Spatial-Aware Decision-Making with Ring Attractors in Reinforcement Learning Systems

Marcos N. Saura, Richard Allmendinger, Wei Pan, Theodore Papamarkou

Motivation

The premises of our research

An agent learns optimal policies by interacting with its environment over time, increasing certainty about its actions as it explores the action space.

– Modern **deep reinforcement learning** treats actions as independent prior to training, **ignoring explicit relationships between actions.**

– This leads to **inefficient action selection in structured environments**.

– **Action spaces tend to have structure**: directional movements, rotation angles, or adjacency between tactical moves in game-like environments.

– Agents tend to perform **inefficient sampling,** rediscovering action relationships, leading to **unstable learning and exploration.**

Idea

Ring attractor for spatial encoding

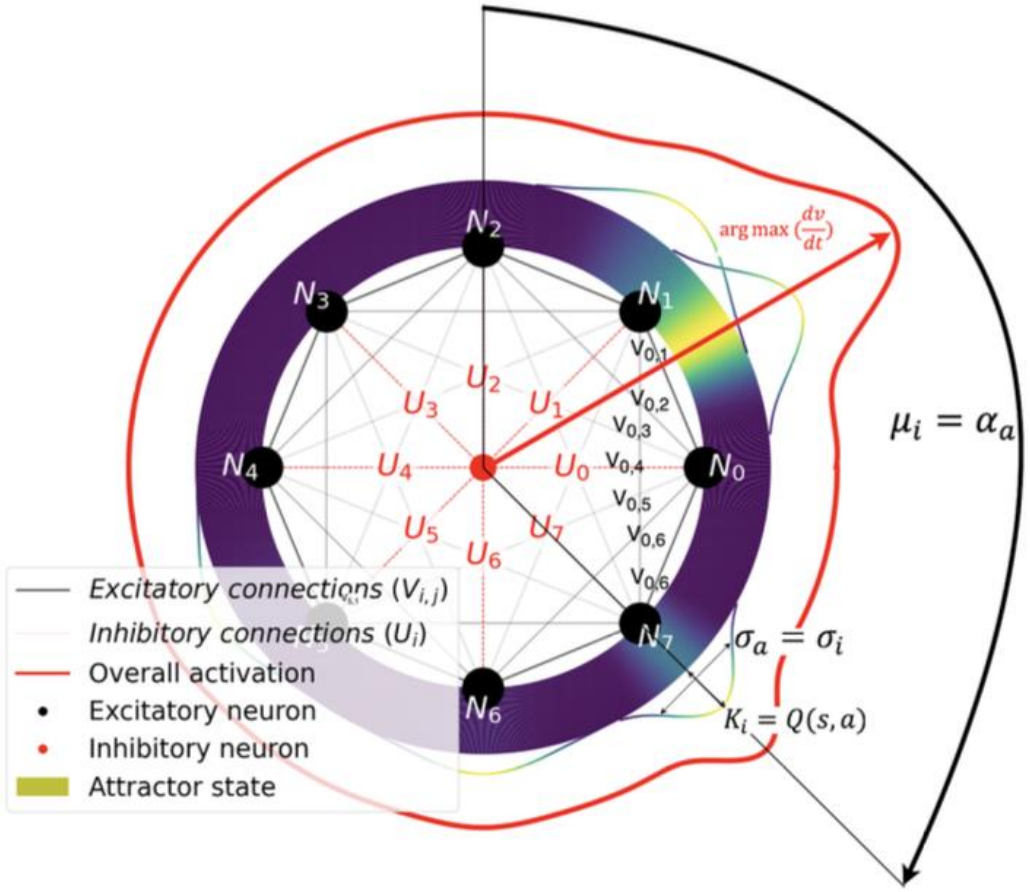
– Ring attractors provide a biologically plausible mechanism to explicitly **encode action space topology and uncertainty.**

– **Actions map to specific ring locations** where distance-weighted connections preserve spatial relationships and enable temporal filtering.

– Ring attractors facilitate the distribution of spatial representations across the neural network, **improving learning speed and accuracy.**

Methods

Continues-time recurrent neural network (CTRNN)



Algorithm 1 CTRNN Ring Attractor Action Selection

Step 1: Compute Q-values and Uncertainty
Compute mean $\bar{Q}(s,a)$ and variance σ_a^2 using Eq. 11
Step 2: Generate Input Action Signals
for each excitatory neuron n **do**
    Generate Gaussian action signal using Eq. 5
end for
Step 3: Reach Attractor State
for timestep t until T **do** ▷ *choose T=50 empirically*
    for each excitatory neuron n **do**
        $\frac{\mathrm{d}v_n}{\mathrm{d}t}$ Update excitatory neurons using Eq. 6
    end for
    $\frac{\mathrm{d}u}{\mathrm{d}t}$ Update inhibitory neuron using Eq. 7
end for
Step 4: Translate Neural Activity V to *action* selected from action space $\mathcal{A}$ using Eq. 8
return action

Algorithm 1: CTRNN ring attractor action selection.

Methods

Continues-time recurrent neural network (CTRNN)

The action selection process as in Algorithm 1:

– Computing **Q-values** and **uncertainty** (Step 1).

