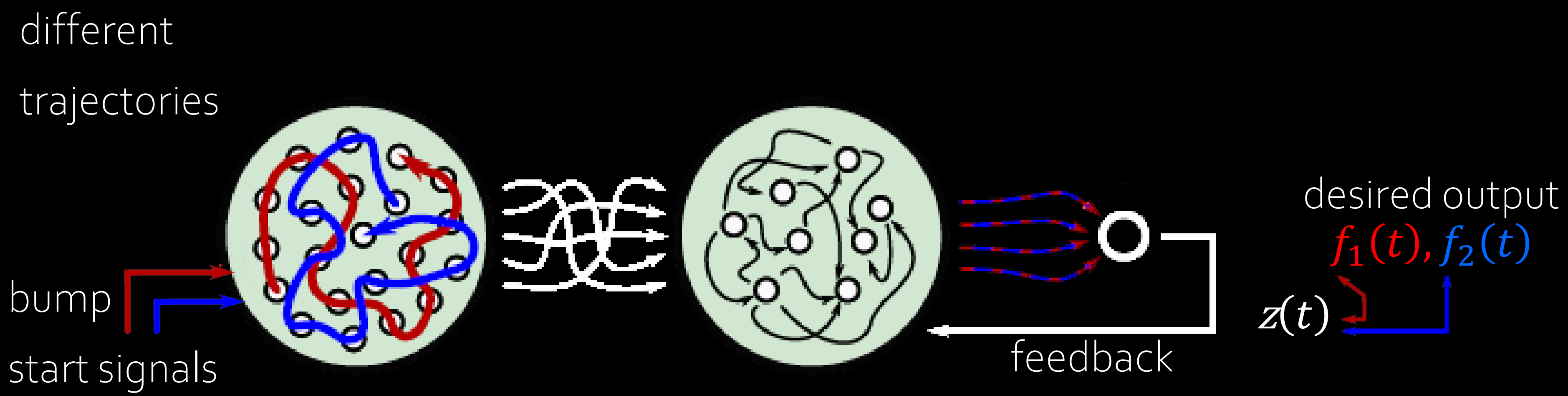


$$\boxed{\begin{aligned} \tau_m \dot{x} &= -\mathrm{x} + [\mathrm{Jx} + \mathrm{e} \ - \mathrm{h}]_+ \\ \tau_a \dot{h} &= -\mathrm{h} + \mathrm{s\ x} \end{aligned}}$$

Model

2. Reservoir network **:** The reservoir learns to approximate a target function f(t) with the output z(t) by linearly combining the firing rate r with readout weights W_{ro}:

