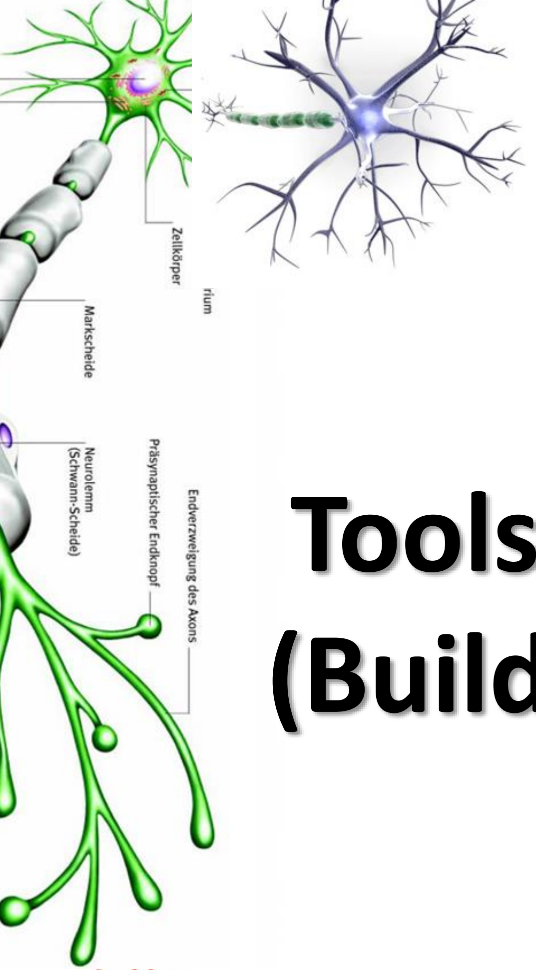
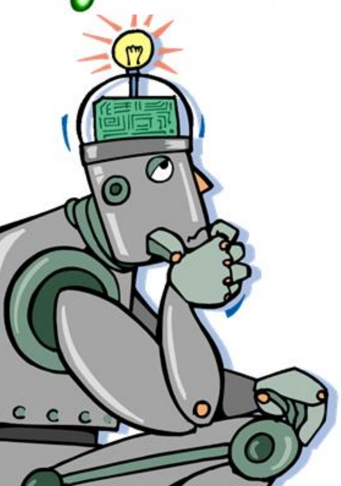
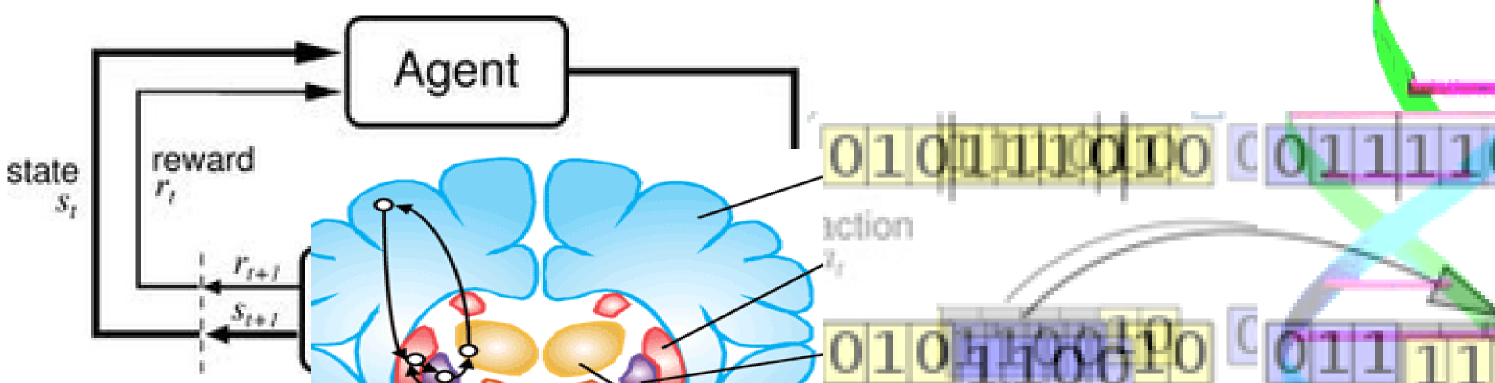


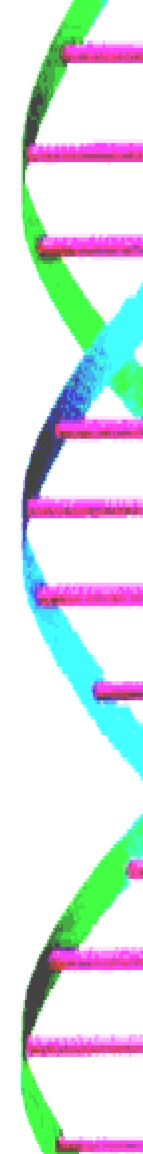
Tools of Artificial Intelligence (Building brains for machines)

RMAI2-U1 (RL_L2)

Poramate Manoonpong







What have we learnt so far?

Contents

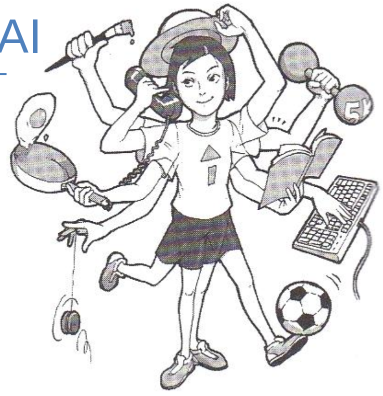
Artificial Intelligence

Artificial neural networks

Reinforcement learning

Evolutionary computation

Embodied AI

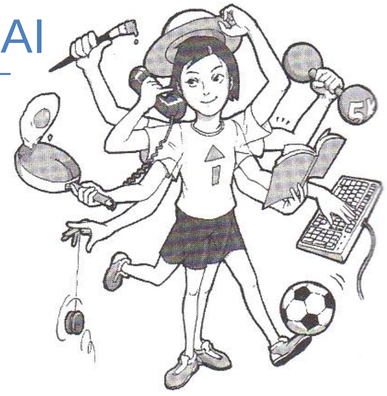


"Intelligence requires a body (actuators/muscles, sensors, structure, materials)!"→ Interactions between body, brain, environment.

What have we learnt so far?

Contents

Embodied AI



Artificial Intelligence

Artificial neural networks

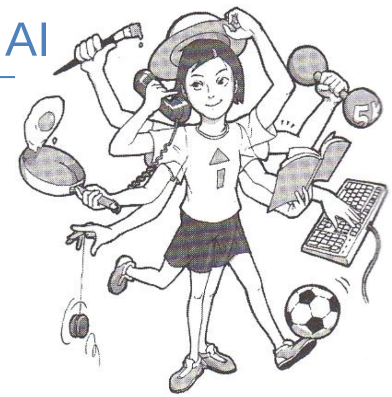
Reinforcement learning

Evolutionary computation

What have we learnt so far?

Contents

Embodied AI

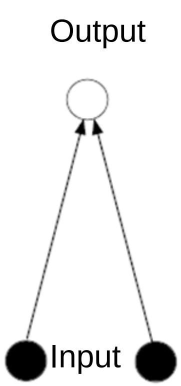


Artificial Intelligence

Artificial neural networks

Reinforcement learning

Evolutionary computation

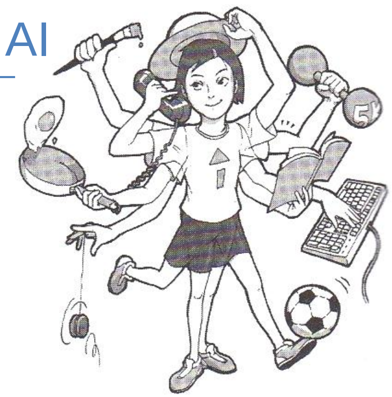


Single layer feedforward net.

What have we learnt so far?

Contents

Embodied AI

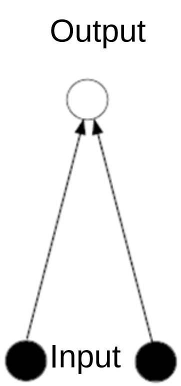


Artificial Intelligence

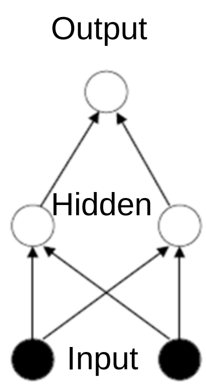
Artificial neural networks

Reinforcement learning

Evolutionary computation



Single layer
feedforward net.

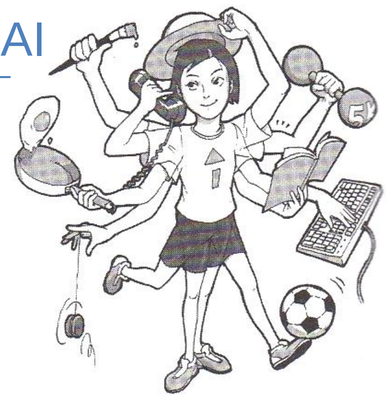


Multilayer
feedforward net.

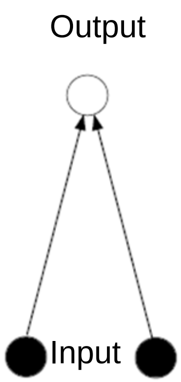
What have we learnt so far?

Contents

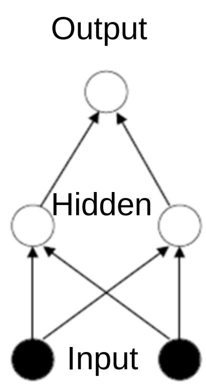
Embodied AI



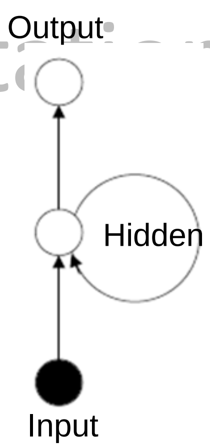
Artificial Intelligence
Artificial neural networks
Reinforcement learning
Evolutionary computation



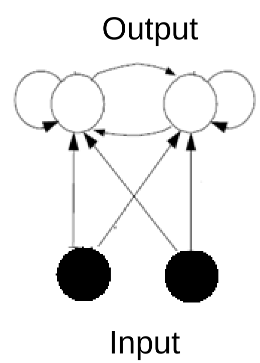
Single layer feedforward net.



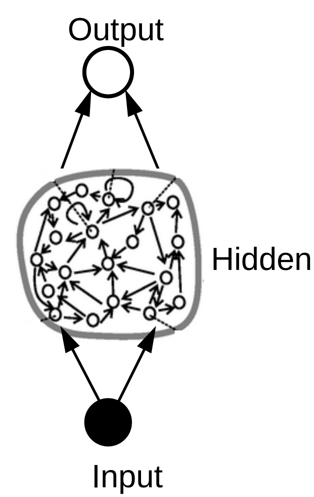
Multilayer feedforward net.



Single recurrent neuron net.



2-recurrent neuron net.

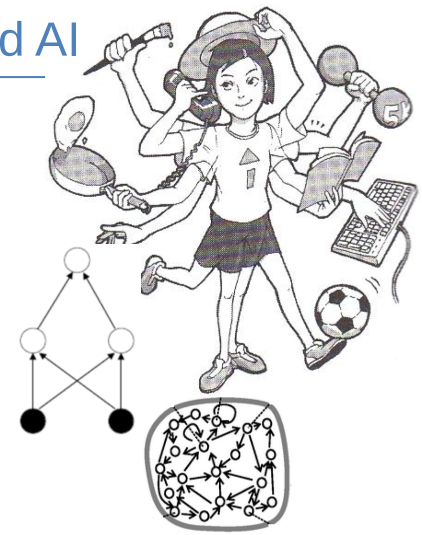


Reservoir computing

What have we learnt so far?

Contents

Embodied AI



Artificial Intelligence

Artificial neural networks

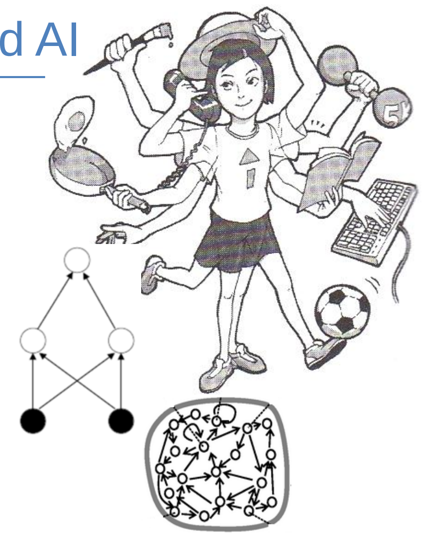
Reinforcement learning

Evolutionary computation

What have we learnt so far?

Contents

Artificial Intelligence
Artificial neural networks
Reinforcement learning
Evolutionary computation



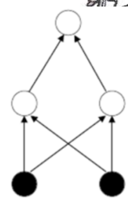
What have we learnt so far?

Contents

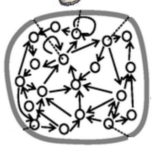
Embodied AI

Artificial Intelligence

Artificial neural networks



Reinforcement learning



Evolutionary computation

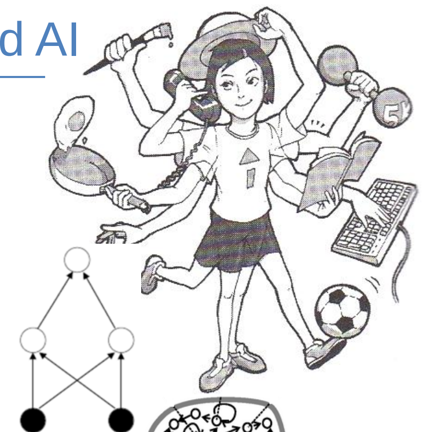
- Q learning (Off-policy TD control)→ State & action pair *(s,a)*

$$\Delta Q(s,a) = \alpha\left(\overbrace{\underbrace{r + \gamma \max_{a_1} Q(s_1,a_1)}_{\textit{Target}} - \underbrace{Q(s,a)}_{\textit{Estimate}}}^{\textit{TD error}}\right)$$

What have we learnt so far?

Contents

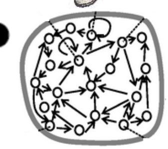
Embodied AI



Artificial Intelligence

Artificial neural networks

Reinforcement learning



Evolutionary computation

- Q learning (Off-policy TD control)→ State & action pair *(s,a)*

$$\Delta Q(s,a) = \alpha\left(r + \gamma \max_{a_1} Q(s_1,a_1) - Q(s,a)\right)$$

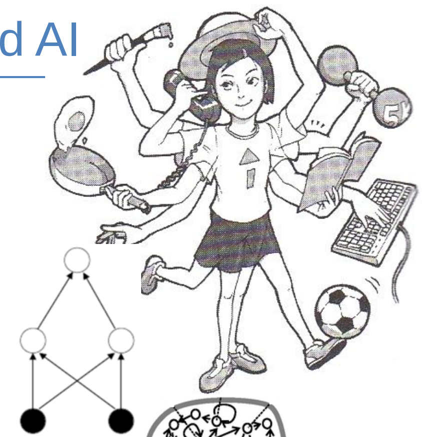
- SARSA (On-policy TD control) → State & action pair *(s,a)*

$$\Delta Q(s,a) = \alpha\left(r + \gamma\, Q(s_1,a_1) - Q(s,a)\right)$$

What have we learnt so far?

Contents

Embodied AI



Artificial Intelligence

Artificial neural networks

Reinforcement learning

Evolutionary computation

- Q learning (Off-policy TD control)→ State & action pair *(s,a)*

$$\Delta Q(s,a) = \alpha \left(r + \gamma \max_{a_1} Q(s_1,a_1) - Q(s,a) \right)$$

- SARSA (On-policy TD control) → State & action pair *(s,a)*

$$\Delta Q(s,a) = \alpha \left(r + \gamma Q(s_1,a_1) - Q(s,a) \right)$$

- Q learning/ SARSA & Neural networks as Q value approximator!

Today's Outline

- Actor-critic RL
- Combinatorial learning

Actor-Critic RL (On-Policy TD control)

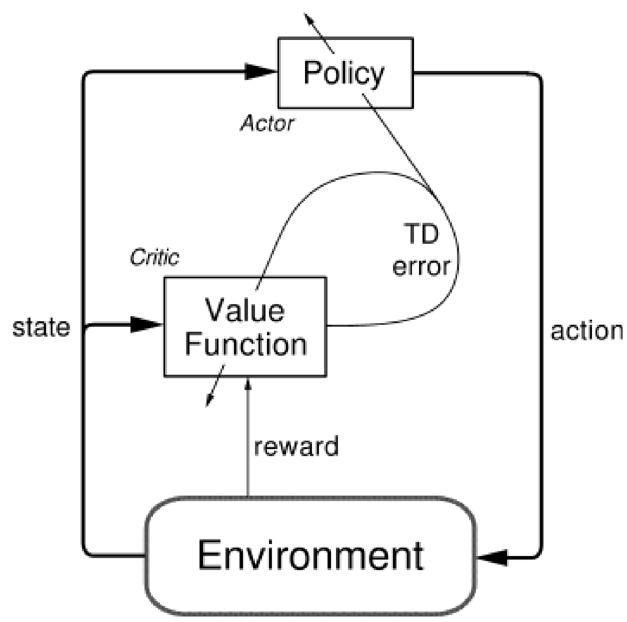
Actor-Critic RL (On-Policy TD control)

- Actor-critic RL (studied by Witten, 1977 and Barto, Sutton, and Anderson, 1983, 1984) is TD learning.

Actor-Critic RL (On-Policy TD control)

- Actor-critic RL (studied by Witten, 1977 and Barto, Sutton, and Anderson, 1983, 1984) is TD learning.

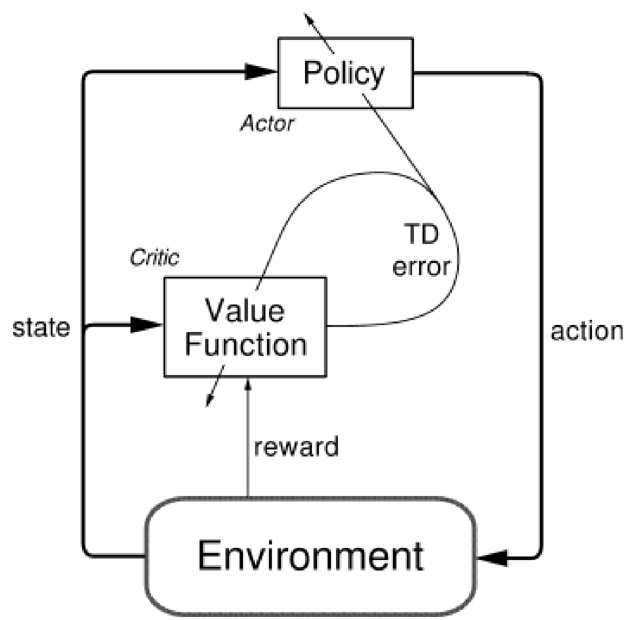
- It has a separate structure: 1) Actor network 2) Critic network



Actor-Critic RL (On-Policy TD control)

- Actor Network: given State compute Action
- Critic Network: given State predict Cumulative future Reward V(s)
- Train Critic & Actor based on prediction error (TD error)

$$\textbf{TDerror} = \delta(t) = \underbrace{r(t) + \gamma V(\mathbf{x}(t))}_{\textit{Target}} - \underbrace{V(\mathbf{x}(t-1))}_{\textit{Estimate}}$$

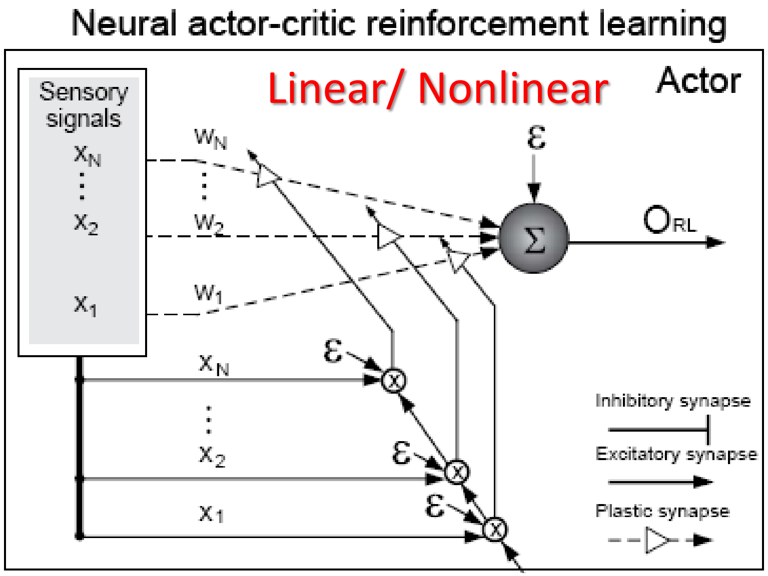


Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks



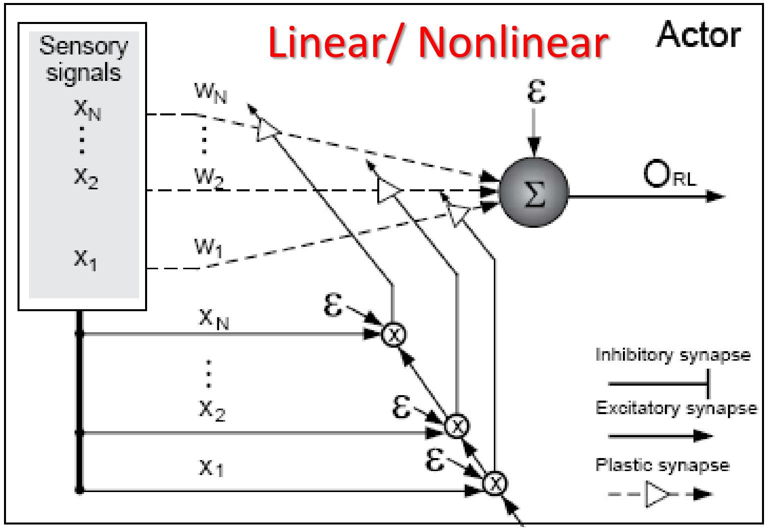
Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \underbrace{\varepsilon(t)}_{\text{exploration}} + \underbrace{\sum_{k=1}^{N} w_k(t) x_k(t)}_{\text{exploitation}}$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

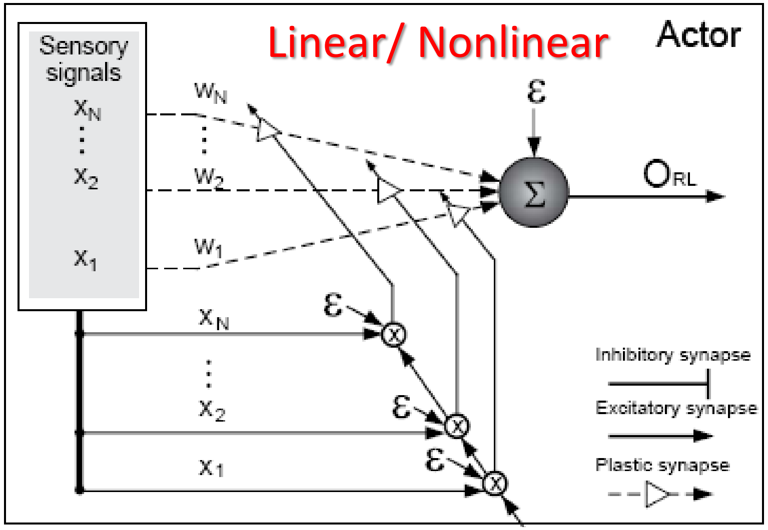
- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

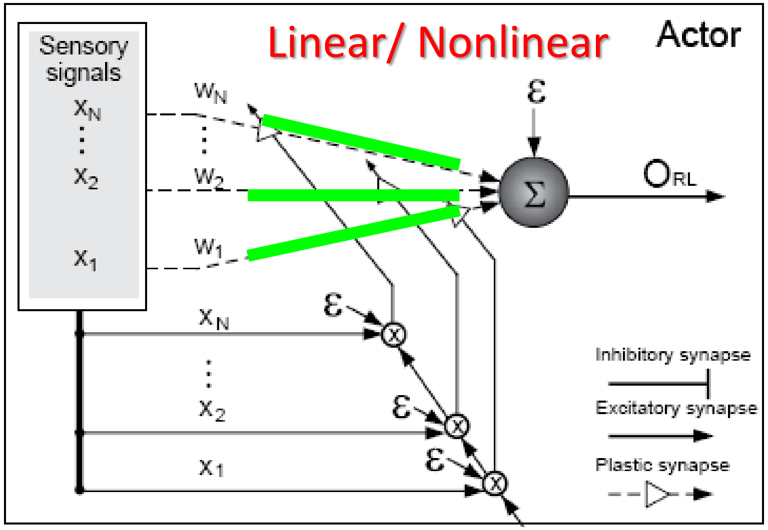
exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t)x_k(t)\varepsilon(t), \quad k = 1, \ldots, N,$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

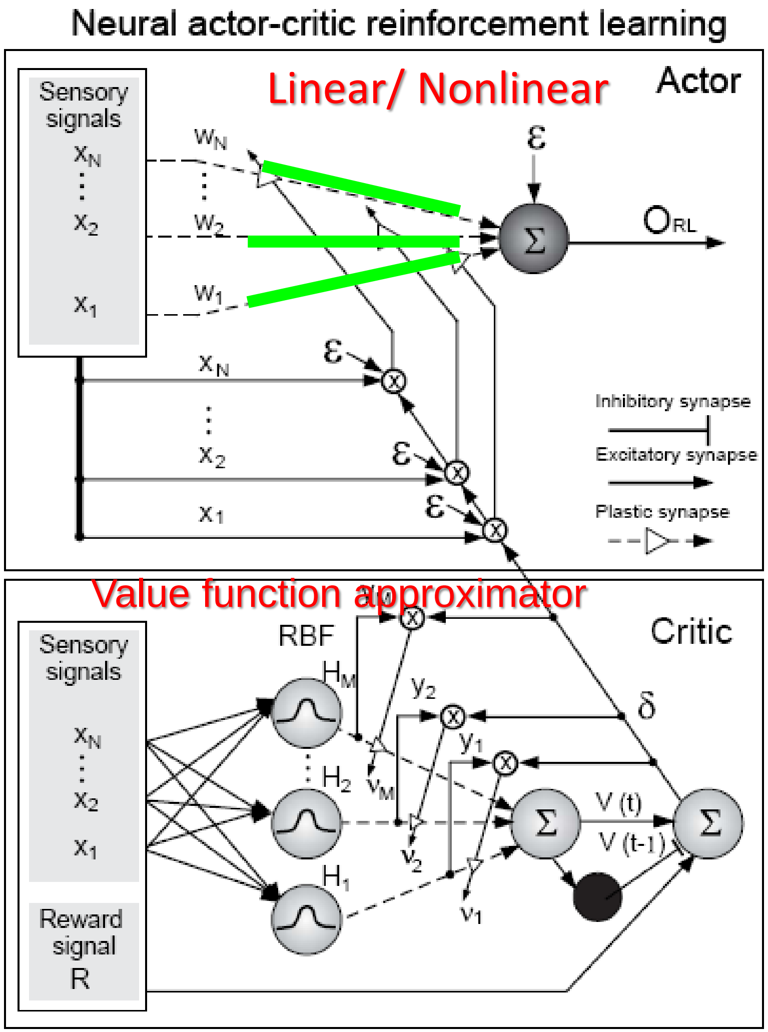
- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t)x_k(t)\varepsilon(t), \quad k = 1, \ldots, N,$$



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

exploration   exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

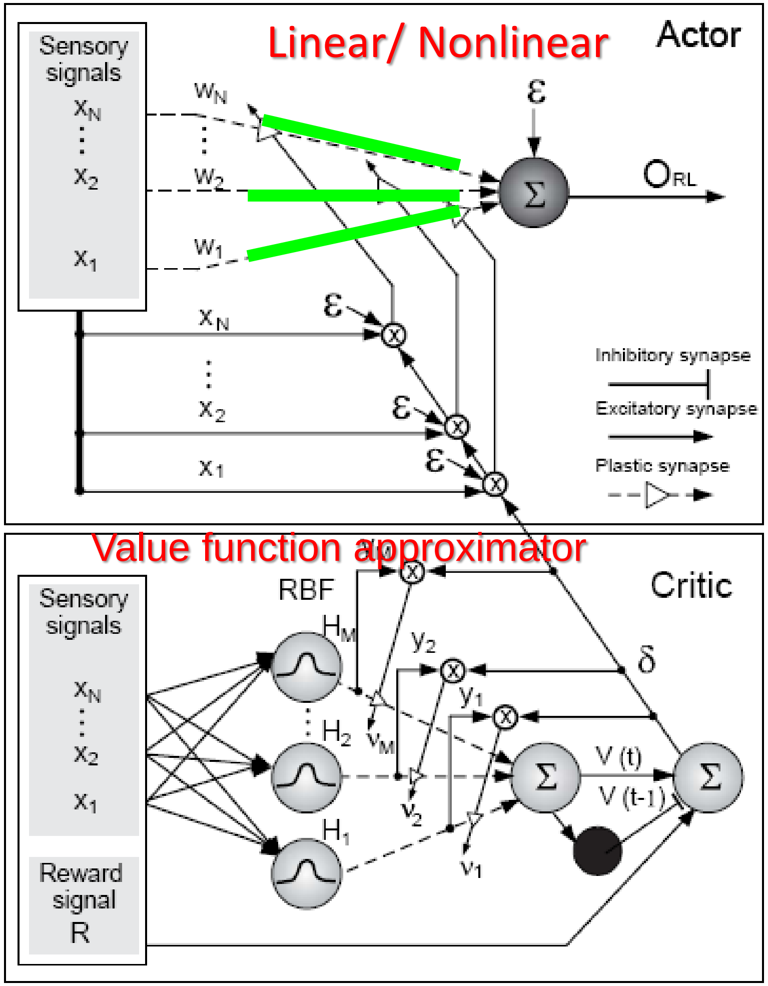
$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t) x_k(t) \varepsilon(t), \quad k = 1, \ldots, N,$$

$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t) y_j(\mathbf{x}(t))$$

expected cumulative future reward

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t) x_k(t) \varepsilon(t), \quad k = 1, \ldots, N,$$

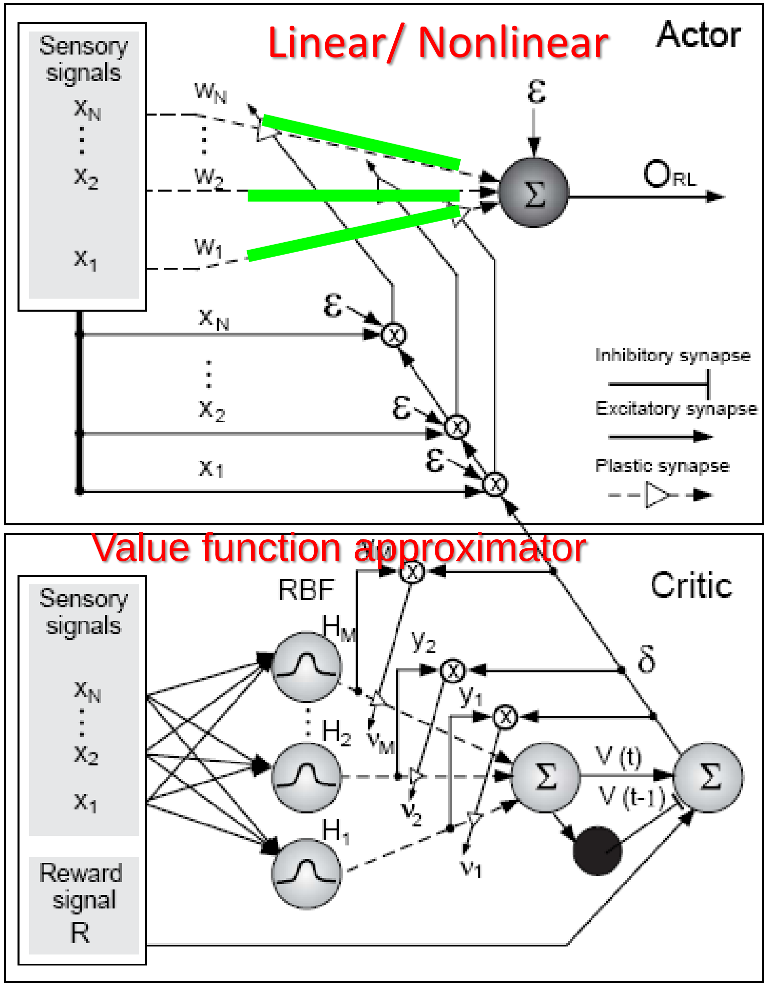
$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t) y_j(\mathbf{x}(t))$$

expected cumulative future reward

RBF

$$y_j(\mathbf{x}(t)) = \frac{a_j(\mathbf{x}(t))}{\sum_{l=1}^{M} a_l(\mathbf{x}(t))}, \quad a_j(\mathbf{x}(t)) = e^{-\|\mathbf{s}_j^T(\mathbf{x}(t) - \mathbf{c}_j)\|^2}$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t) x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t) x_k(t) \varepsilon(t), \quad k = 1, \ldots, N,$$

$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t) y_j(\mathbf{x}(t))$$

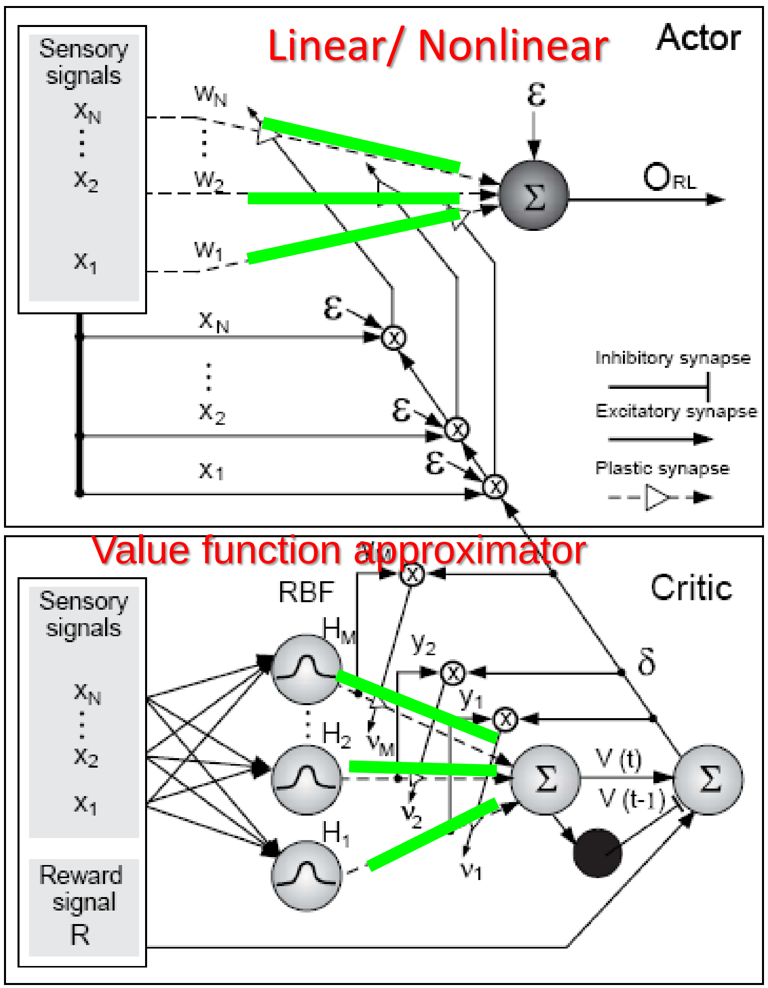
expected cumulative future reward

RBF

$$y_j(\mathbf{x}(t)) = \frac{a_j(\mathbf{x}(t))}{\sum_{l=1}^{M} a_l(\mathbf{x}(t))}, \quad a_j(\mathbf{x}(t)) = e^{-\|\mathbf{s}_j^T(\mathbf{x}(t) - \mathbf{c}_j)\|^2}$$

$$\Delta v_j(t) = \lambda\delta(t) y_j(\mathbf{x}(t)), \quad j = 1, \ldots, M,$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

$$O_{\mathrm{RL}}(t) = \overbrace{\varepsilon(t)}^{\text{exploration}} + \overbrace{\sum_{k=1}^{N} w_k(t) x_k(t)}^{\text{exploitation}}$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t) x_k(t)\varepsilon(t), \quad k = 1, \ldots, N,$$