– Generating **Gaussian input signals** (Step 2):

$$x_n(Q(s,a)) = \sum_{a=1}^{A} \frac{Q(s,a)}{\sqrt{2\pi\sigma_a}} \exp\left(-\frac{(\alpha_n - \alpha_a(a))^2}{2\sigma_a^2}\right)$$

– Evolving the **attractor dynamics** (Step 3):

$$\frac{\mathrm{d}v_n}{\mathrm{d}t} = \frac{1}{\tau}\left(max\left(0, \left(\sum_{m=1}^{m=N} w_{m,n}^{(E\to E)} v_m + x_n(Q) + w^{(I\to E_u)} u\right)\right)\right) - v_n$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{1}{\tau}\left(max\left(0, \left(u + \sum_{n=1}^{N} w_n^{(E_n\to I)} v_n\right)\right)\right) - u,$$

– Neural **activity to action selection** (Step 4):

$$\text{action} = \text{argmax}\{\mathbf{V}\}\frac{N^{(A)}}{N^{(E)}}$$

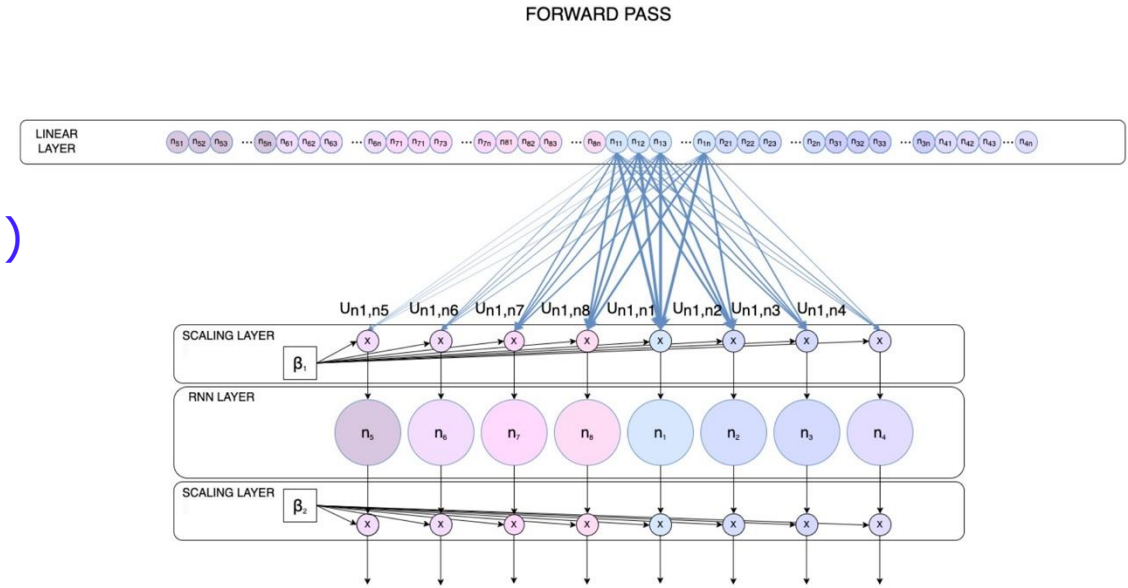
Algorithm 1 CTRNN Ring Attractor Action Selection

Step 1: Compute Q-values and Uncertainty
Compute mean $\bar{Q}(s,a)$ and variance σ_a^2 using Eq. 11
Step 2: Generate Input Action Signals
for each excitatory neuron n **do**
    Generate Gaussian action signal using Eq. 5
end for
Step 3: Reach Attractor State
for timestep t until T **do** ▷ *choose T=50 empirically*
    for each excitatory neuron n **do**
        $\frac{\mathrm{d}v_n}{\mathrm{d}t}$ Update excitatory neurons using Eq. 6
    end for
    $\frac{\mathrm{d}u}{\mathrm{d}t}$ Update inhibitory neuron using Eq. 7
end for
Step 4: Translate Neural Activity V to *action* selected from action space $\mathcal{A}$ using Eq. 8
return action

Algorithm 1: CTRNN ring attractor action selection.

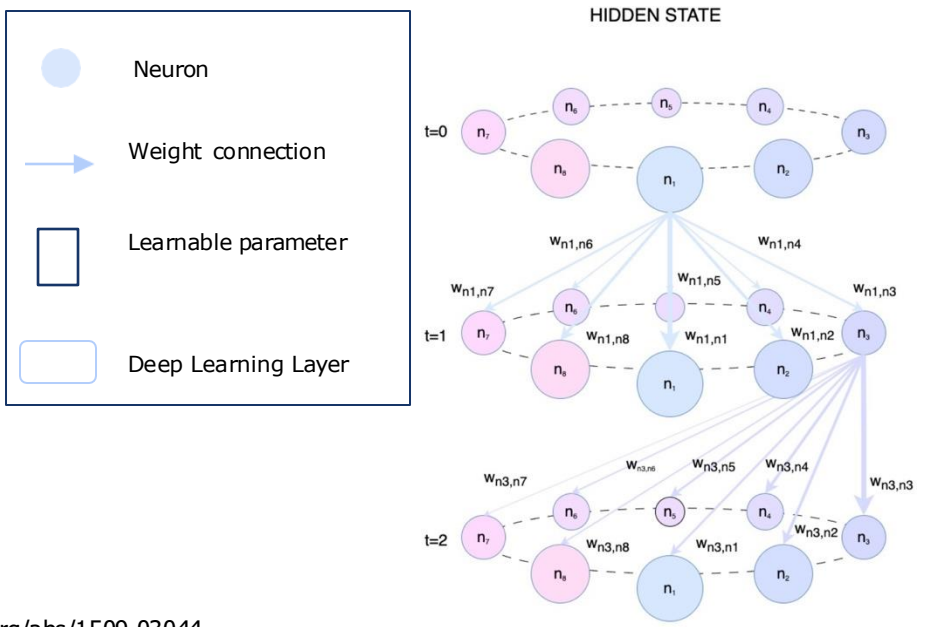
Methods

Ring attractor Deep Learning representation (DL-RNN)



Recurrent neural networks (RNN): RNN have been the architecture of choice to integrate the ring attractor within a Deep Learning algorithm. In recent studies [11] RNNs show good performance at modelling sequential data and capturing temporal dependencies for decision-making. In this context we can use RNNs mirroring the temporal dynamics encoded by ring attractors.

Attractor state as recurrent connections: The recurrent connections within the RNN allow the model to retain information over time. As the network processes sequential inputs, information from previous time steps is integrated into the current state of the network through the hidden state. We model those connections to resemble the spatial distribution of the ring.



