Overcoming Limitations of Deep Learning

AI Lecture Spring 2020

Yoonsuck Choe, Ph.D.

1. Professor, Department of Computer Science & Eng., Faculty of Neuroscience, Texas A&M University

Overview

- Limitations of Deep Learning
  - Practical limits
  - Fundamental limits
- Overcoming Fundamental Limits of Deep Learning
  - Meaning
  - Consciousness
  - Open-ended improvement

Part 1: Practical Limits of Deep Learning

Practical Limits of Deep Learning

- Requires massive amounts of (labeled) data.
- Long training time. Large trained models.
- Catastrophic forgetting.
- Designing good model is done mostly manually.
- Vulnerable to adversarial inputs.
- Hard to explain how it works / what it learned.

Overcoming Practical Limits of DL

Pretty much well known problems, and solutions emerging.

- Data: Active learning, Core sets, data augmentation, etc.
- Computing time: Train with reduced data. Compact models.
- Large trained models: Compression, distillation
- Catastrophic forgetting: Various approaches, not perfect yet.
- Issue of manual design: AutoML, NAS, ENAS, Evolution, etc.
- Adversarial inputs: Adversarial training, defensive distillation, ...
- Explainability: DARPA XAI effort - explanation generation, Bayesian program induction, semantic associations, etc.

Part 2: Fundamental Limits of Deep Learning

Fundamental Limits of Deep Learning

Questions from a brain and cognitive science perspective:

- Do deep neural networks have inherent meaning?
- Can deep neural networks become conscious?
- Can deep neural networks improve open-endedly?

Fundamental Limits of Deep Learning

Why are these relevant questions?

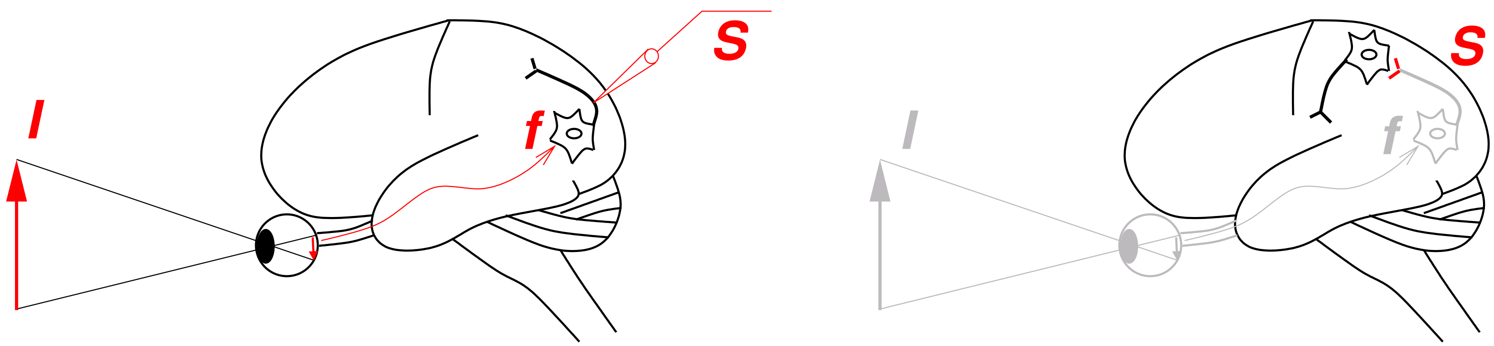
- Do deep neural networks have inherent meaning?
  - Information does not have inherent meaning, and meaningless representations lead to brittleness.
- Can deep neural networks become conscious?
  - Fundamental question of weak vs. strong AI.
- Can deep neural networks improve open-endedly?
  - Current DL excels in specific tasks, and is confined to the brain. Can it go beyond the immediate tasks, beyond the confines of its brain?

Part 2.1. Meaning

Meaning in Neural Networks

- Do neural networks possess meaning?
- Aren't they just information processors?
  - Shannon information by definition does not have meaning.
- Semantic embedding (e.g. Word2Vec) allows meaning-level manipulation.
- However, is meaning inherent to the neural network and can it be decoded from within?
- Strategy: consider how the brain does it – meaning of neural code.

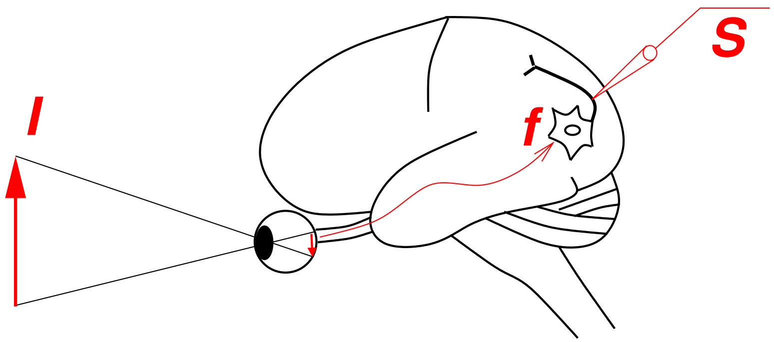
How to Understand the Neural Code?



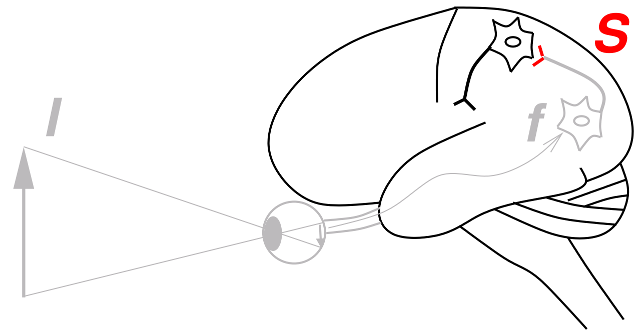
(a) From the OUTSIDE (b) From the INSIDE

- How can **we** understand the neural code? (X)

How to Understand the Neural Code?