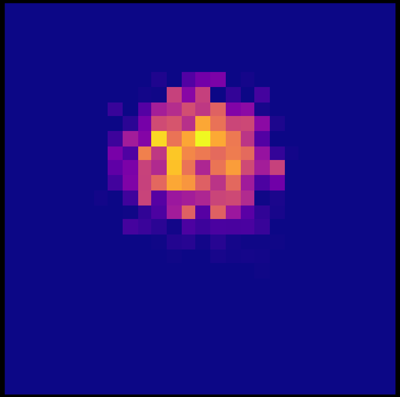


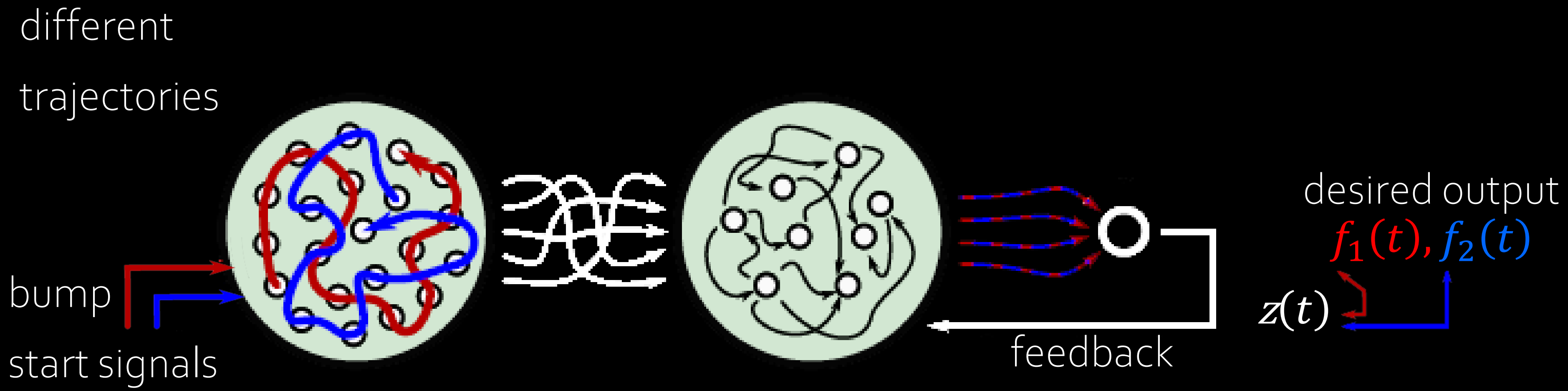


$$\tau\dot{u} = -\mathrm{u} + \lambda W_{rec}r + W_{fb}z + cW_{attr}x\ , \qquad r = \tanh(u) + \ \xi$$
$$z = W_{ro}r + \eta \equiv \ \hat{z} + \eta$$

Model

3. Learning rule: We use the reward-modulated Hebbian rule (which we also learned in lectures) for the readout weights $\boldsymbol{W_{ro}}$:





$$\Delta W_{ro}(\mathrm{t}) = \eta(t)M(t)\big(z(t) - \bar{z}(t)\big)r^{\intercal}(t),\ \ \eta(t) = \eta_0/(1 + t/\tau_n)$$

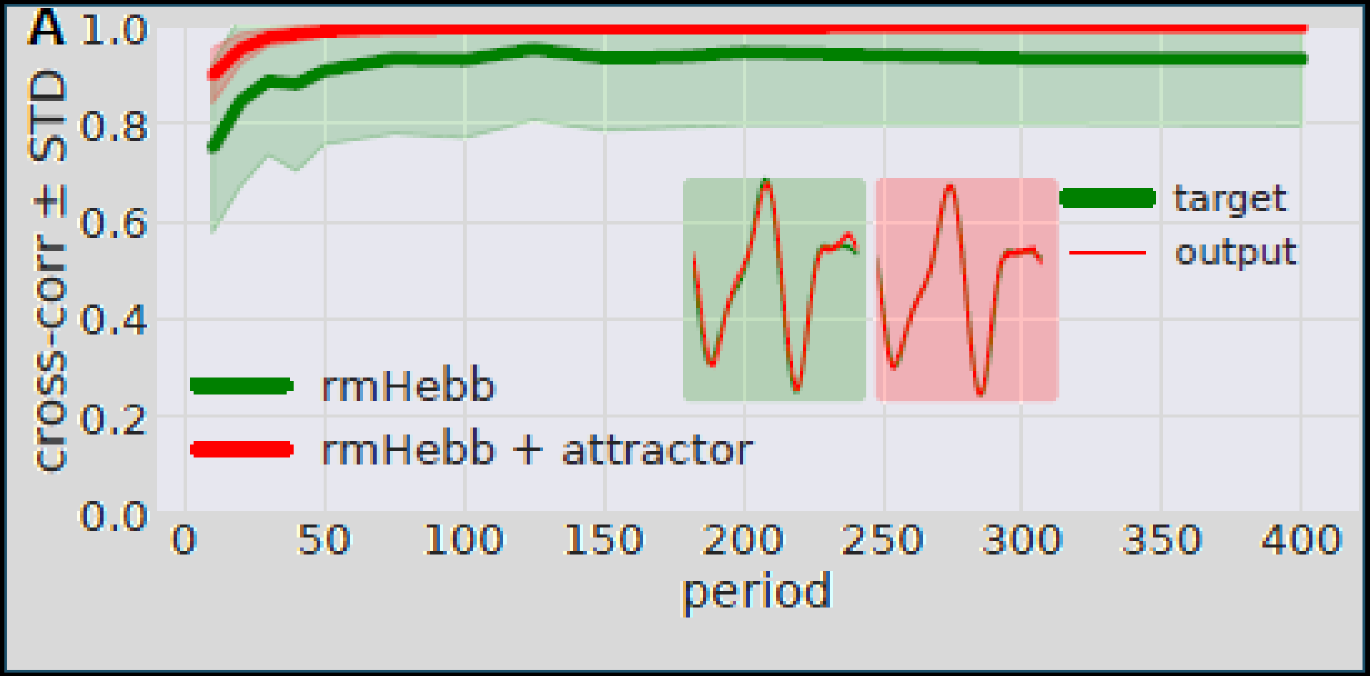
$$P(t) = -\|f(t) - z(t)\|^2, \qquad M(t) = \begin{cases} 1. & P(t) > \bar{P}(t). \\ 0. & P(t) > \bar{P}(t). \end{cases}$$