Cue signal input as a forward pass: These attractor states encode information about the current context or task, and their dynamics are influenced by both the current input and the network's hidden state. This temporal evolution controlled by a learnable time constant tau allows the model to adapt its behavior over time, responding to changes in the environment or input signals.

[11] Li X, et al (2015 Recurrent reinforcement learning: A hybrid approach. arxiv.org/abs/1509.03044

Methods

Ring attractor Deep Learning representation (DL-RNN)

To enable **end-to-end training**, we developed a recurrent layer that preserves **ring topology** through **structured weight matrices**.

$$V(s)_{m,n} = \frac{1}{\tau}\Phi_\theta(s)^T w_{m,n}^{I\to H} = \frac{1}{\tau}\Phi_\theta(s)^T e^{\frac{d(m,n)}{\lambda}}$$

$$d(m,n) = \min\left(|m - n\frac{M}{N}|, N - |m - n\frac{M}{N}|\right)$$

$$U(v)_{m,n} = h(v)^T w_{m,n}^{H\to H} = h(\Phi_\theta)^T e^{\frac{d(m,n)}{\lambda}}$$

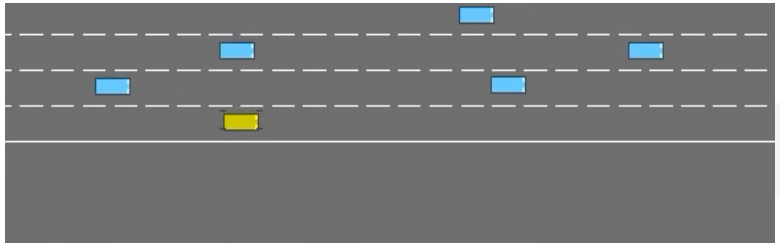
$$d(m,n) = \min\left(|m-n|, N - |m-n|\right)$$

The layer outputs action-values via Q(s,a) = β tanh(V(s) + U(v)), where **τ** controls input contribution into the **attractor state** and β scales outputs to match environment rewards.

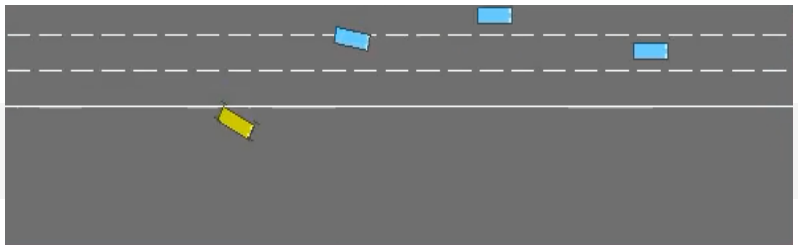
Results

At a glance: Performance after 30 minutes of gameplay

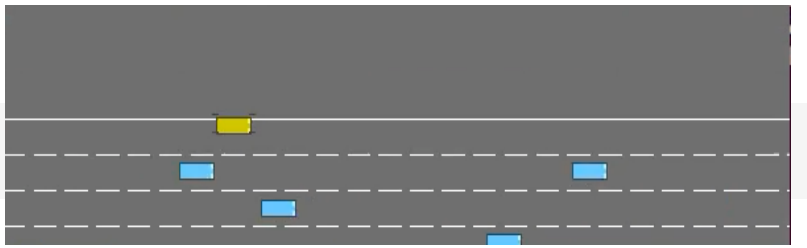
Learning: 15 Minutes of Gameplay



Baseline Model Double DQN (DDQN)