$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t) y_j(\mathbf{x}(t))$$

expected cumulative future reward

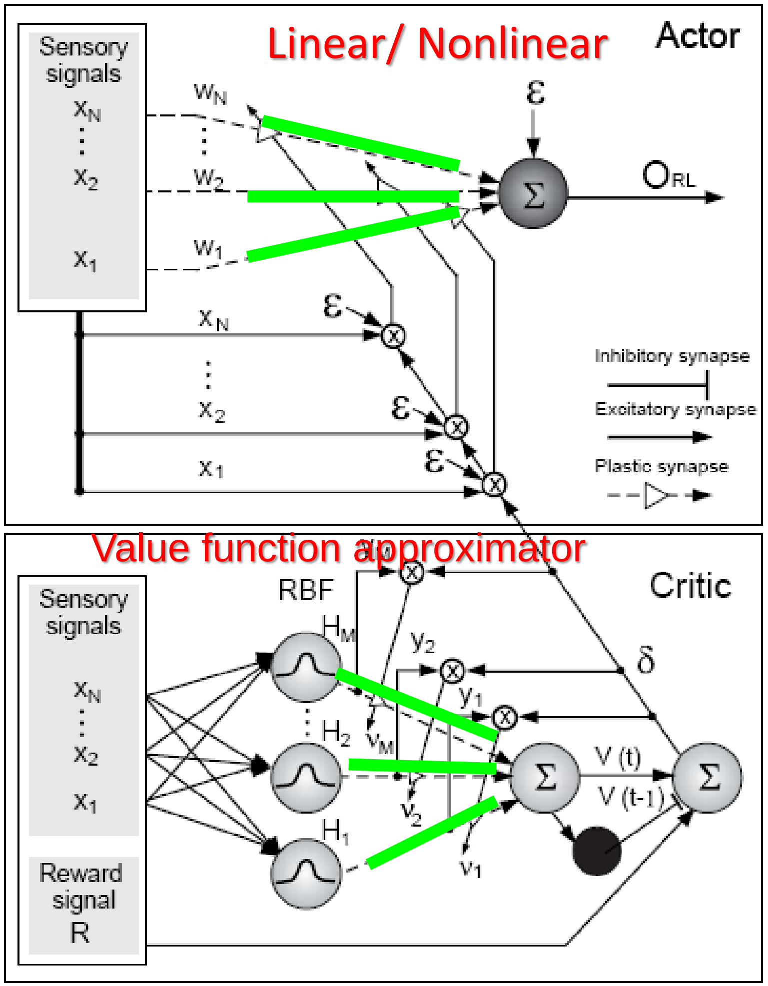
RBF

$$y_j(\mathbf{x}(t)) = \frac{a_j(\mathbf{x}(t))}{\sum_{l=1}^{M} a_l(\mathbf{x}(t))}, \quad a_j(\mathbf{x}(t)) = e^{-\|\mathbf{s}_j^T(\mathbf{x}(t) - \mathbf{c}_j)\|^2}$$

$$\Delta v_j(t) = \lambda\delta(t) y_j(\mathbf{x}(t)), \quad j = 1, \ldots, M,$$

$$\delta(t) = r(t) + \gamma V(\mathbf{x}(t)) - V(\mathbf{x}(t-1))$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- Continuous Actor-Critic RL & Neural Networks

exploration exploitation

$$O_{\mathrm{RL}}(t) = \varepsilon(t) + \sum_{k=1}^{N} w_k(t)x_k(t)$$

$$\varepsilon(t) = \xi\sigma(t) \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}(t))}{V_{\max} - V_{\min}}\right]\right]$$

$$\Delta w_k = \alpha\delta(t)x_k(t)\varepsilon(t), \quad k = 1, \ldots, N,$$

$$V(\mathbf{x}(t)) = \sum_{j=1}^{M} v_j(t)y_j(\mathbf{x}(t))$$

expected cumulative future reward

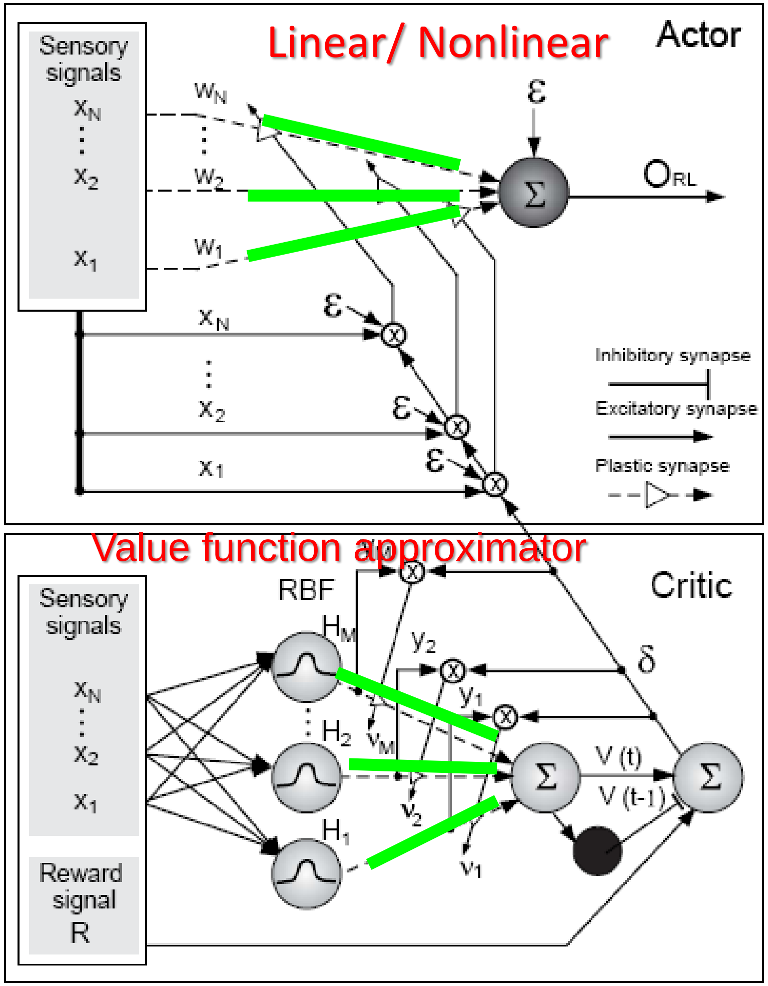
RBF

$$y_j(\mathbf{x}(t)) = \frac{a_j(\mathbf{x}(t))}{\sum_{l=1}^{M} a_l(\mathbf{x}(t))}, \quad a_j(\mathbf{x}(t)) = e^{-\|\mathbf{s}_j^T(\mathbf{x}(t)-\mathbf{c}_j)\|^2}$$

$$\Delta v_j(t) = \lambda\delta(t)y_j(\mathbf{x}(t)), \quad j = 1, \ldots, M,$$

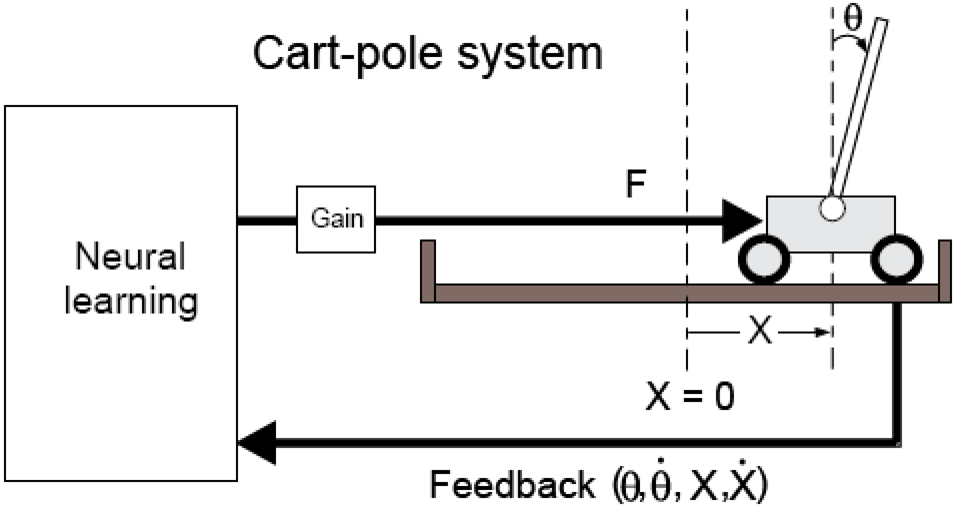
$$\delta(t) = r(t) + \gamma V(\mathbf{x}(t)) - V(\mathbf{x}(t-1))$$

Neural actor-critic reinforcement learning



Actor-Critic RL (On-Policy TD control)

- **Example1:** Dynamical Control Problem (Pole Balancing)



Manoonpong et al., 2013

Actor-Critic RL (On-Policy TD control)

- **Example1:** Dynamical Control Problem (Pole Balancing)

Reward values:
r = -1 at failure, 0 otherwise

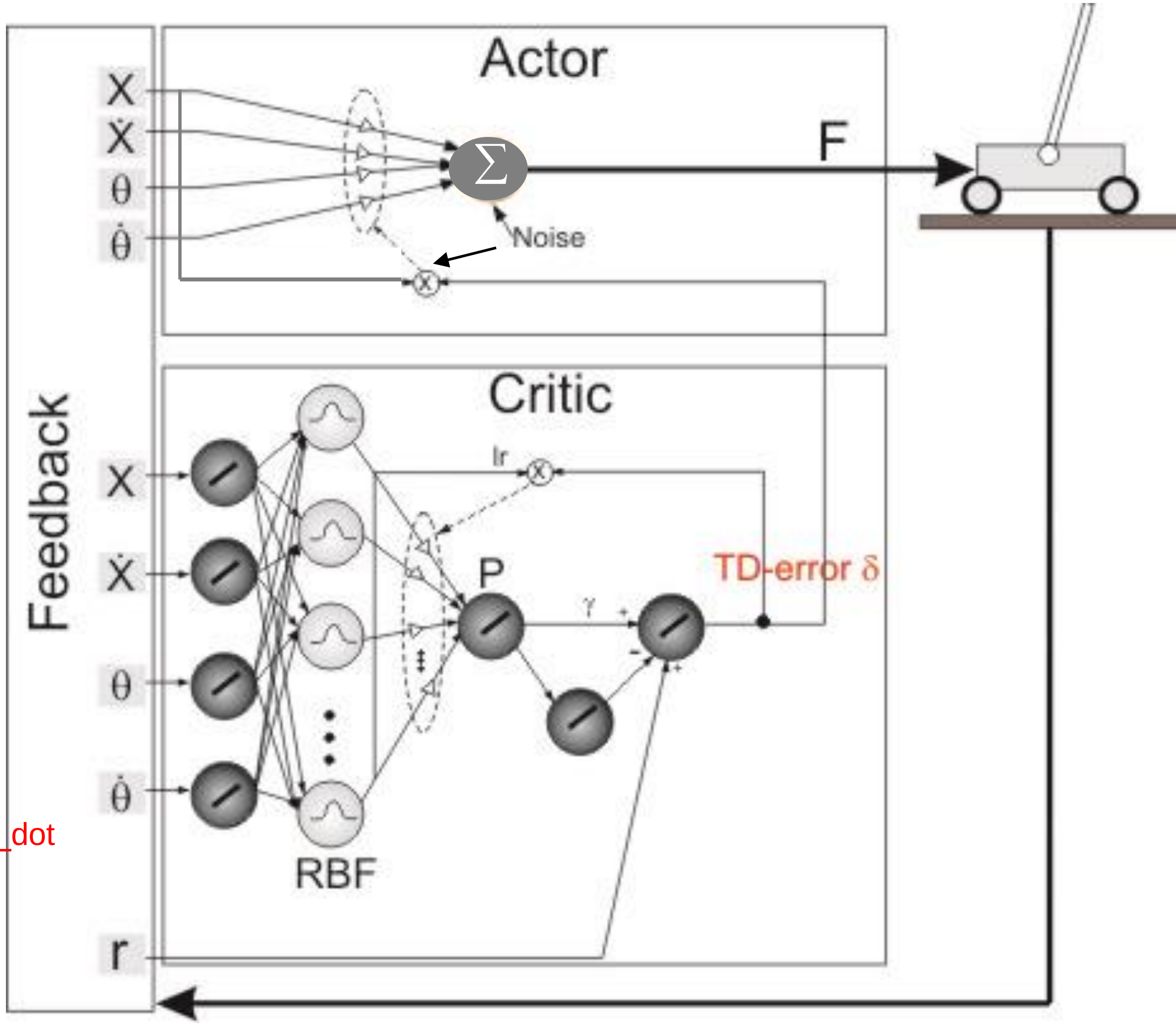
Exploration mechanism:
Vmax = 0
Vmin = -1
ξ Scale factor of exploration = 5.0

Actor:
α Learning rate = 0.5

Critic:
λ Learning rate = 0.5
RBF = 162 hidden neurons

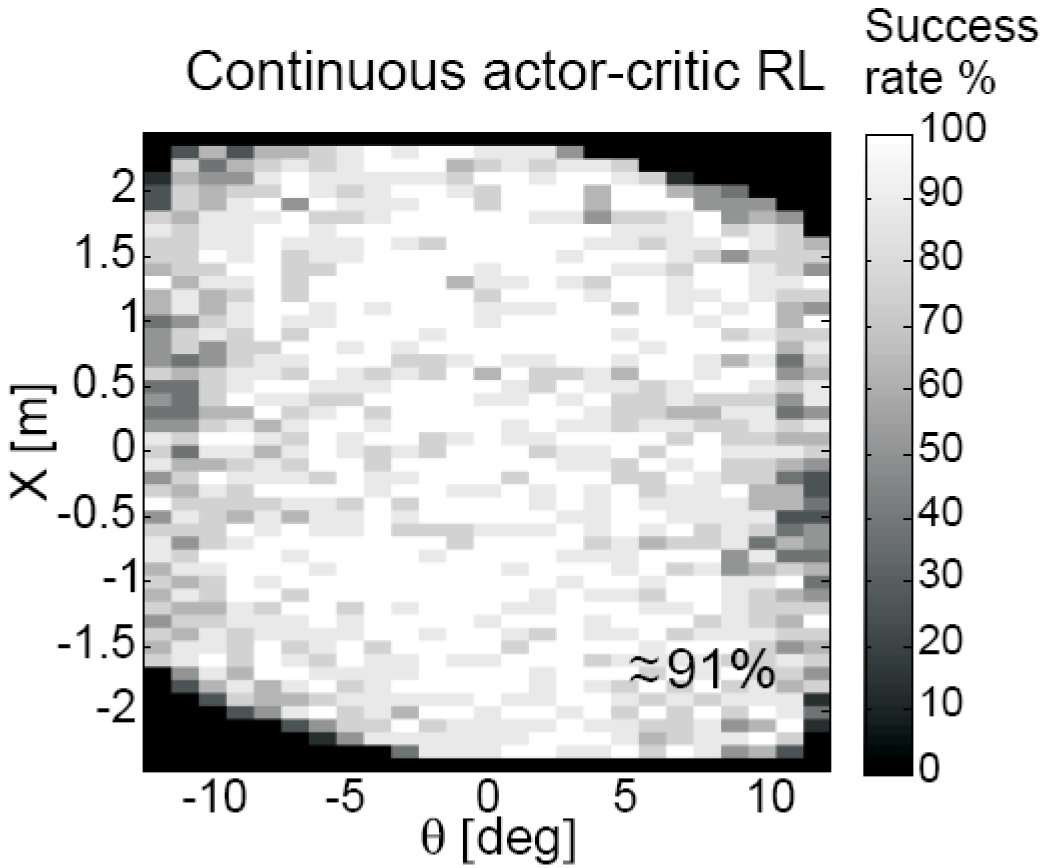
3 for x
3 for x_dot
6 for theta
3 for theta_dot

TD error:
γ Discount factor = 0.95



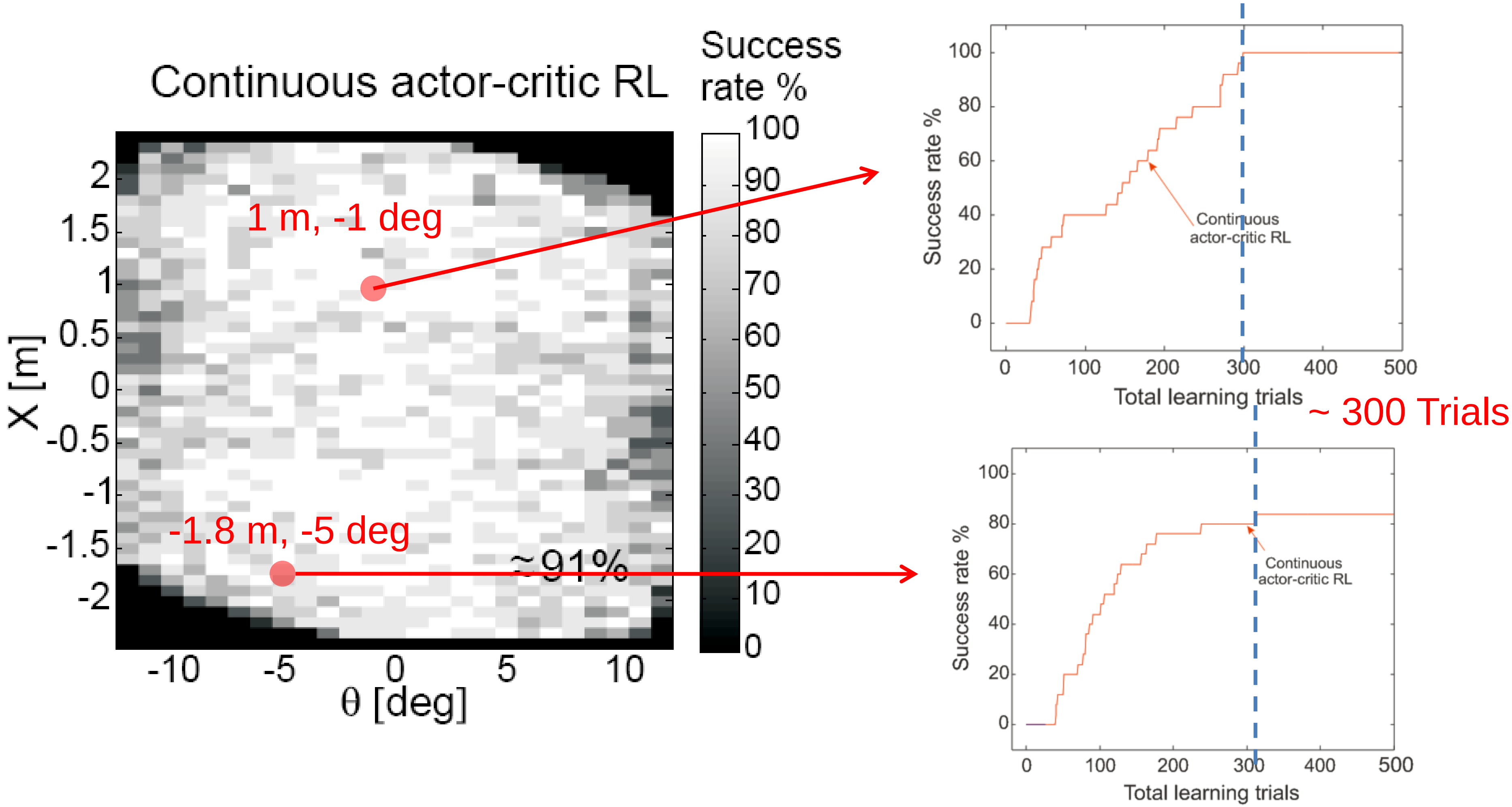
Actor-Critic RL (On-Policy TD control)

- **Example1:** Dynamical Control Problem (Pole Balancing)



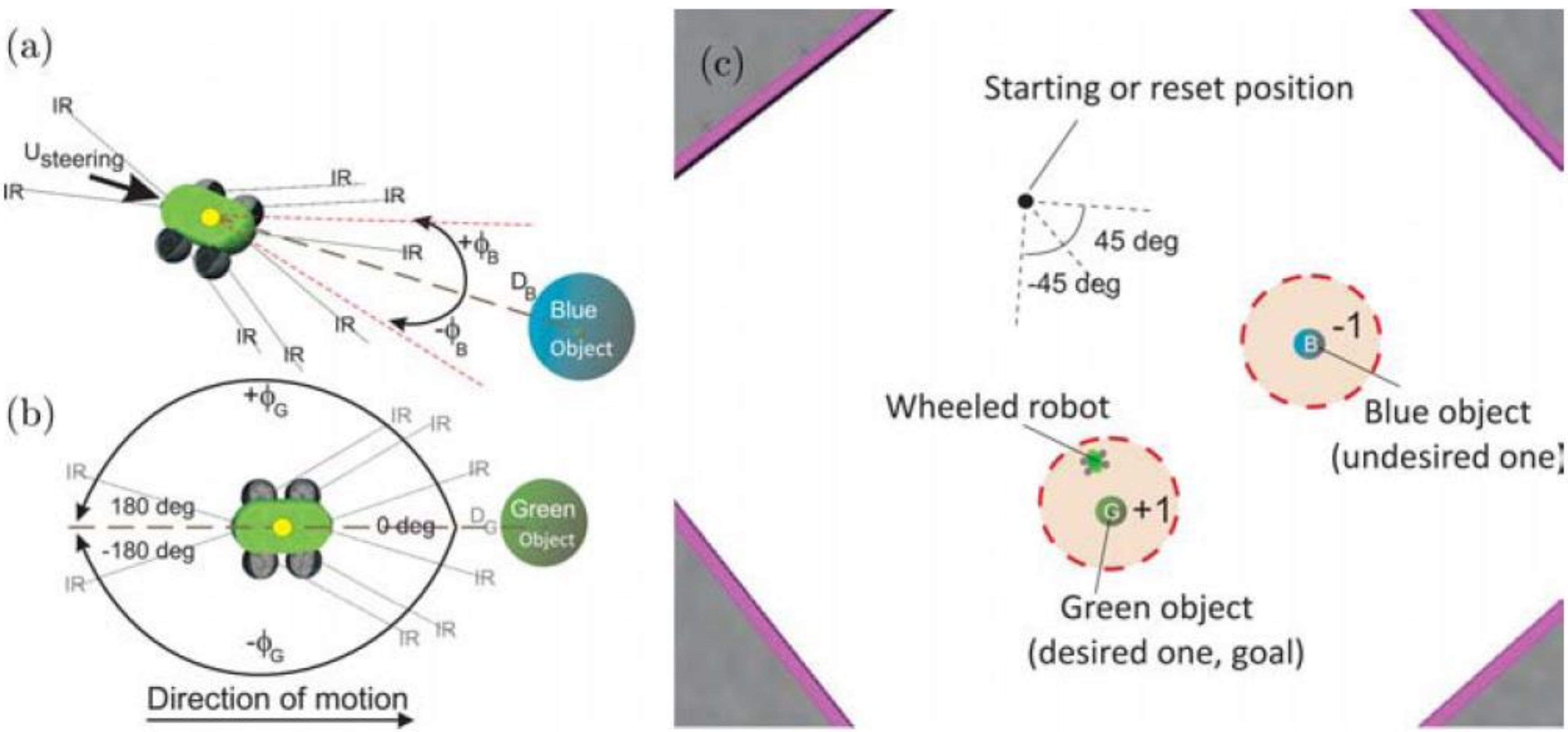
Actor-Critic RL (On-Policy TD control)

- **Example1:** Dynamical Control Problem (Pole Balancing)



Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)



Manoonpong et al., 2013

Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

It is not so easy to steer!. If you have not learnt

Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

Reward values:
r = -1 at blue object, +1 at green object

Exploration mechanism:
Vmax = 50
Vmin = 0
ξ Scale factor of exploration = 5.0

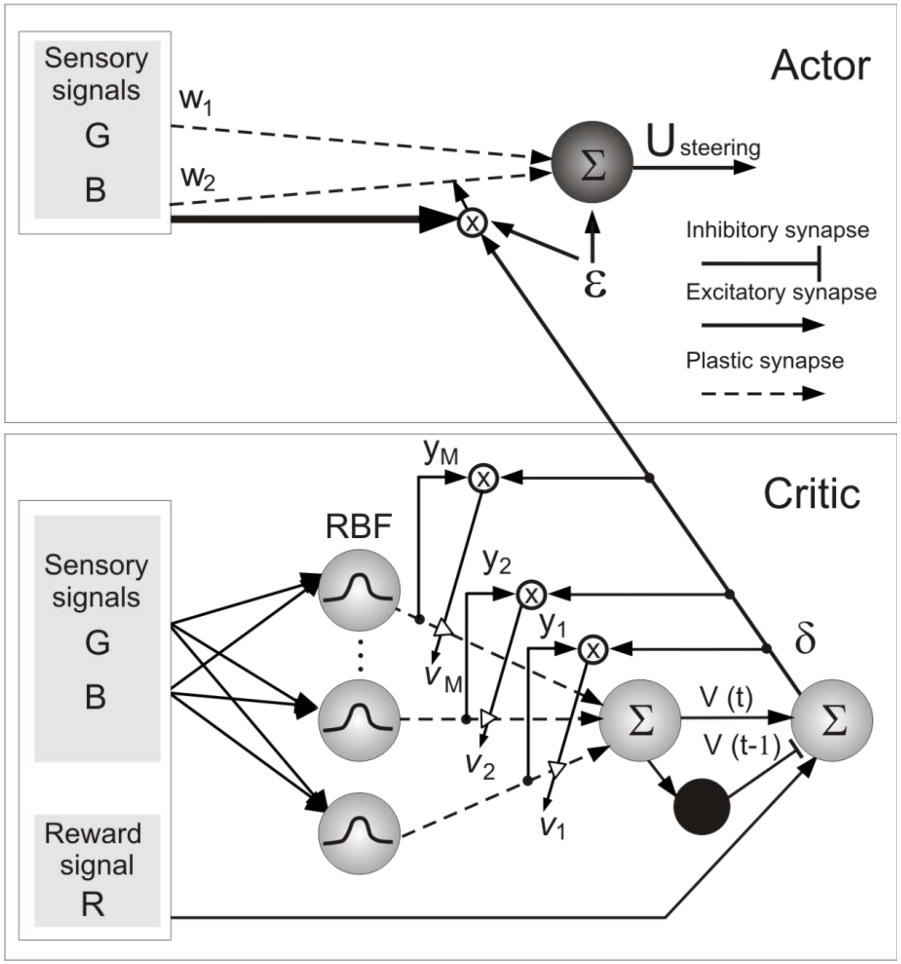
Actor:
α Learning rate = 0.001

Critic:
λ Learning rate = 0.7
RBF = 16 hidden neurons

G= 4
B =4

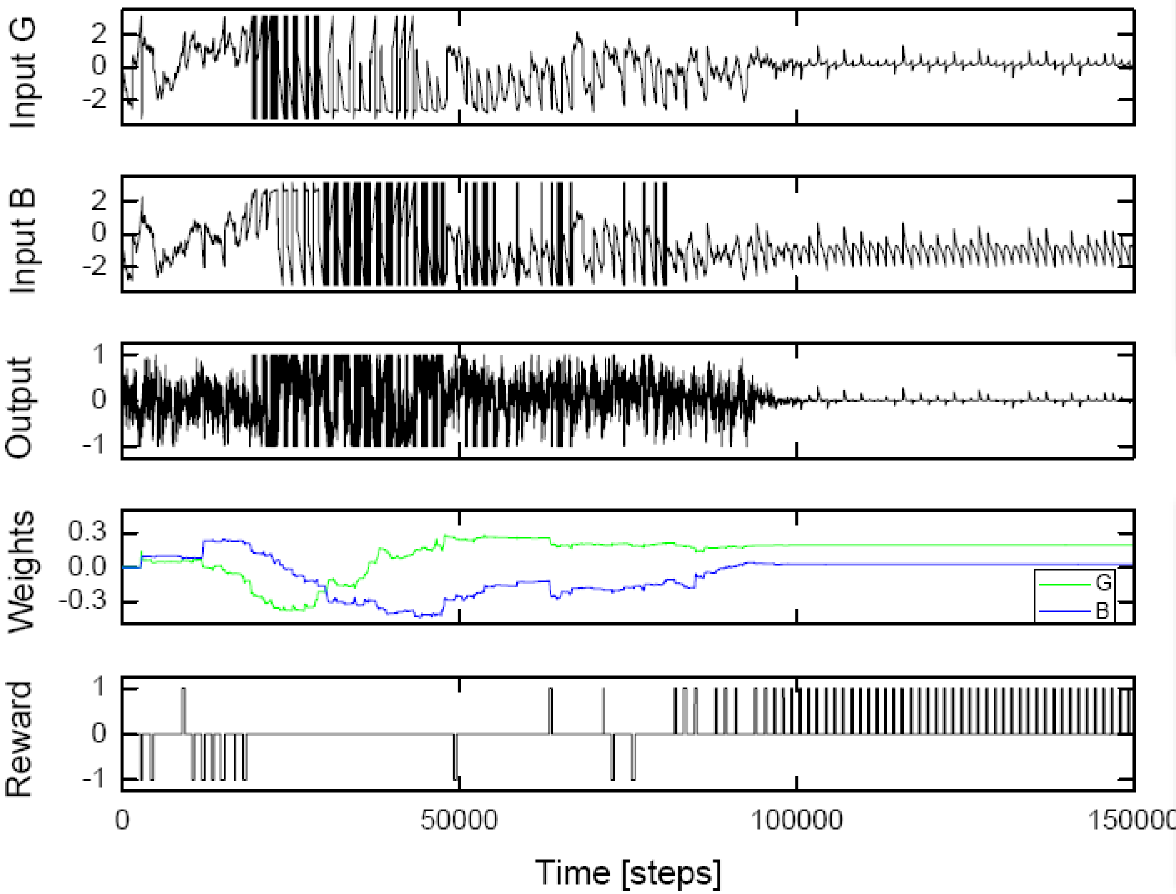
TD error:
γ Discount factor = 0.98

Neural actor-critic learning

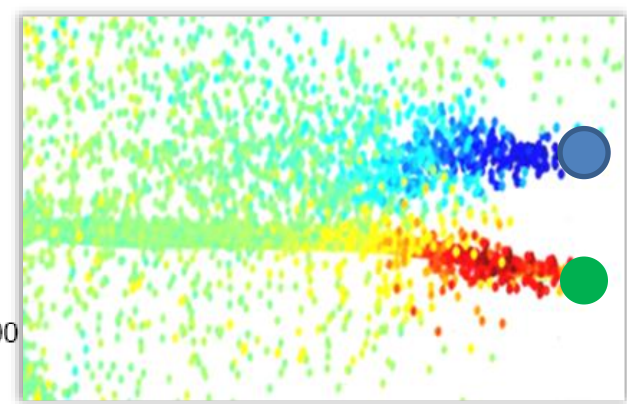


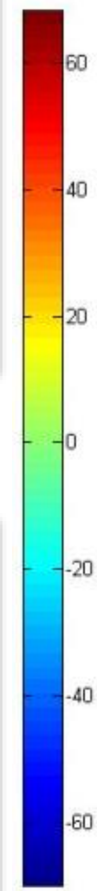
Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)



Value function
(Expected cumulative reward)

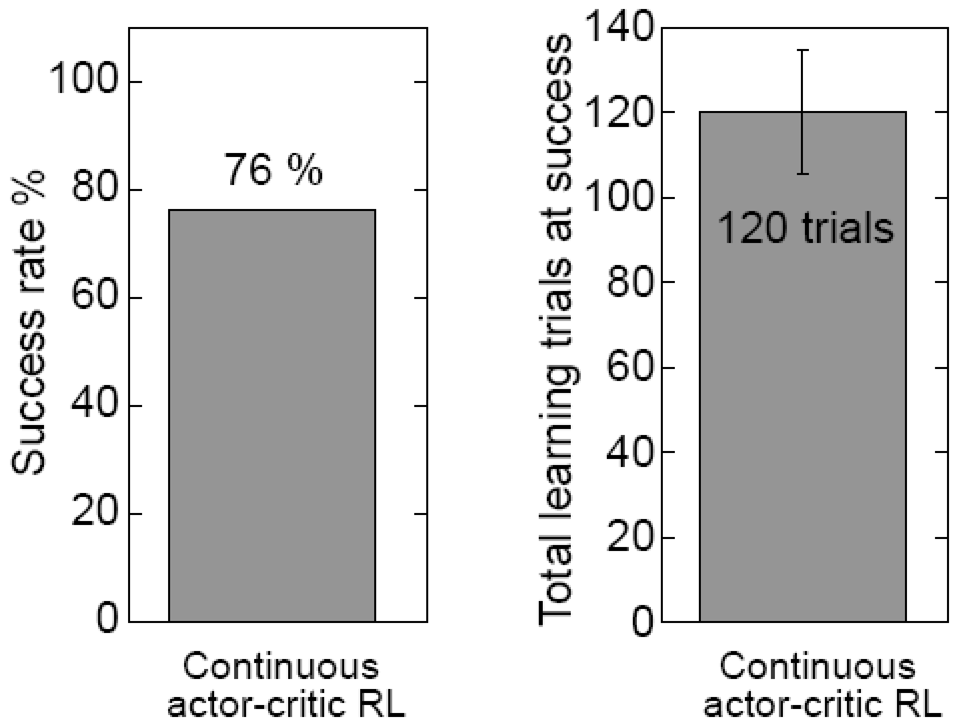




Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

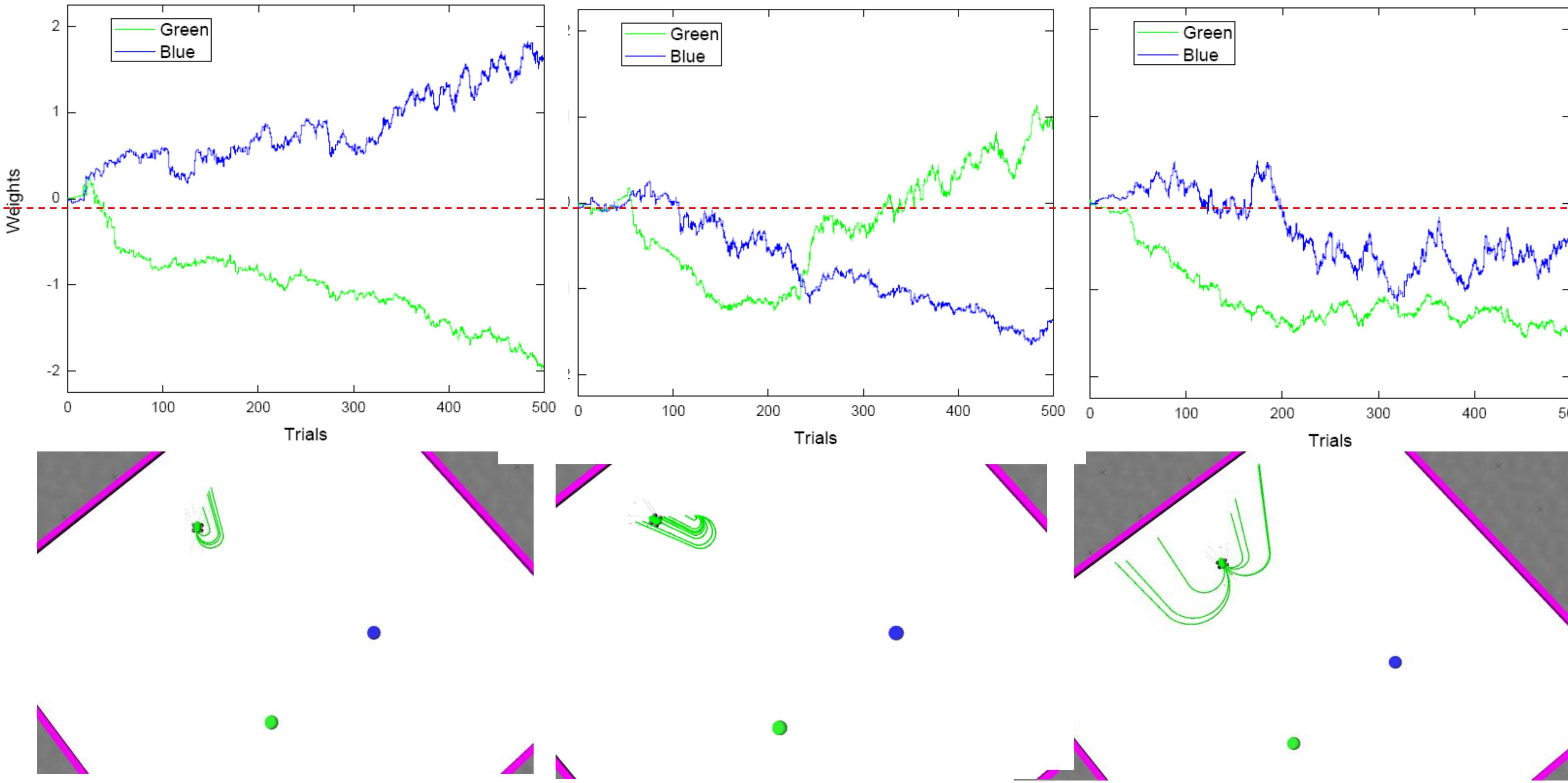
50 Experiments



What's the problem of 24%?