(a) From the OUTSIDE



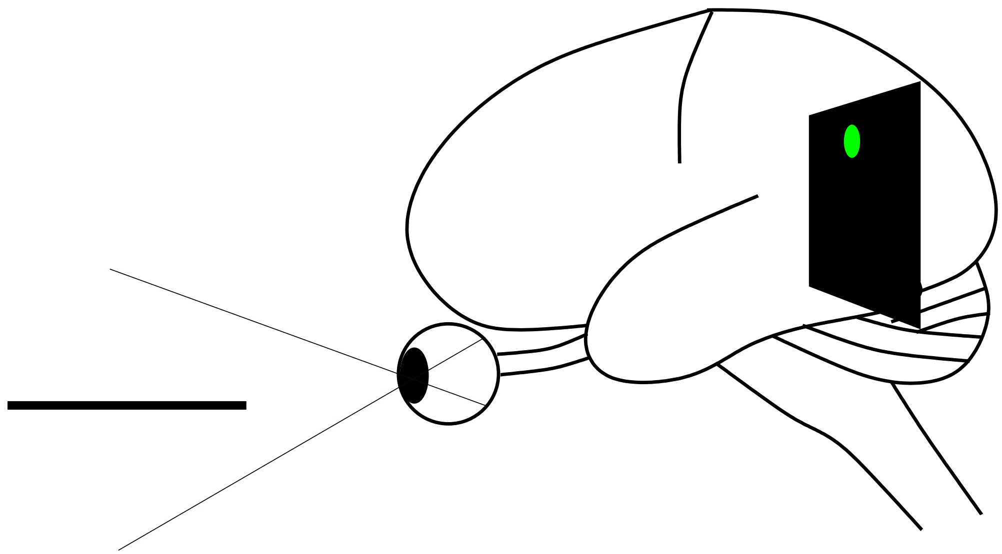
(b) From the INSIDE

- How can **we** understand the neural code? (X)
- How can **the brain itself** understand its neural code? (O)

Understanding the Neural Code, by the Brain

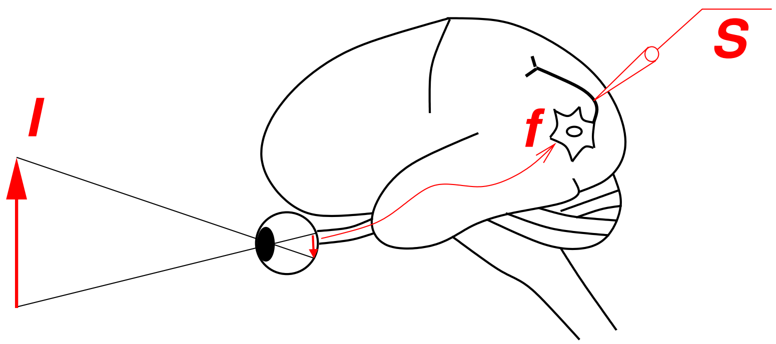
- What do these blinking lights mean?
- This is the BRAIN's perspective.
  - Seems impossible to solve!

Understanding the Neural Code, by Us

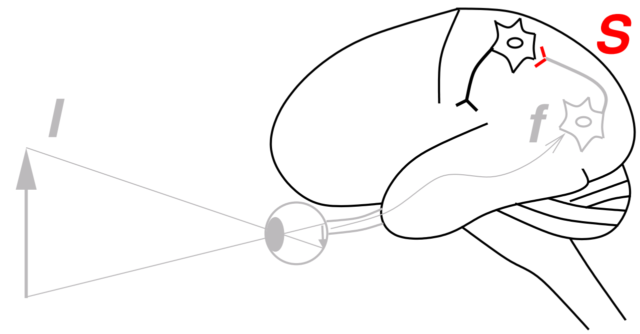


- Now we can understand the meaning.
- This is OUR perspective.
  - However, this methodology is not available to the brain!

How to Understand the Neural Code?