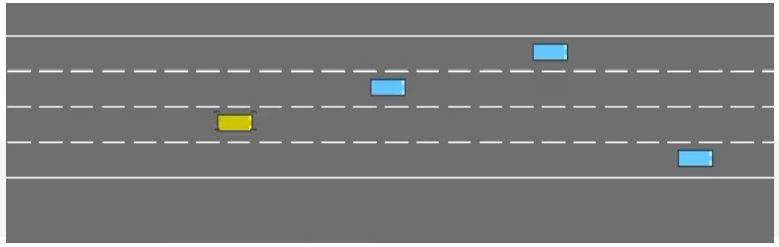


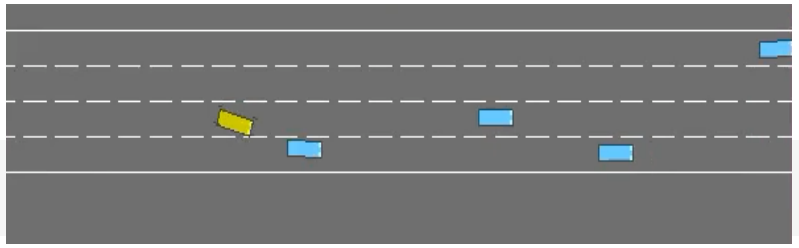
Ablation Model Standard–RNN + DDQN

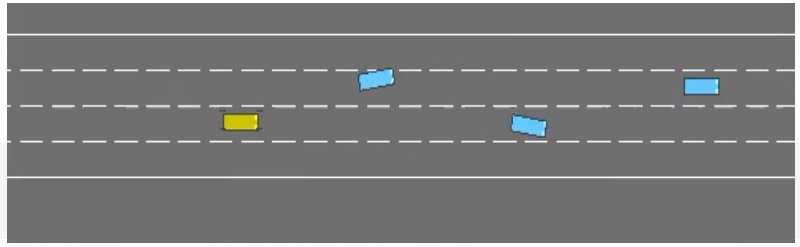


Ring Attractor (RA) Model RA–RNN + DDQN

Learning: 30 Minutes of Gameplay

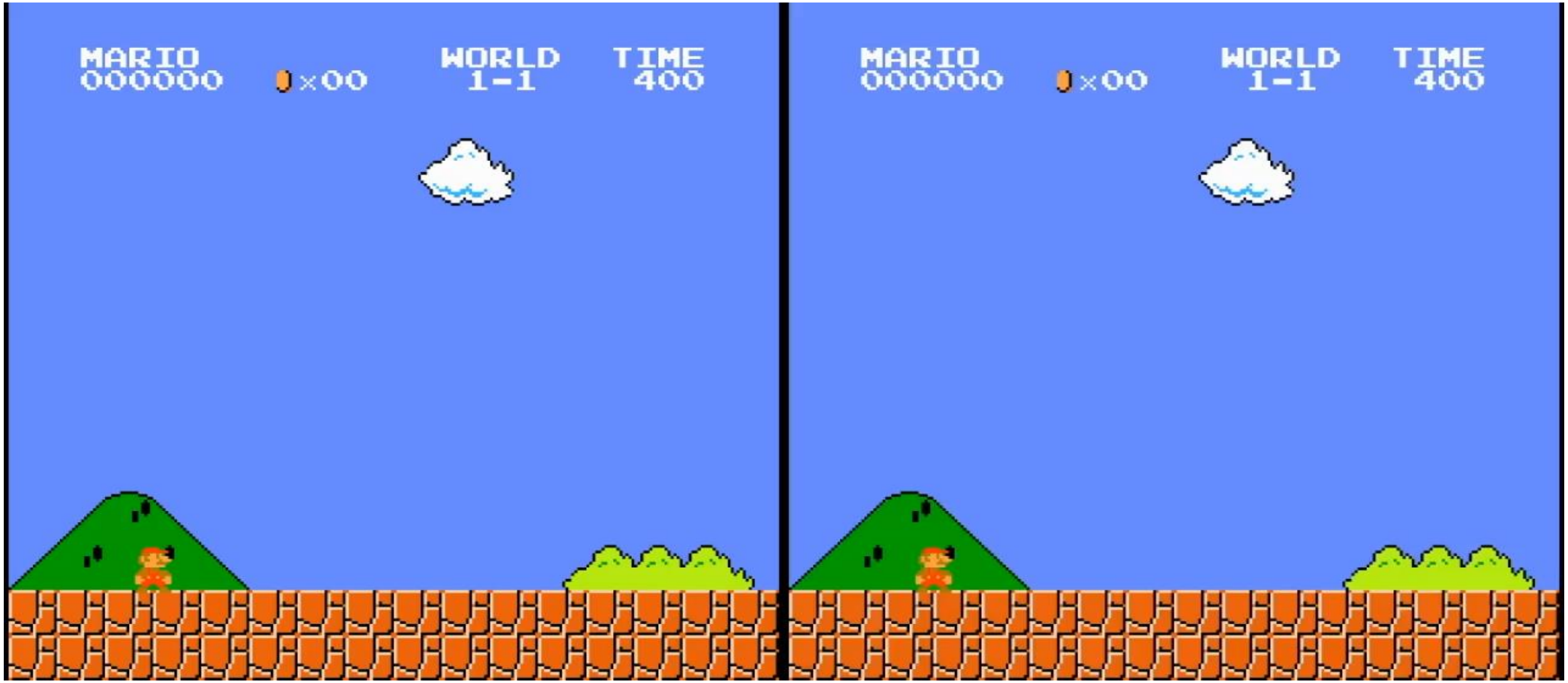






Results

At a glance: Performance after 2 hours of gameplay



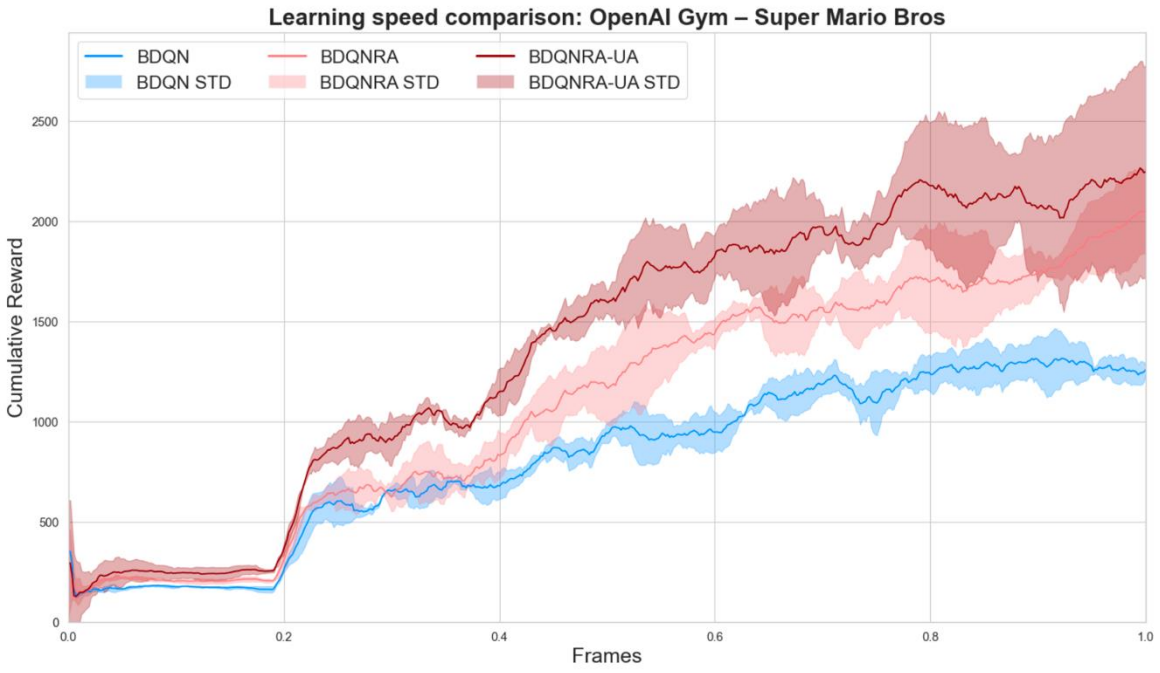
Ring Attractor (RA) Model RA+DDQN: ~ 80% Wins