Advantages of this Model

- Reward-modulated Hebbian learning with attractor input reaches FORCE performance.
- Attractor input allows realistic frequency of weight.
- Working memory translates to efficient reservoir learning of multiple signals.
- We will discuss about these in next slides…

Reaching FORCE learning performance

- For simple tasks, Hebbian learning is proper too:

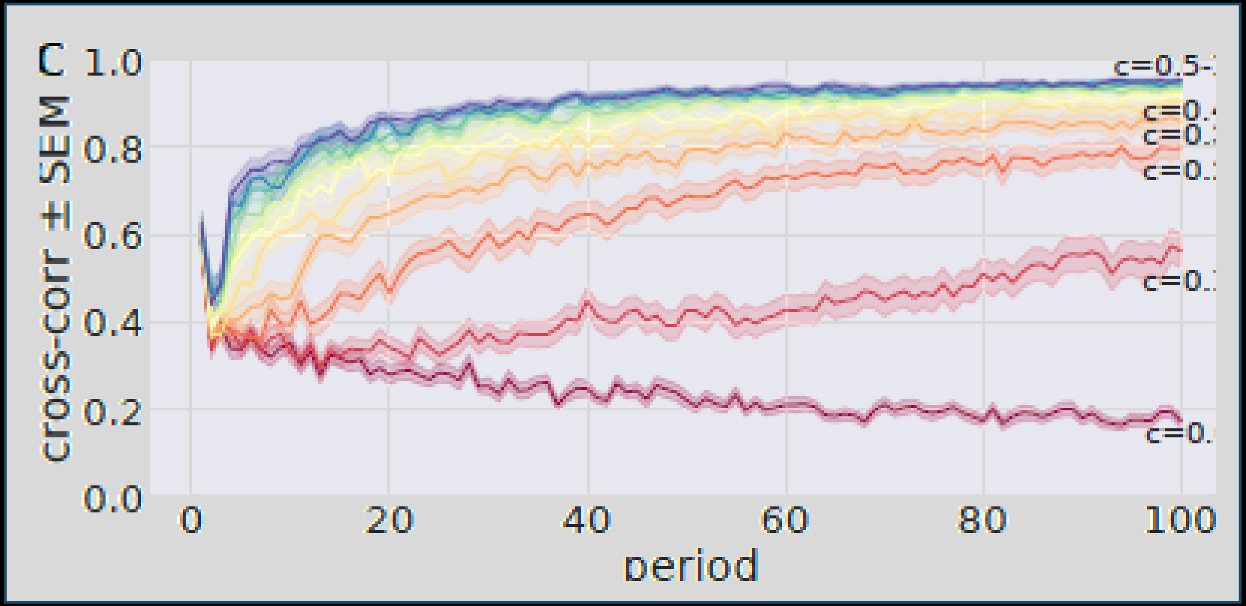


Reaching FORCE learning performance

- But for target signals that extend over 10 seconds, the reward-modulated Hebbian rule achieves a performance of 1 after 200 trials if combined with input from the attractor network (fig. 2B) but fails completely without the attractor network.

Reaching FORCE learning performance

- Effect of Working memory on performance:

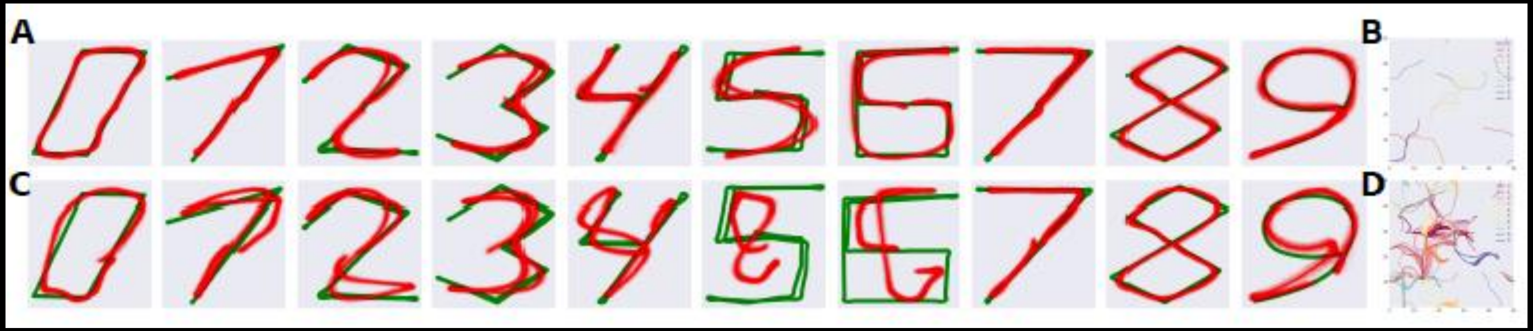


Attractor input allows realistic frequency of weight updates

- FORCE learning needs a feedback signal at every time step.
- Standard reward-modulated Hebbian learning can support very small delays, but fails if updates are less frequent than every few ms.
- But in this model, proposed updates are summed up in the background, but applied only at the end of a one-second trial. We find that even with such a temporally sparse update, learning is still possible.

Working memory translates to efficient reservoir learning of multiple signals

- it is well known that reservoir networks can learn multiple tasks given different initial conditions with both FORCE and the reward-modulated Hebbian rule. we want to prove that these networks do the same with our approach.

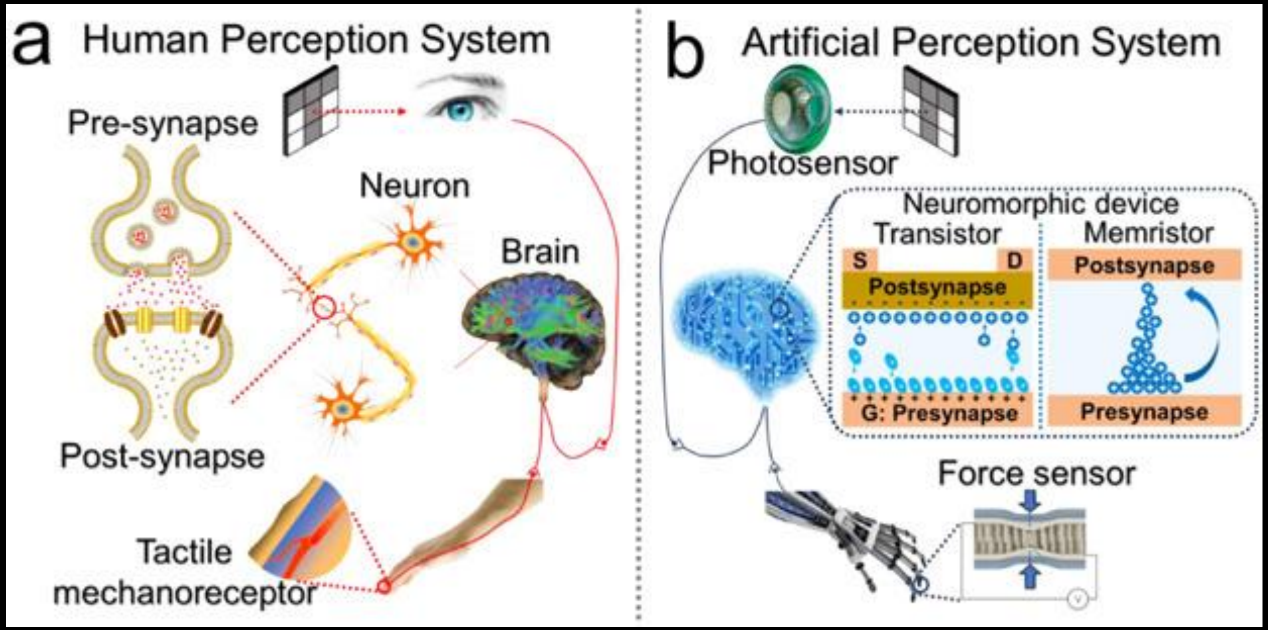


Conclusion

- We showed that a dynamic working memory can facilitate learning of complex tasks with biologically plausible three-factor learning rules. Our results indicate that, when combined with a bump attractor, reservoir computing with reward-modulated learning can be as efficient as FORCE, a widely used but biologically unrealistic rule.

Conclusion

- Apart from the biological relevance, the proposed method might be used for real-world applications of reservoir computing as it is computationally less expensive than FORCE. It might also be an interesting alternative for learning in neuromorphic devices.



Thanks for your attention!

Feel free to ask any questions!

- And here are two cool neuroscientific jokes:

1. Why was the neuron sent to the principal's office?

   It had trouble controlling its impulses.

2. What is the neuron's favorite television channel?

   The Ion Channel.