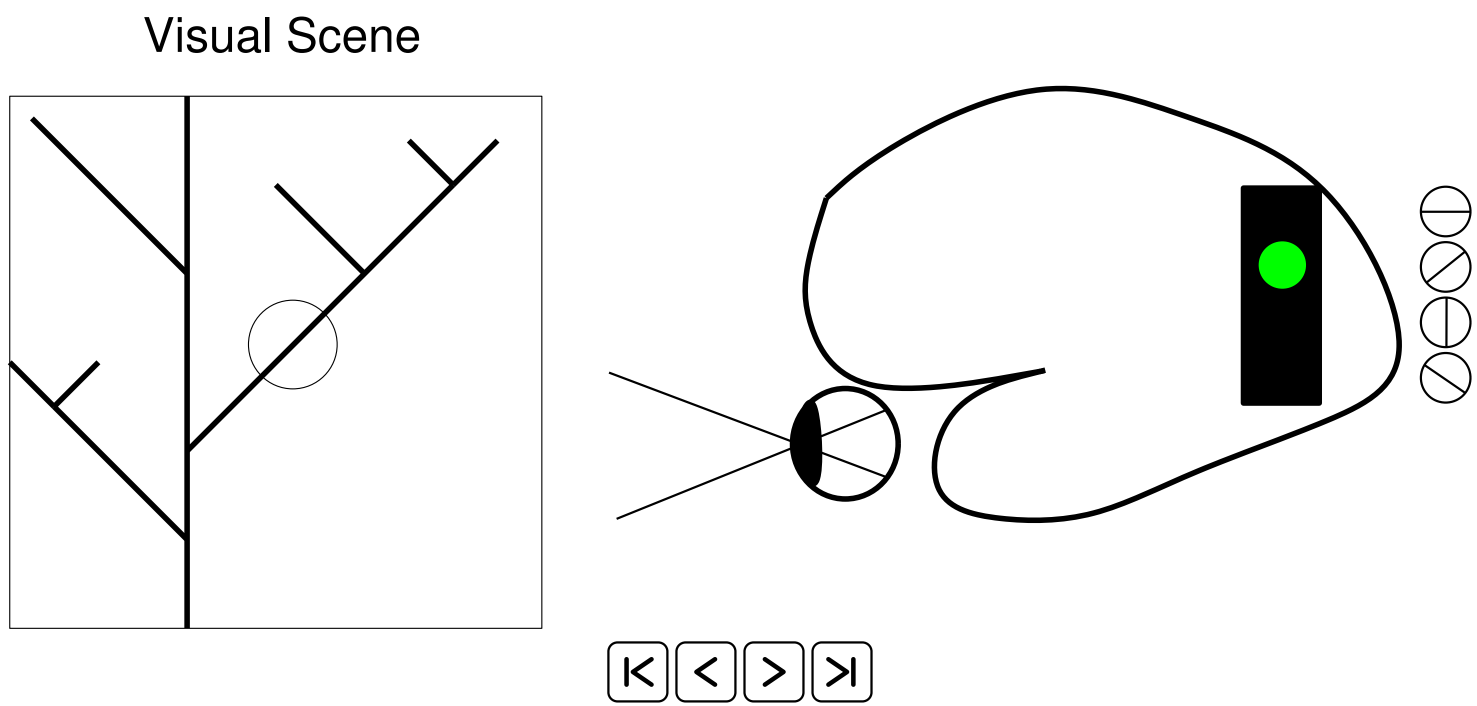
(a) From the OUTSIDE



(b) From the INSIDE

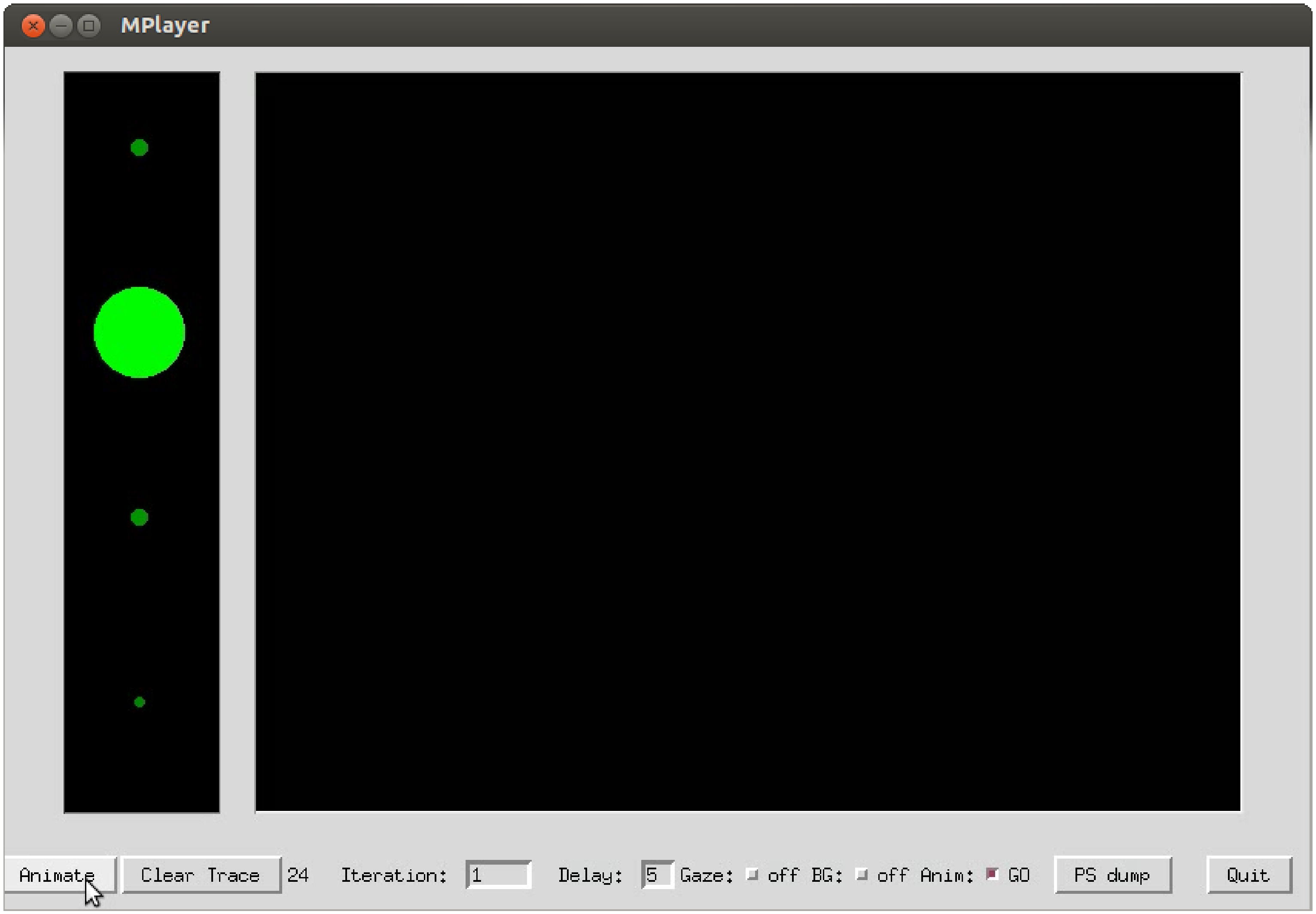
- How can **we** understand the brain? (X)
- How can **the brain itself** understand itself? (O)
  - Solution: sensorimotor learning – not obvious when wrong question asked (Choe and Smith 2006; Choe et al. 2007) Cf. Buzsaki's "Inside-Out approach". *Rhythms of the Brain* (2006).

Sensorimotor Learning to the Rescue



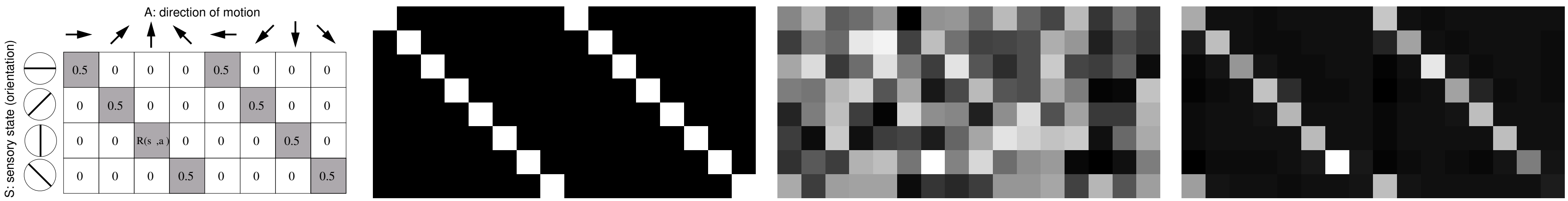
- Property of motor output that maintains internal state invariant
- Same as property of encoded sensory information.