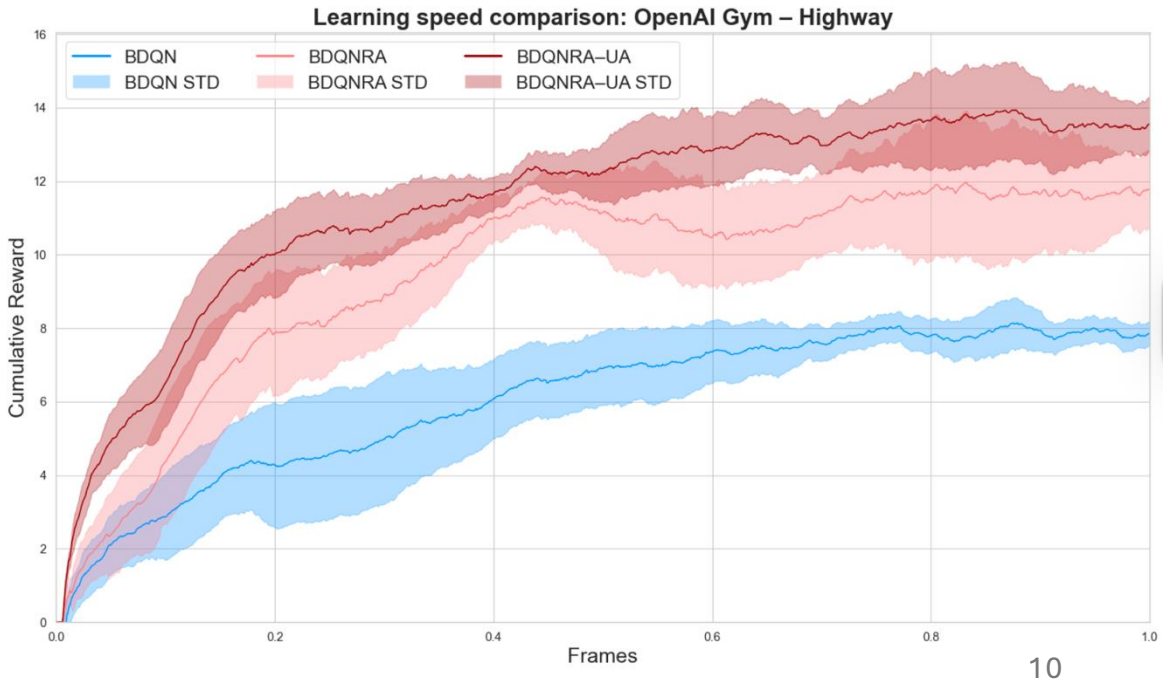
Baseline Model Double DQN (DDQN): 0% Wins

Results

CTRNN model: Uncertainty quantification

- The CTRNN ring attractor model achieves **78% average improvement** over baselines in benchmarked environments.
- Integrating ring attractors with **uncertainty quantification** further **accelerates the learning** rate.
- Results apply to both **discrete** (Super Mario) and **continuous** (Highway) action spaces.