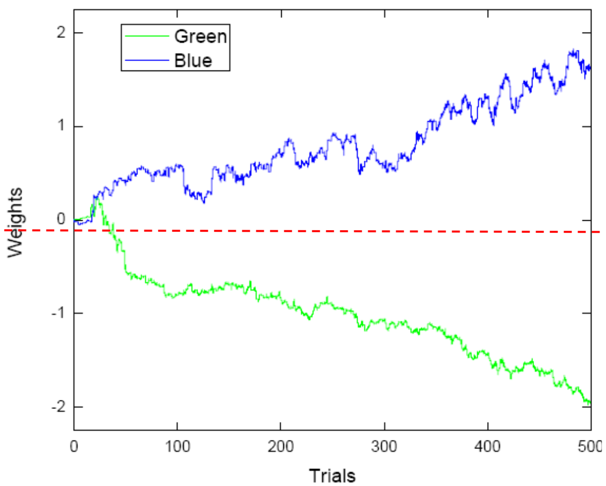
Actor-Critic RL (On-Policy TD control)

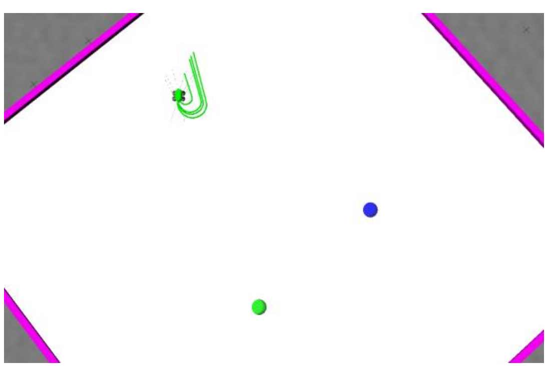
- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

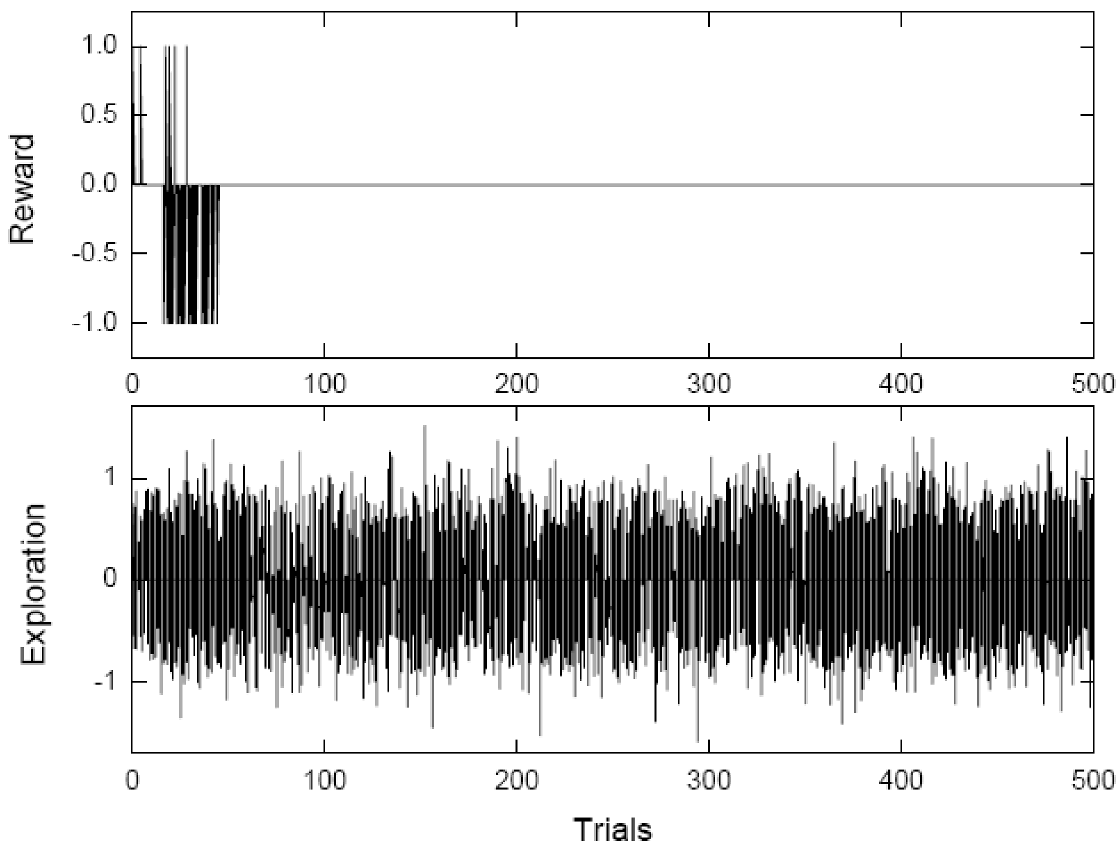


Actor-Critic RL (On-Policy TD control)

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

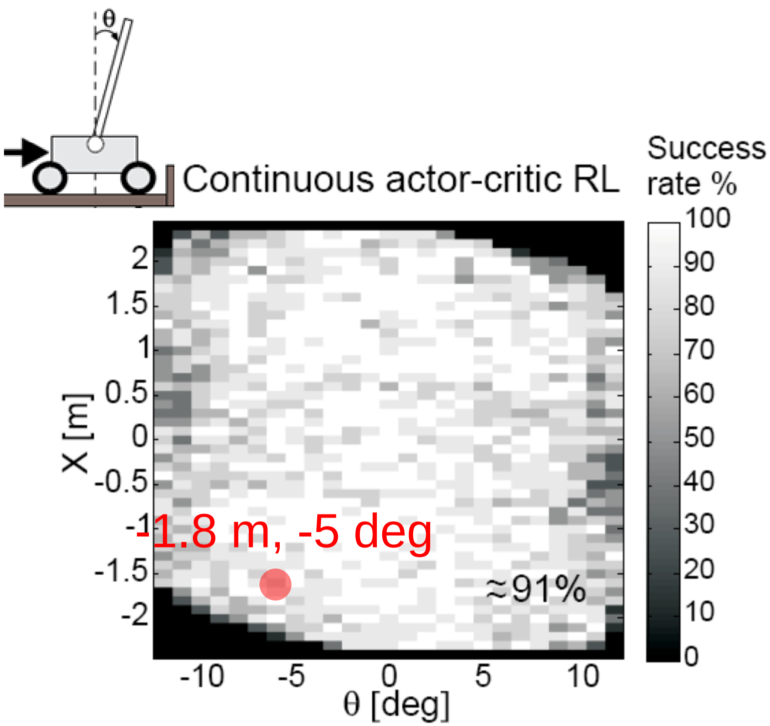


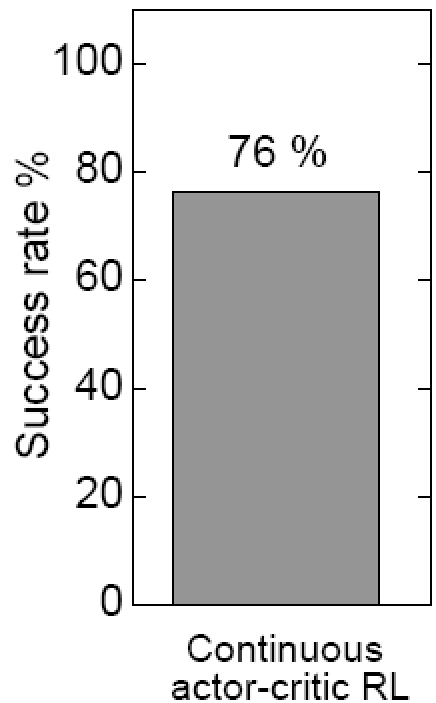


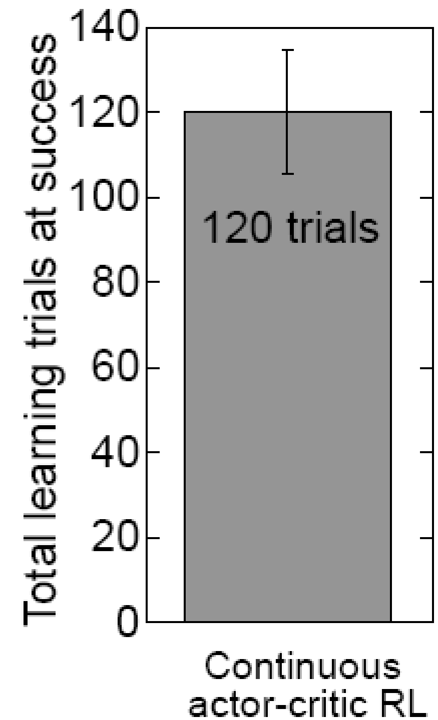


Actor-Critic RL (On-Policy TD control)

- **Example1:** Dynamical Control Problem (Pole Balancing)
- **Example2:** Goal-directed Behavior Control Problem (Mobile Robot)







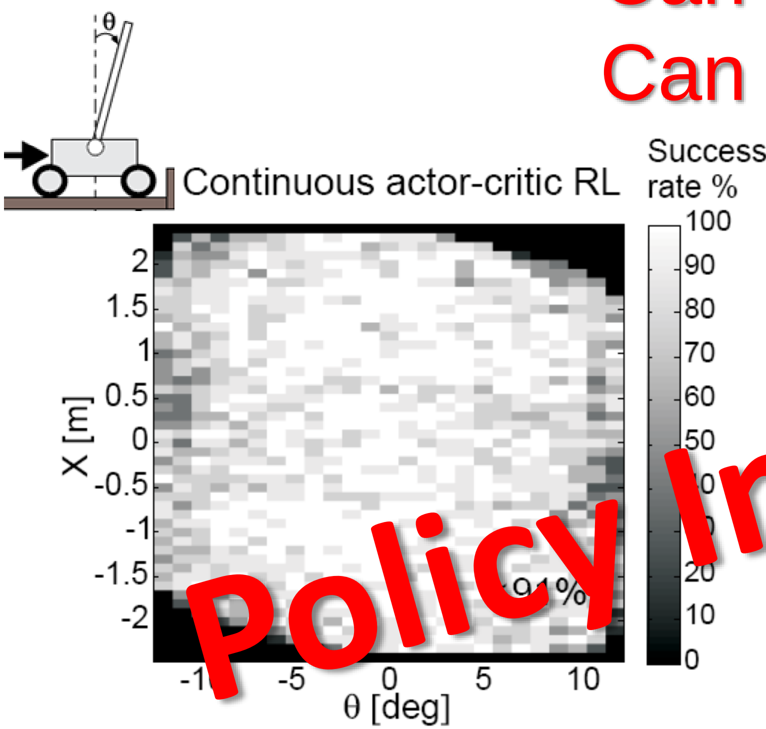
e.g., ~300 Trials at -1.8 m, -5 deg

Actor-Critic RL (On-Policy TD control)

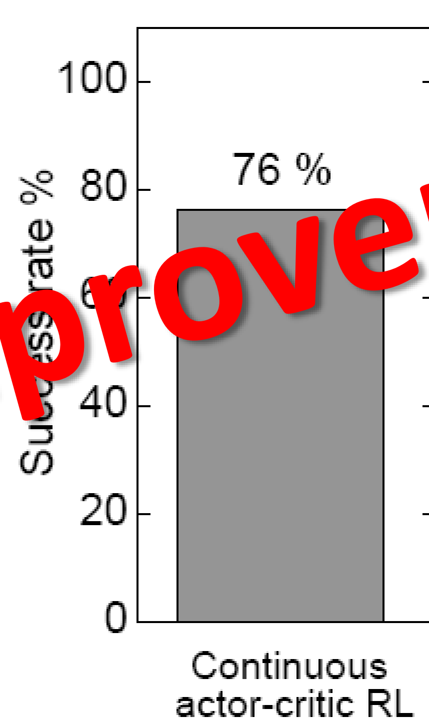
- **Example1:** Dynamical Control Problem (Pole Balancing)
- **Example2:** Goal-directed Behavior Control Problem (Mobile Robot)

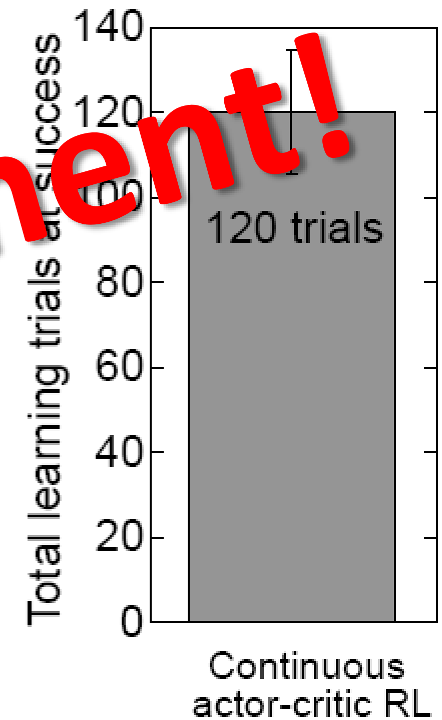
Can we reduce the number of trials?

Can we increase success rate?



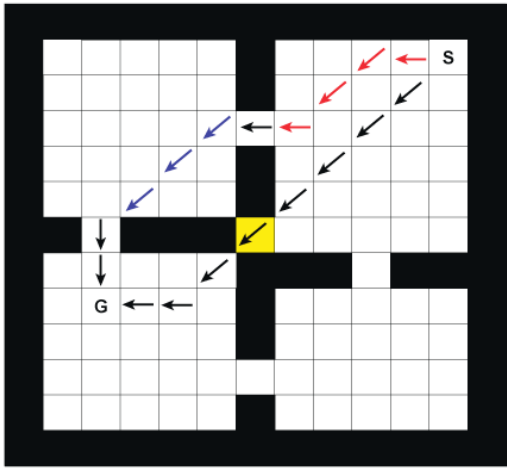
e.g., ~300 Trials at -1.8 m, -5 deg

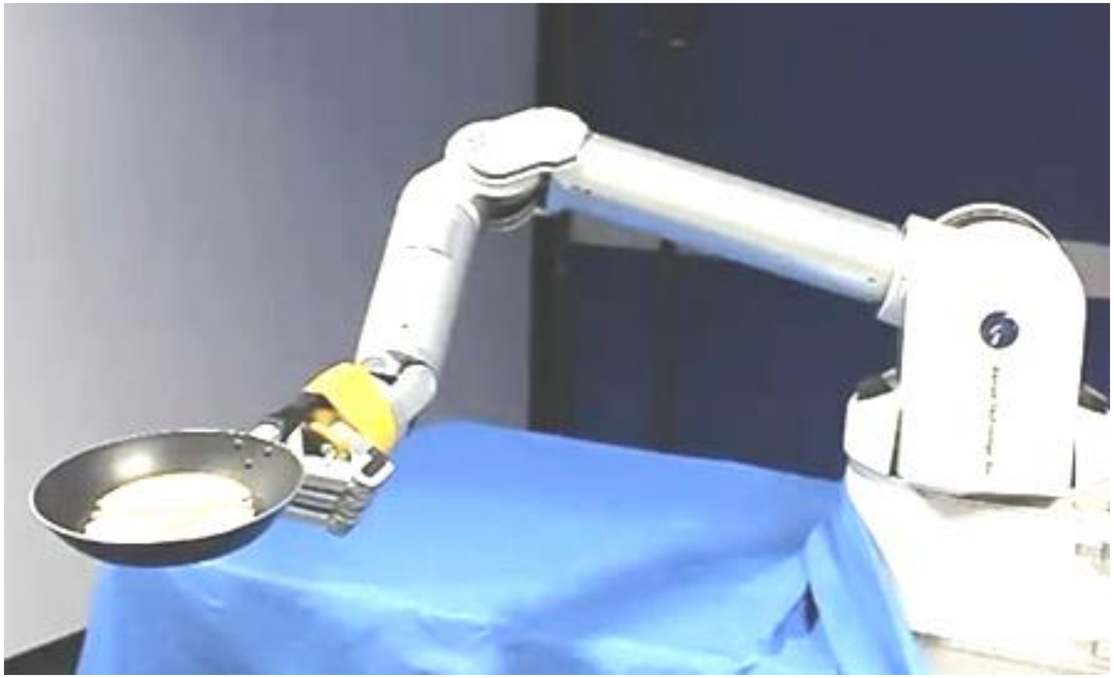




Policy Improvement!

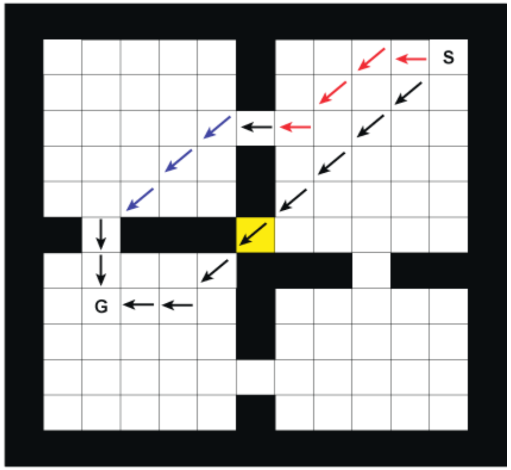
Different Approaches for Policy Improvement





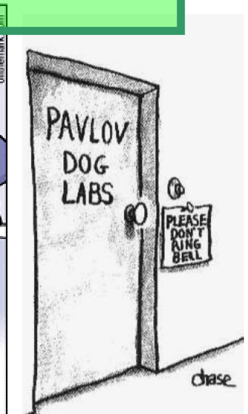
- **Imitative Reinforcement Learning → Prior knowledge** (Price & Boutilier 2003, Latzke et al, 2006)
- **Hierarchical RL → Sub goals (Options)** (Botvinick et al, 2008)
- **Self-Organizing Exploration in Reinforcement Learning → Exploration** (Simón Bize)
- ……

Different Approaches for Policy Improvement



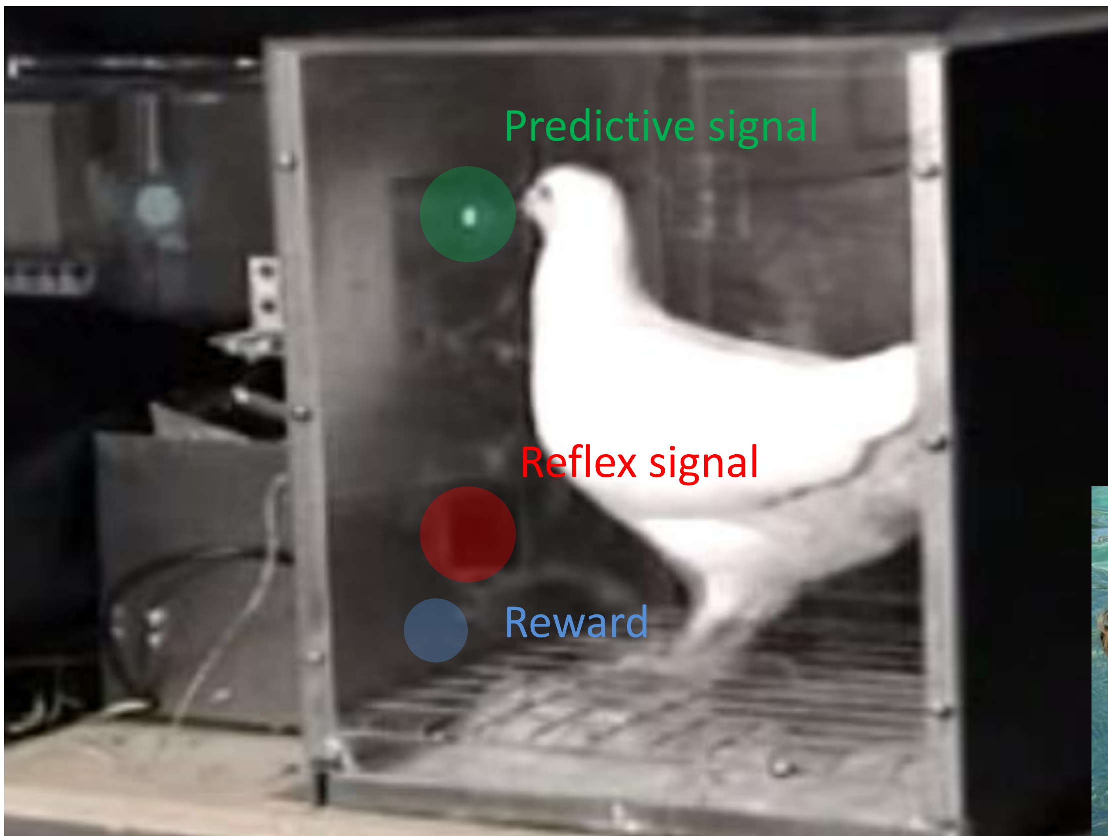
- **Imitative Reinforcement Learning → Prior knowledge** (Price & Boutilier 2003, Latzke et al, 2006)
- **Hierarchical RL → Sub goals (Options)** (Botvinick et al, 2008)
- **Self-Organizing Exploration in Reinforcement Learning → Exploration** (Simón Bize)
- ……

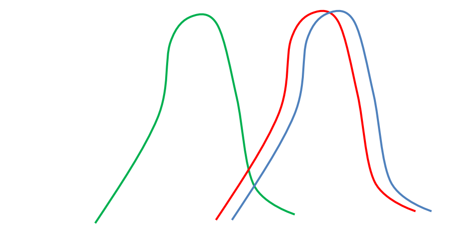
Learning in Biological Systems



Combinatorial Learning in Animals

Operant conditioning (Reward based learning) / Classical conditioning (Correlation based learning)

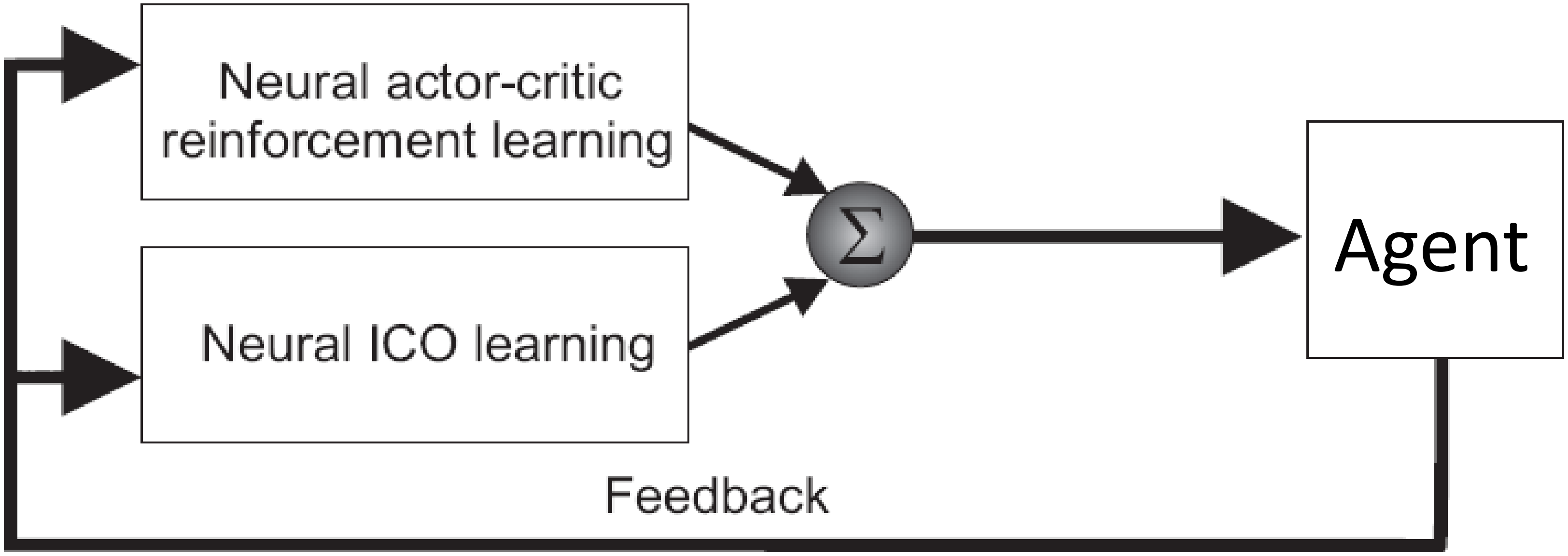




Bio-inspired Combinatorial Learning

How can we make a model of such combinatorial learning?

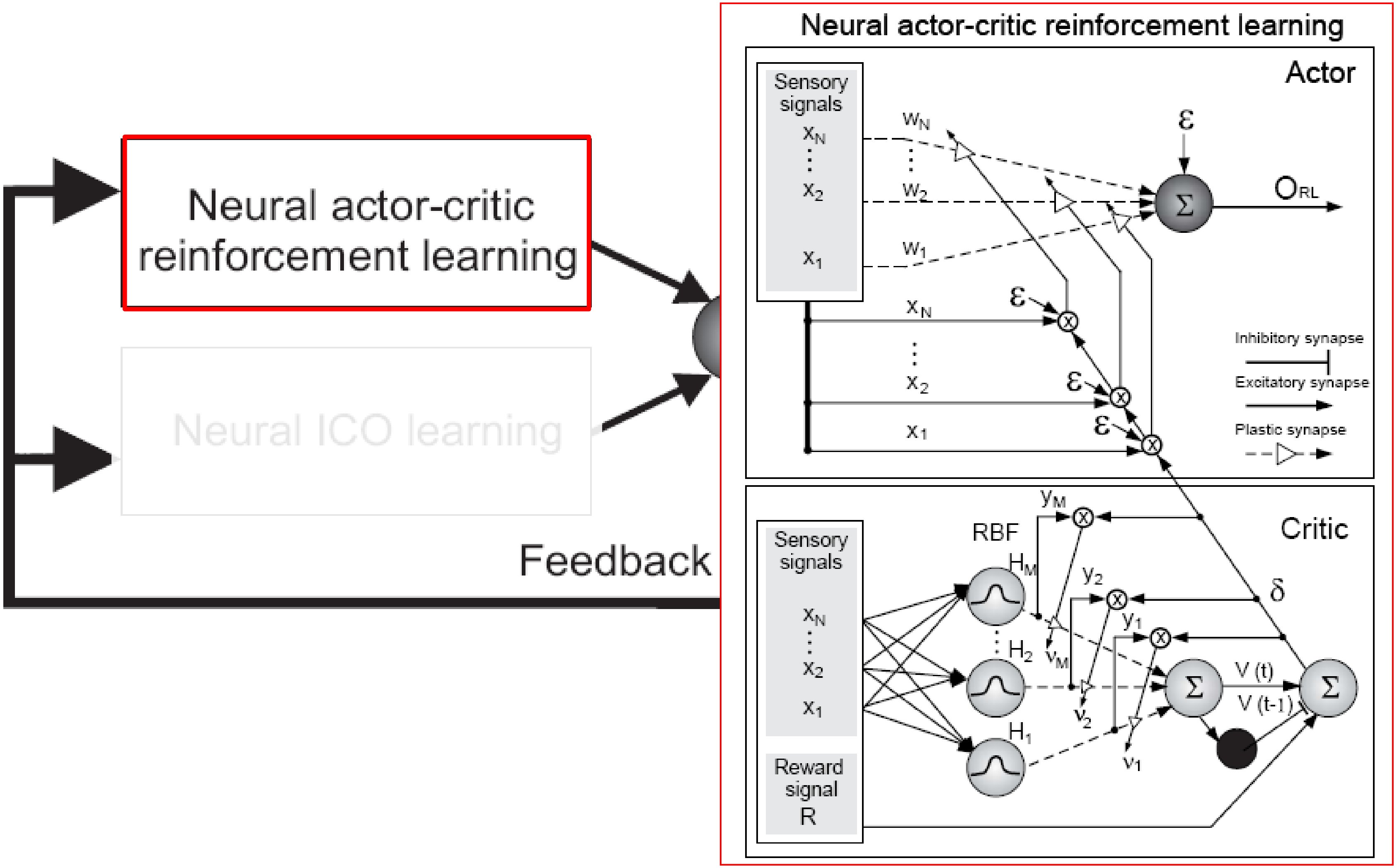
Bio-inspired Combinatorial Learning



"Parallel Combination"