Understanding, Inside the Brain

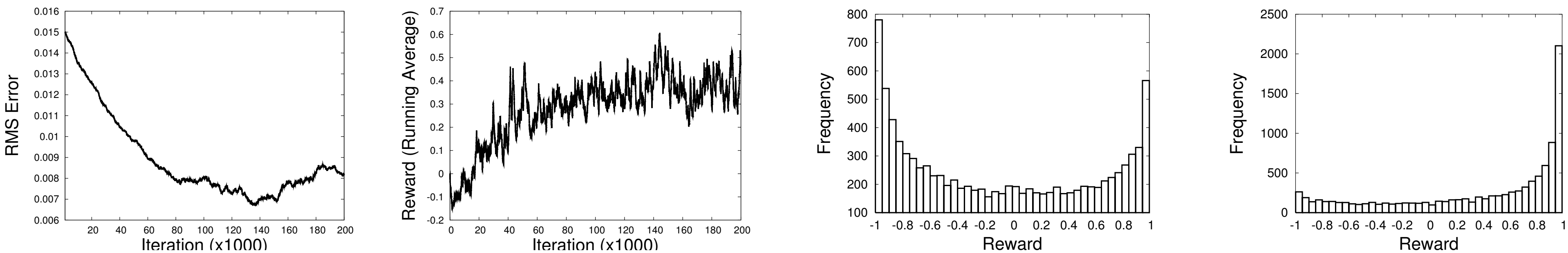


Choe et al., *Int'l J. of Humanoid Robotics* 2007

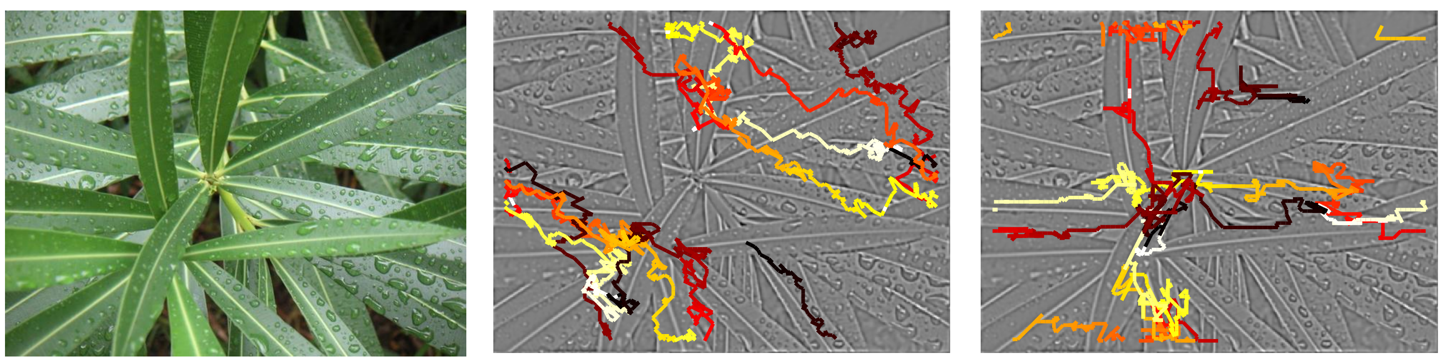
Results



(a) Reward Table $R(s, a)$ (b) Ideal R (c) Initial R (d) Final R



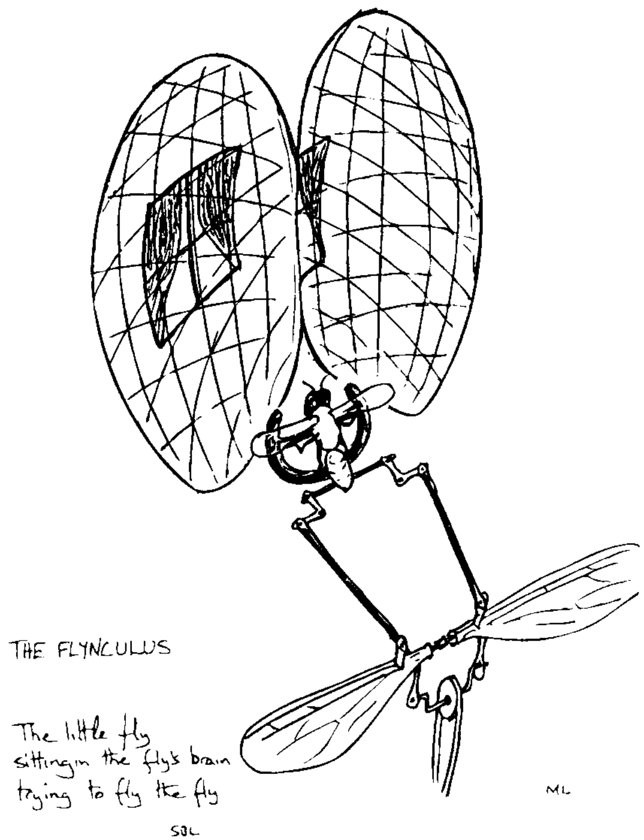
(e) RMSE (f) Avg. Reward (g) Initial R dist. (h) Final R dist.



(i) Input (j) Initial Eye Trajectory (k) Final Eye Trajectory

Choe et al., *Int'l J. of Humanoid Robotics* 2007

Applications to Optic Flow



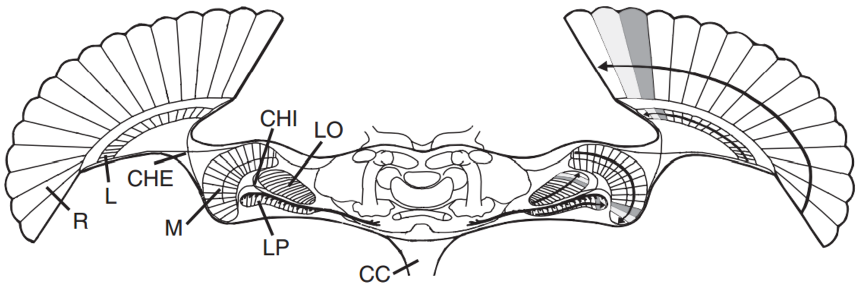
Same principle applied to the fly visual system:

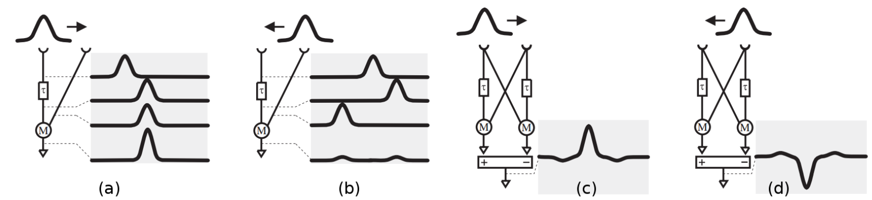
1. Fly Optic flow detectors (LPTC, Lobula Plate Tangential cells)
2. Learning the meaning of LTPC spikes: reinforcement learning based on internal state invarnance