Results

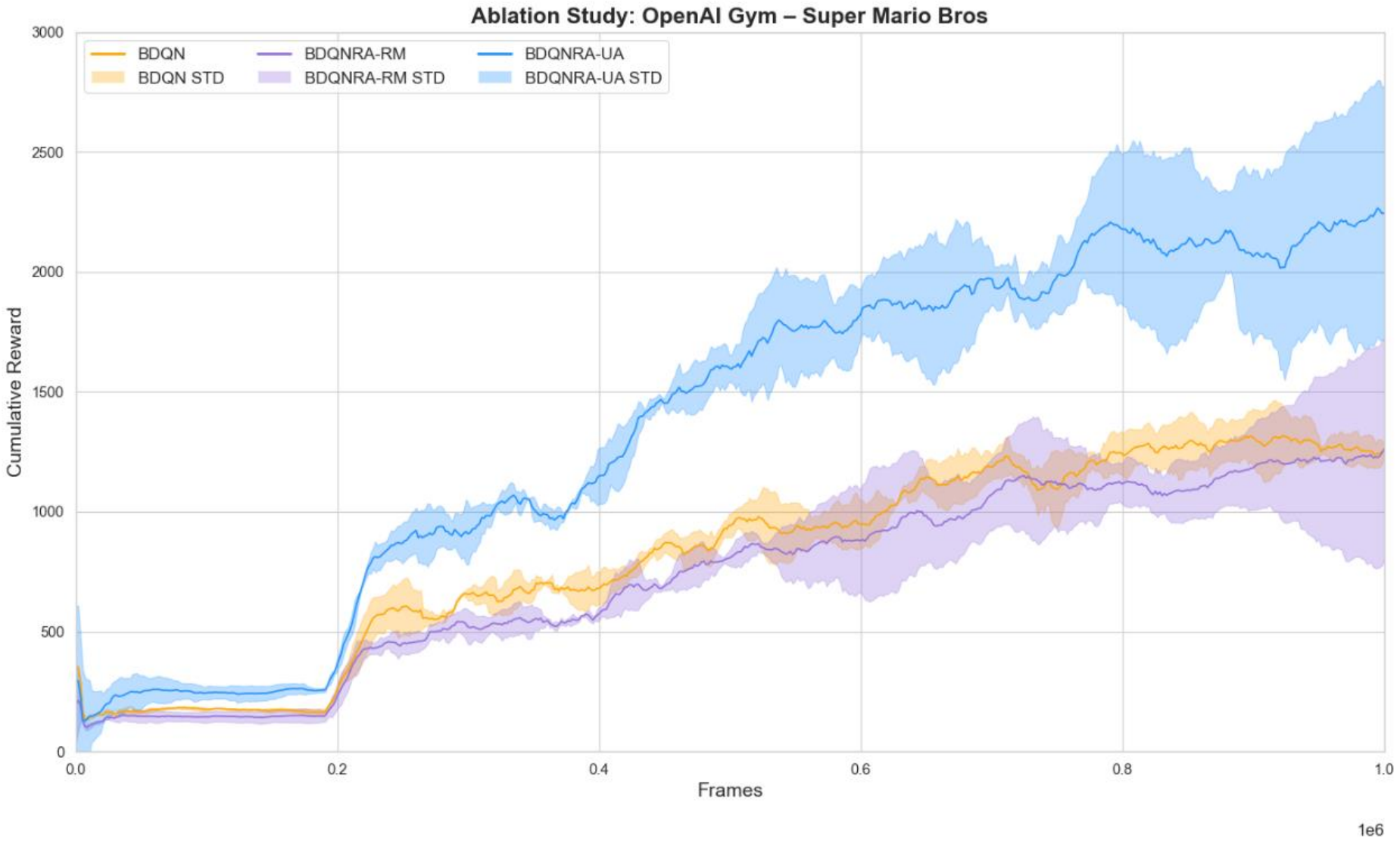
DL-RNN model: Atari 100K benchmark

- +53% average improvement over state-of-the-art baselines.
- Strong gains in both spatially structured and decision-making games like Asterix (+110%) and Boxing (+105%).

Game Environment	Ring	Agent Human	CURL	Reported SPR	EffZero	Implemented EffZero	EffZeroRA
Alien	Double	7127.7	558.2	801.5	808.5	738.1	**1098.8**
Asterix	Single	8503.3	734.5	977.8	25557.8	14839.3	**31037.3**
Bank Heist	Double	753.1	131.6	380.9	351.0	362.8	**460.5**
BattleZone	Double	37187.5	14870.0	**16651.0**	13871.2	11908.7	15672.0
Boxing	Double	12.1	1.2	35.8	52.7	30.5	**62.4**
Chopper C.	Double	7387.8	1058.5	974.8	1117.3	1162.4	**1963.0**
Crazy Climber	Single	35829.4	12146.5	42923.6	83940.2	83883.0	**100649.7**
Freeway	Double	29.6	26.7	24.4	21.8	22.7	**31.3**
Frostbite	Double	4334.7	1181.3	1821.5	296.3	287.5	**354.8**
Gopher	Double	2412.5	669.3	715.2	3260.3	2975.3	**3804.0**
Hero	Double	30826.4	6279.3	7019.2	9315.9	9966.4	**11976.1**
Jamesbond	Double	302.8	471.0	365.4	**517.0**	350.1	416.4
Kangaroo	Double	3035.0	872.5	3276.4	724.1	689.2	**1368.8**
Krull	Double	2665.5	4229.6	3688.9	5663.3	6128.3	**9282.1**
Kung Fu M.	Double	22736.3	14307.8	13192.7	30944.8	27445.6	**49697.7**
Ms Pacman	Single	6951.6	1465.5	1313.2	1281.2	1166.2	**2028.0**
Private Eye	Double	69571.3	**218.4**	124.0	96.7	94.3	155.8
Road Runner	Double	7845.0	5661.0	669.1	17751.3	19203.1	**29389.3**
Seaquest	Double	42054.7	384.5	583.1	1100.2	1154.7	**1532.8**
Human–normalised Score							
	Mean	1.000	0.428	0.638	1.101	0.959	**1.454**
	Median	1.000	0.242	0.434	0.420	0.403	**0.531**