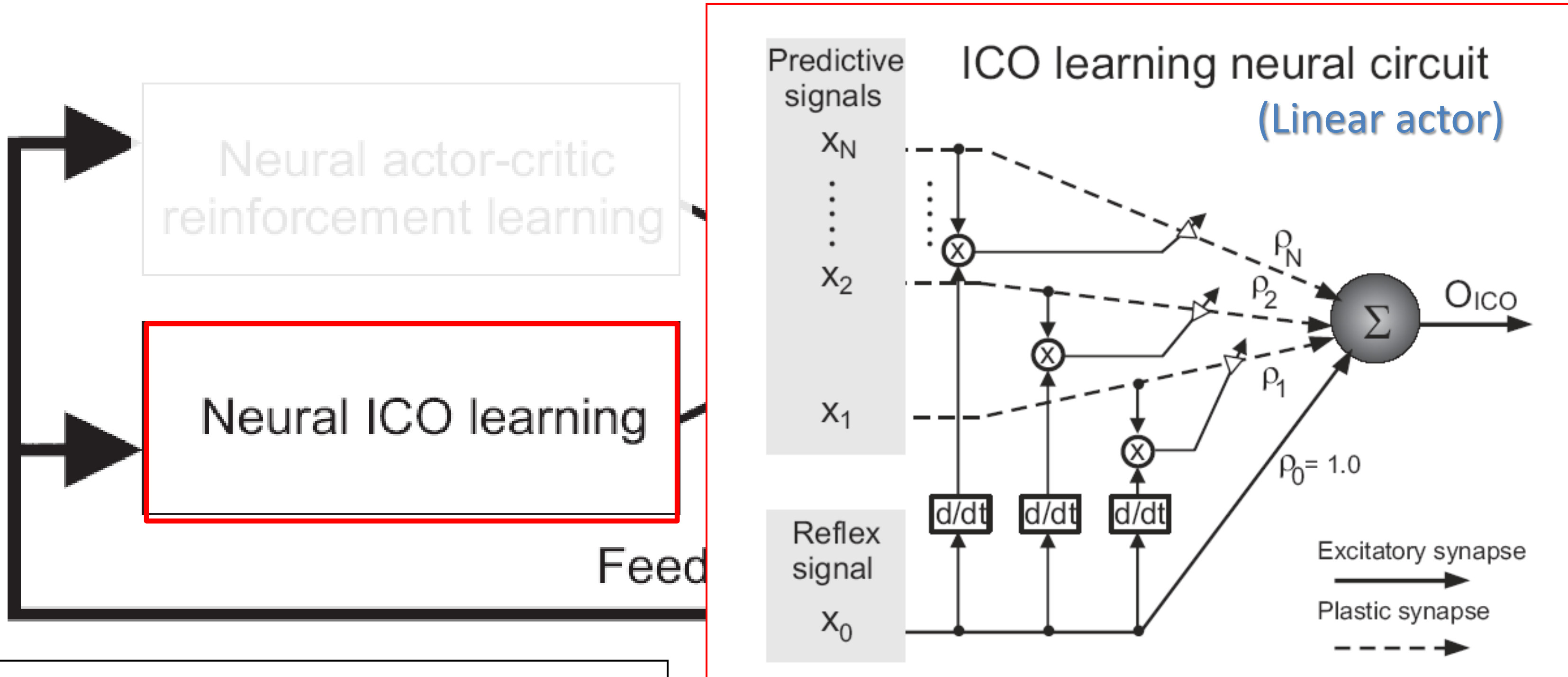
Manoonpong et al., 2013

Bio-inspired Combinatorial Learning



Bio-inspired Combinatorial Learning

Unsupervised learning

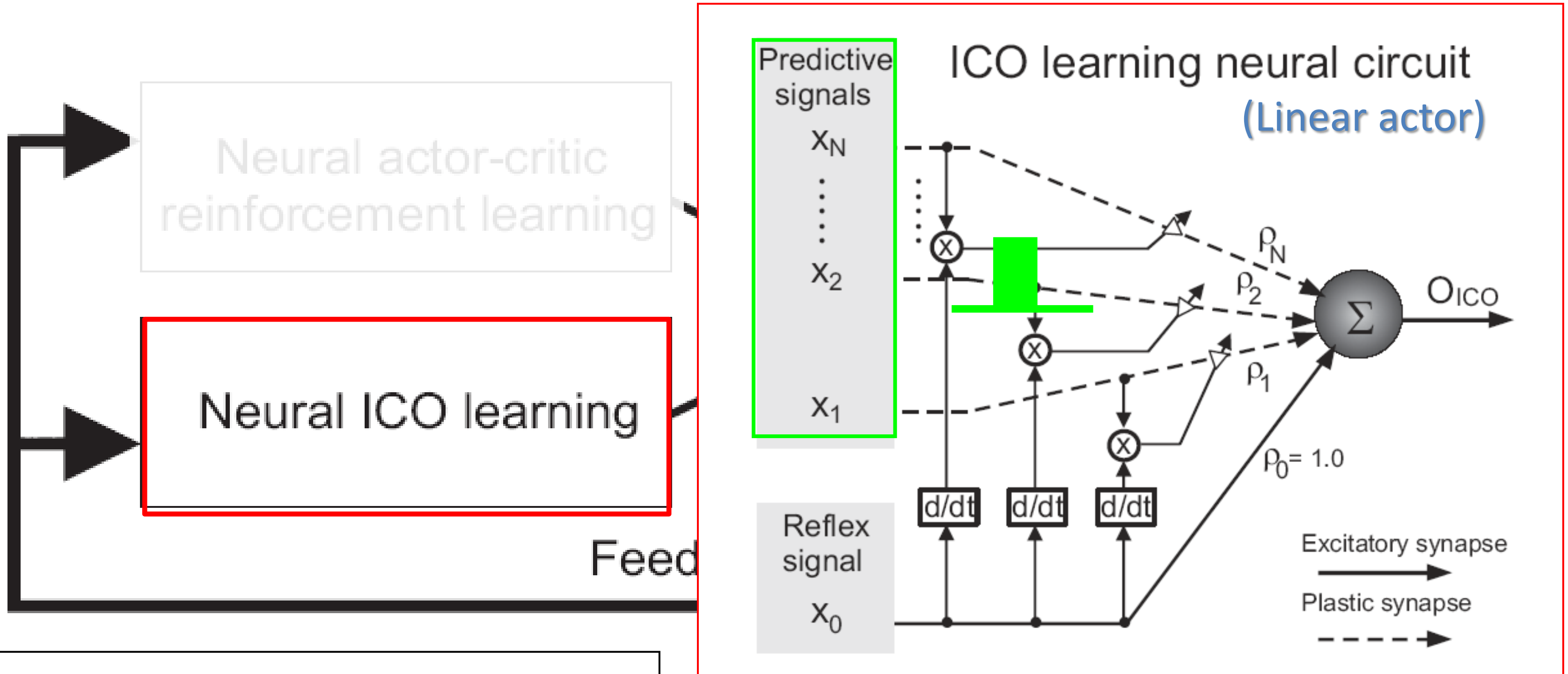


$$O_{ICO}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t)$$

$$\Delta \rho_k = \mu x_k(t) \frac{dx_0(t)}{dt}, \quad k = 1, \ldots, N$$

Bio-inspired Combinatorial Learning

Unsupervised learning

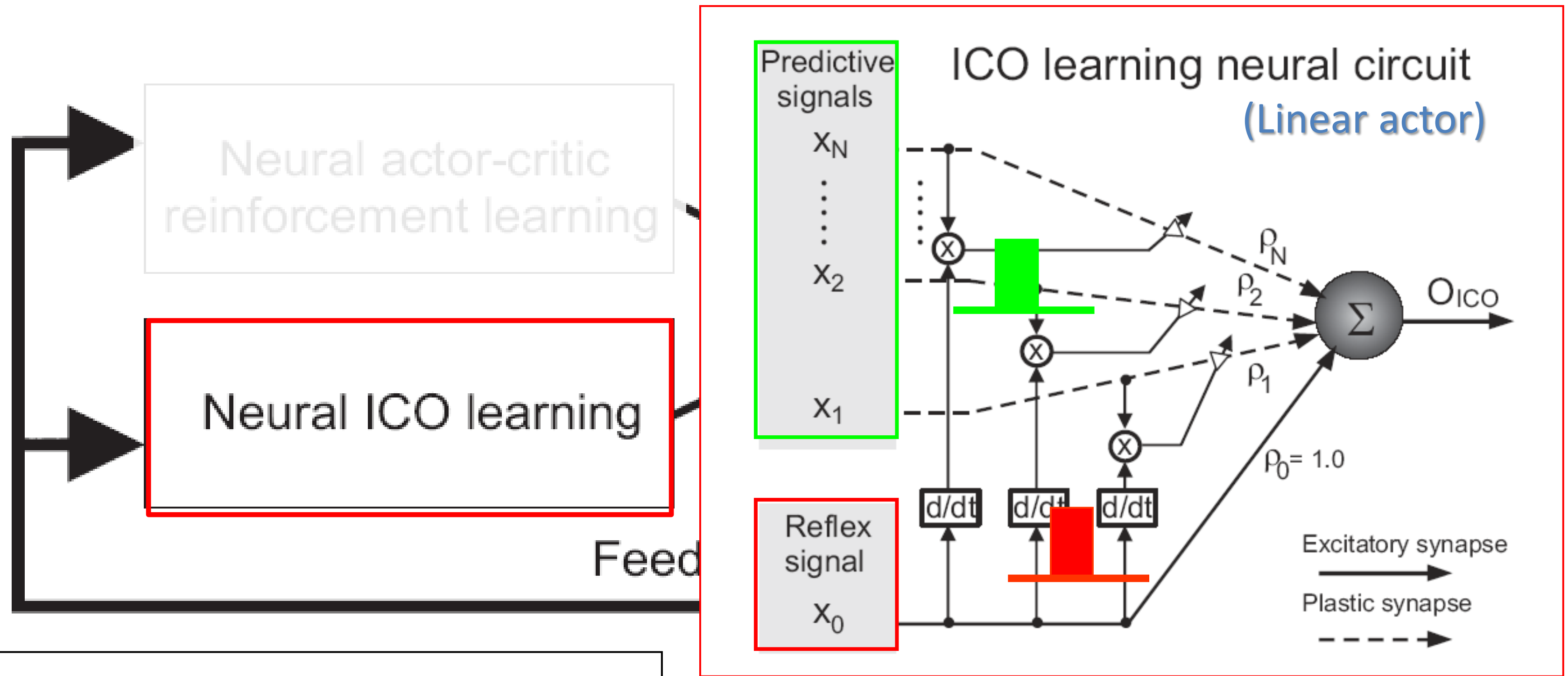


$$O_{ICO}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t)$$

$$\Delta \rho_k = \mu x_k(t) \frac{dx_0(t)}{dt}, \quad k = 1, \ldots, N$$

Bio-inspired Combinatorial Learning

Unsupervised learning

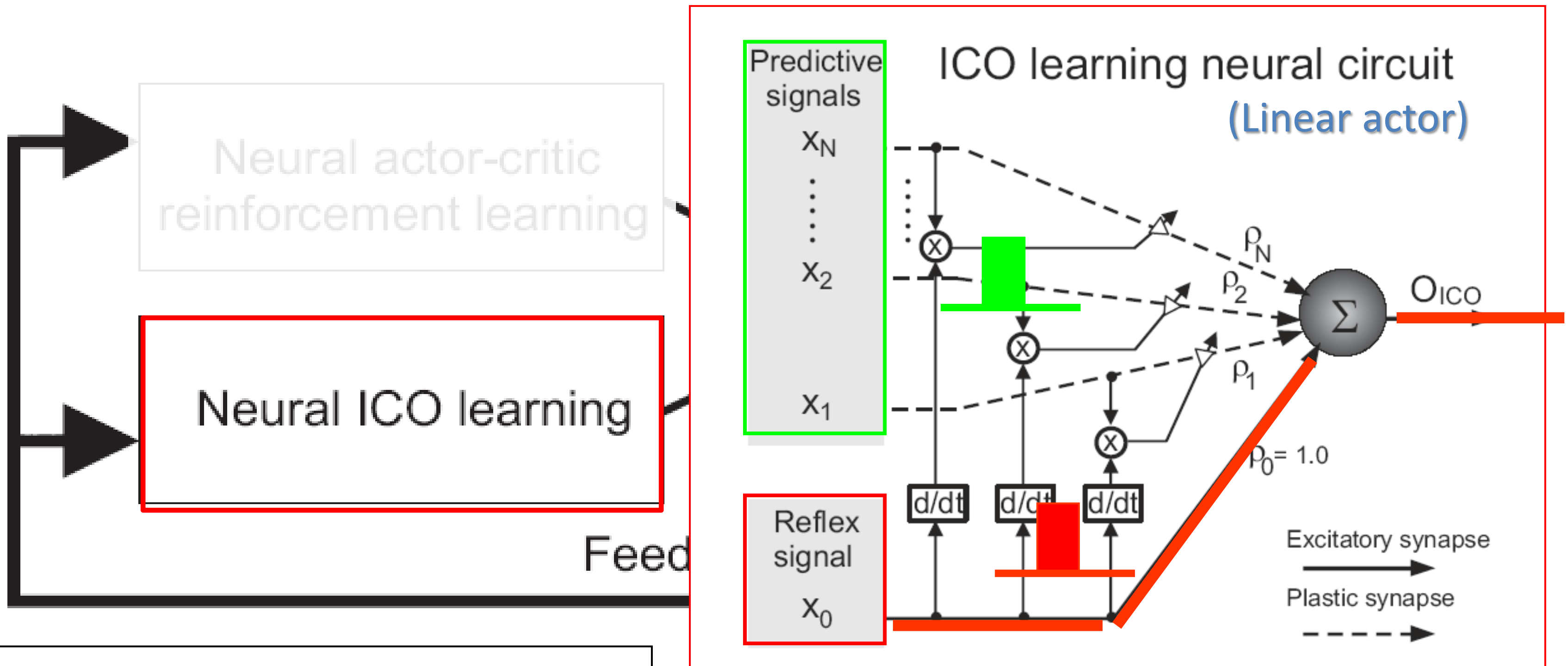


$$O_{ICO}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t)$$

$$\Delta \rho_k = \mu x_k(t) \frac{dx_0(t)}{dt}, \qquad k = 1, \ldots, N$$

Bio-inspired Combinatorial Learning

Unsupervised learning

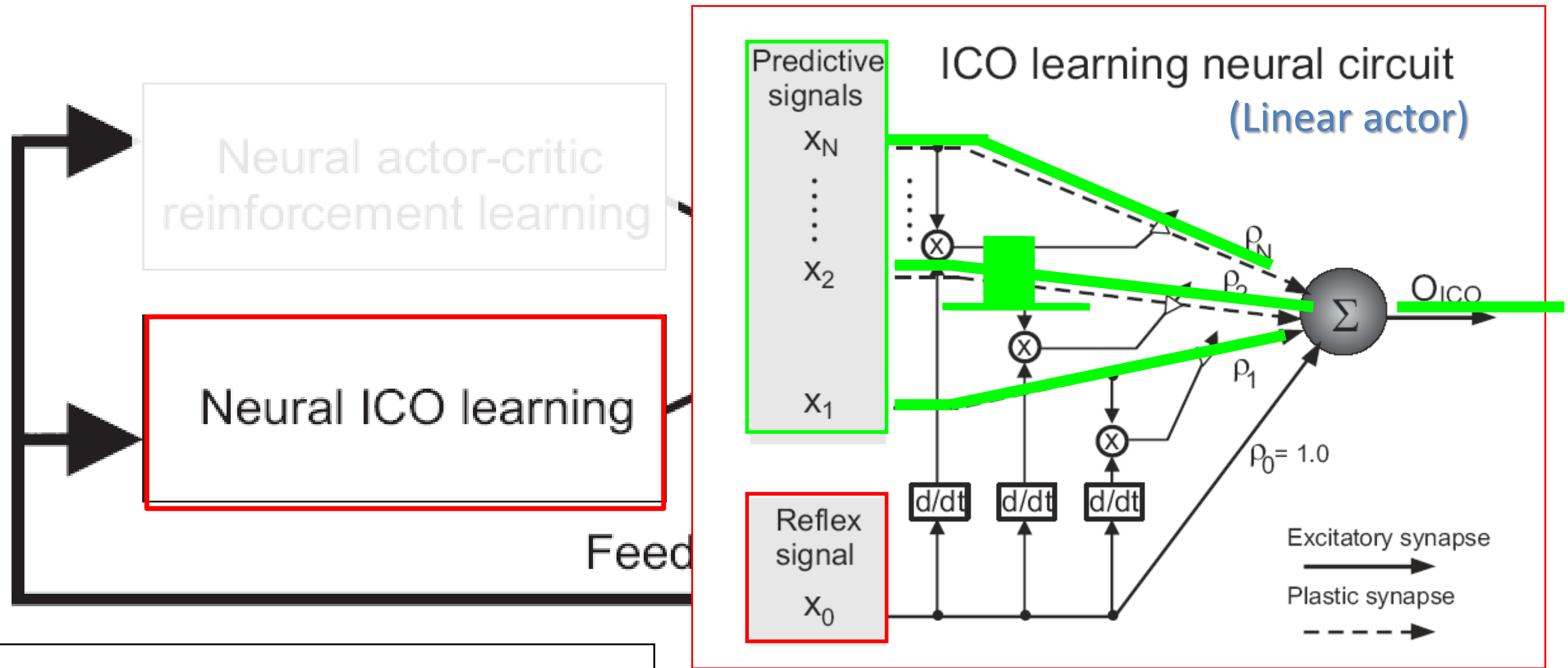


$$O_{ICO}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t)$$

$$\Delta \rho_k = \mu x_k(t) \frac{dx_0(t)}{dt}, \quad k = 1, \ldots, N$$

Bio-inspired Combinatorial Learning

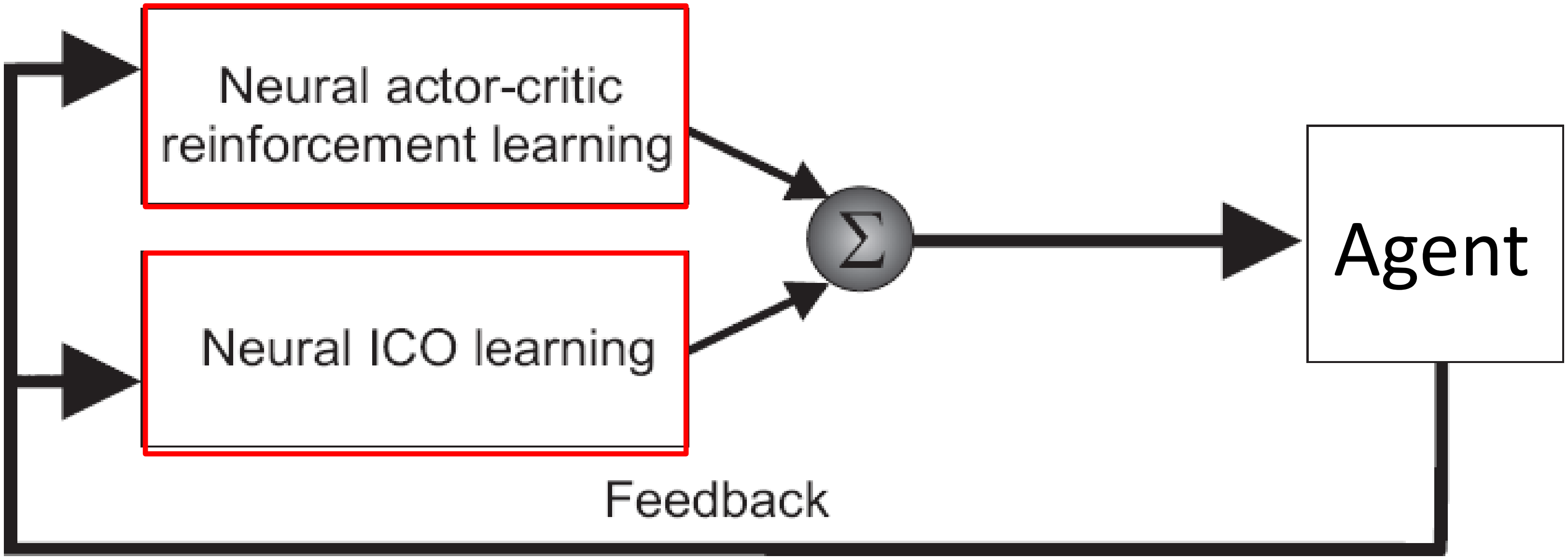
Unsupervised learning



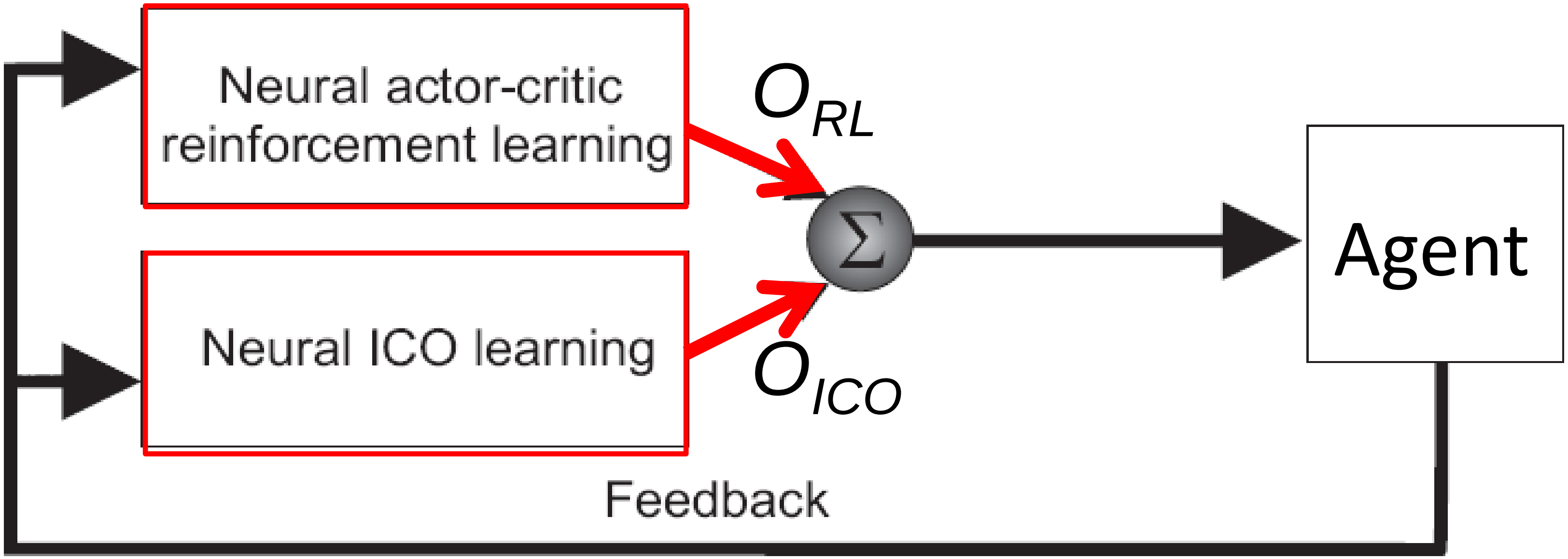
$$O_{ICO}(t) = \rho_0 x_0(t) + \sum_{k=1}^{N} \rho_k(t) x_k(t)$$

$$\Delta \rho_k = \mu x_k(t) \frac{dx_0(t)}{dt}, \quad k = 1, \ldots, N$$

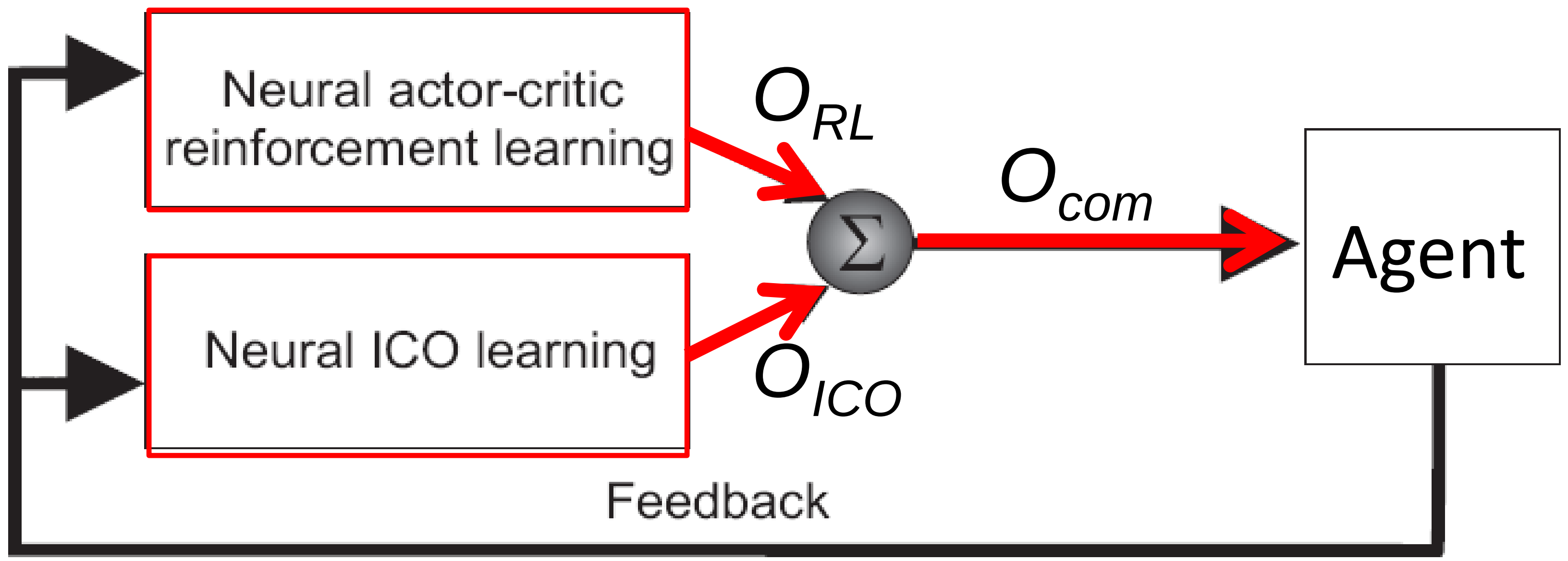
Bio-inspired Combinatorial Learning



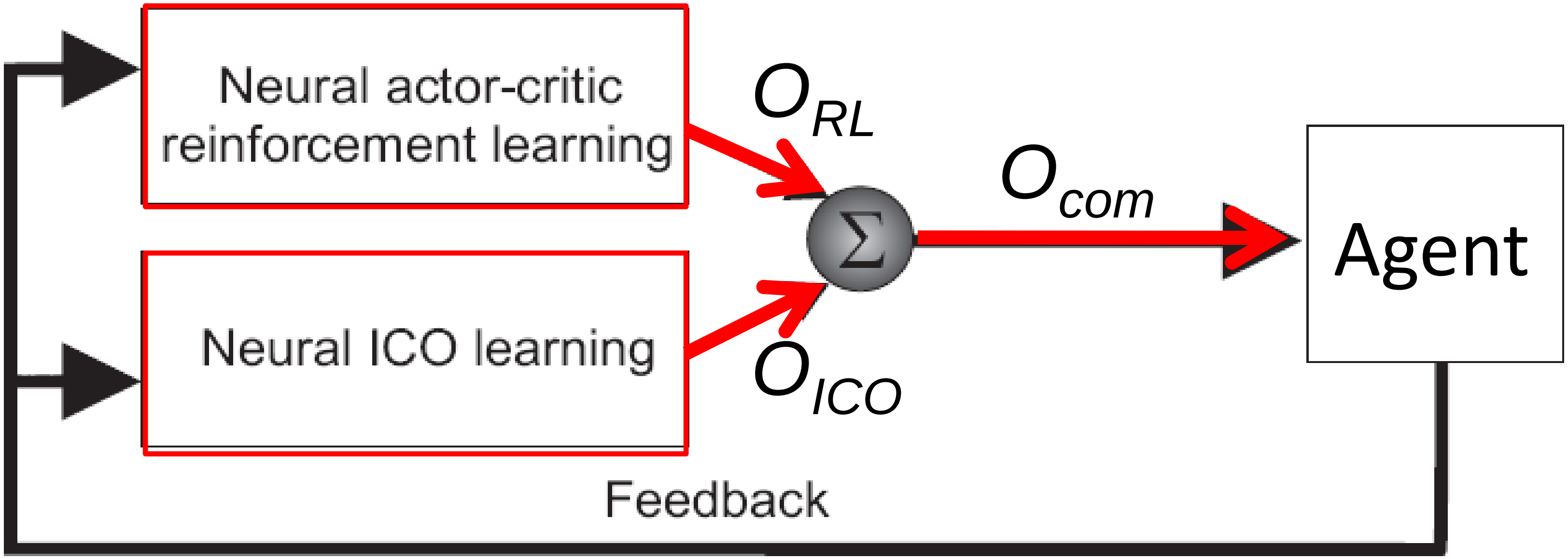
Bio-inspired Combinatorial Learning



Bio-inspired Combinatorial Learning

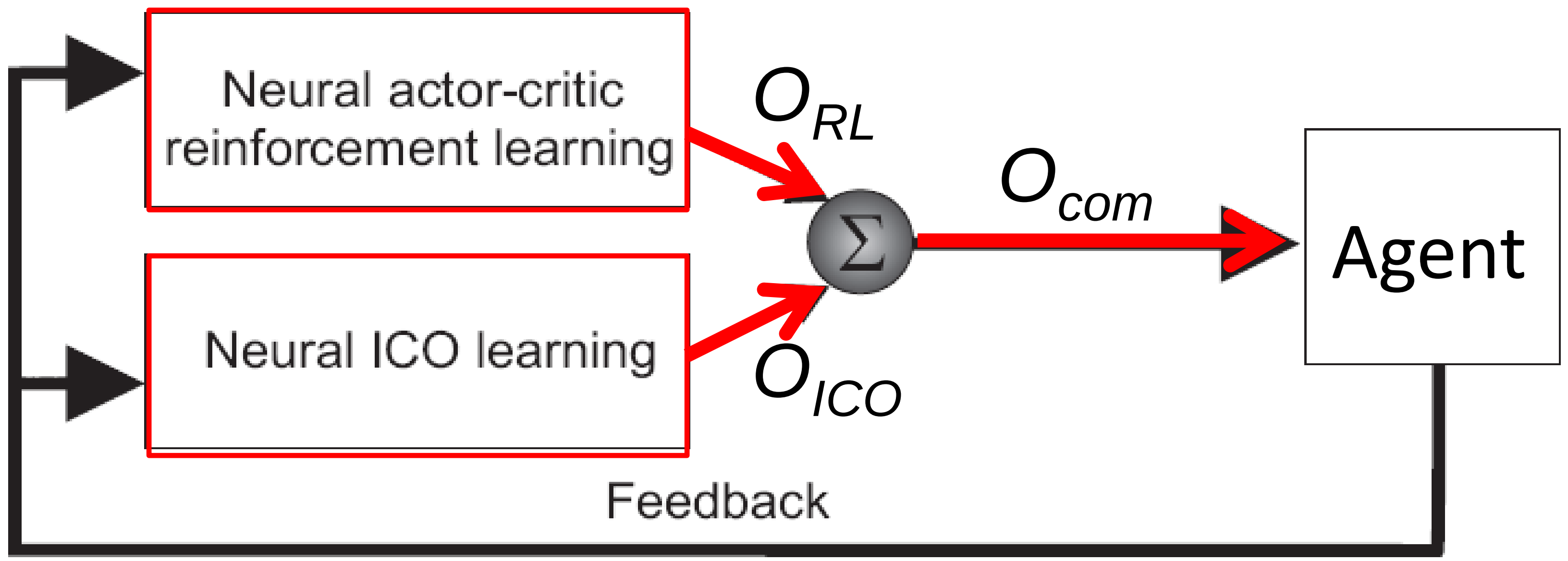


Bio-inspired Combinatorial Learning



$$O_{\mathrm{COM}}(t) = \zeta \cdot (O_{\mathrm{ICO}}(t) + O_{\mathrm{RL}}(t)),$$

Bio-inspired Combinatorial Learning



$$O_{\mathrm{COM}}(t) = \zeta \cdot (O_{\mathrm{ICO}}(t) + O_{\mathrm{RL}}(t)),$$

Scale factor→ constant (e.g., 0.5) or adaptive!

Bio-inspired Combinatorial Learning

The algorithm is:

Bio-inspired Combinatorial Learning

The algorithm is:

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number

ICO weights Actor weights Critic weights

Bio-inspired Combinatorial Learning

The algorithm is:

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number

Repeat:

At time step t

(1) observe reflex signal x_0 and sensory signals x_k which are the state $\mathbf{x}$

(2) compute control output

$$O_{\text{ICO}} \leftarrow \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k$$

$$O_{\text{RL}} \leftarrow \varepsilon + \sum_{k=1}^{N} w_k x_k$$

$$O_{\text{COM}} \leftarrow \zeta \cdot (O_{\text{ICO}} + O_{\text{RL}})$$

Bio-inspired Combinatorial Learning

The algorithm is:

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number

Repeat:

At time step t

(1) observe reflex signal x_0 and sensory signals x_k which are the state $\mathbf{x}$

(2) compute control output

ICO $$O_{\text{ICO}} \leftarrow \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k$$

Actor (RL) $$O_{\text{RL}} \leftarrow \varepsilon + \sum_{k=1}^{N} w_k x_k$$

Combine (ICO + RL) $$O_{\text{COM}} \leftarrow \zeta \cdot (O_{\text{ICO}} + O_{\text{RL}})$$

Bio-inspired Combinatorial Learning

The algorithm is:

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number

Repeat:

At time step t

(1) observe reflex signal x_0 and sensory signals x_k which are the state $\mathbf{x}$

(2) compute control output

ICO $$O_{\mathrm{ICO}} \leftarrow \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k$$

Actor (RL) $$O_{\mathrm{RL}} \leftarrow \varepsilon + \sum_{k=1}^{N} w_k x_k$$

Combine (ICO + RL) $$O_{\mathrm{COM}} \leftarrow \zeta \cdot (O_{\mathrm{ICO}} + O_{\mathrm{RL}})$$

(3) perform action $\leftarrow O_{com}$

Bio-inspired Combinatorial Learning

The algorithm is:

Initialize ρ_k, w_k, and v_j to 0.0; ε = Gaussian random number

Repeat:

At time step t

(1) observe reflex signal x_0 and sensory signals x_k which are the state $\mathbf{x}$

(2) compute control output

ICO $$O_{\mathrm{ICO}} \leftarrow \rho_0 x_0 + \sum_{k=1}^{N} \rho_k x_k$$

Actor (RL) $$O_{\mathrm{RL}} \leftarrow \varepsilon + \sum_{k=1}^{N} w_k x_k$$

Combine (ICO + RL) $$O_{\mathrm{COM}} \leftarrow \zeta \cdot (O_{\mathrm{ICO}} + O_{\mathrm{RL}})$$

(3) perform action **← O_{com}**

(4) observe reward R, new state $\mathbf{x}'$ and new reflex signal x'_0

Bio-inspired Combinatorial Learning

(5) obtain value function by computing **Critic (RL)**

Bio-inspired Combinatorial Learning

(5) obtain value function by computing

Critic (RL)

Inputs (new state)

$$a_j \leftarrow e^{-\|\mathbf{s}_j^T(\mathbf{x}-\mathbf{c}_j)\|^2}$$

$$y_j \leftarrow \frac{a_j}{\sum_{l=1}^{M} a_l}$$

$$V \leftarrow \sum_{j=1}^{M} v_j y_j$$

RBF network as
a value function approximator

Bio-inspired Combinatorial Learning

(5) obtain value function by computing **Critic (RL)**

Inputs (new state)

$$a_j \leftarrow e^{-\|\mathbf{s}_j^T(\mathbf{x}'-\mathbf{c}_j)\|^2}$$

$$y_j \leftarrow \frac{a_j}{\sum_{l=1}^{M} a_l}$$

RBF network as
a value function approximator

$$V \leftarrow \sum_{j=1}^{M} v_j y_j$$

(6) compute $\varepsilon \leftarrow \xi\sigma \cdot \min\left[1, \max\left[0, \frac{V_{\max}-V(\mathbf{x}')}{V_{\max}-V_{\min}}\right]\right]$ **Exploration**

Bio-inspired Combinatorial Learning

(5) obtain value function by computing **Critic (RL)**

Inputs (new state)

$$a_j \leftarrow e^{-\|\mathbf{s}_j^T(\mathbf{x}'-\mathbf{c}_j)\|^2}$$

$$y_j \leftarrow \frac{a_j}{\sum_{l=1}^{M} a_l}$$

RBF network as
a value function approximator

$$V \leftarrow \sum_{j=1}^{M} v_j y_j$$

(6) compute $\varepsilon \leftarrow \xi\sigma \cdot \min\left[1, \max\left[0, \frac{V_{\max}-V(\mathbf{x}')}{V_{\max}-V_{\min}}\right]\right]$ **Exploration**

(7) compute $\delta \leftarrow R + \gamma V(\mathbf{x}') - V(\mathbf{x})$ **TD error**

Previous state

Bio-inspired Combinatorial Learning

(5) obtain value function by computing **Critic (RL)**

Inputs (new state)

$$a_j \leftarrow e^{-\|\mathbf{s}_j^T(\mathbf{x}'-\mathbf{c}_j)\|^2}$$

RBF network as a value function approximator

$$y_j \leftarrow \frac{a_j}{\sum_{l=1}^{M} a_l}$$

$$V \leftarrow \sum_{j=1}^{M} v_j y_j$$

(6) compute $\varepsilon \leftarrow \xi\sigma \cdot \min\left[1, \max\left[0, \frac{V_{\max} - V(\mathbf{x}')}{V_{\max} - V_{\min}}\right]\right]$ **Exploration**

(7) compute $\delta \leftarrow R + \gamma V(\mathbf{x}') - V(\mathbf{x})$ **TD error**

(8) update control parameters

Previous state

ICO weights $\rho_k \leftarrow \rho_k + \mu x'_k (x'_0 - x_0)$

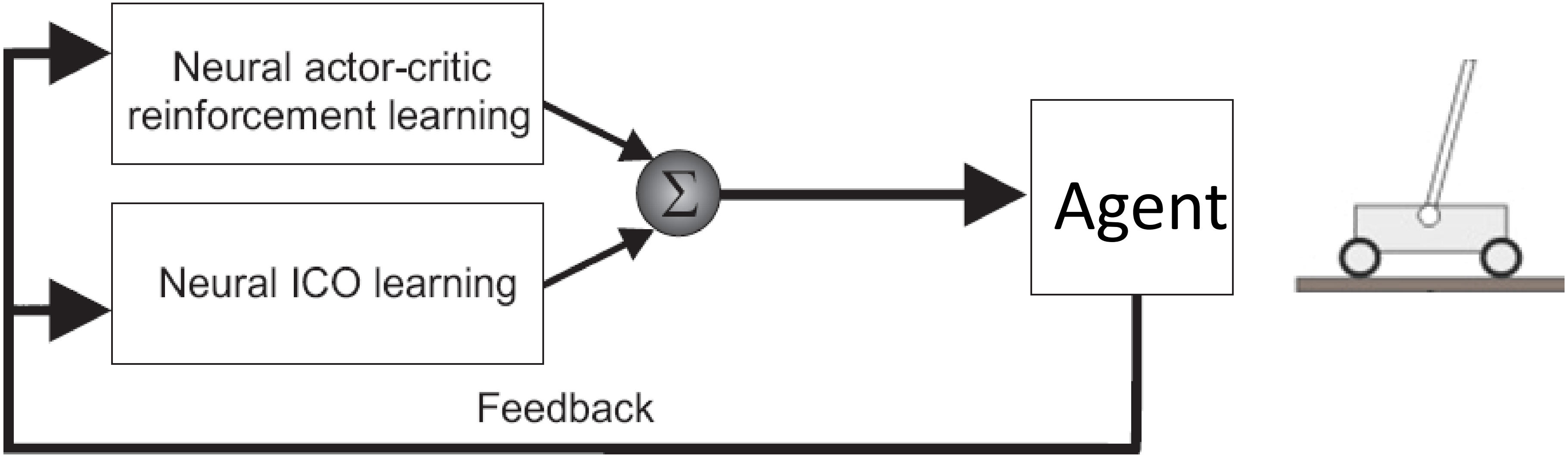
Actor weights $w_k \leftarrow w_k + \alpha\delta x'_k \varepsilon$

Critic weights $v_j \leftarrow v_j + \lambda\delta y_j$

Until: Successful control policy is found or the maximum number of trials is reached.

Bio-inspired Combinatorial Learning

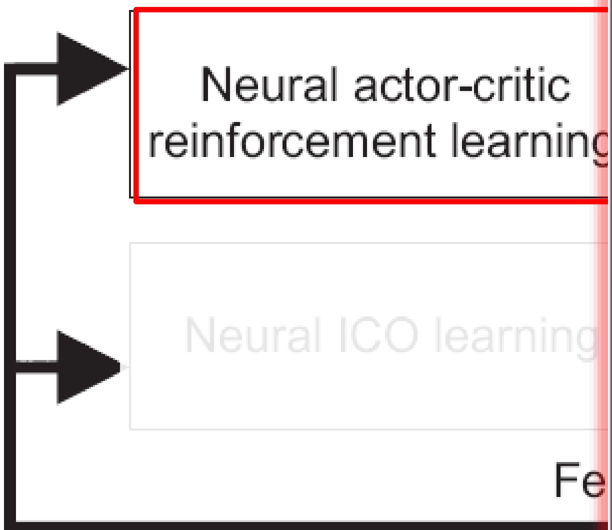
- **Example1:** Dynamical Control Problem (Pole Balancing)

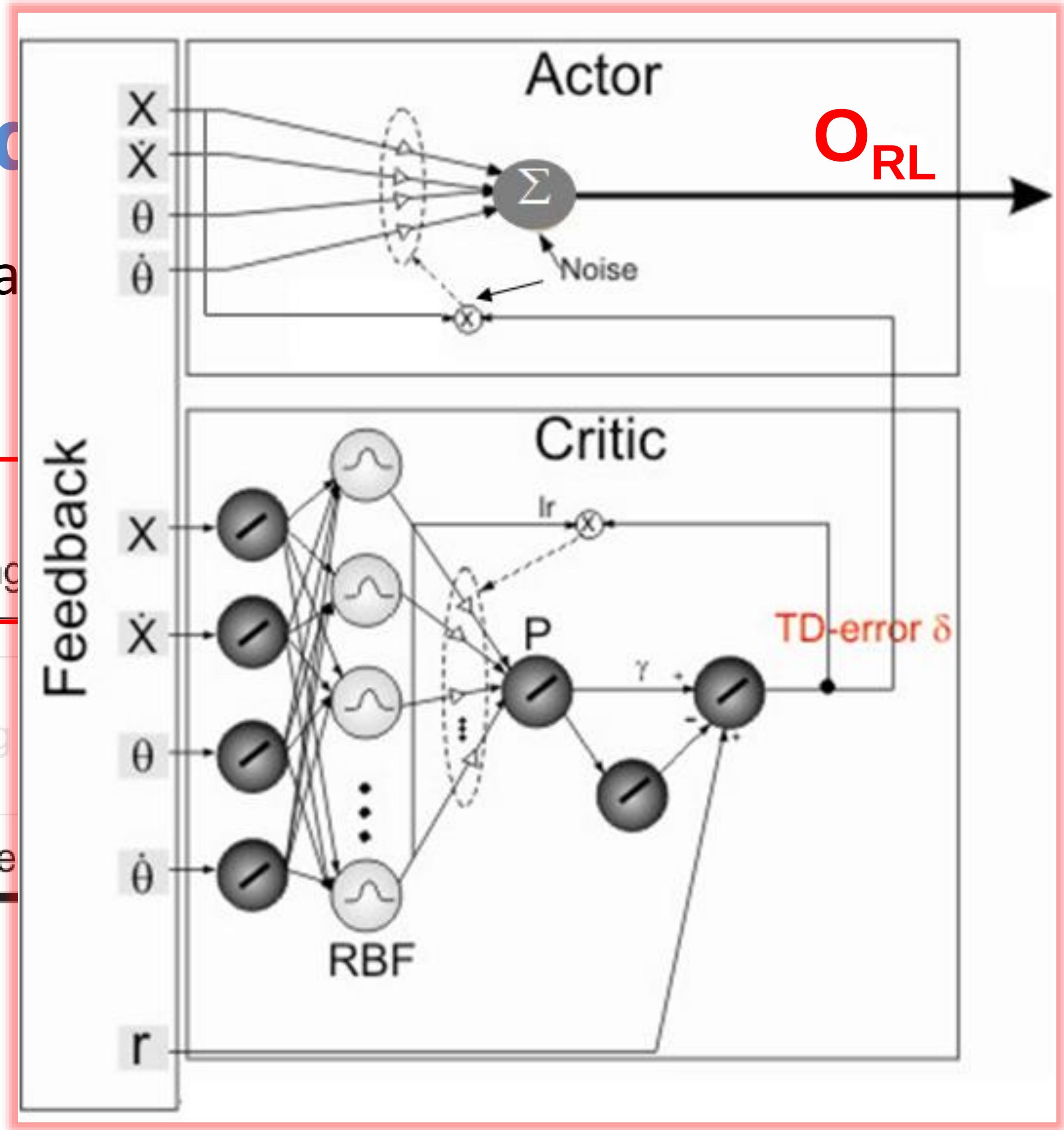


Manoonpong et al., 2013

Bio-inspirec

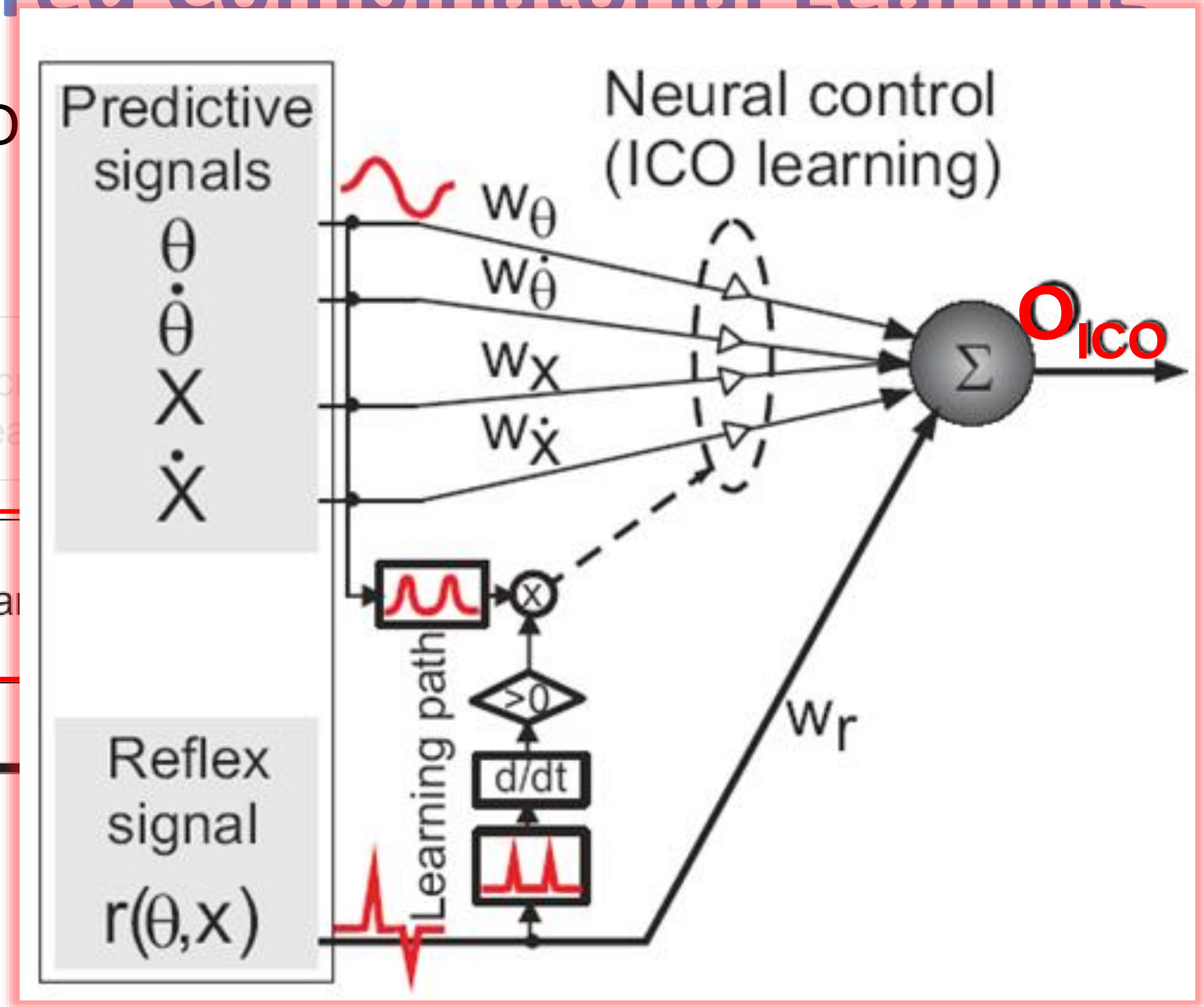
- **Example1:** Dyna





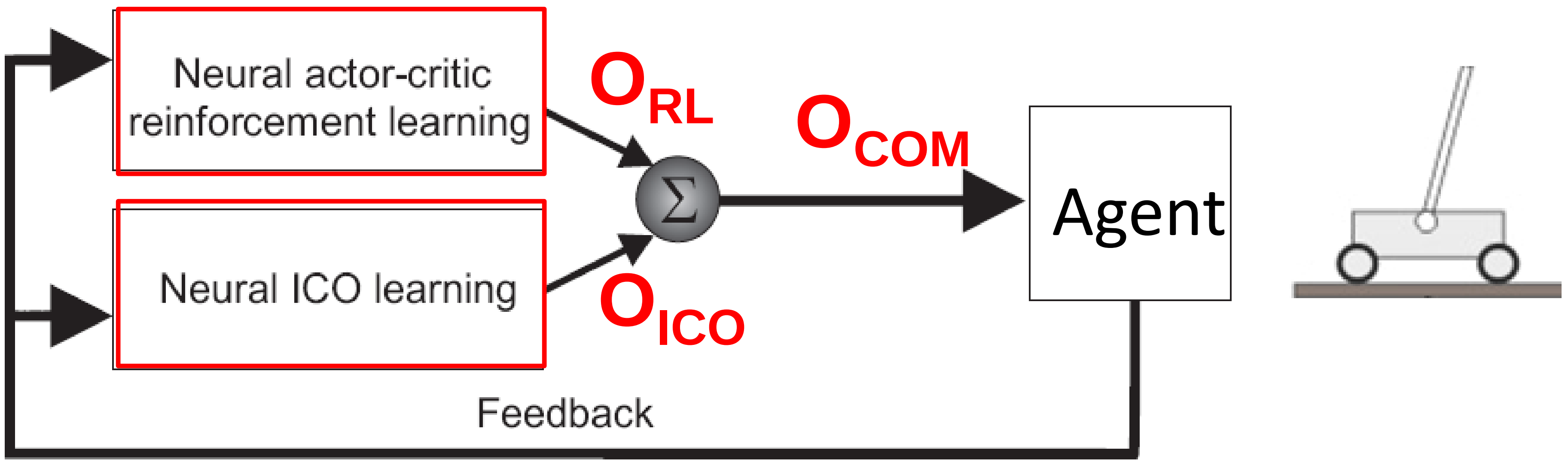
Bio-inspired Combinatorial Learning

- **Example1:** D



Bio-inspired Combinatorial Learning

- **Example1:** Dynamical Control Problem (Pole Balancing)