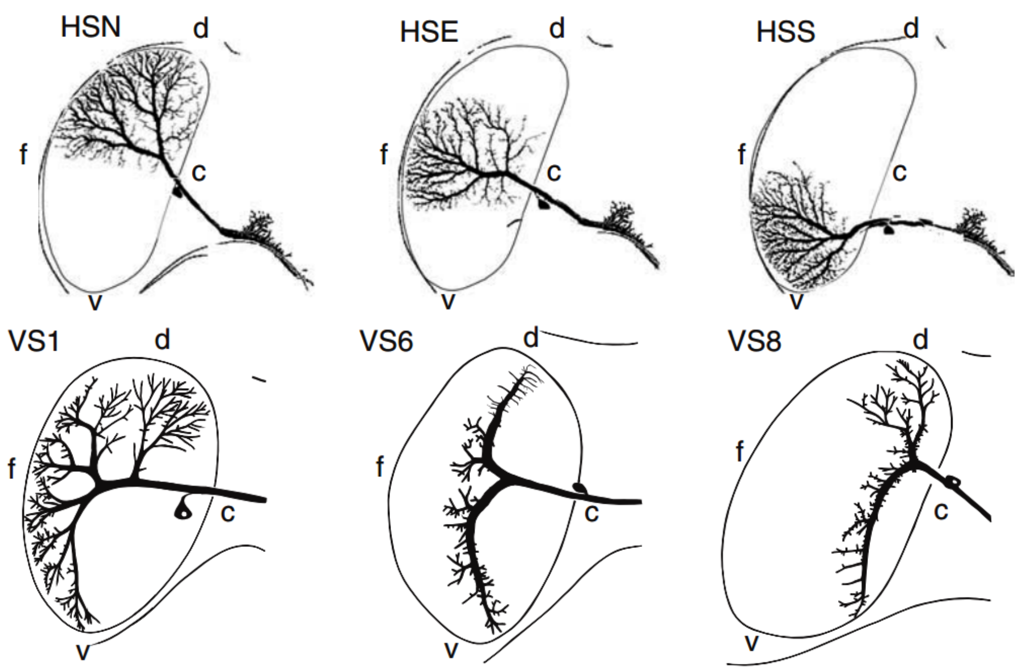
Cartoon from Rieke et al. (1997)

Parulkar and Choe IJCNN 2016 (Parulkar and Choe 2016).

Fly Visual System



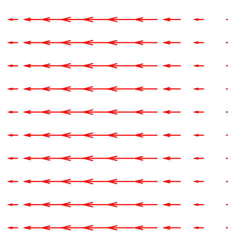




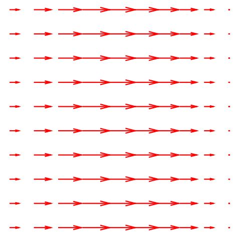
- Lamina (L) and Medulla (M):
  - Elementary Motion Detectors (EMD)
- Lobula Plate (LP) Tangential Cells (LPTCs)
  - HS and VS detect complex motion.

Borst and Egelhaaf (1989); Taylor and Krapp (2007)

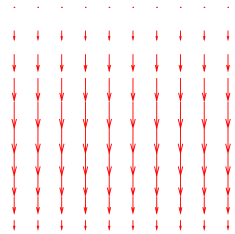
Fly Visual System Model



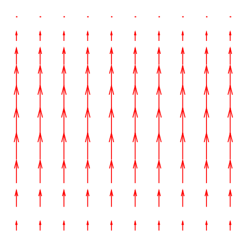
Rotation: Yaw Right to Left (RYRL)



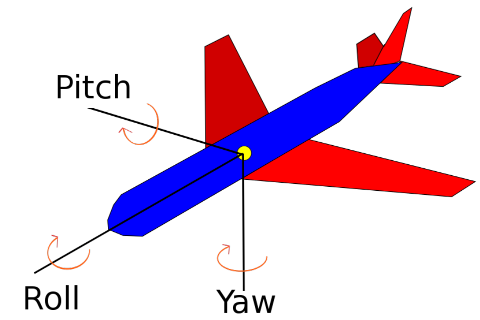
Rotation: Yaw Left to Right (RYLR)

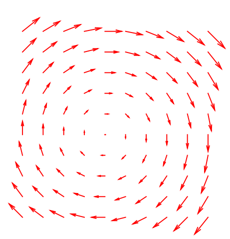


Rotation: Pitch Up to Down (RPUD)

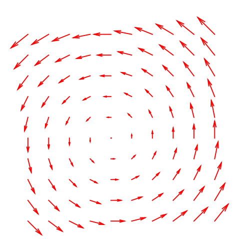


Rotation: Pitch Down to Up (RPDU)

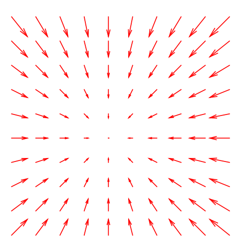




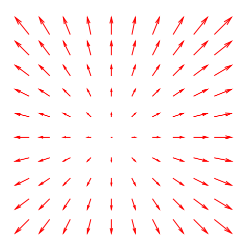
Rotation: Roll Clockwise (RRCL)



Rotation: Roll X-clockwise (RRAC)



Translation: Radiate in (TLRI)



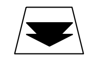
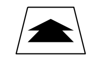
Translation: Radiate out (TLRO)

- Initial optic flow computation: Lucas and Kanade (1981) method.
- HS: simple horizontal motion; VS: matched filter (roll and pitch [Krapp 2000])

Learning the Reward Table $R(s, a)$

Direction of Motion (a)

Optical Flow States (s)

$R(s,a)$	$\rightarrow$	$\leftarrow$	$\uparrow$	$\downarrow$	$\circlearrowleft$	$\circlearrowright$		
RYRL	1	0	0	0	0	0	0	0
RYLR	0	1	0	0	0	0	0	0
RPUD	0	0	1	0	0	0	0	0
RPDU	0	0	0	1	0	0	0	0
RRCL	0	0	0	0	1	0	0	0
RRAC	0	0	0	0	0	1	0	0
TLRI	0	0	0	0	0	0	1	0
TLRO	0	0	0	0	0	0	0	1

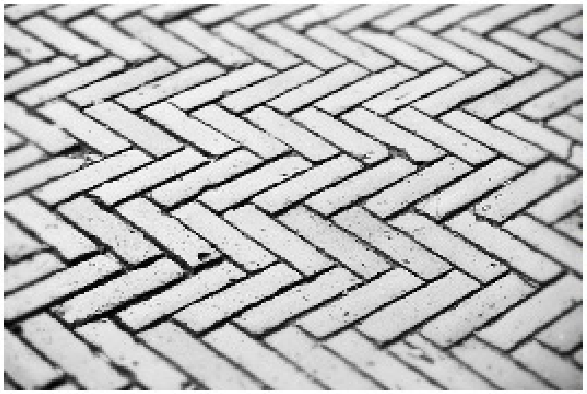
- Action is selected based on $P(a|s) = R(s, a)$.
- Learning (α: learning rate):

$$R_{t+1}(s_t, a_t) = R_t(s_t, a_t) + \alpha\rho_{t+1}, \text{ where}$$

$$\rho_{t+1} = 1/\sqrt{\sum_i (r_{t+1,i} - r_{t,i})^2}$$

Finally, $R(s, a)$ is normalized over all a.