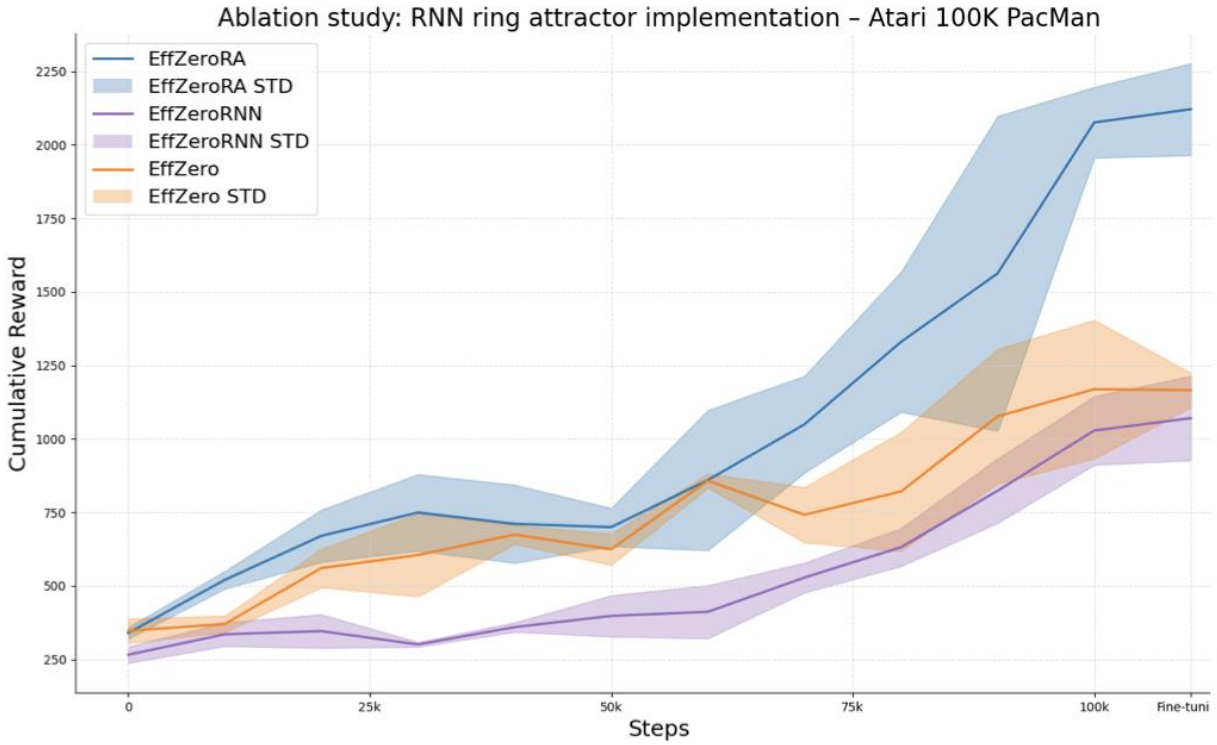
Results

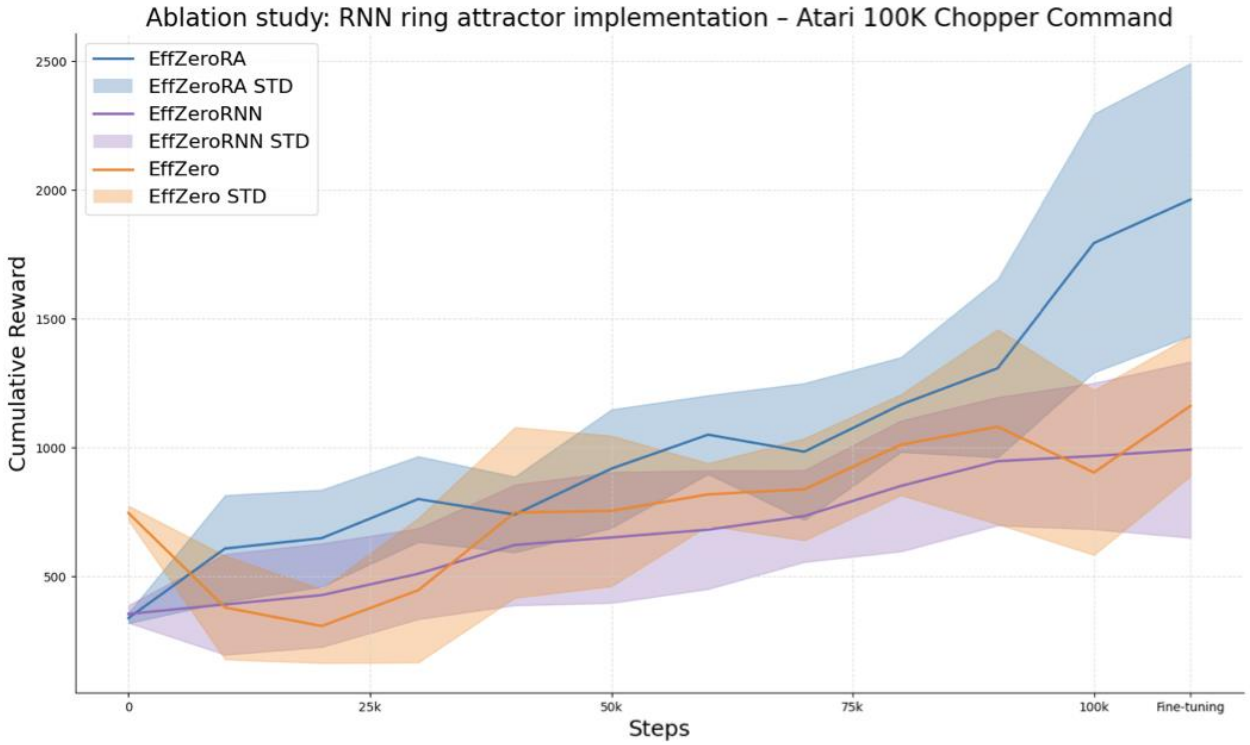
Ablation studies: CTRNN model with randomly assigned actions



Results

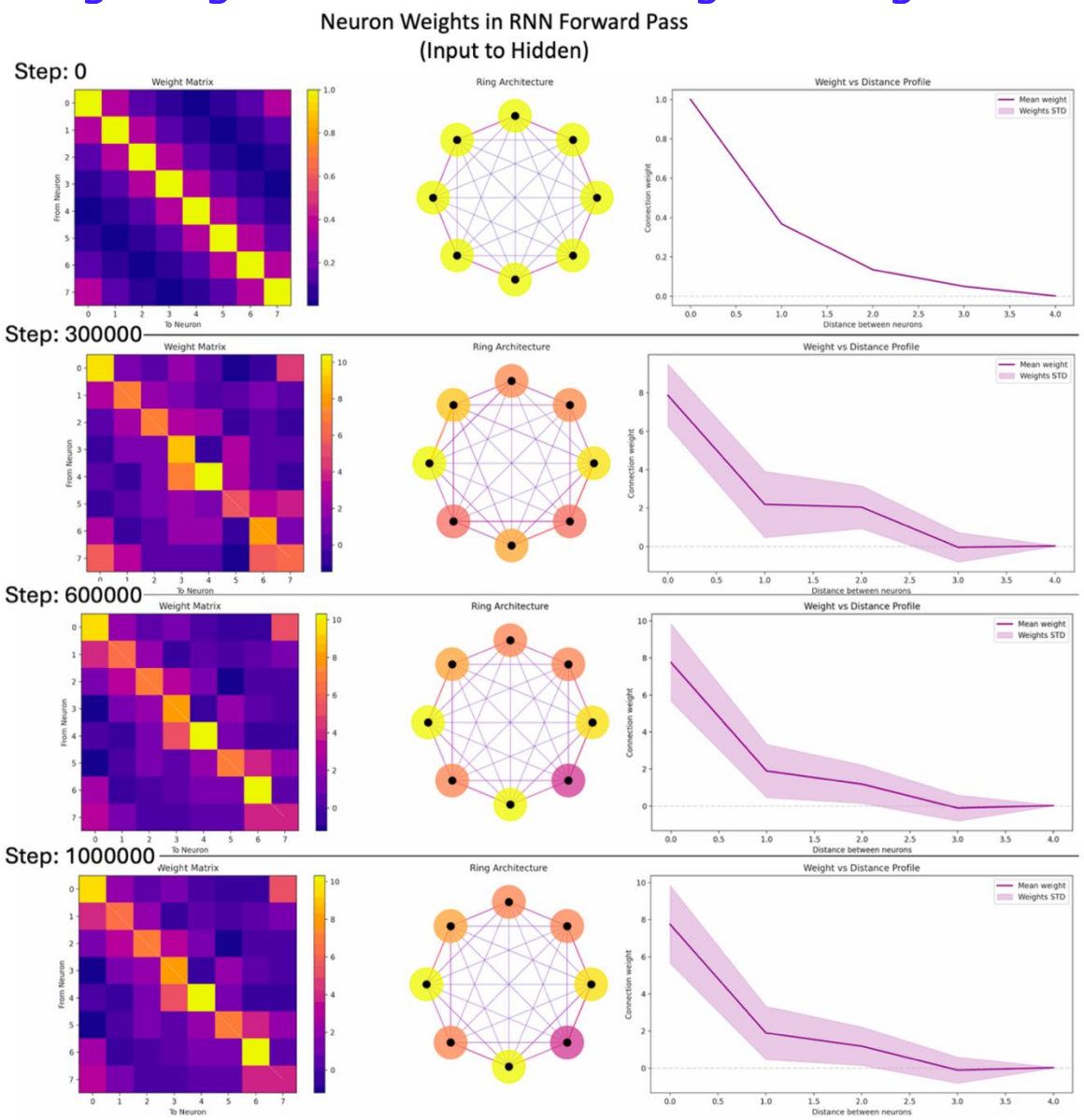
Ablation studies: DL-RNN model without ring shaped connectivity