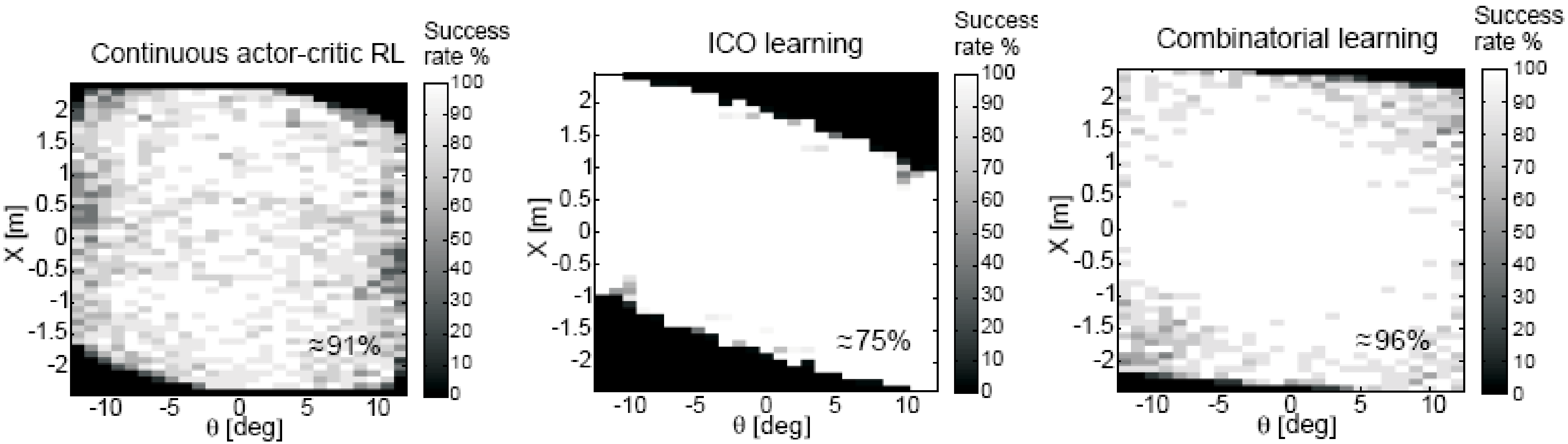


$$O_{\mathrm{COM}}(t) = \zeta \cdot (O_{\mathrm{ICO}}(t) + O_{\mathrm{RL}}(t)),$$

0.5

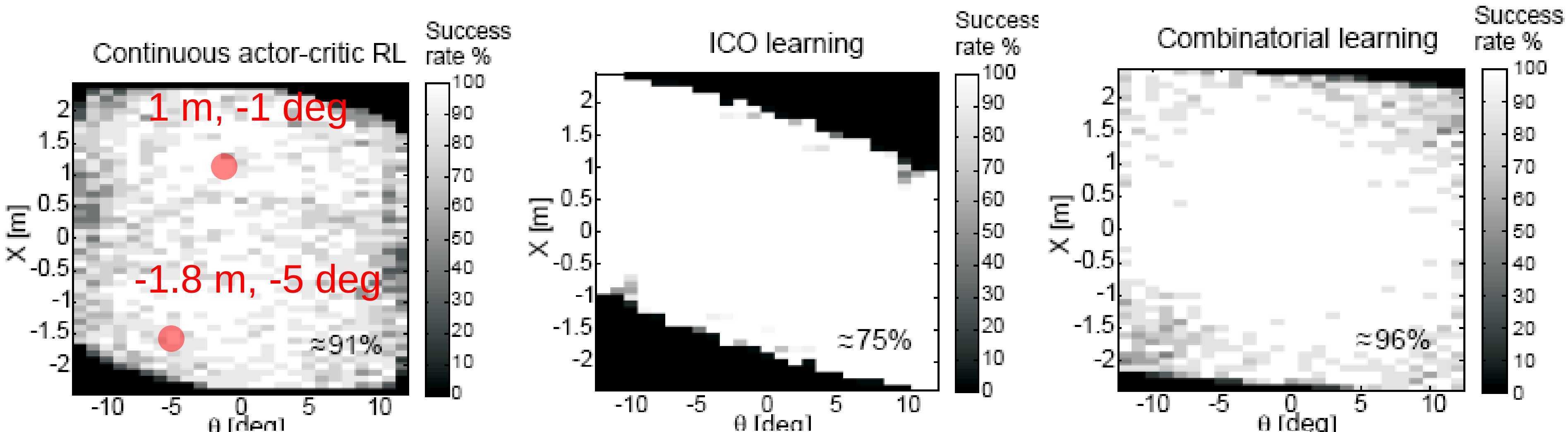
Bio-inspired Combinatorial Learning

- **Example1:** Dynamical Control Problem (Pole Balancing)

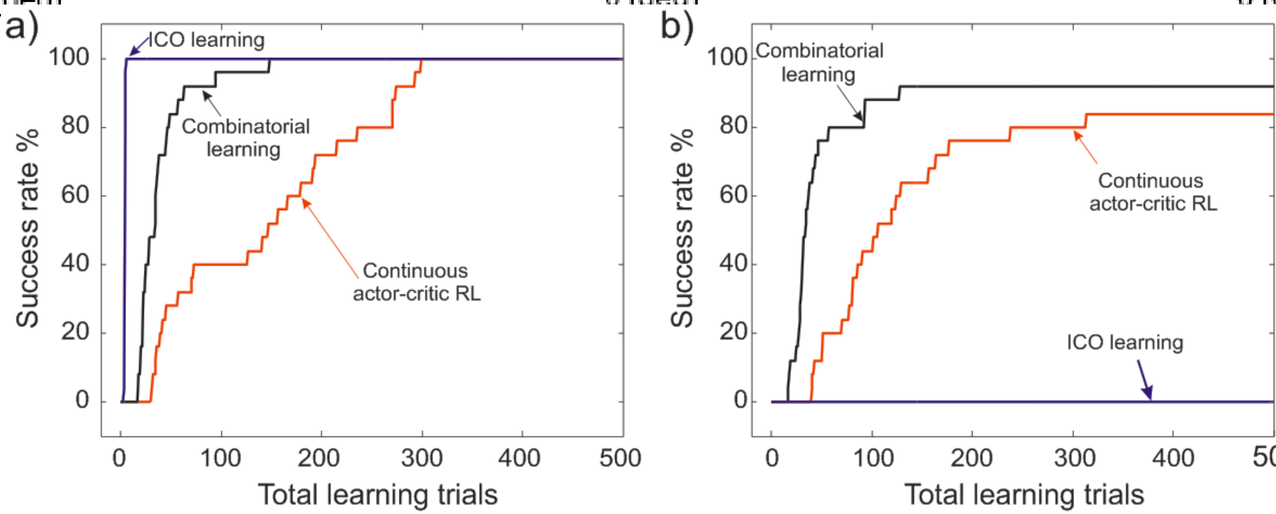


Bio-inspired Combinatorial Learning

- **Example1:** Dynamical Control Problem (Pole Balancing)



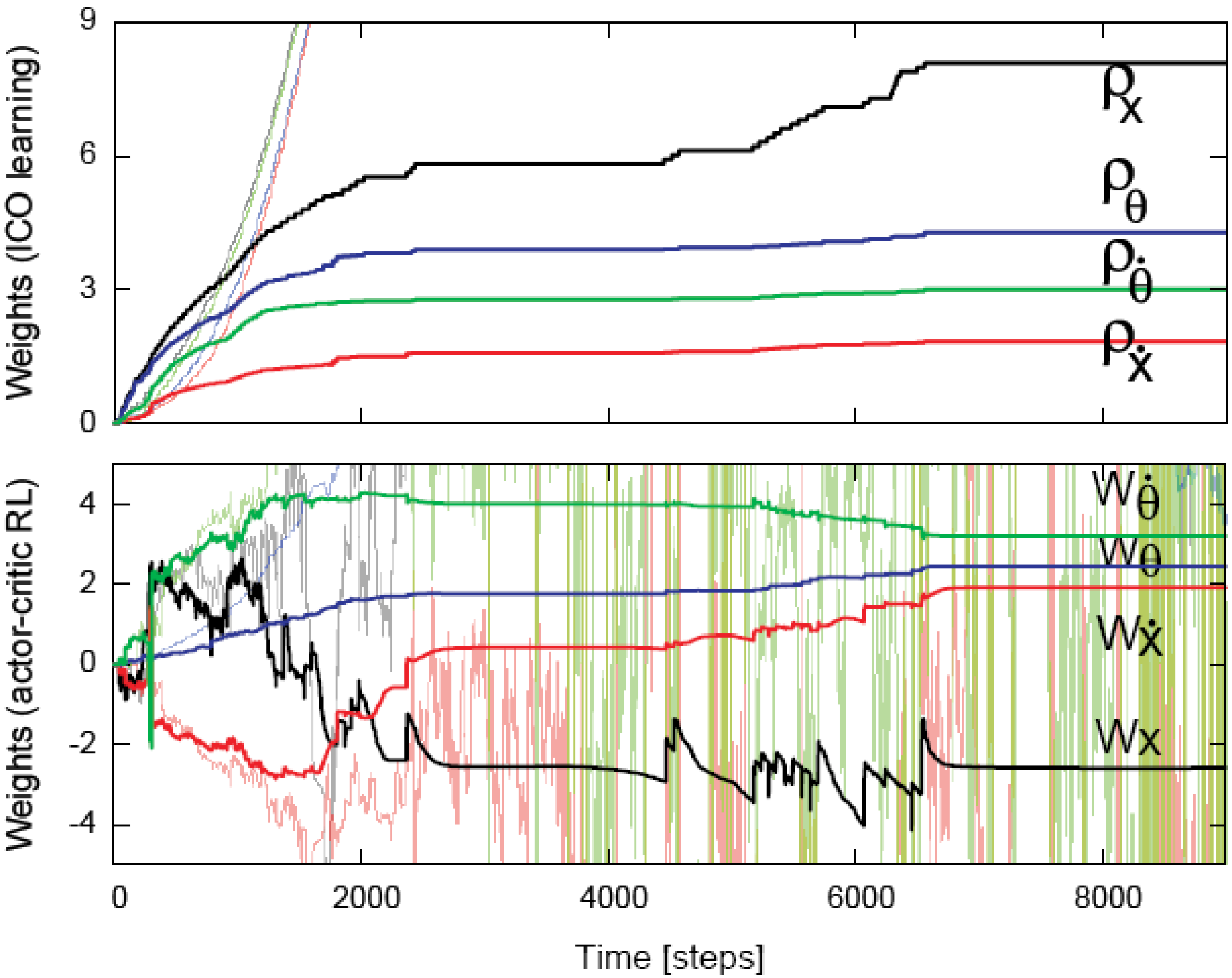
1 m, -1 deg

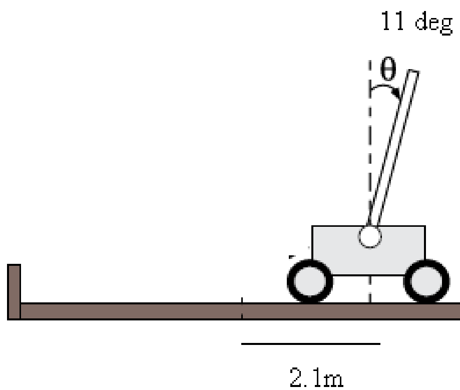


-1.8 m, -5 deg

Bio-inspired Combinatorial Learning

- **Example1:** Dynamical Control Problem (Pole Balancing)

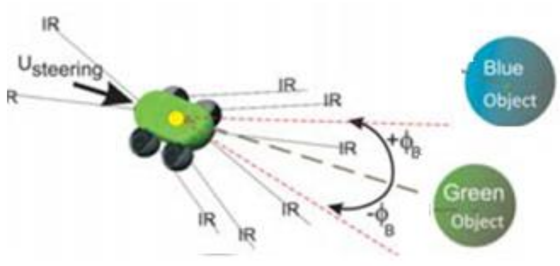


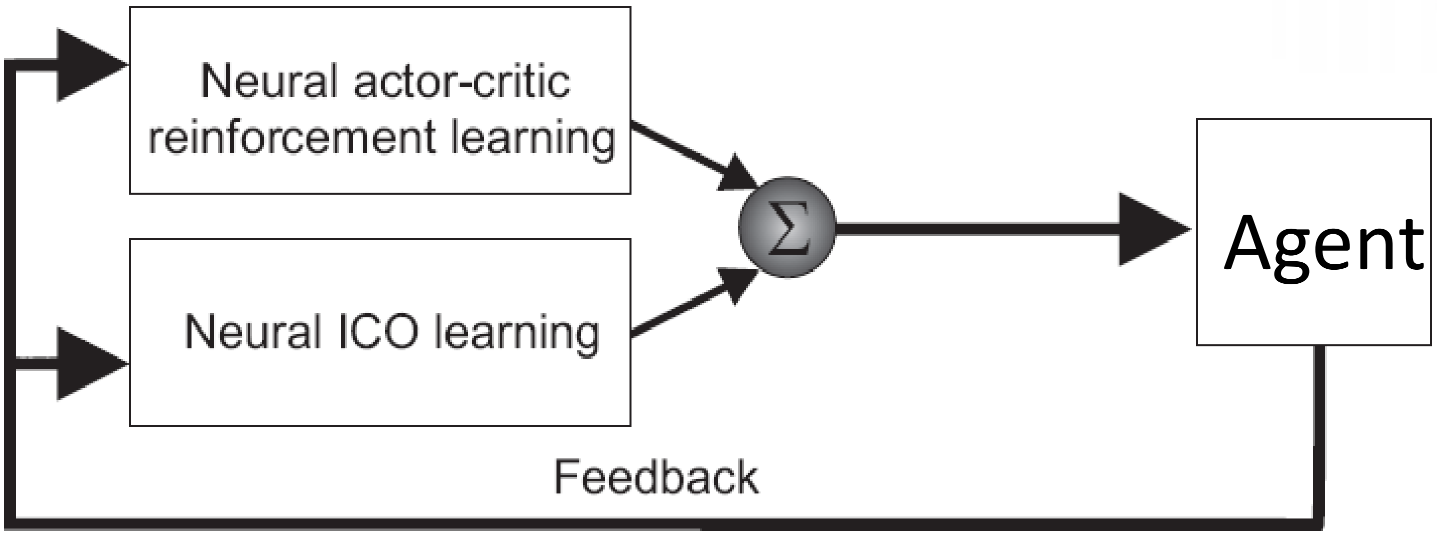


x = 2.1m, θ = 11deg

Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

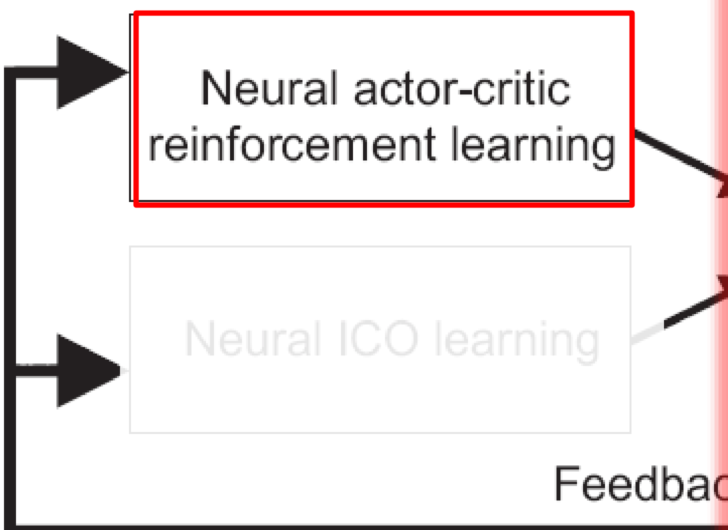


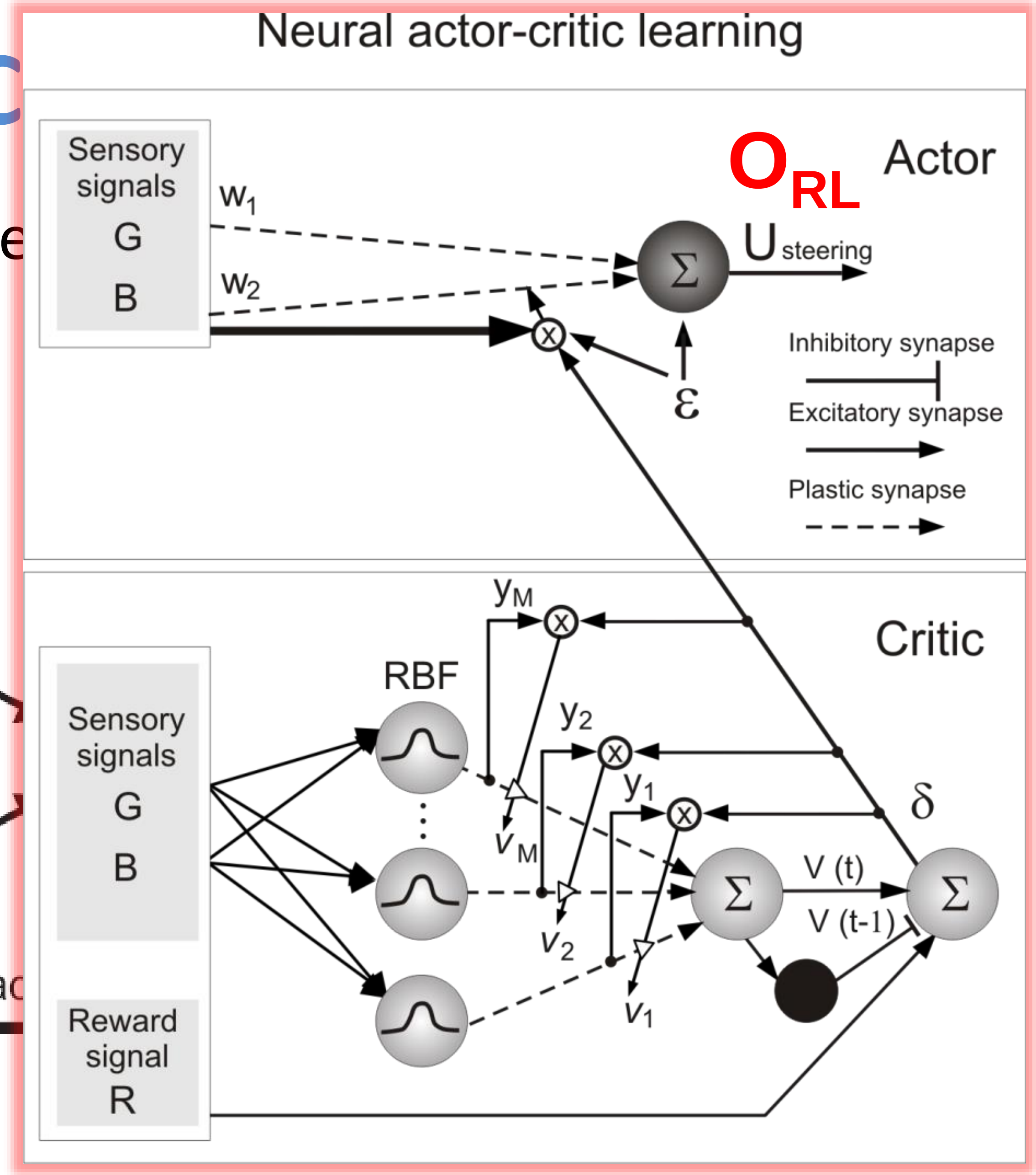


Manoonpong et al., 2013

Bio-inspired C

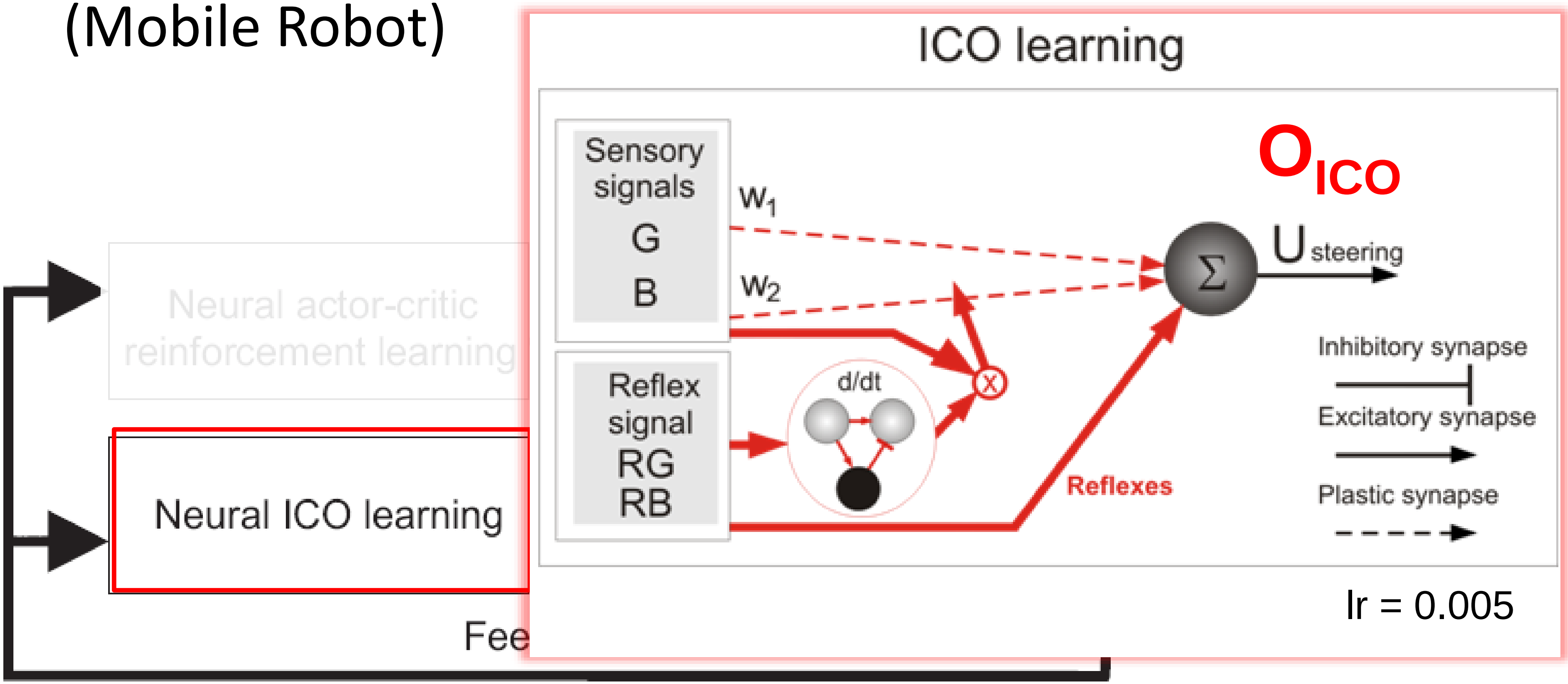
- **Example2:** Goal-dire
(Mobile Robot)





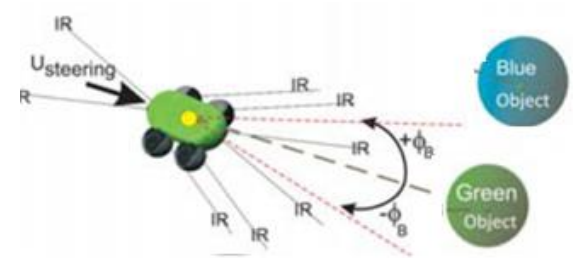
Bio-inspired Combinatorial Learning

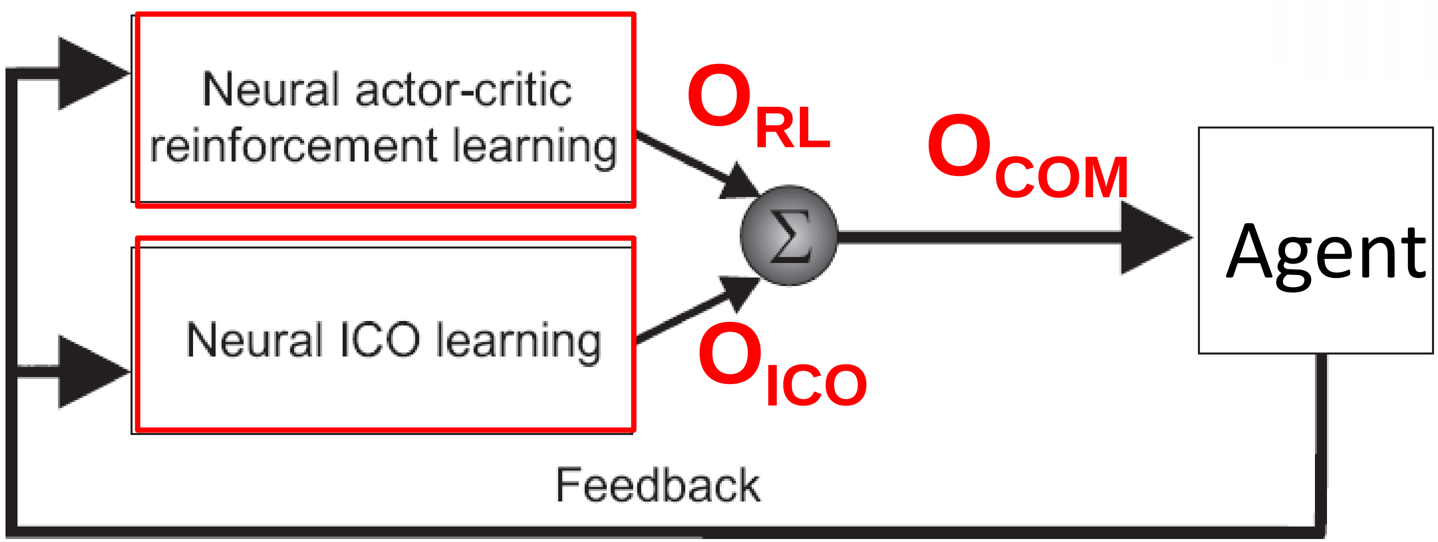
- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)



Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)



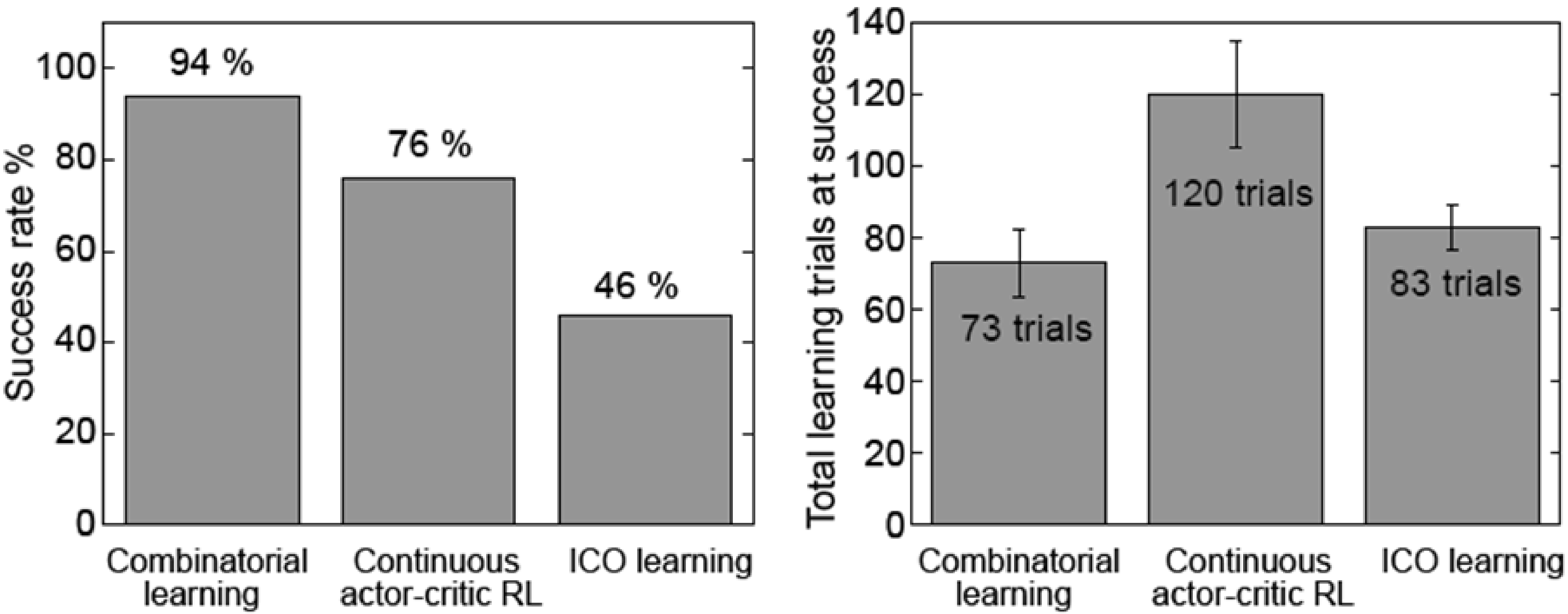


$$O_{\mathrm{COM}}(t) = \zeta \cdot (O_{\mathrm{ICO}}(t) + O_{\mathrm{RL}}(t)),$$

0.5

Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

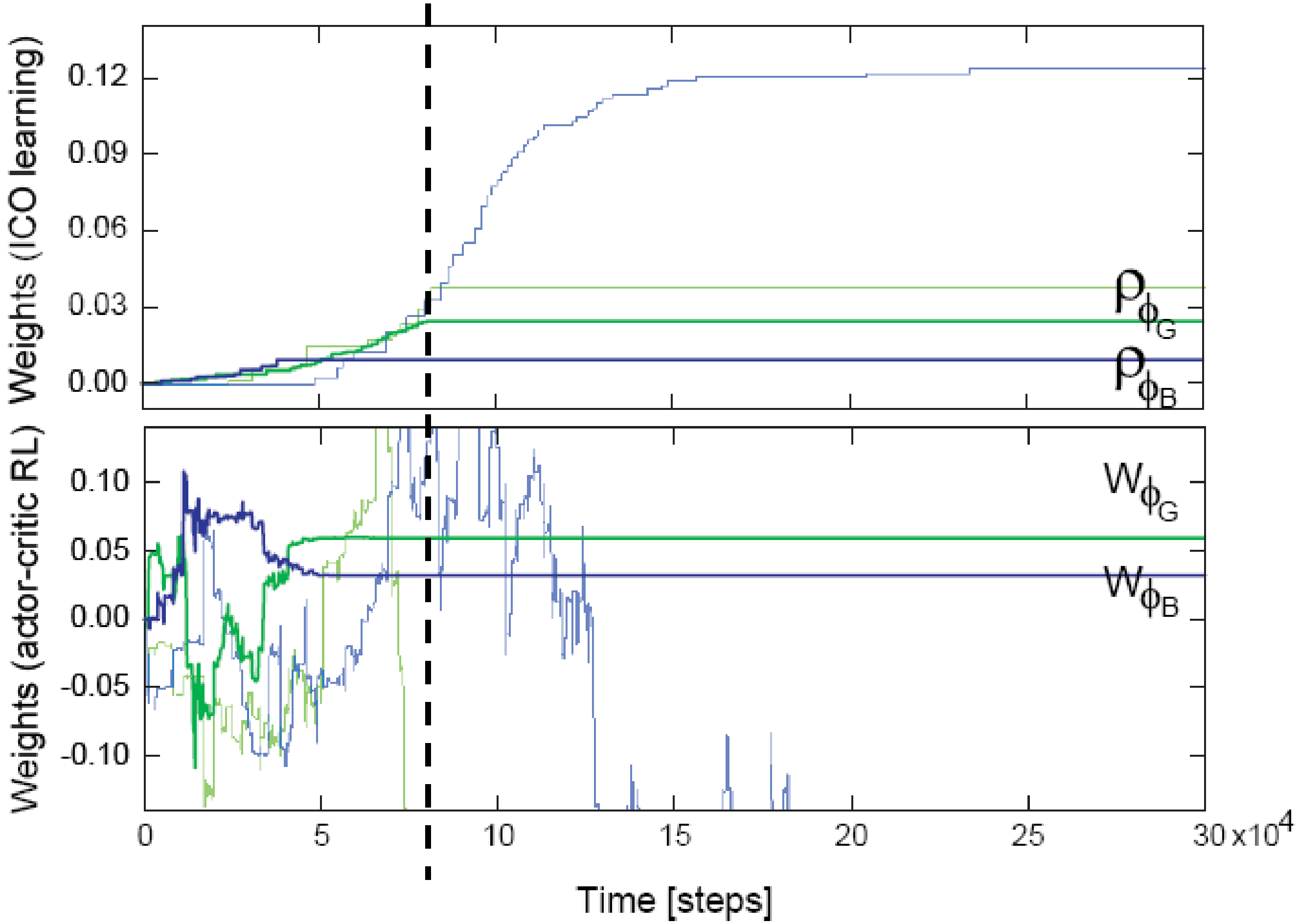


50 Experiments

Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

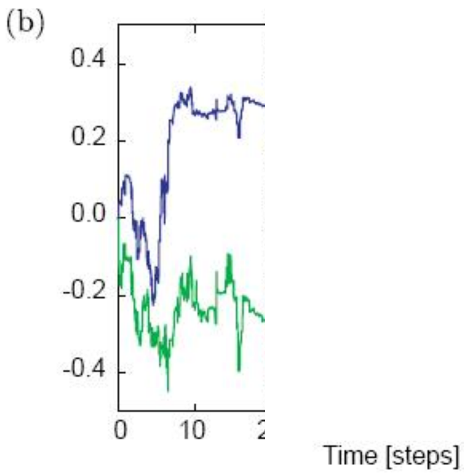
~7.5x10^4 time steps or ~50 trials (learning stops)

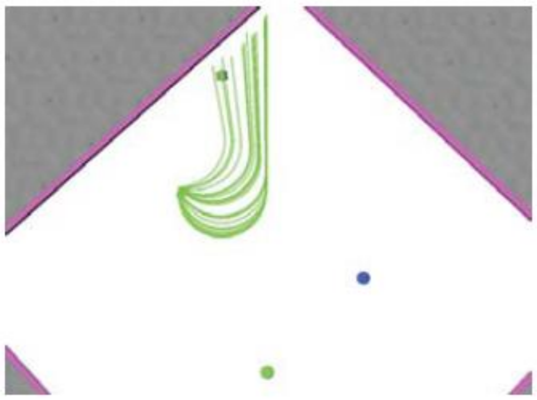


Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

AC-RL



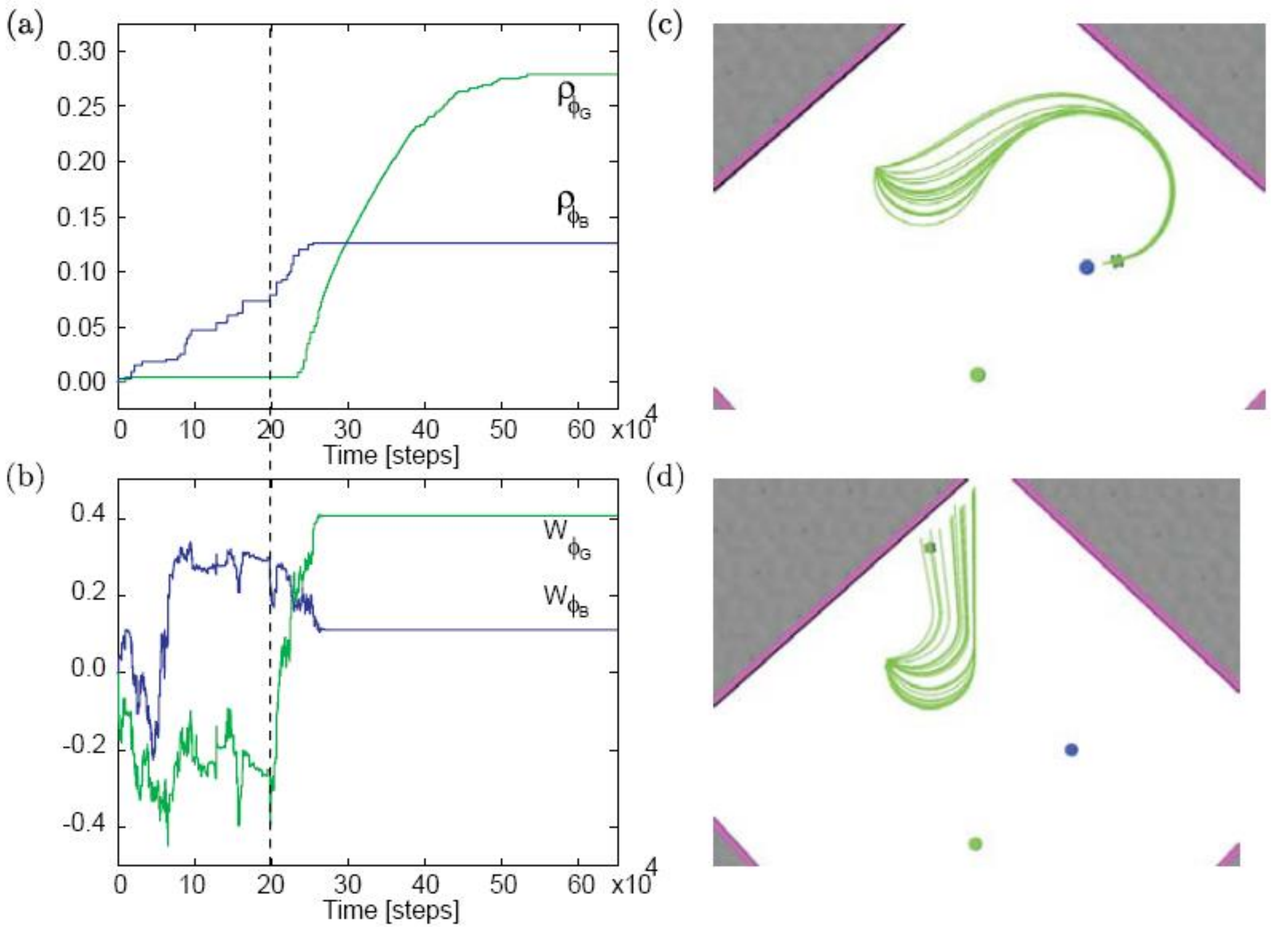


Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)

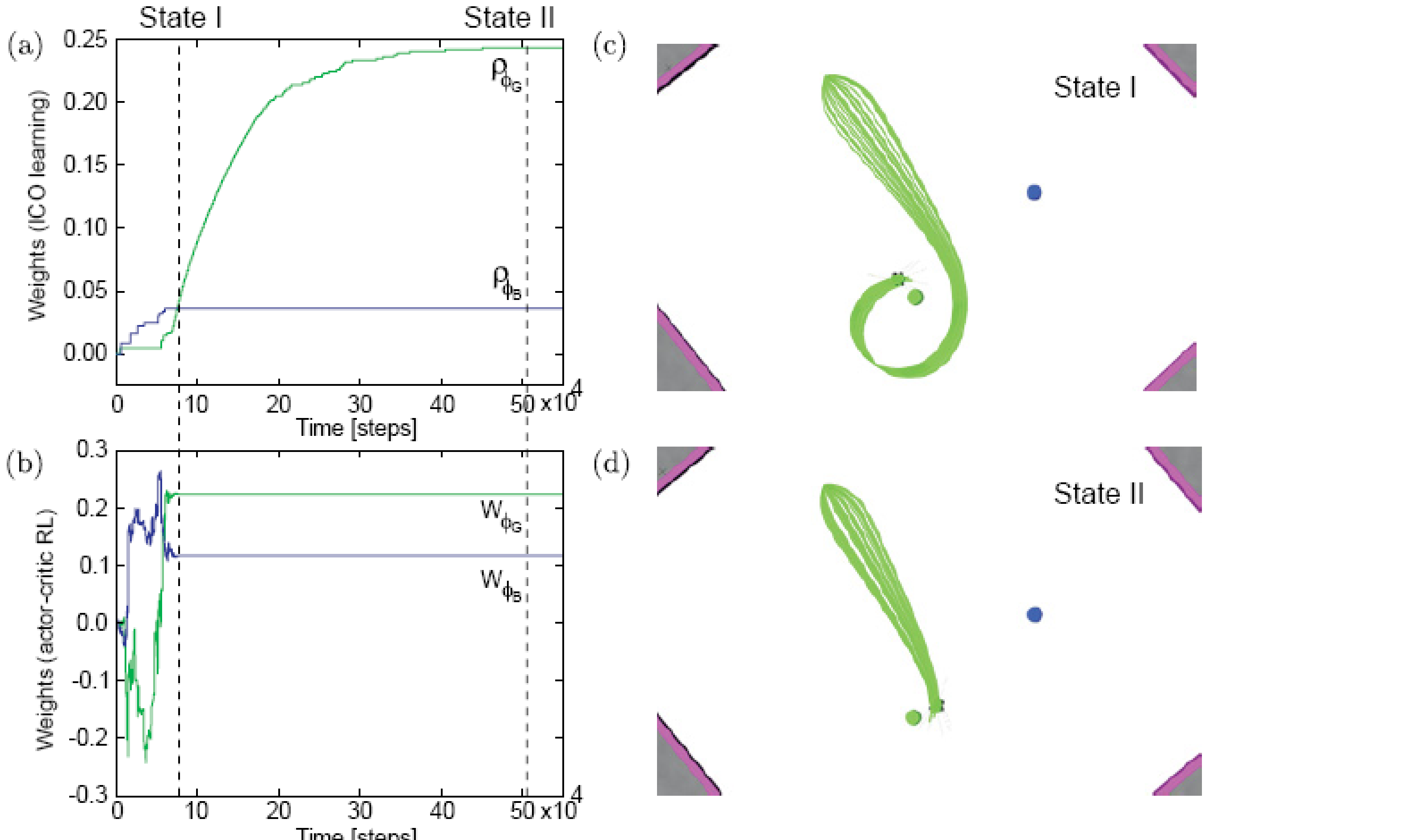
ICO

AC-RL



Bio-inspired Combinatorial Learning

- **Example2:** Goal-directed Behavior Control Problem: (Mobile Robot)



Learning alone !