Experiments and Results: Input



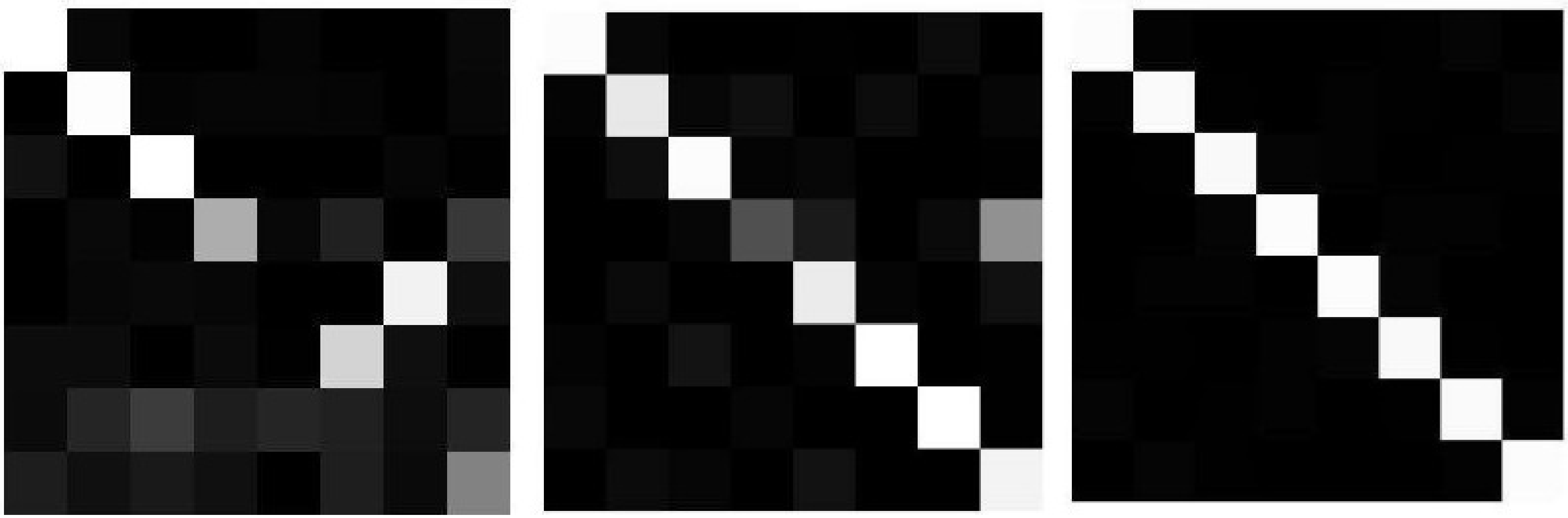
(a) Synthetic

(b) Natural 1

(c) Natural 2

- Model fly trained on three different inputs above.

Experiments and Results: Learned R



(a) Synthetic (b) Natural 1 (c) Natural 2

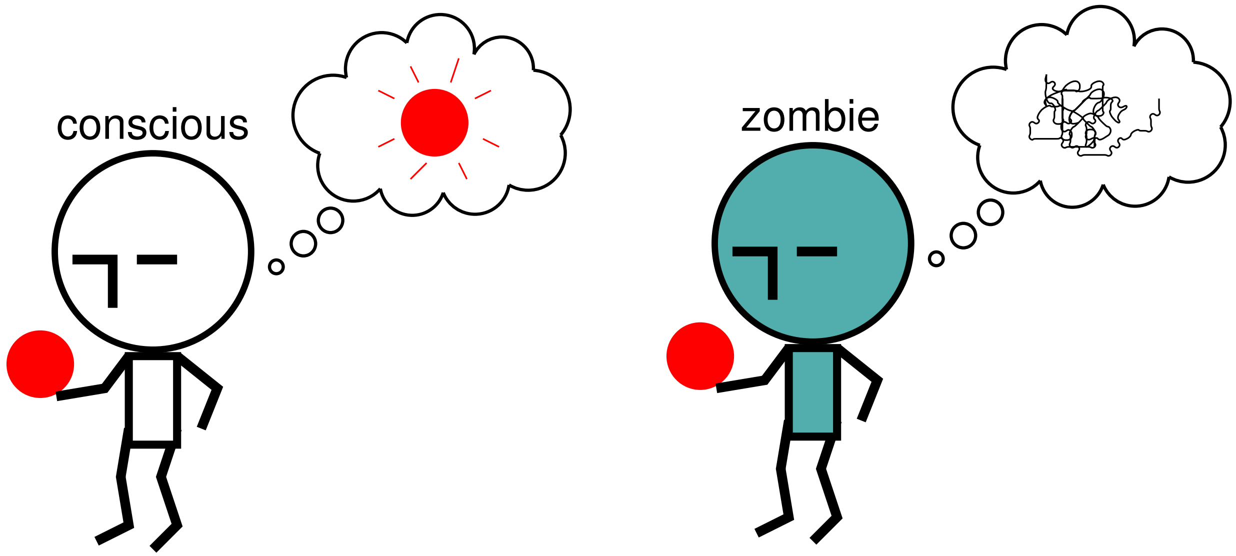
- All three inputs lead to near-ideal $R(s, a)$.
- Given a certain internal state, action that has the same encoded property as that state is generated.

Summary: Meaning

- Motor exploration is key to autonomous grounding of meaning.
- Meaning is in large part based on motor primitives, not perceptual features.
- Very simple criterion of internal state invariance can be used to learn the sensorimotor meaning.
- Implications on deep learning: Purely perception-based meaning is untenable. Need the network to interact with the environment.