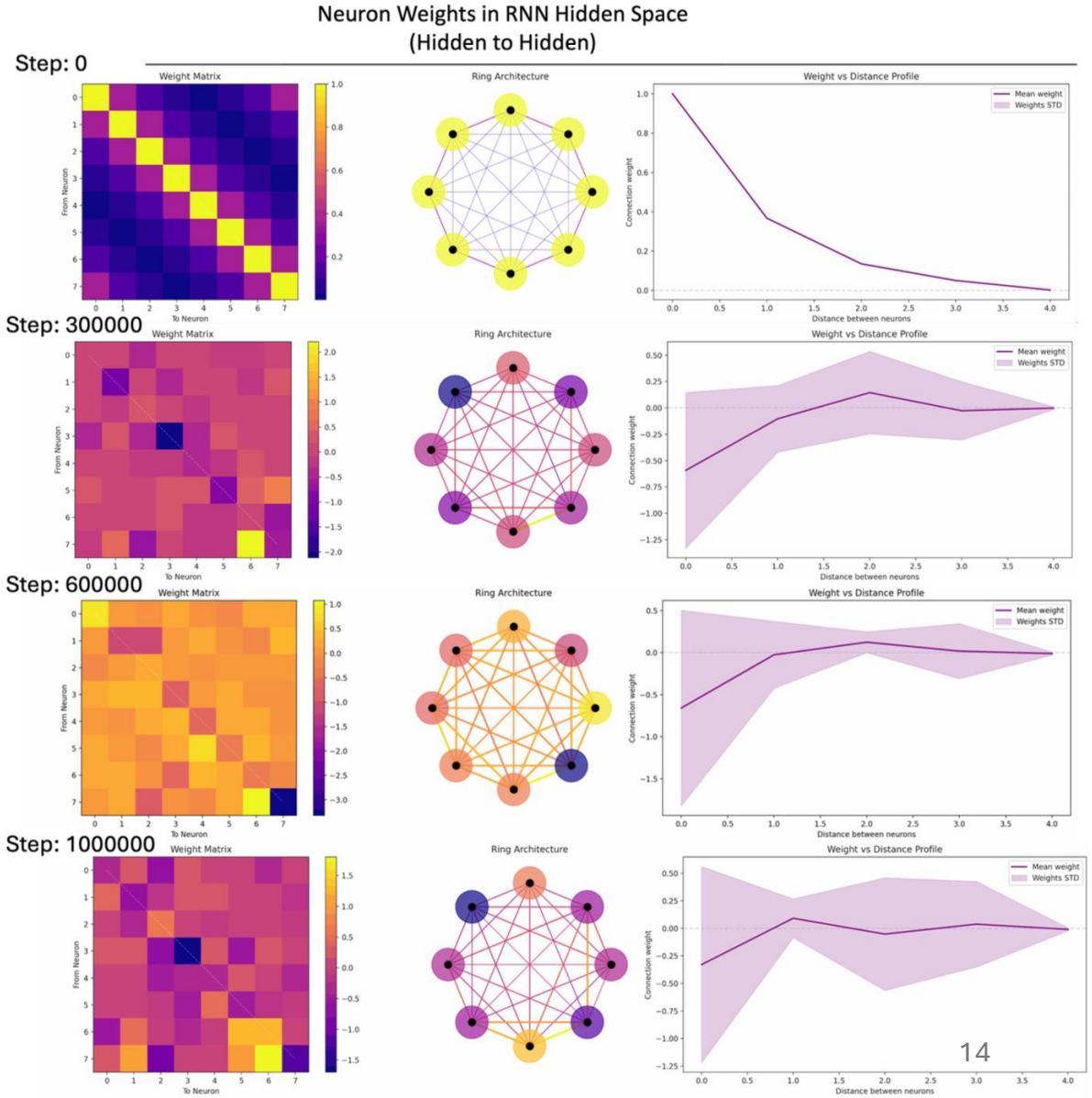




Results

Ring weigthts evolution during training





Thank you

Marcos N. Saura, Richard Allmendinger, Wei Pan, Theodore Papamarkou