Learning alone !

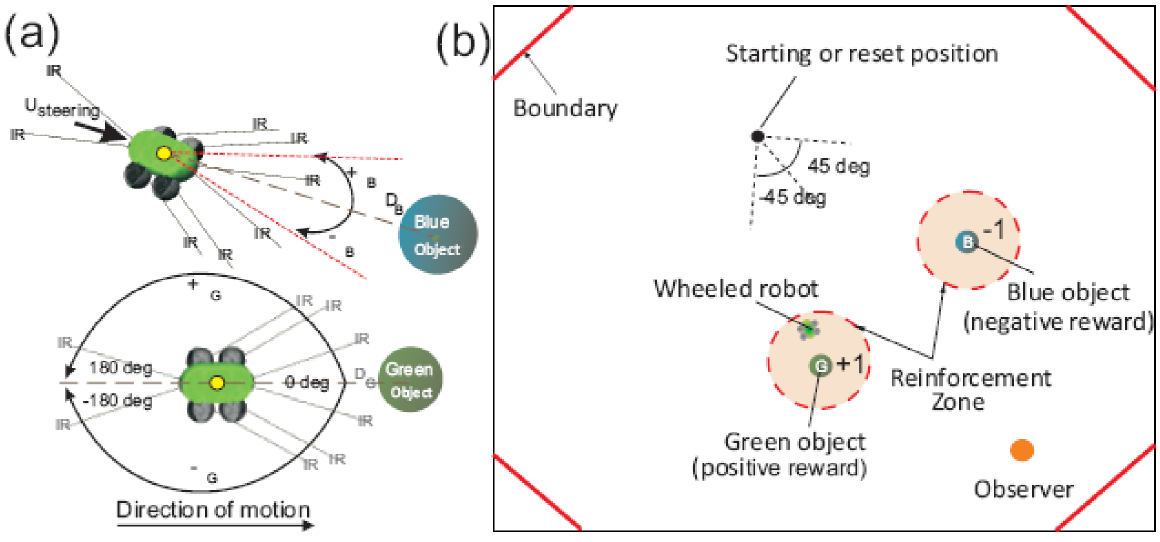
Combinatorial learning!

Learning alone !

Combinatorial learning!

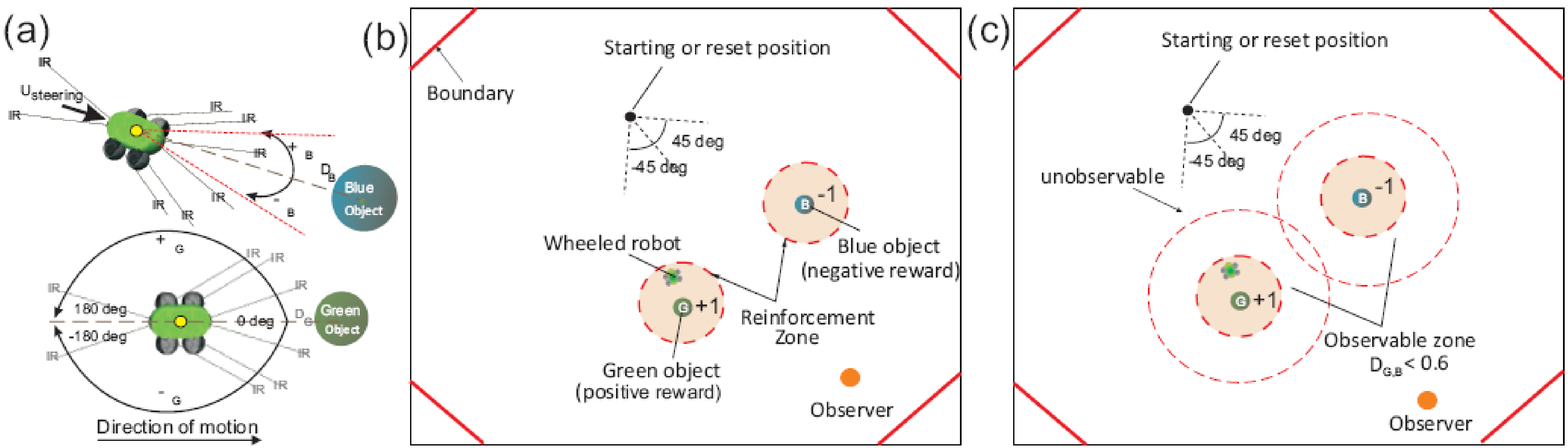
Bio-inspired Combinatorial Learning

So far the mobile robot can observe all states at all time !→ “fully observable case”



Bio-inspired Combinatorial Learning

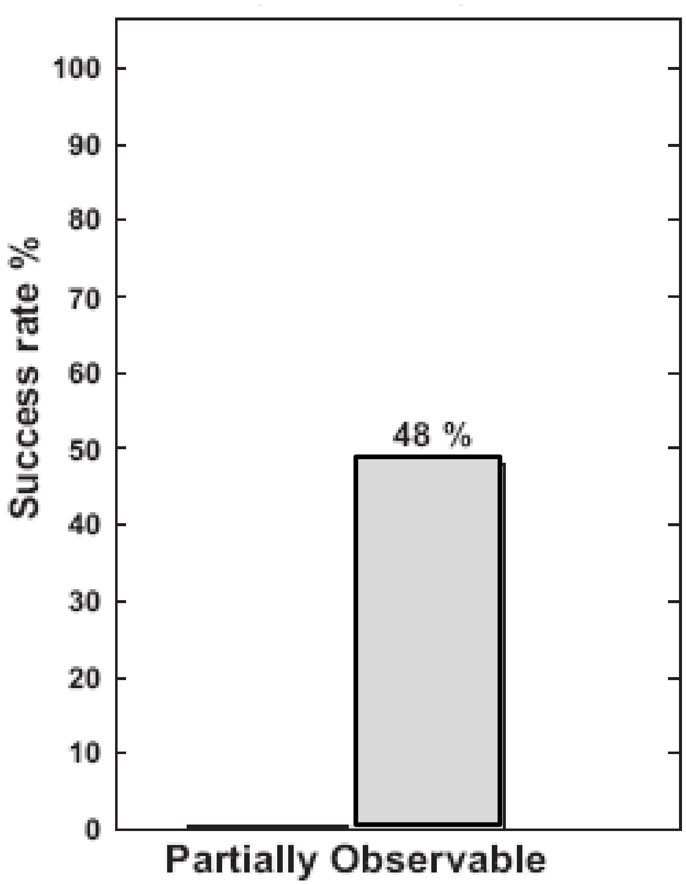
So far the mobile robot can observe all states at all time !→ “fully observable case”

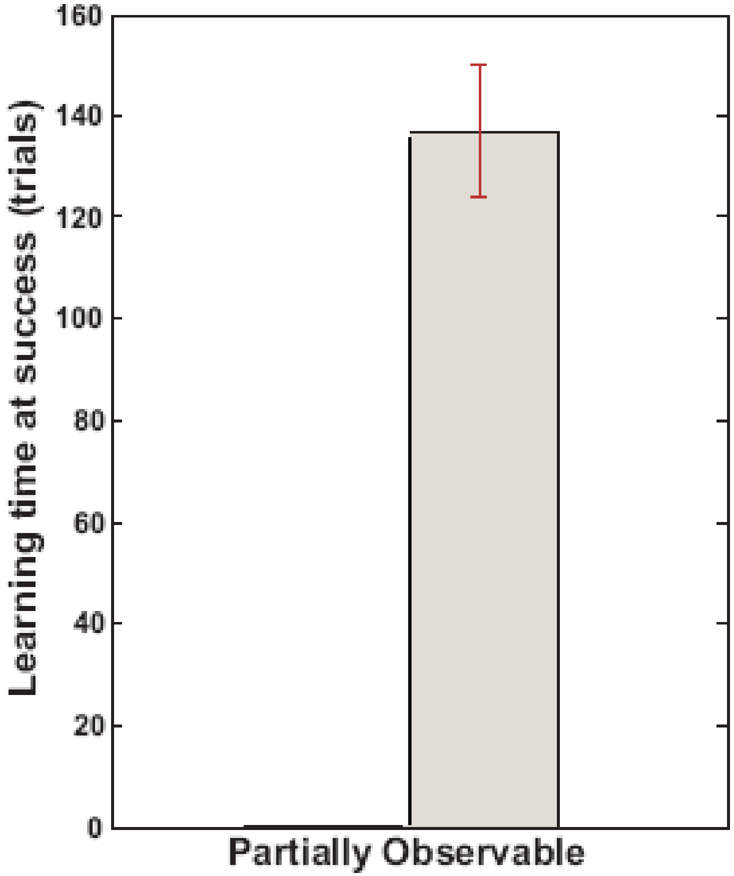


→ What happen if all states cannot be observed at all time (Partially Observable State)?

Bio-inspired Combinatorial Learning

- Partially Observable State

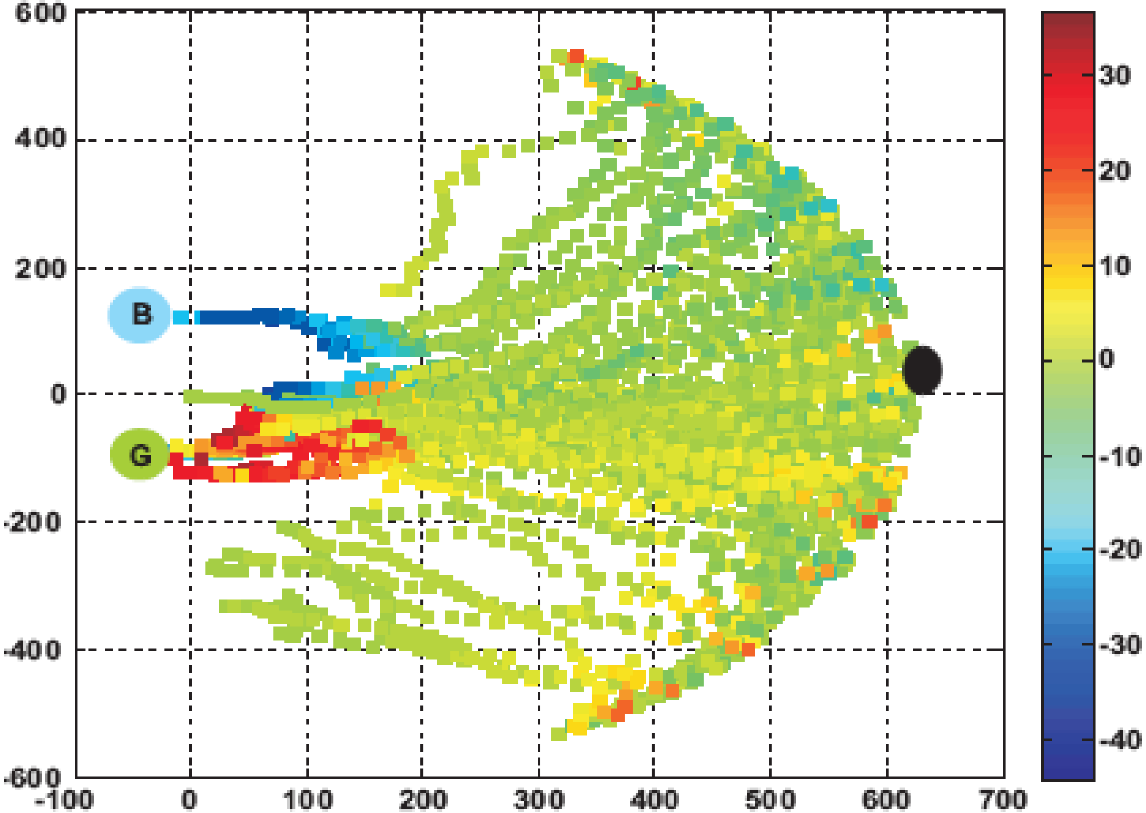




Bio-inspired Combinatorial Learning

- Partially Observable State

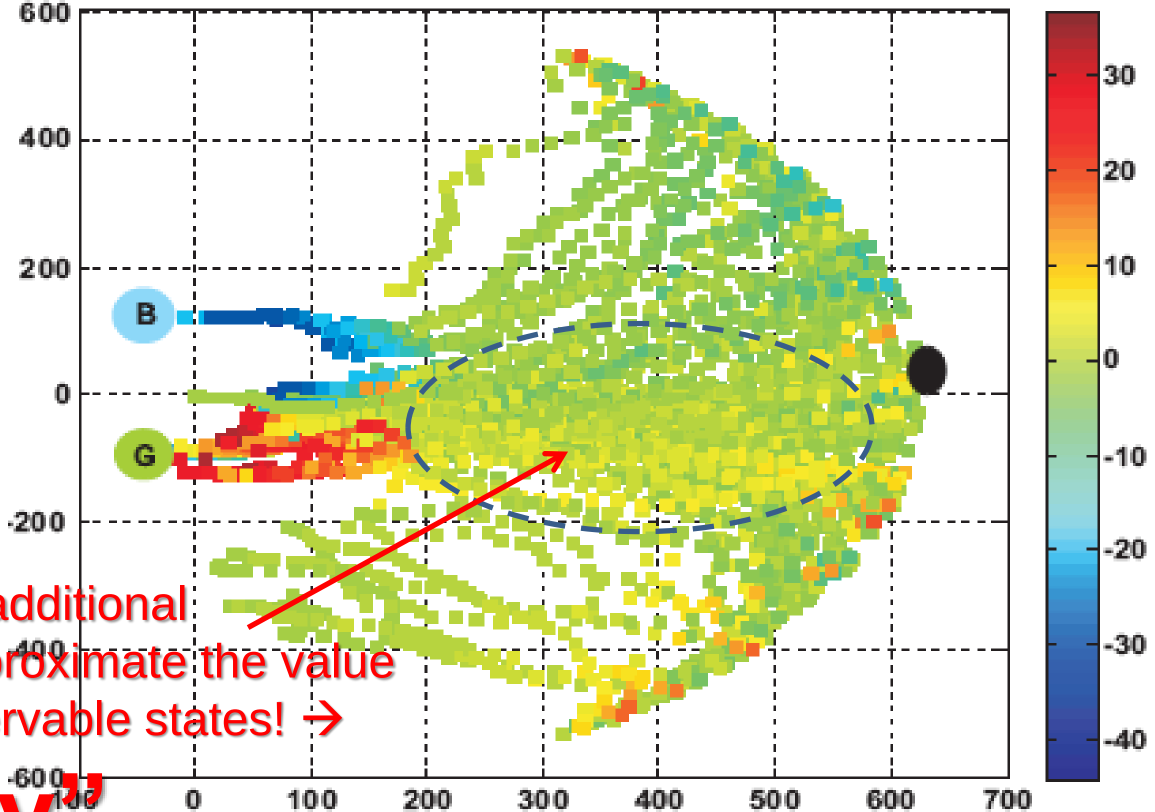
Value function (Expected cumulative reward)



Bio-inspired Combinatorial Learning

- Partially Observable State

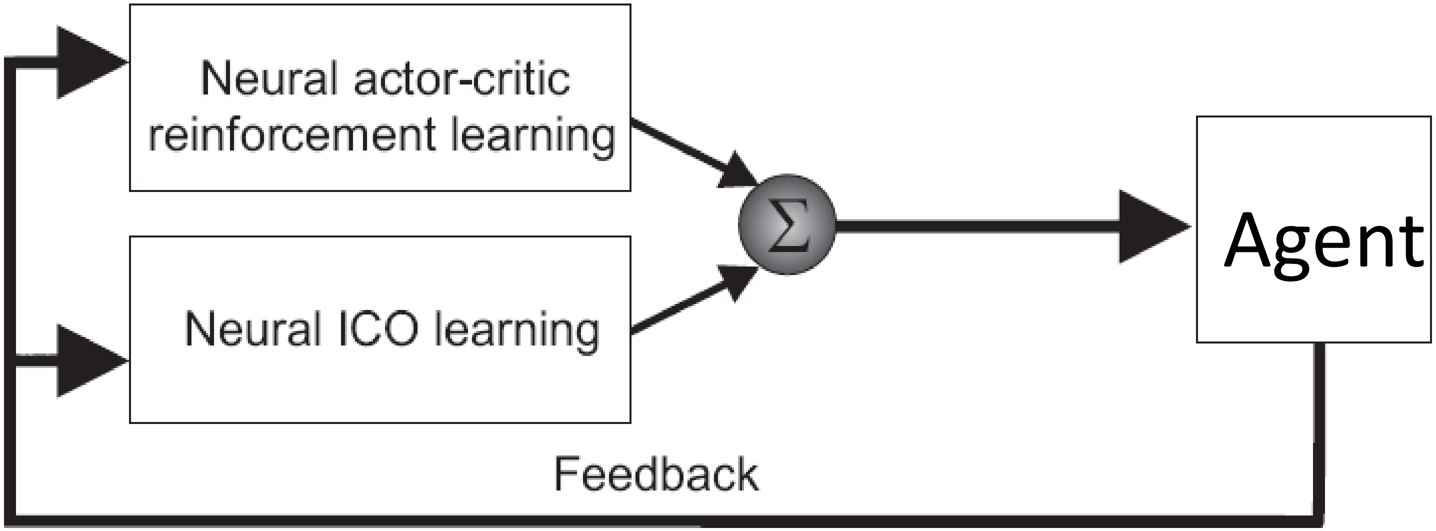
Value function (Expected cumulative reward)



This requires an additional mechanism to approximate the value function in unobservable states! → **"Memory"**

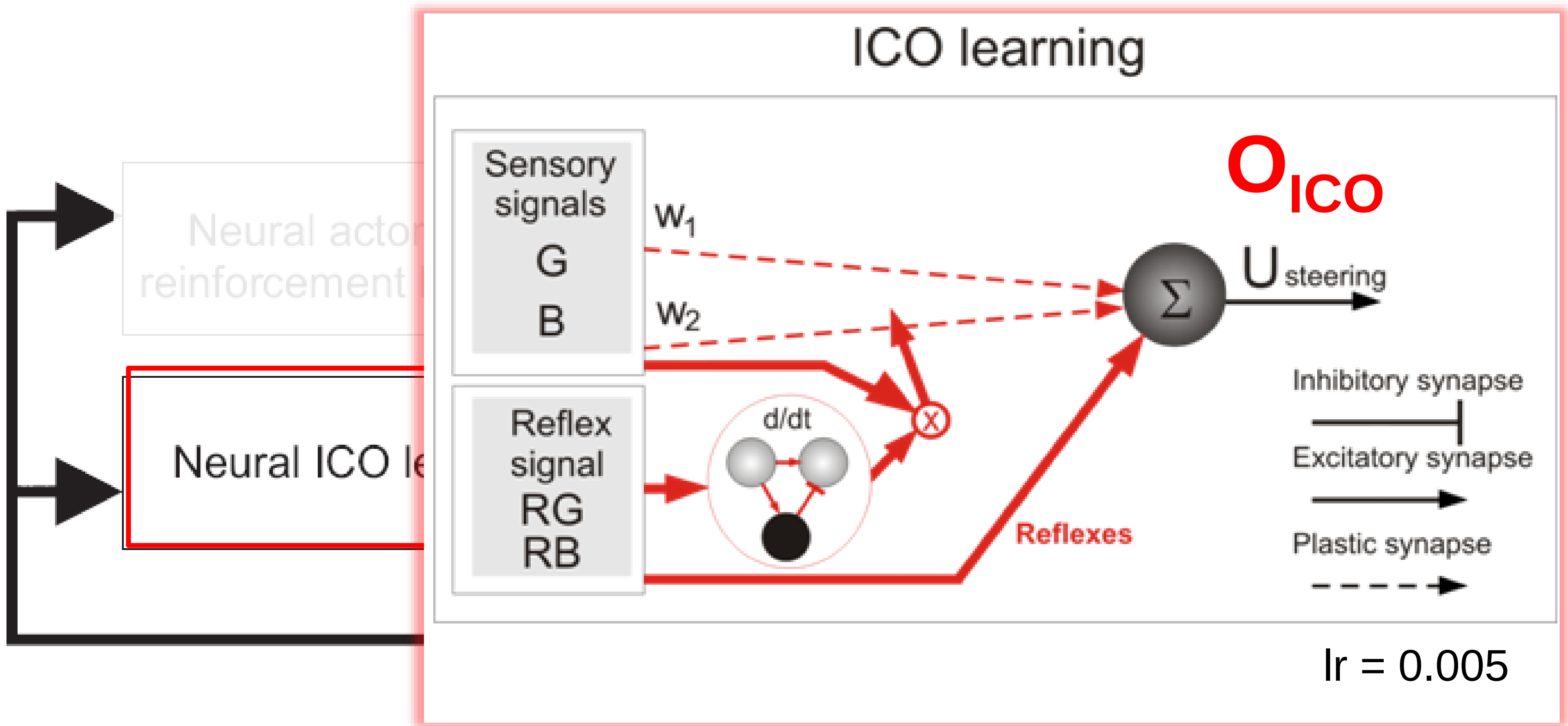
Bio-inspired Combinatorial Learning

- Partially Observable State



Bio-inspired Combinatorial Learning

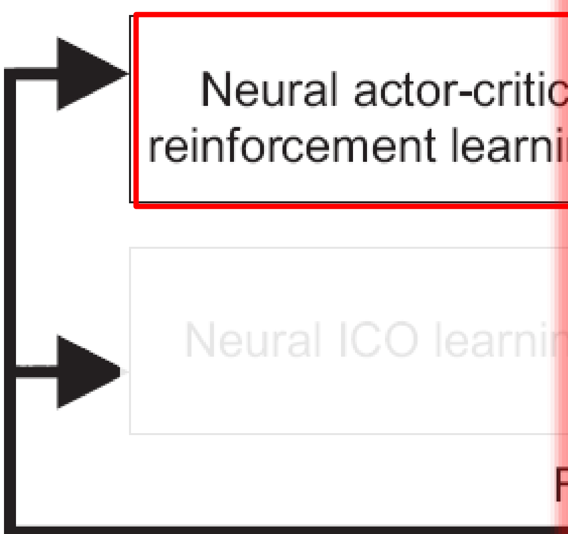
- Partially Observable State

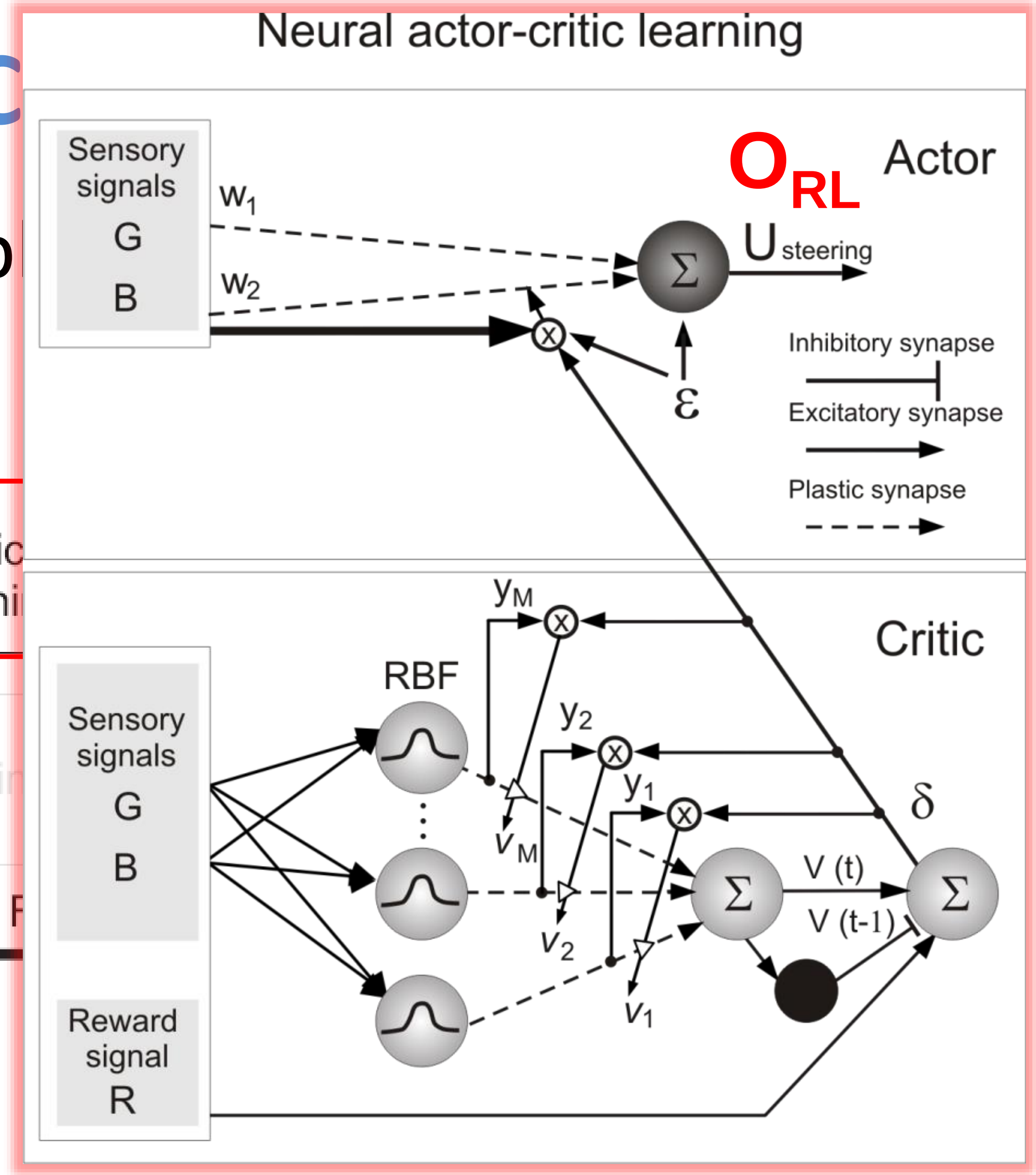


Remain unchanged!

Bio-inspired C

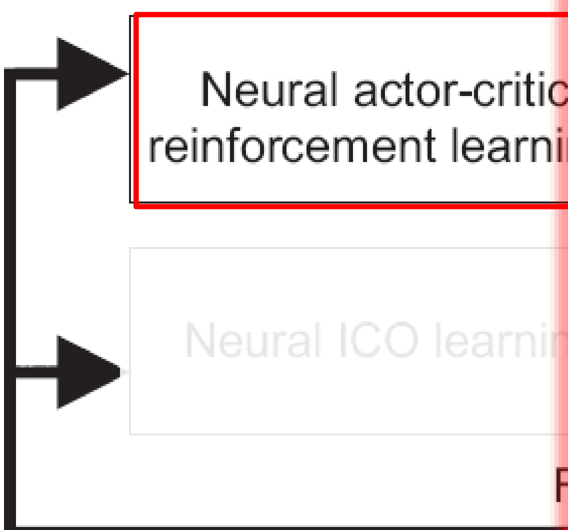
- Partially Observabl

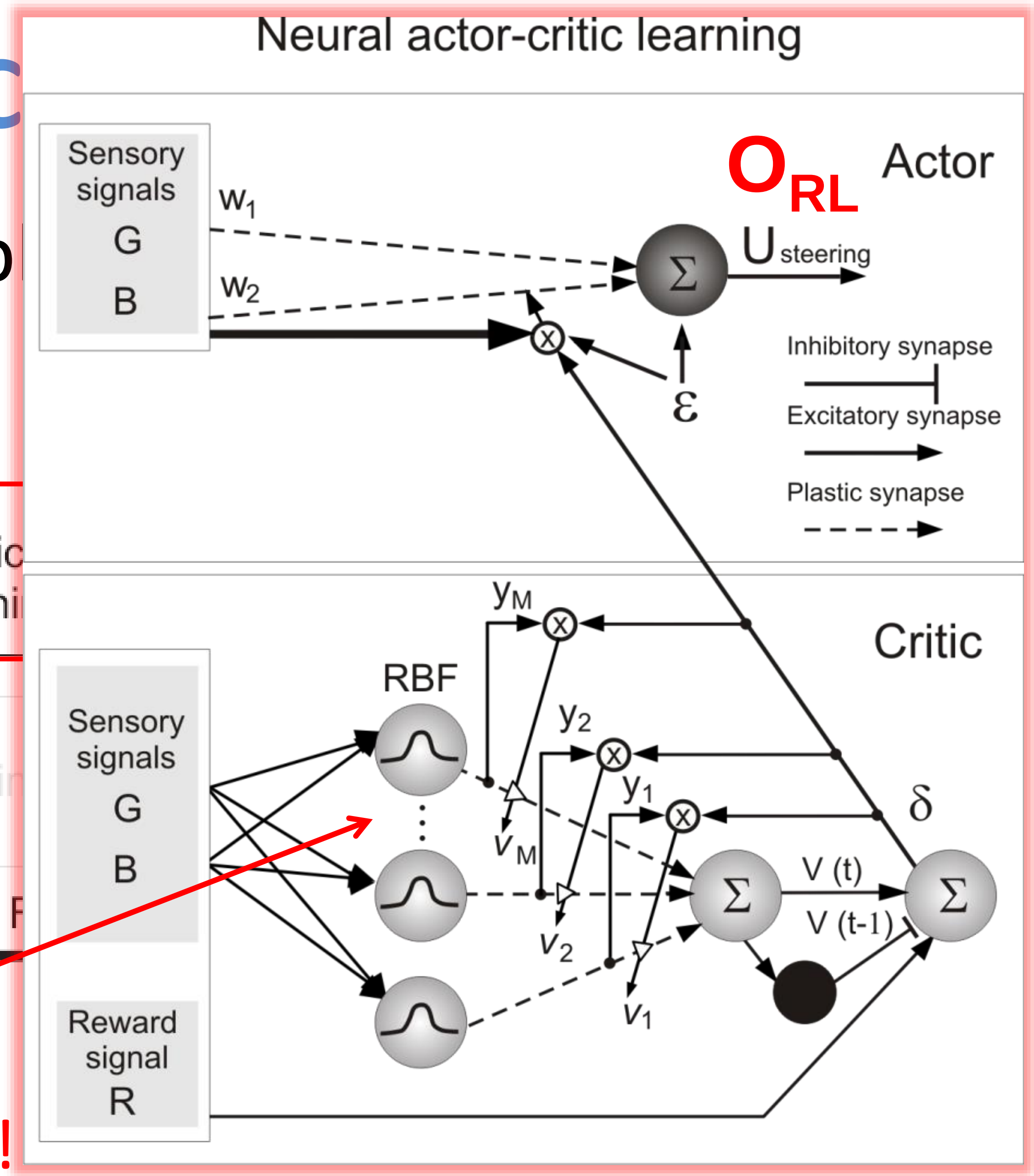




Bio-inspired C

- Partially Observabl

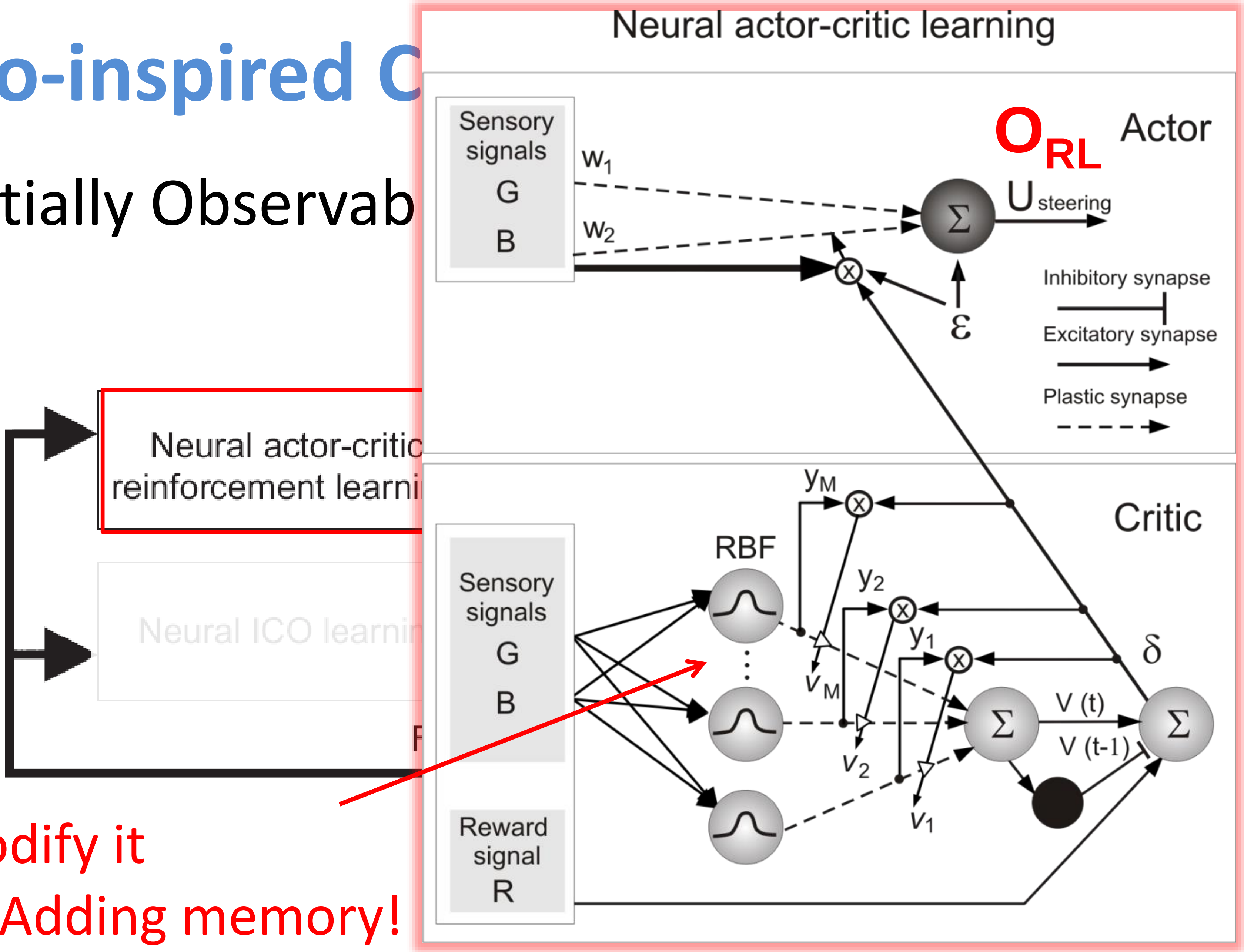




RBF is just nonlinear mapping, no memory!