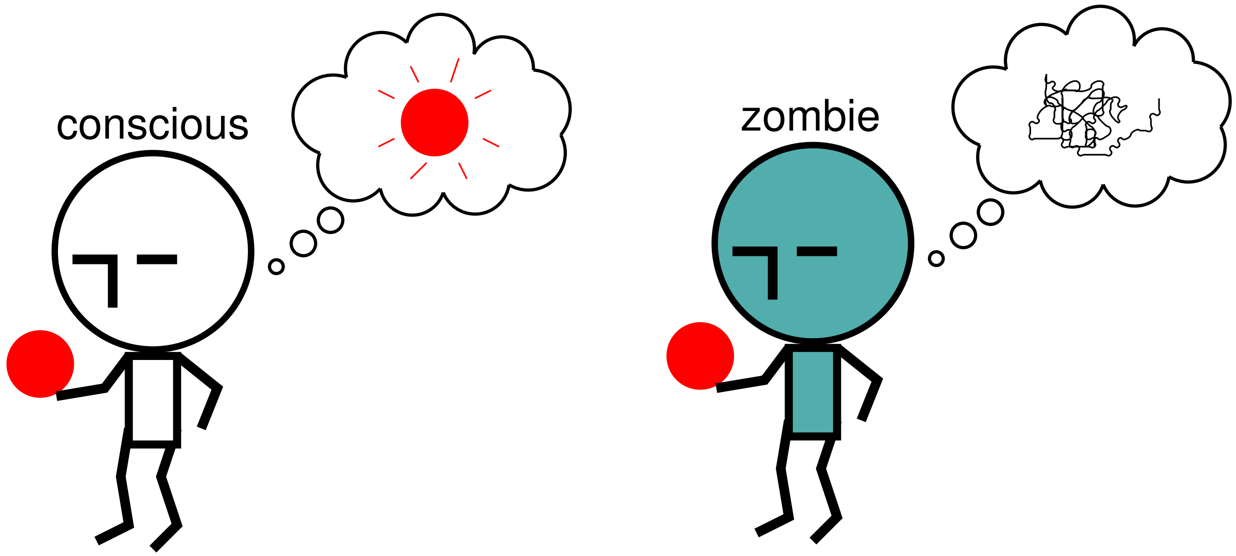
Part 2.2. Consciousness

The Question of Consciousness



- How did consciousness evolve? (X)

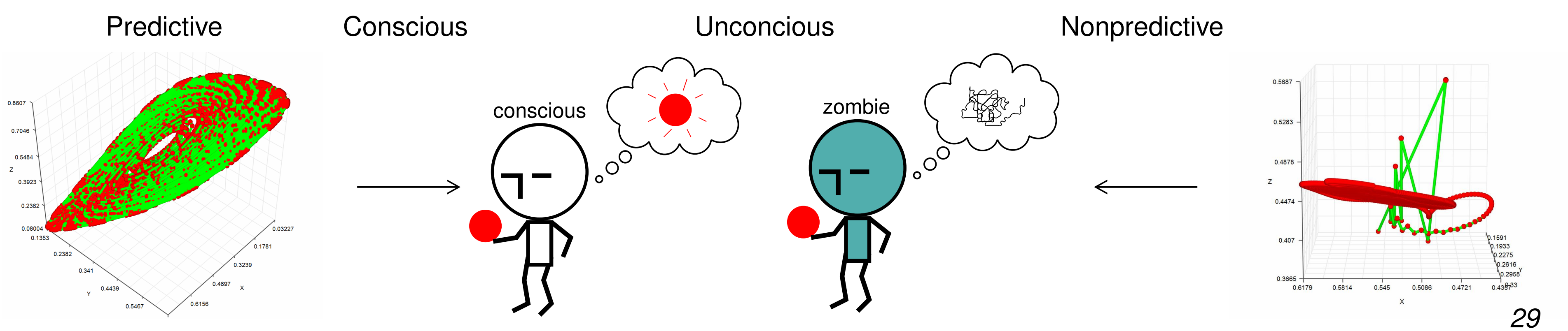
The Question of Consciousness



- How did consciousness evolve? (X)
- How did the **necessary conditions** of consciousness evolve? (O)

How did Consciousness Evolve?

- How did consciousness evolve? (X)
- How did the **necessary conditions** of consciousness evolve? (O)
  - Former is subjective, latter is objective.
  - Predictive dynamics found to be key (Choe et al. 2012)



Necessary Condition for Consciousness

- Are there future events that are 100% predictable?

Necessary Condition for Consciousness

- Are there future events that are 100% predictable?
- What if I say there are such events?

Necessary Condition for Consciousness

- Are there future events that are 100% predictable?
- What if I say there are such events?
- I will clap my hands in the next 5 seconds.

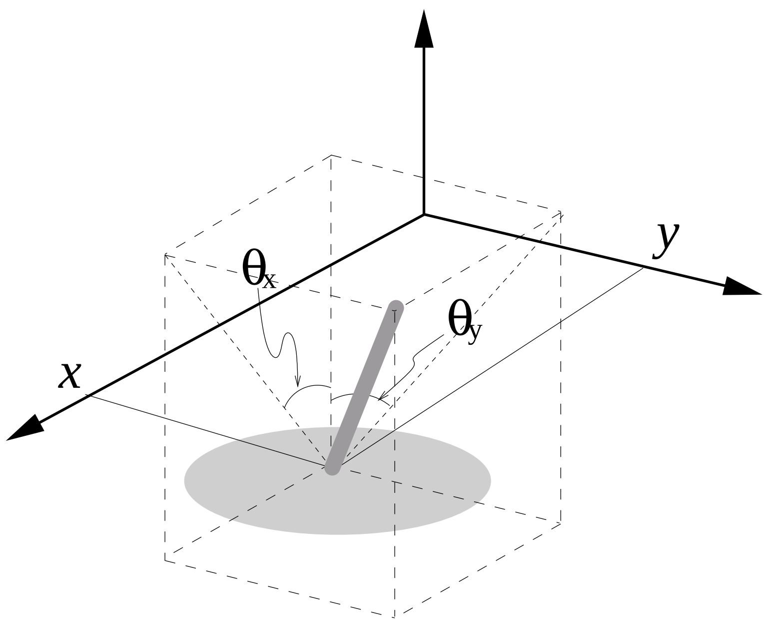
Necessary Condition for Consciousness

- Are there future events that are 100% predictable?
- What if I say there are such events?
- I will clap my hands in the next 5 seconds.
- "My" actions are 100% predictable, and this (authorship) is a key property of the self, the subject of consciousness.

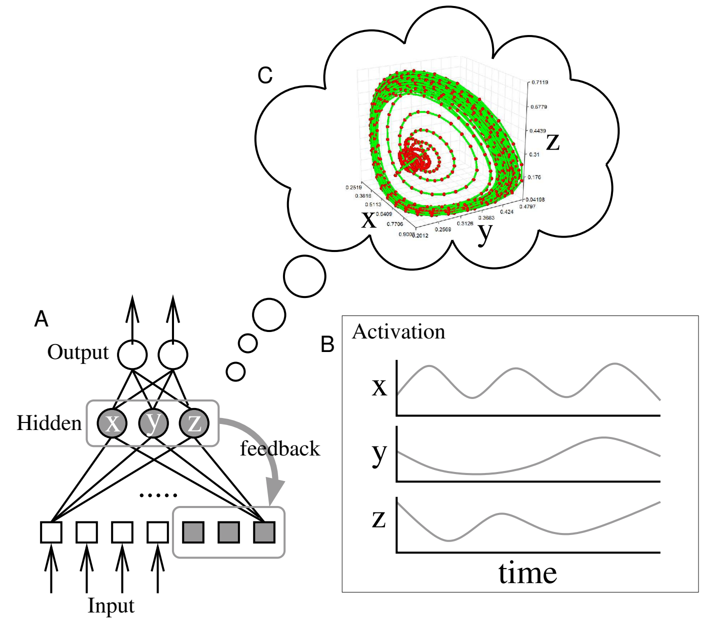
Necessary Condition for Consciousness

- Are there future events that are 100% predictable?
- What if I say there are such events?
- I will clap my hands in the next 5 seconds.
- “My” actions are 100% predictable, and this (authorship) is a key property of the self, the subject of consciousness.
- Thus, the brain dynamics have to be predictable.