Bio-inspired C

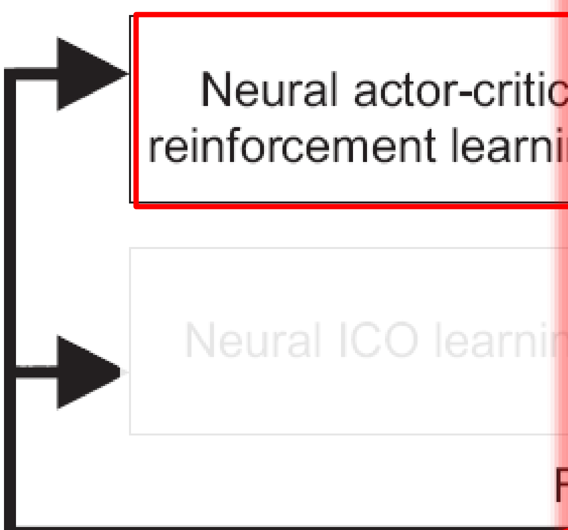
- Partially Observabl

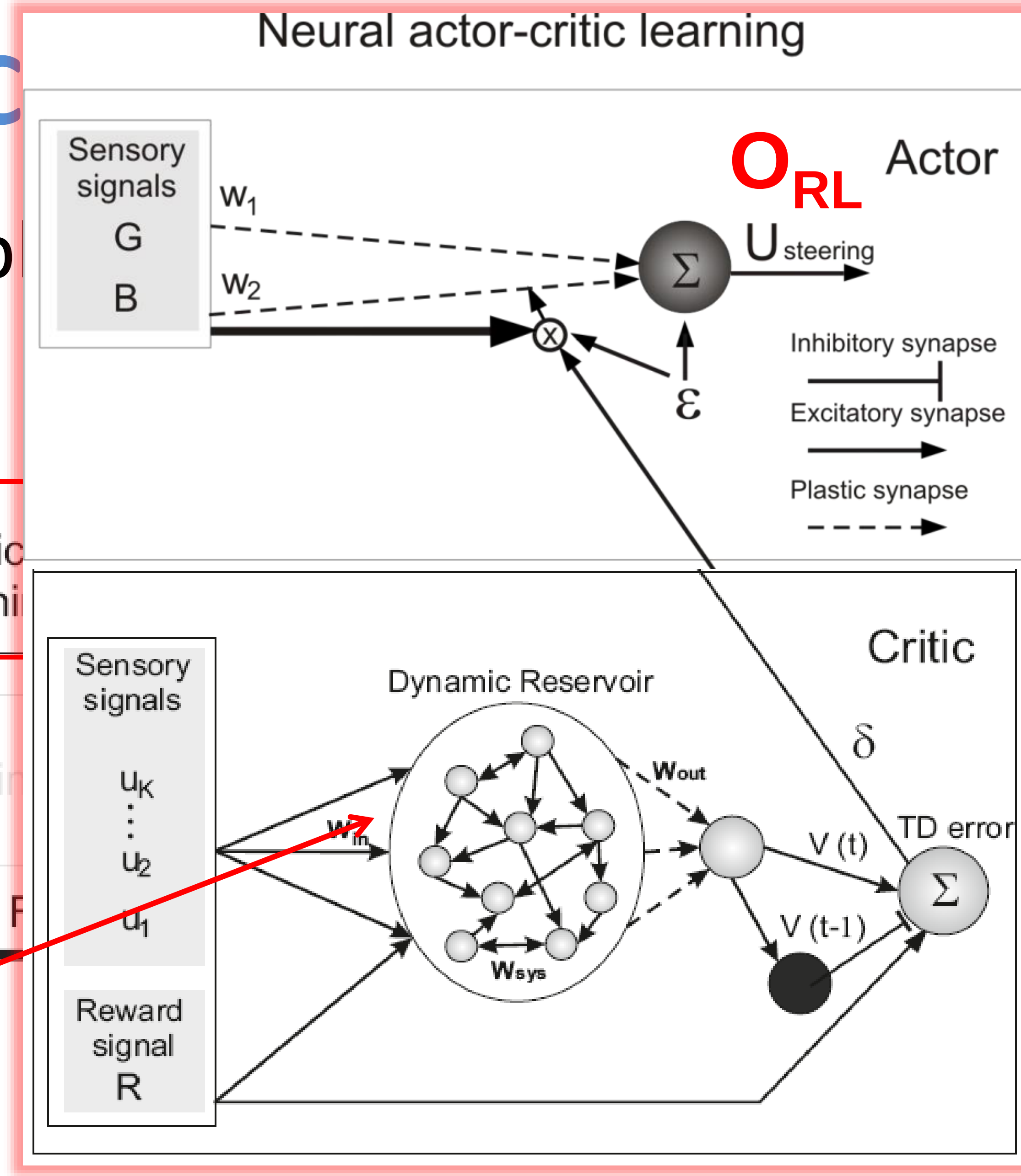


Modify it
→ Adding memory!

Bio-inspired C

- Partially Observabl



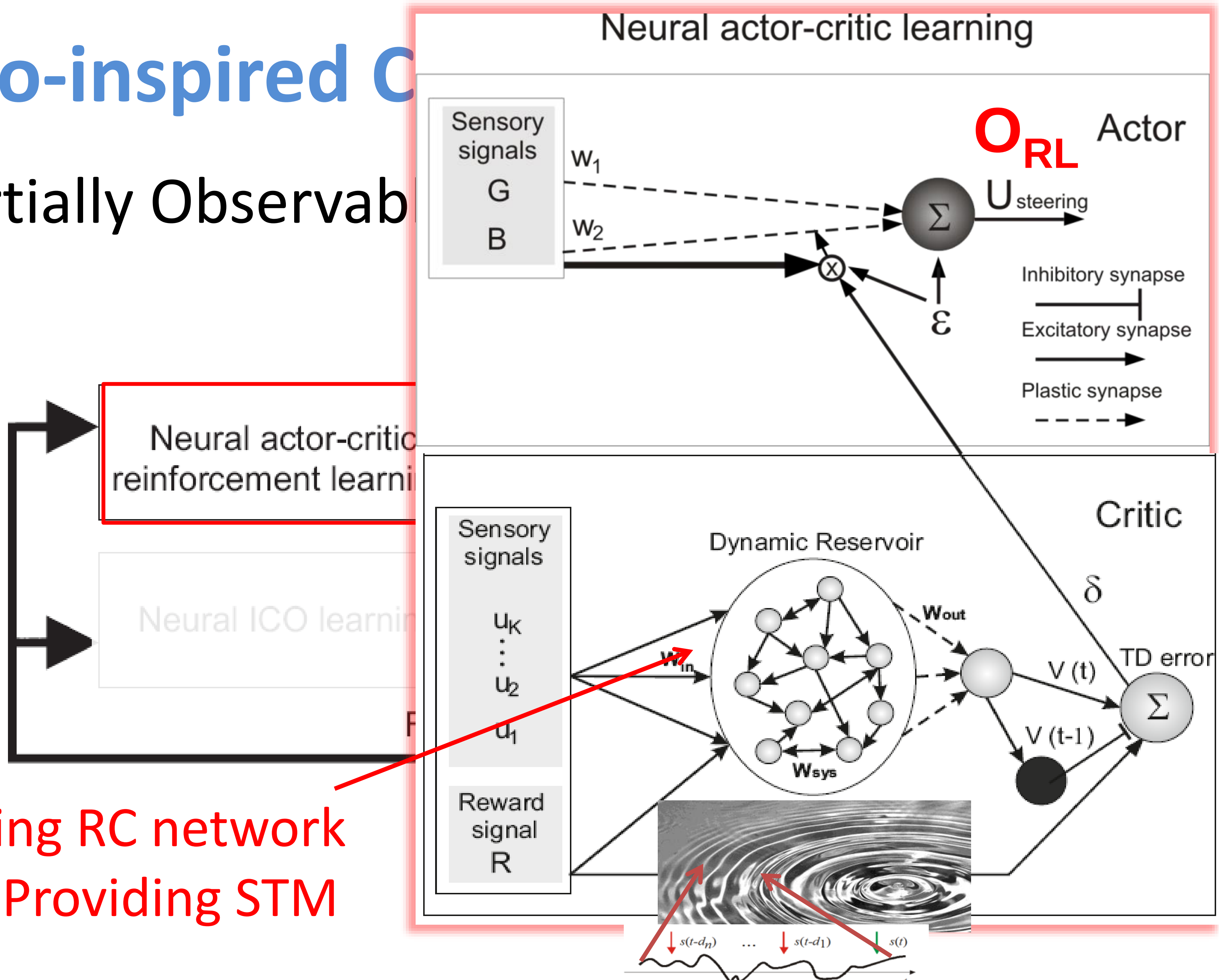


Using RC network
→ Providing STM

Dasgupta et al., 2013

Bio-inspired C

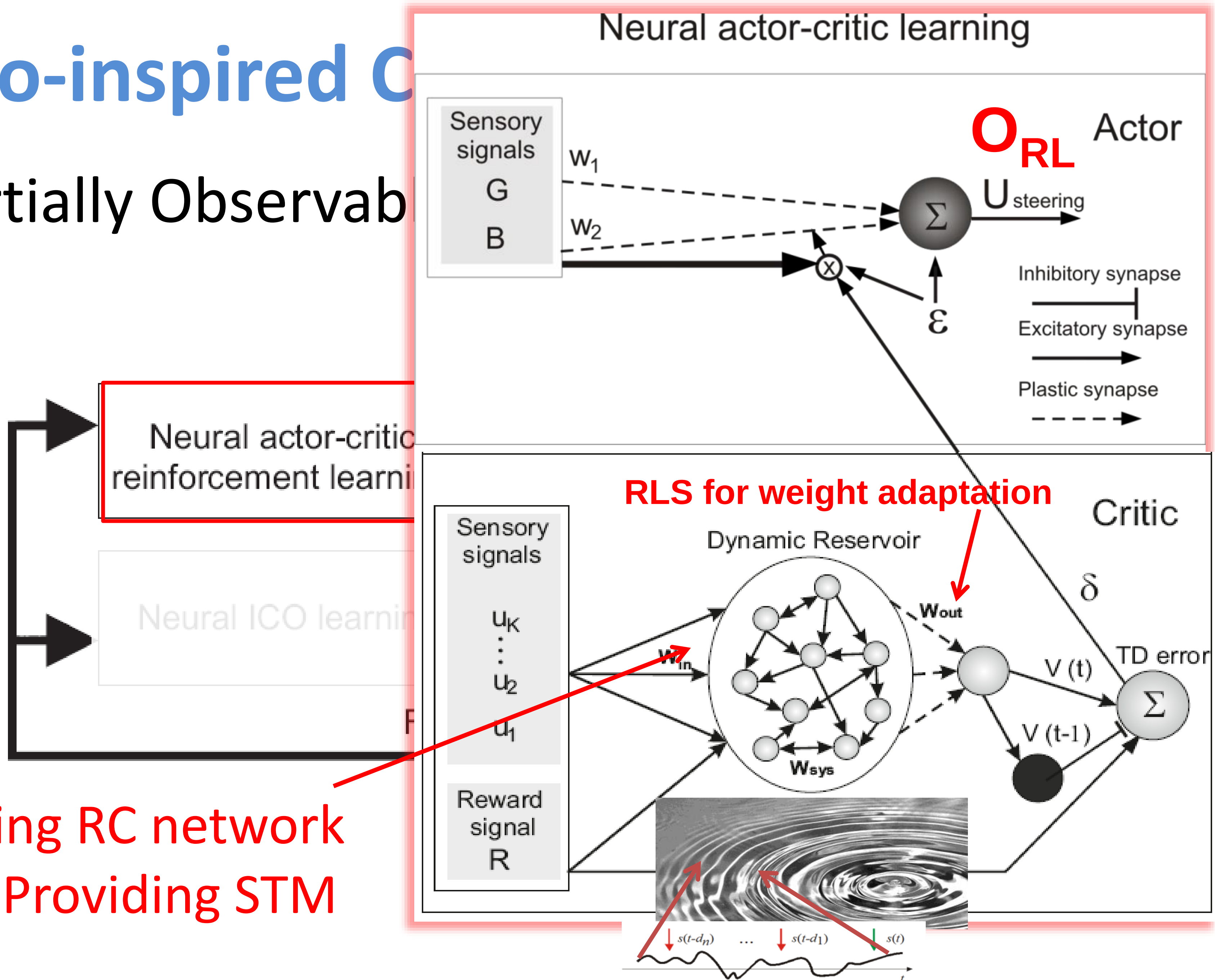
- Partially Observabl



Using RC network
→ Providing STM

Bio-inspired C

- Partially Observabl

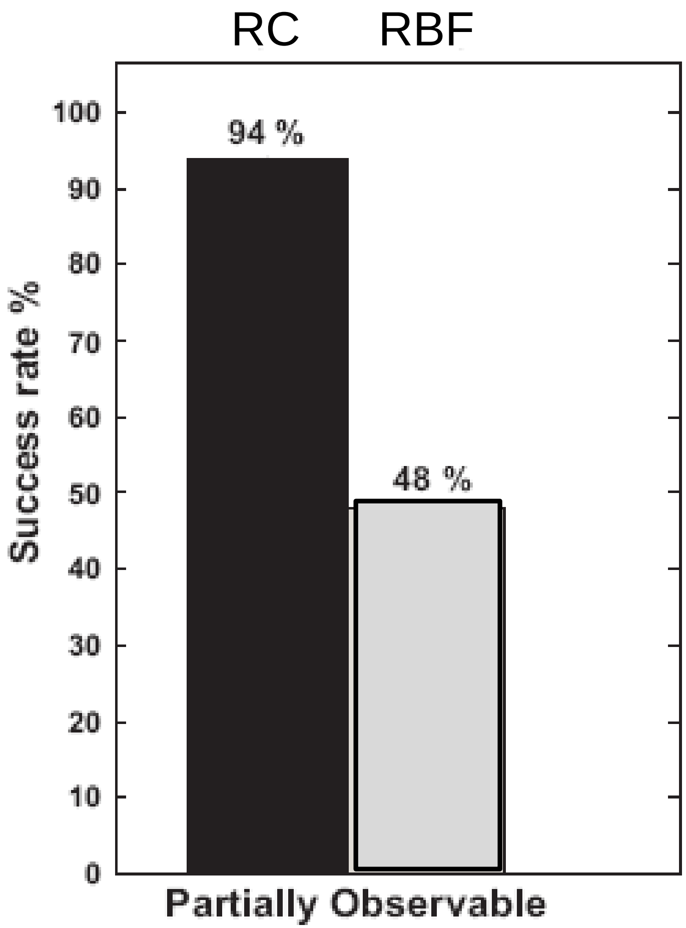


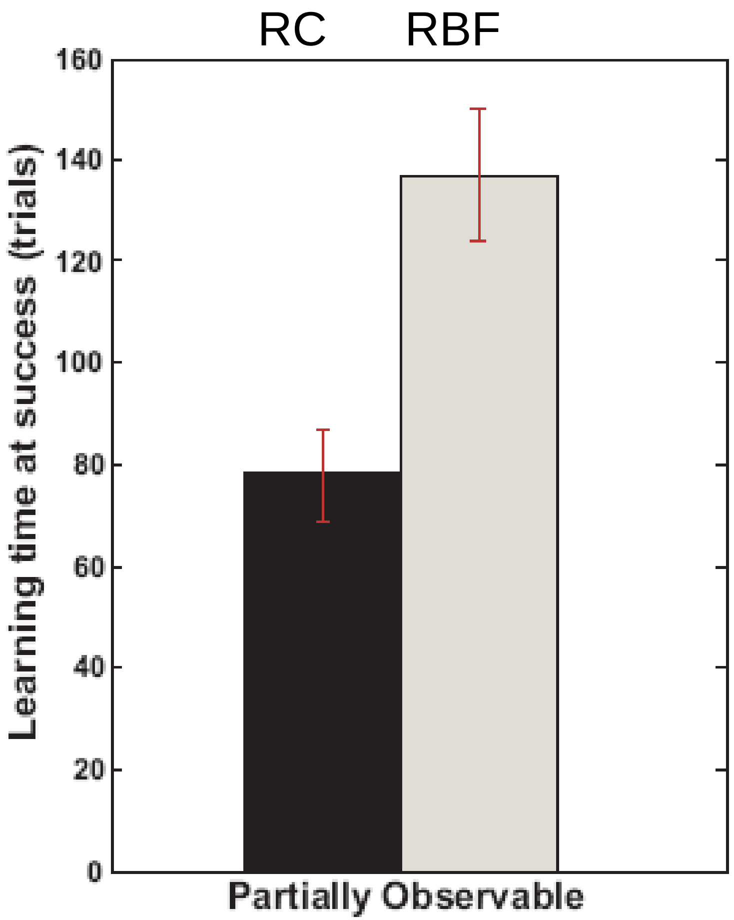
Using RC network

→ Providing STM

Bio-inspired Combinatorial Learning

- Partially Observable State





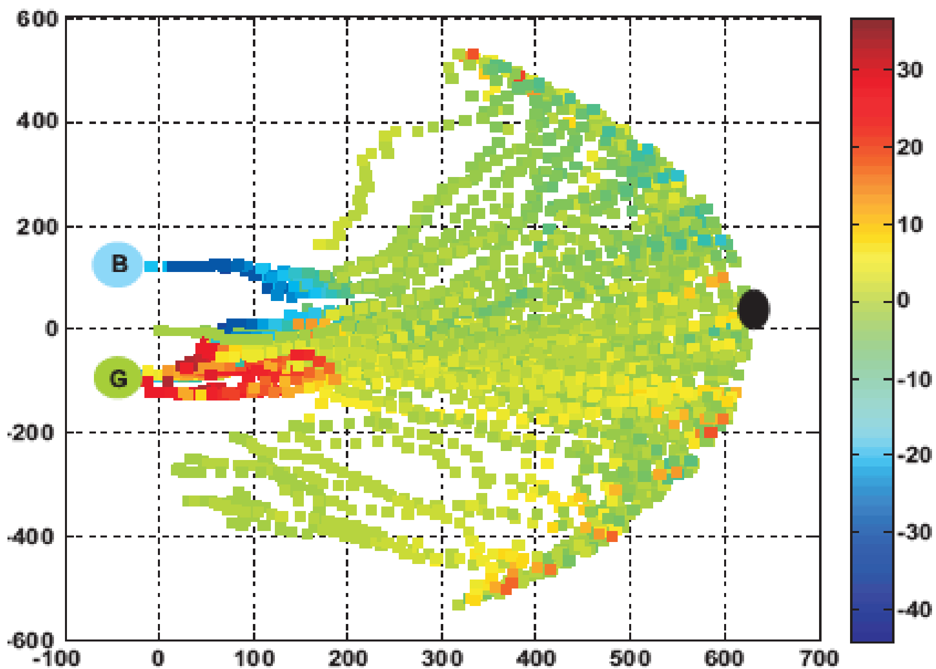
Dasgupta et al., 2013

Bio-inspired Combinatorial Learning

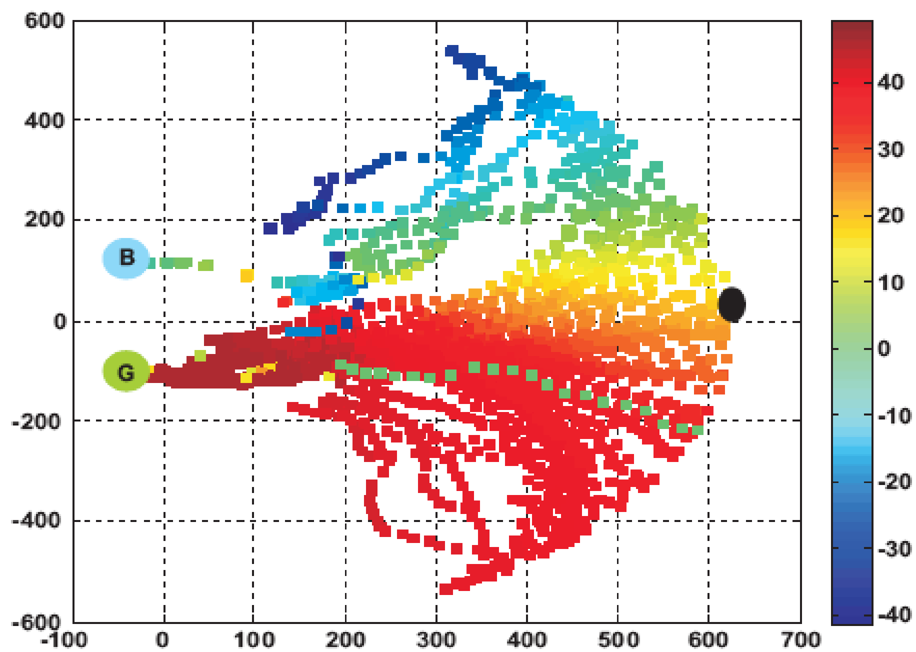
- Partially Observable State

Value function (Expected cumulative reward)

RBF critic net



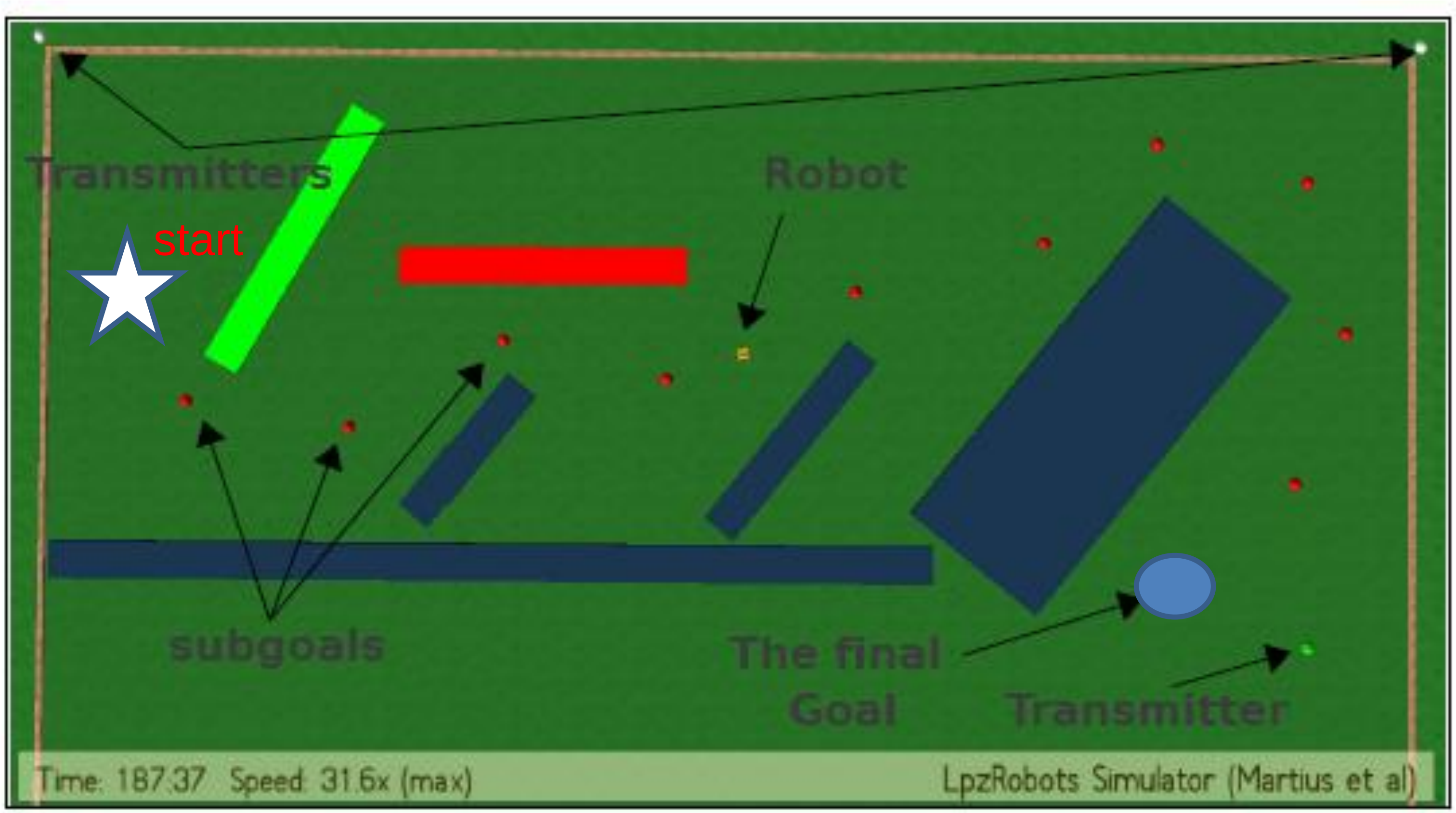
RC critic net



Dasgupta et al., 2013

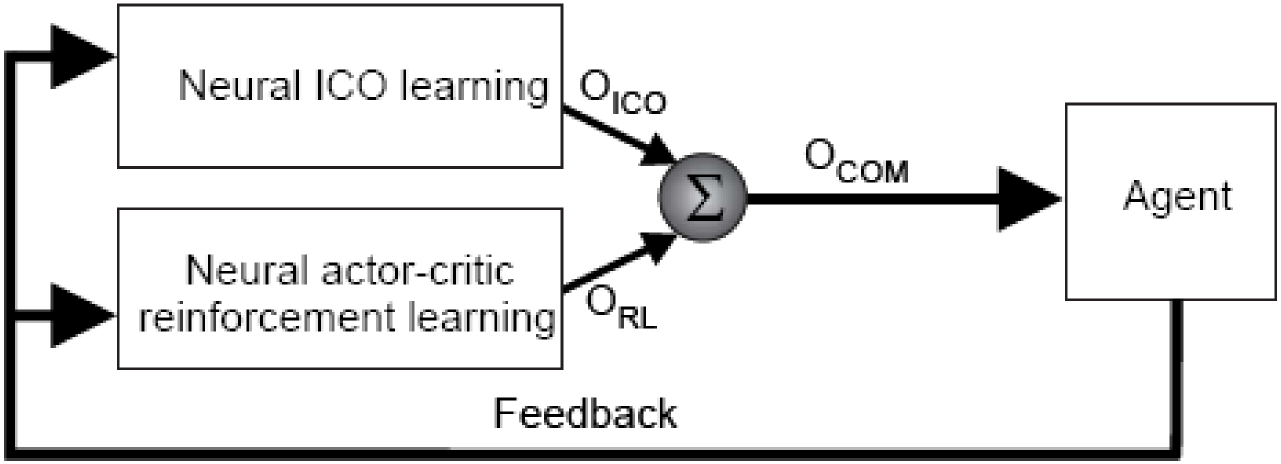
Bio-inspired Combinatorial Learning

- Complex navigation: Landmark to landmark!



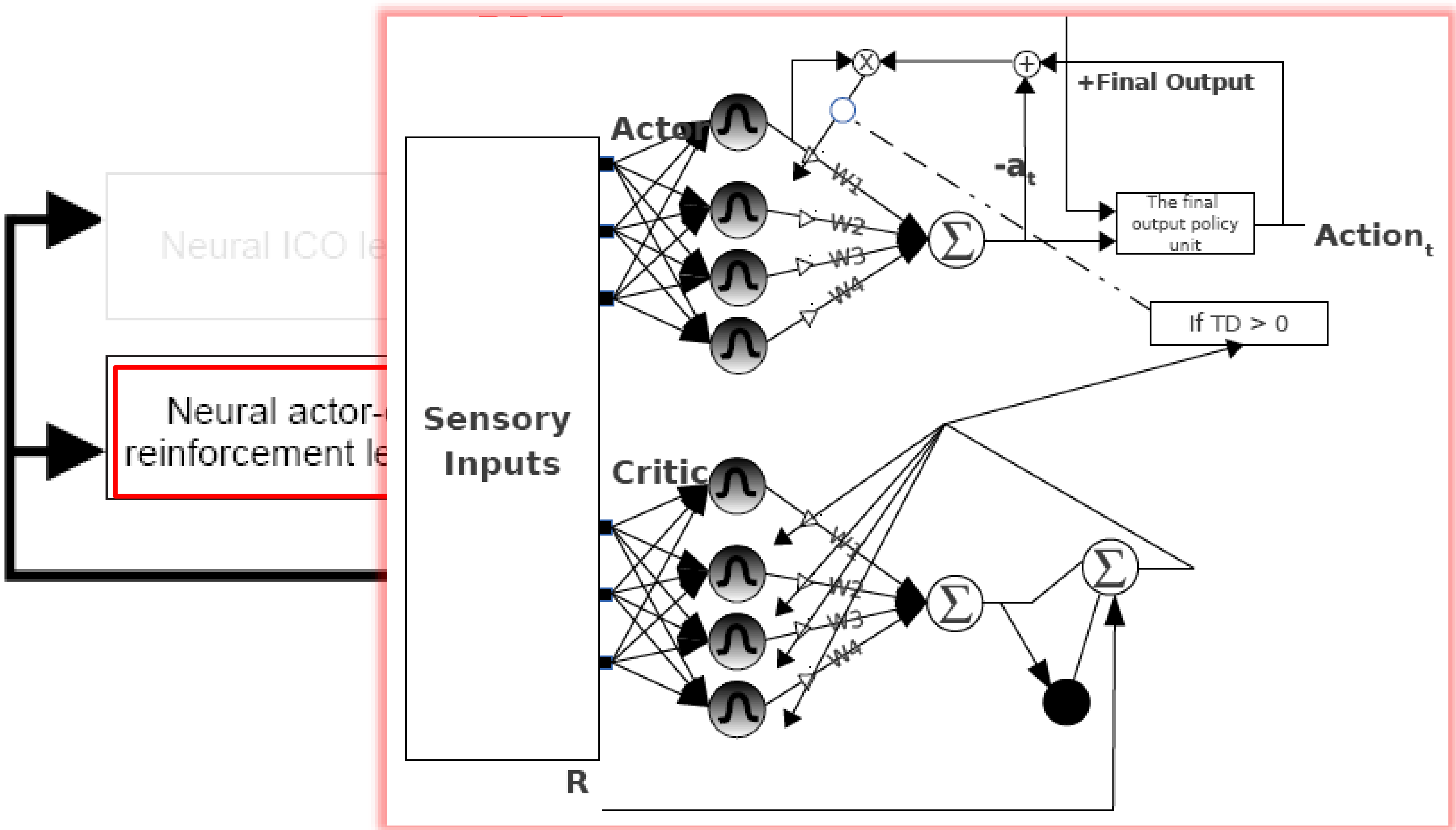
Bio-inspired Combinatorial Learning

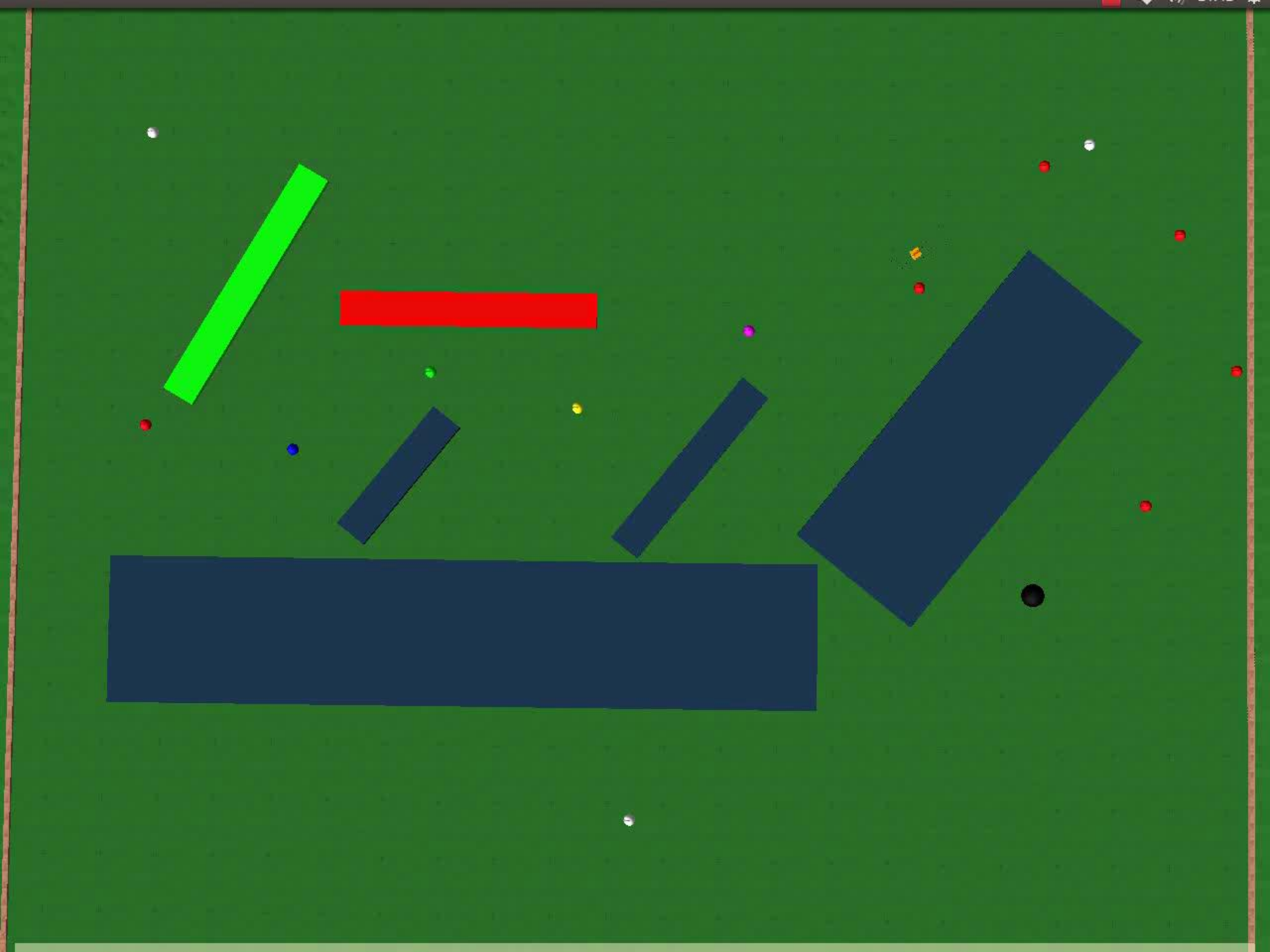
- Complex navigation: Landmark to landmark!



Bio-inspired Combinatorial Learning

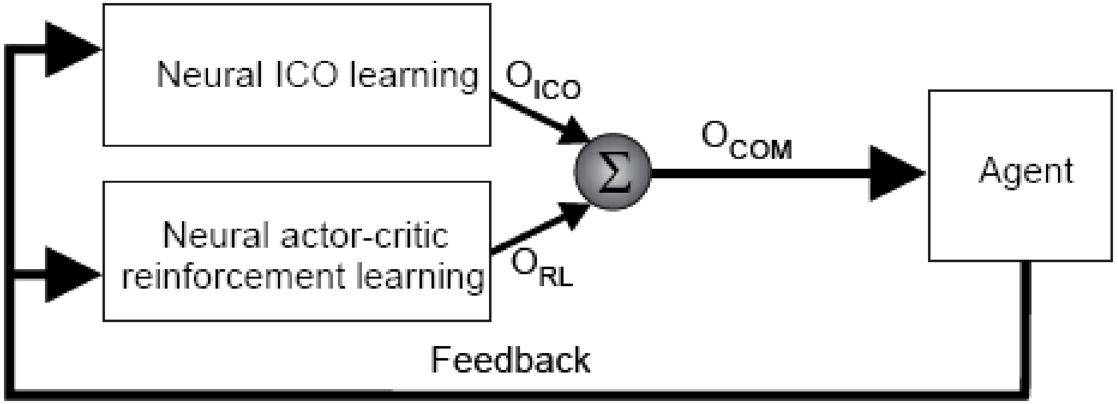
- Complex navigation: Landmark to landmark!





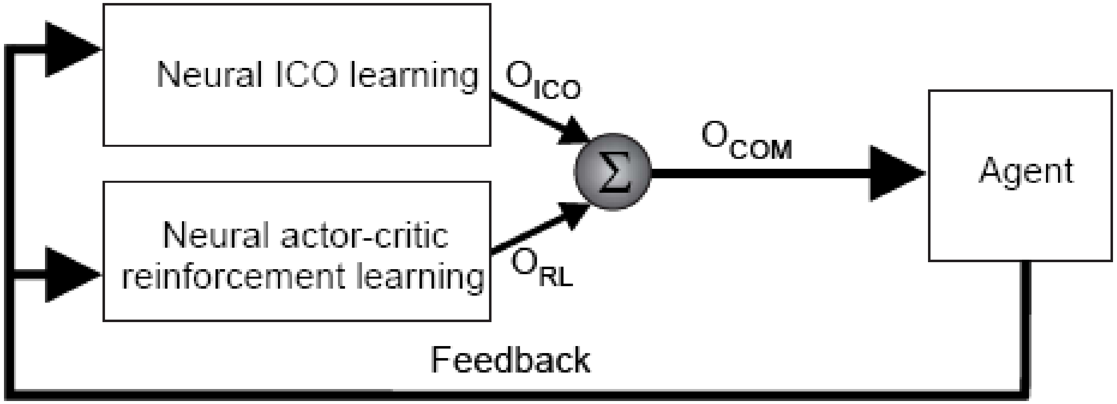
Bio-inspired Combinatorial Learning

- Parallel Combination Model

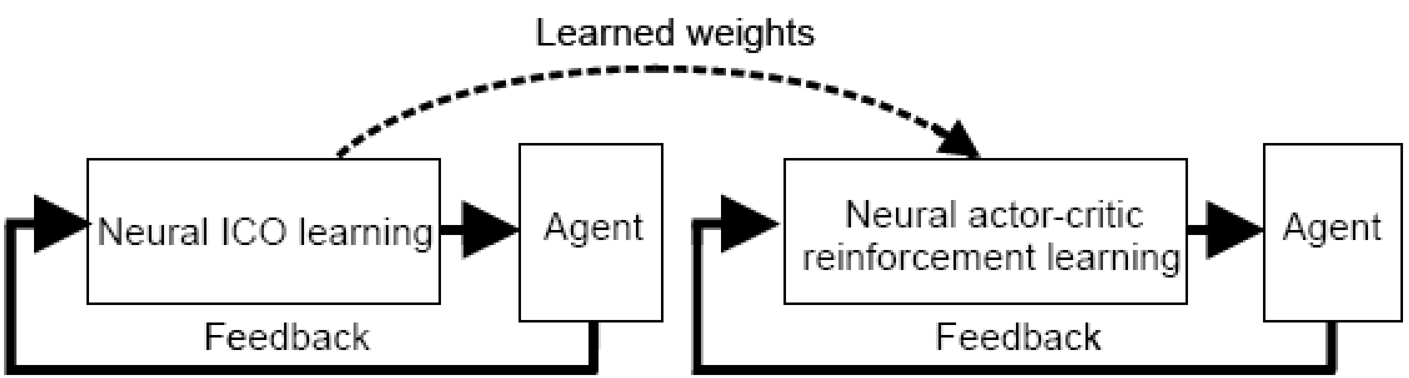


Bio-inspired Combinatorial Learning

- Parallel Combination Model



- Sequential Combination Model



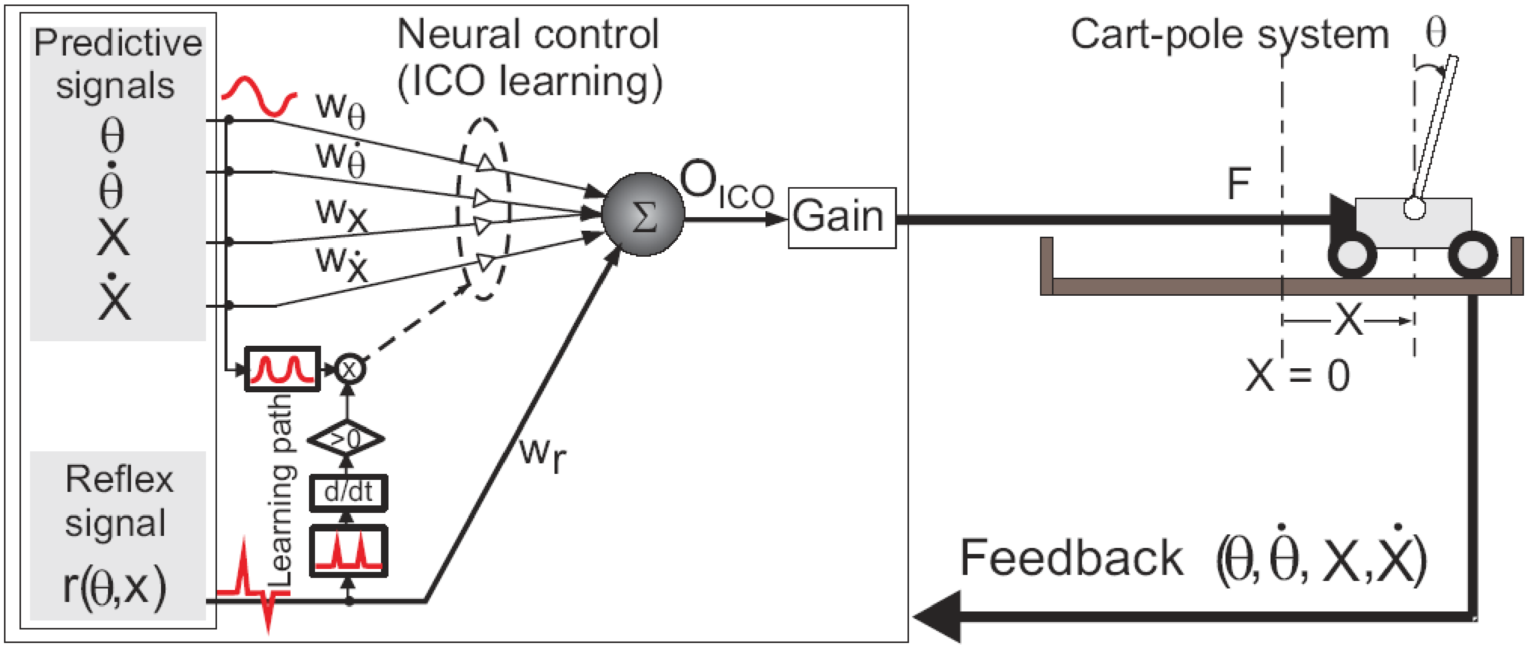
Bio-inspired Combinatorial Learning

- Sequential Combination Model: Pole Balancing Task

Bio-inspired Combinatorial Learning

- Sequential Combination Model: Pole Balancing Task

Step 1: ICO learning



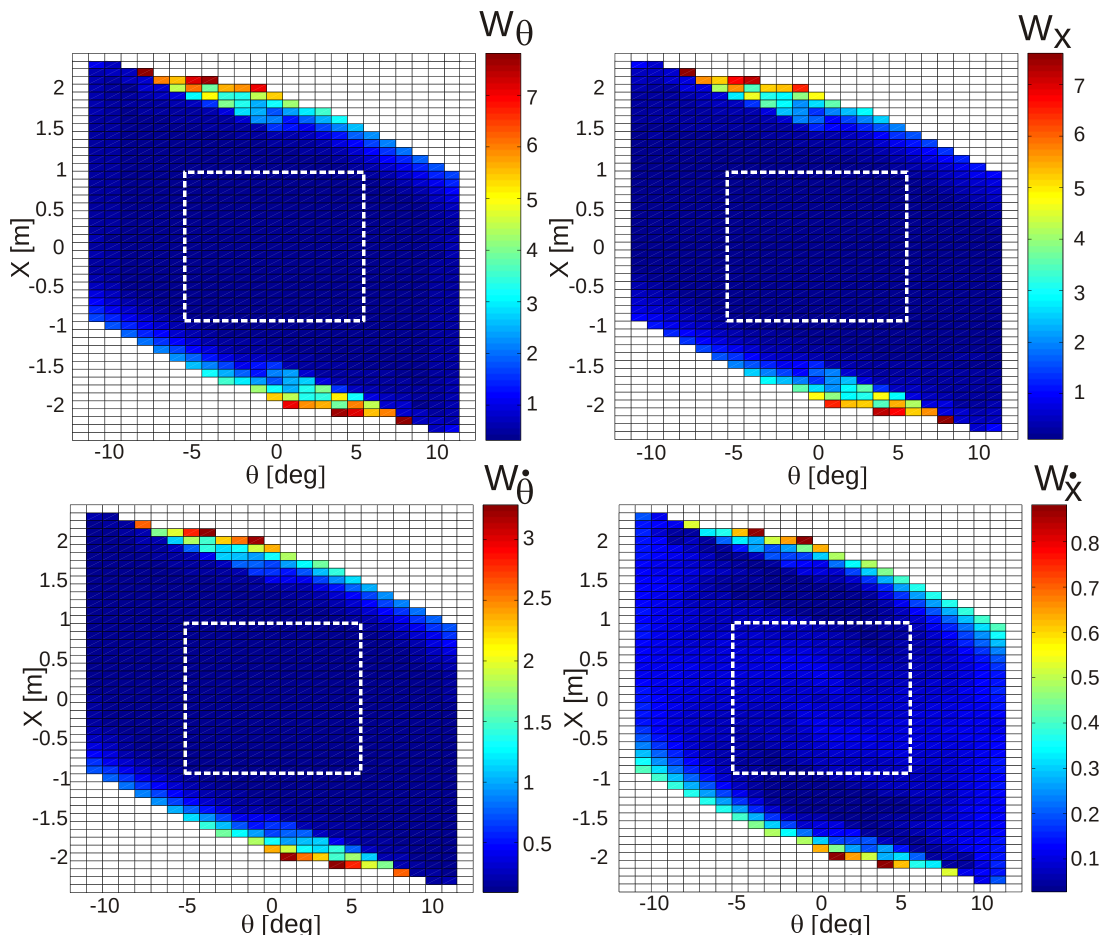
Manoonpong et al., 2010

Bio-inspired Combinatorial Learning

- Sequential Combination Model: Pole Balancing Task

Step 1: ICO learning

$$w_\theta\,,\ w_x\,,\ w_{\dot{\theta}}\,,\ w_{\dot{x}}$$

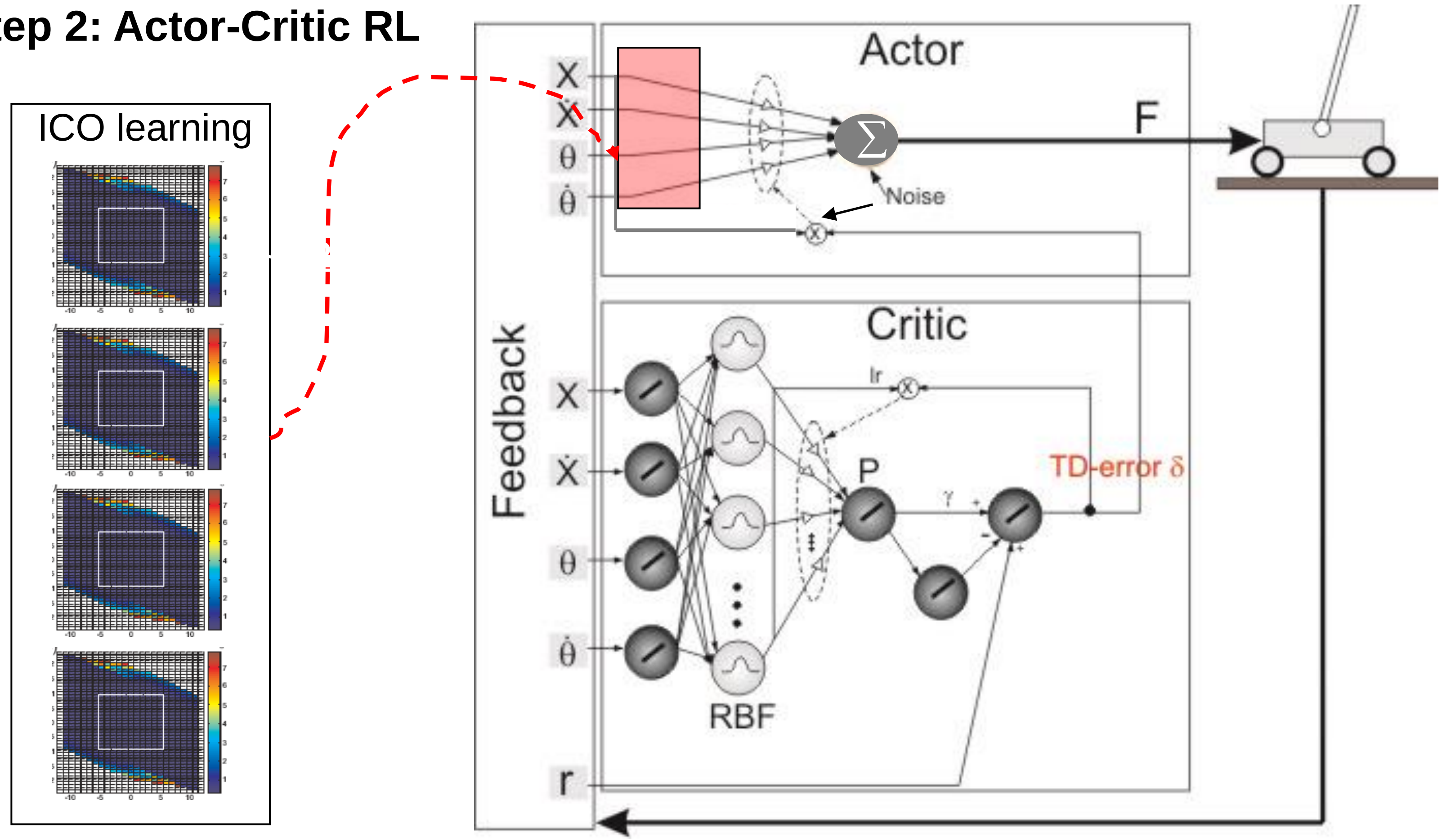


Prior knowledge for RL

Bio-inspired Combinatorial Learning

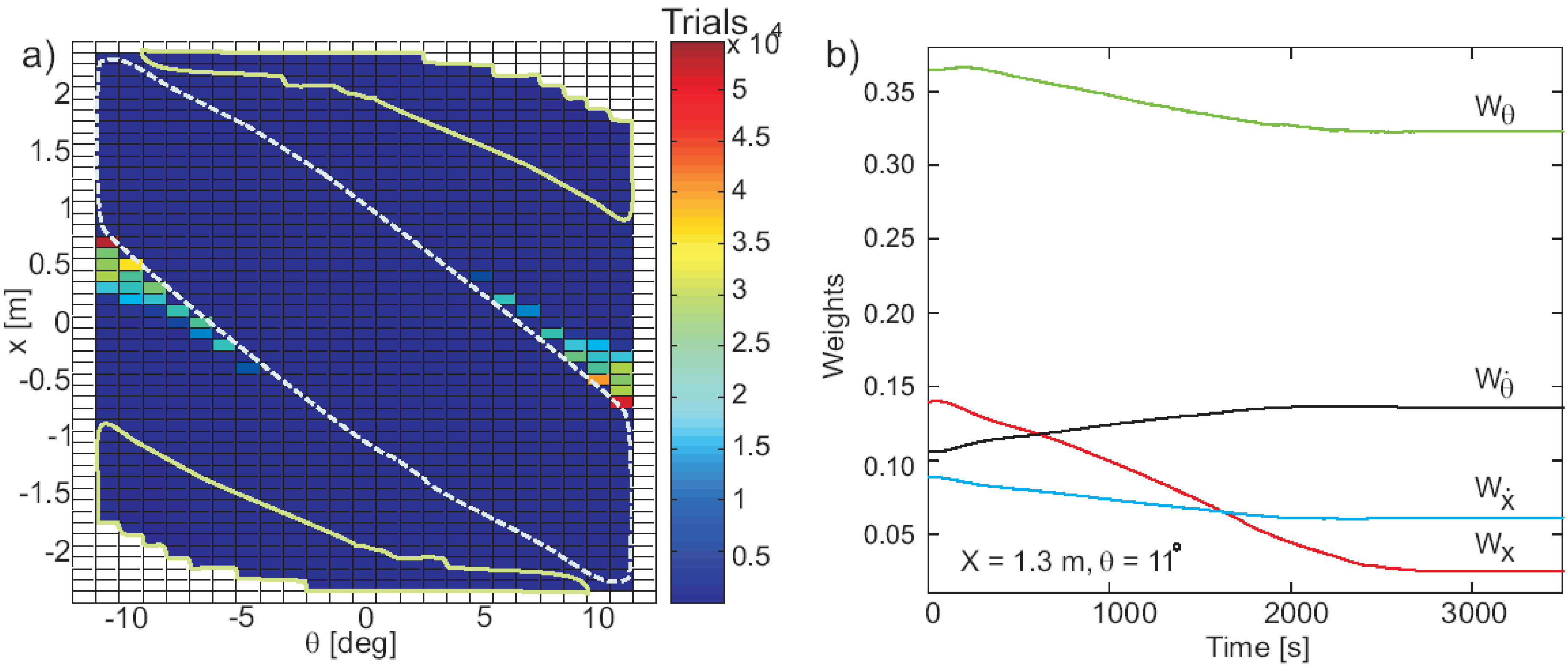
- Sequential Combination Model: Pole Balancing Task

Step 2: Actor-Critic RL



Bio-inspired Combinatorial Learning

- Sequential Combination Model: Pole Balancing Task



Manoonpong et al., 2010

Summary

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator
- Combinatorial learning: On-policy TD control
  - ICO learning: Classical conditioning
  - Actor-critic RL: Operant conditioning

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator
- Combinatorial learning: On-policy TD control
  - ICO learning: Classical conditioning
  - Actor-critic RL: Operant conditioning
  - Parallel Combination with a linear actor and an RBF critic network (no memory)→ Fully observable state

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator
- Combinatorial learning: On-policy TD control
  - ICO learning: Classical conditioning
  - Actor-critic RL: Operant conditioning
  - Parallel Combination with a linear actor and an RBF critic network (no memory)→ Fully observable state
  - Parallel Combination with a linear actor and an RC critic neural network (STM) → Partially observable state

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator
- Combinatorial learning: On-policy TD control
  - ICO learning: Classical conditioning
  - Actor-critic RL: Operant conditioning
  - Parallel Combination with a linear actor and an RBF critic network (no memory)→ Fully observable state
  - Parallel Combination with a linear actor and an RC critic neural network (STM) → Partially observable state
  - Parallel Combination with RBF actor and critic networks for complex navigation

Summary

- Actor-critic RL: On-policy TD control
  - Actor is a controller
  - Critic is a value function approximator
- Combinatorial learning: On-policy TD control
  - ICO learning: Classical conditioning
  - Actor-critic RL: Operant conditioning
  - Parallel Combination with a linear actor and an RBF critic network (no memory)→ Fully observable state
  - Parallel Combination with a linear actor and an RC critic neural network (STM) → Partially observable state
  - Parallel Combination with RBF actor and critic networks for complex navigation
  - Sequential Combination with a linear actor and an RBF critic network

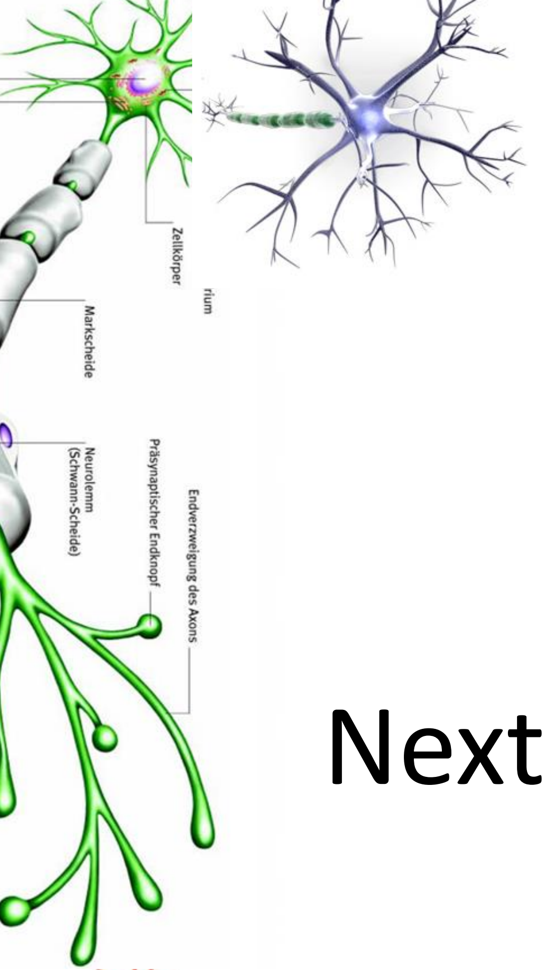
Reading Materials of Today!

Combinatorial learning:

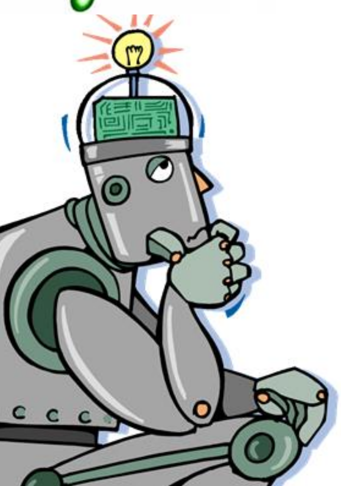
Manoonpong, P., Woergoetter, F., and Morimoto, J. (2010) Extraction of Reward-Related Feature Space Using Correlation-Based and Reward-Based Learning Methods. In Proc. 17th International Conference on Neural Information Processing, Sydney, Australia, November 22-25 (ICONIP’10),Part I, LNCS 6443, pp. 414-421.

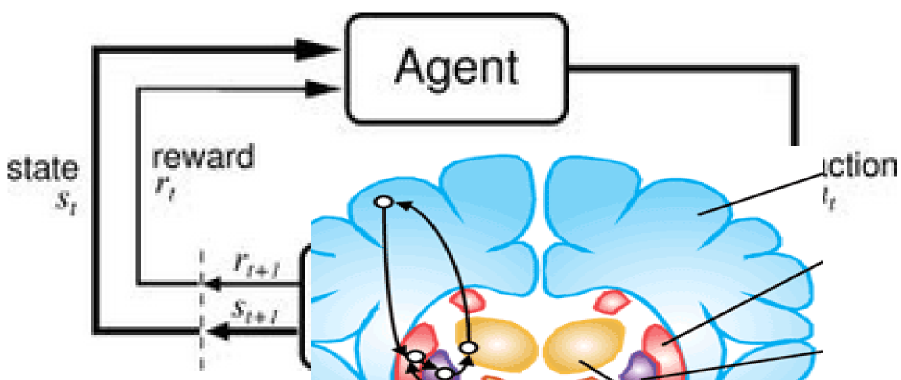
Manoonpong, P., Kolodziejski, C., Woergoetter, F., and Morimoto J. (2013) Combining Correlation-Based and Reward-Based Learning in Neural Control for Policy Improvement. Advances in Complex Systems, doi: 10.1142/S021952591350015X

S. Dasgupta, F. Woergoetter, J. Morimoto, P. Manoonpong (2013) Neural combinatorial learning of goal-directed behavior with reservoir critic and reward modulated hebbian plasticity, in: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC2013), IEEE, Manchester, UK, 2013, pp. 993-1000.



Next time: Deep RL, Dr. Nat!





Rule: Ludo game

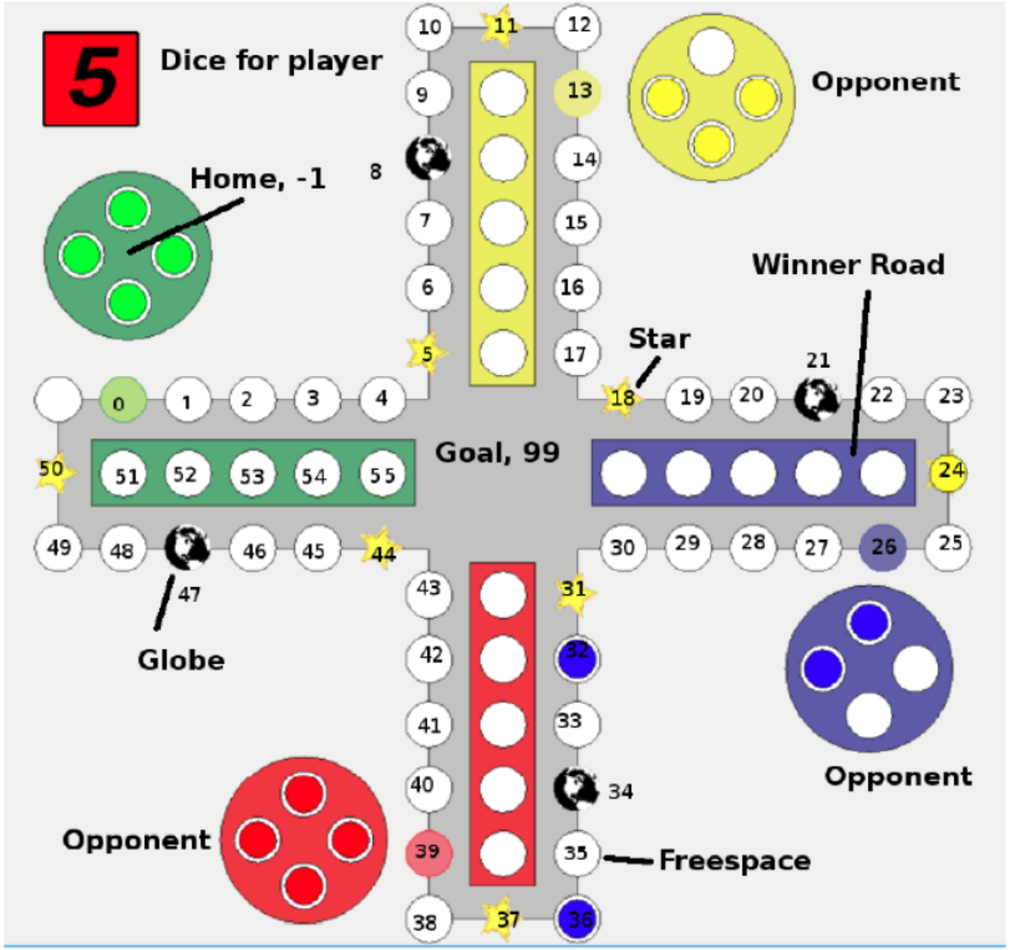
1. In the beginning of game, all 4 pieces (of each player) are at home position and a player needs to dice roll 6 to release a piece

2. The player is declared a winner when all four pieces reach the goal position after passing through all positions on the board

3. Players can hit pieces of opponent players, sending it back home, when they share same un-safe positions

4. Globes and winner road are safe positions where a piece cannot be hit

5. Star positions teleport a piece to the next star position



LUDO RULES

- http://spilregler.dk/ludo/

- **Start**

The players take turns rolling the dice.
When hitting 6s, you have the right to move a piece out on the court.
The players take turns clockwise.

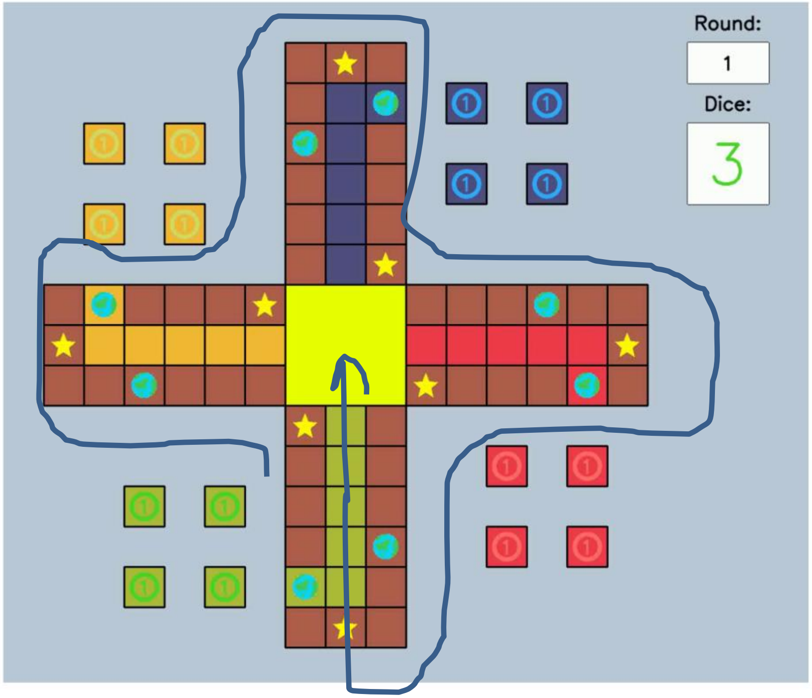
- **The course of the game**

A 6 gives the right to an extra throw.
The player moves the number of squares corresponding to the eyes on the dice.

If you land on a field where there is already a piece, the first piece must return to the starting field.

If, on the other hand, there are two pieces on the field, it is the latter who must return to the start.

One can not fail to move a piece. I.e. one has to move a piece, even if it means one has to go back to the start.

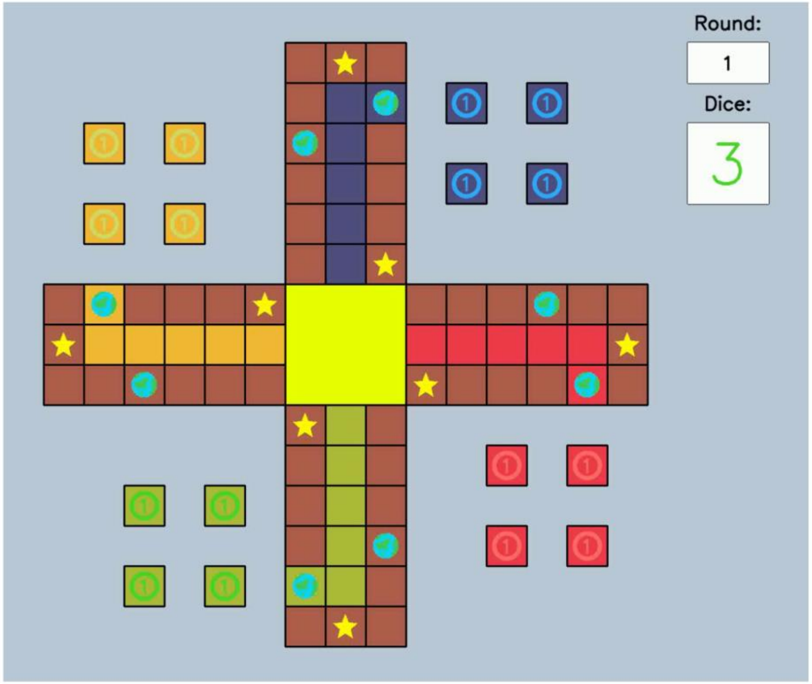


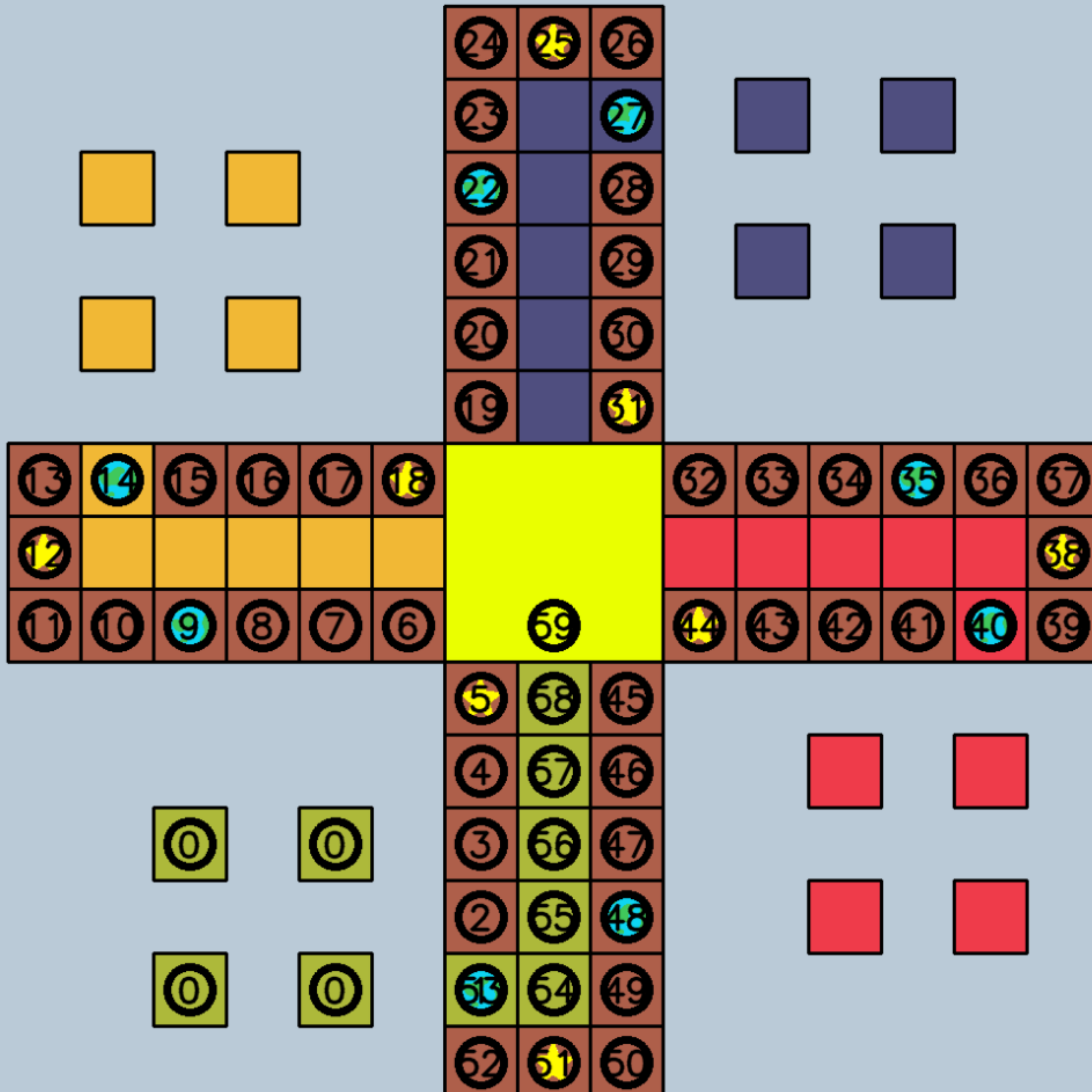
The starting area is where the four pieces start. To take a piece out of the starting area, you must hit a six. If you have all your pieces in the starting area, you get up to three strokes with the dice before the turn is given.

The target area is where the four pieces are to be followed. Each player has his own goal area and no other player is allowed to take his pieces in there. To reach the end of the target area, it must hit precisely - otherwise you have to move your piece in the opposite direction.

Globe fields protect the pieces from being hit home. If an opponent's piece lands on a protected piece, it is knocked home. However, there is an exception to the colored globe fields. For example, only red pieces can be protected on the red globe field, regardless of the number of pieces you have on the field. If you have two pieces standing on the opponent's colored globe, they can both be beaten home.

Star fields act as shortcuts that can bring the pieces faster to the target area. If a piece lands on a star, it must be moved to the next star. If the star in front of the target area is landed on, the piece is moved directly to the target.



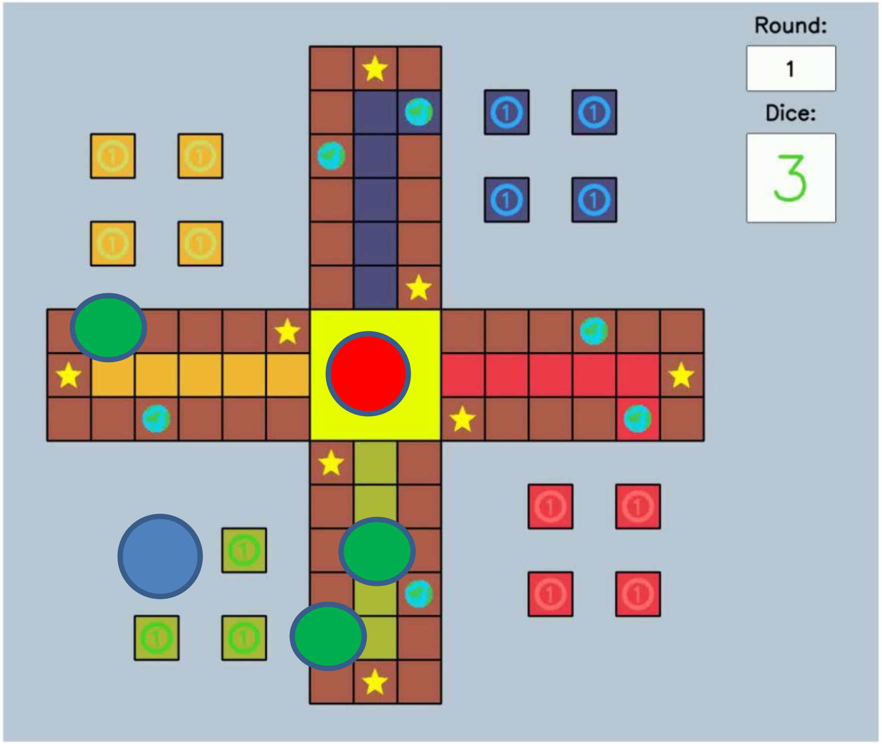


24 25 26
23 27
22 28
21 29
20 30
19 31
13 14 15 16 17 18 32 33 34 35 36 37
12 38
11 10 9 8 7 6 59 44 43 42 41 40 39
5 58 45
4 57 46
0 0 3 56 47
2 55 48
0 0 53 54 49
52 51 50

Q learning

4 States for each player:

- HOME: state in which the token is home, which also is the starting position. A
- GOAL: state in which the token has reached goal.
- SAFE: state where the token is safe but not at home or goal.
- Danger: state in which the token is in danger of being knocked home before the next turn.



10 Actions for all players:

- 1. Move out: Moving the token out of start.
- 2. Normal: Moving the eyes of the dice.
- 3. Goal: Moving into goal position.
- 4. Star: Moving to a star and jump.
- 5. Globe: Moving to a globe.
- 6. Protect: Moving to a token of same color as yourself.
- 7. Kill: Moving to token of opposite color and sending them home.
- 8. Die: Moving to token of opposite color and sending your own token home
- 9. Goal zone: Moving into the goal stretch.
- 10. Nothing: If token can't be moved with current dice roll.

		A	C	T	I	O	N				
		A0	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**
	S0	0	0	0	0	0	0	0	0	0	0
	S1	0	0	0	0	0	0	0	0	0	0
	S2	0	0	0	0	0	0	0	0	0	0
	S3	0	0	0	0	0	0	0	0	0	0
	S4	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	0	0	0	0	0
S	**S7**	0	0	0	0	0	0	0	0	0	0
T	**S8**	0	0	0	0	0	0	0	0	0	0
A	**S9**	0	0	0	0	0	0	0	0	0	0
T	**S10**	0	0	0	0	0	0	0	0	0	0
E	**S11**	0	0	0	0	0	0	0	0	0	0
	S12	0	0	0	0	0	0	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0
	S14	0	0	0	0	0	0	0	0	0	0
	S15	0	0	0	0	0	0	0	0	0	0

Table 1. Q-tabel initialized to 0

REWARDS									
Nothing	Move_out	Normal	Goal	Star	Globe	Protect	Kill	Die	Goal_zone
0	0.25	0.01	0.8	0.5	0.4	0.3	0.4	0.5	0.4

$$\Delta Q(s,a) = \alpha\left(r + \gamma \max_{a_1} Q(s_1,a_1) - Q(s,a)\right)$$

A C T I O N

S T A T E

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
S0	-0.08627	0.45842	0	0	0	0	0	0	0	0
S1	0	0	0.22163	1.13863	0.65245	0.59999	0.51370	0.51692	-0.30578	0.50974
S2	0	0	-0.0868	0.50049	0.68493	0.62209	0.45629	0.54760	-0.46693	0.42193
S3	0	0	0	0.13705	0	0.27505	0	0	0	0
S4	-0.08238	0.40285	0.09782	0.03011	0.67288	0.54486	0.14172	0	0	0.09812
S5	0	0	0.09013	1.20583	0.61914	0.55280	0.55518	0.58936	-0.62924	0.56009
S6	0	0	0.13549	0.95859	0.71783	0.55390	0.44287	0.51752	-0.43600	0.54592
S7	0	0	0.19347	0.30112	0.14792	0.28309	0	0	0	0.09832
S8	-0.08238	0.42984	0	0	0	0	0	0	0	0
S9	0	0	0.15121	0.96164	0.67222	0.54783	0.40839	0.53011	-0.28931	0.54762
S10	0	0	0.06159	1.02187	0.66140	0.57641	0.54273	0.55363	-0.49376	0.54134
S11	0	0	0	0	0	0	0	0	0	0
S12	-0.06512	0.38027	0.16358	0.34113	0.66374	0.51014	0	0	0	0.16627
S13	0	0	0.09880	1.10723	0.61790	0.55275	0.49569	0.57030	-0.52957	0.56439
S14	0	0	0.19834	0.96178	0.69718	0.56351	0.40456	0.54475	-0.36227	0.51221
S15	0	0	0.23957	0	0.51709	0	0	0	0	0.09448

Table 5. Updated Q-table after training

Short discussion about the report of LUDO

- **Evaluation:** Individual written report based on project and evaluated according to the Danish 7-point grading scale with external co-examiner

- Assessment: individual max 11 page report; implement one AI technique, compare to a second for Ludo Game

Guideline for the report & template:

Short discussion about the report of LUDO

- Presentation — the Report (11 pages max)

Your paper will have the following sections. The % figures indicate roughly how much of your page allowance should be used for each main section; they are guidelines, not rules!

- **Abstract** (brief summary of your report) at the beginning of a manuscript
- **Introduction:** why is this interesting? why do you select this method? **(5%)**
- **Method:** give enough information for replication , e.g., algorithm as pseudo code, state space representation, inputs/outputs of your learning algorithm, etc. **(40%)** - Your own method (in details) & a second one (in short description) - The two methods can be different algorithms, e.g. GA vs. Q-learning, or they can be two instances of the same algorithm using different game representations.
- **Results:** state measurements chosen, statistical data, learning curves, etc. **(20%)**
- **Analysis & Discussion:** interpretation and analysis of the results **(25%)**
- **Conclusion:** report the main conclusions of your study **(5%)**
- **Acknowledgements (5%)**
- **References:** citation !!!