Could the Necessary Condition Evolve?



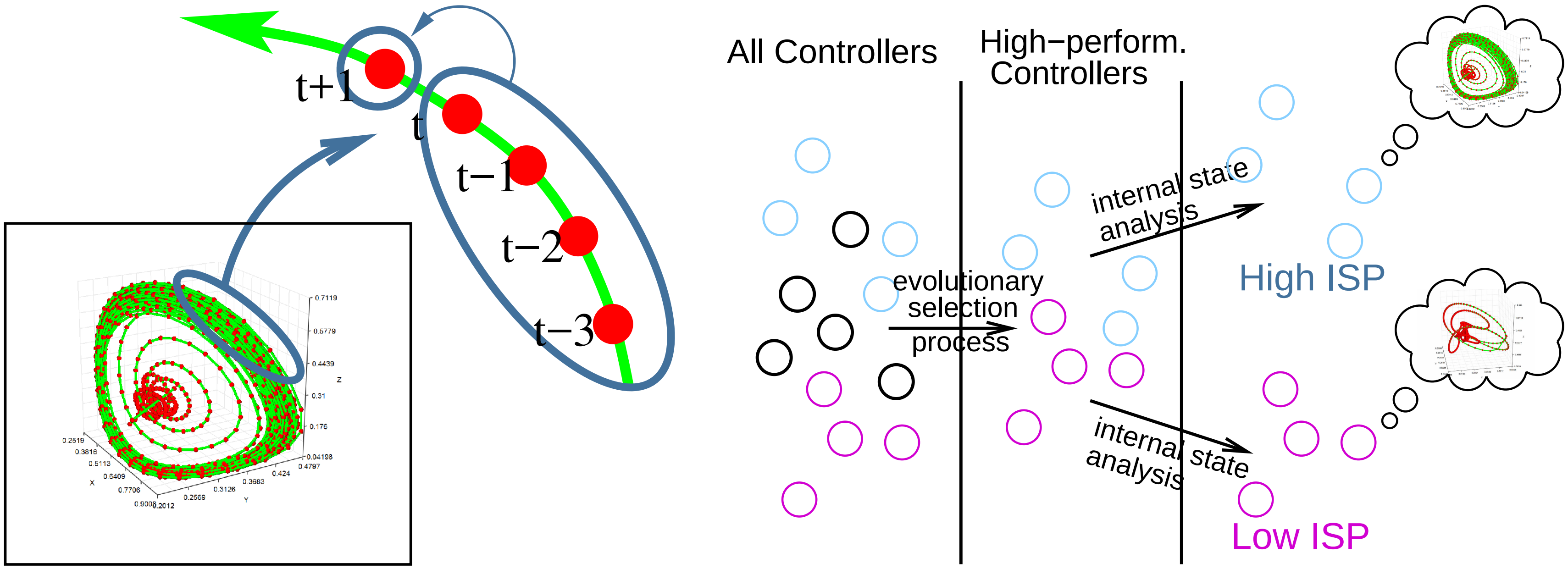
(a) Task



(b) Controller

- Pole balancing task.
- Evolved neural network controller.

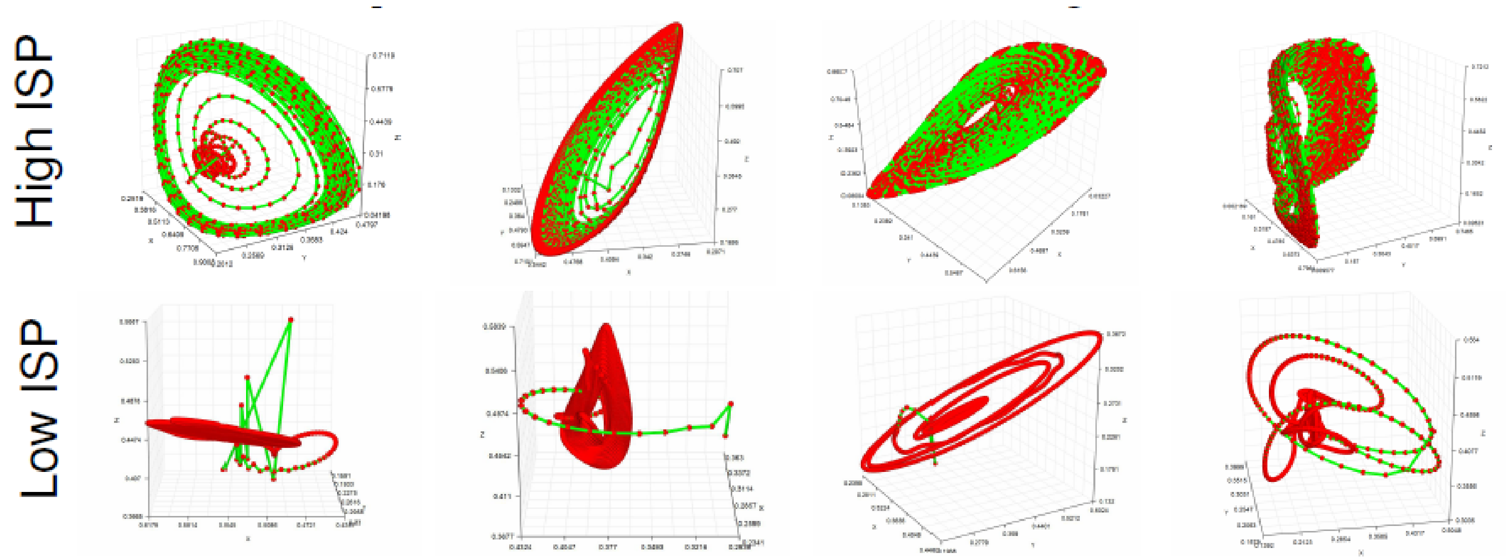
Could the Necessary Condition Evolve?



(a) Measure ISP (b) Overview

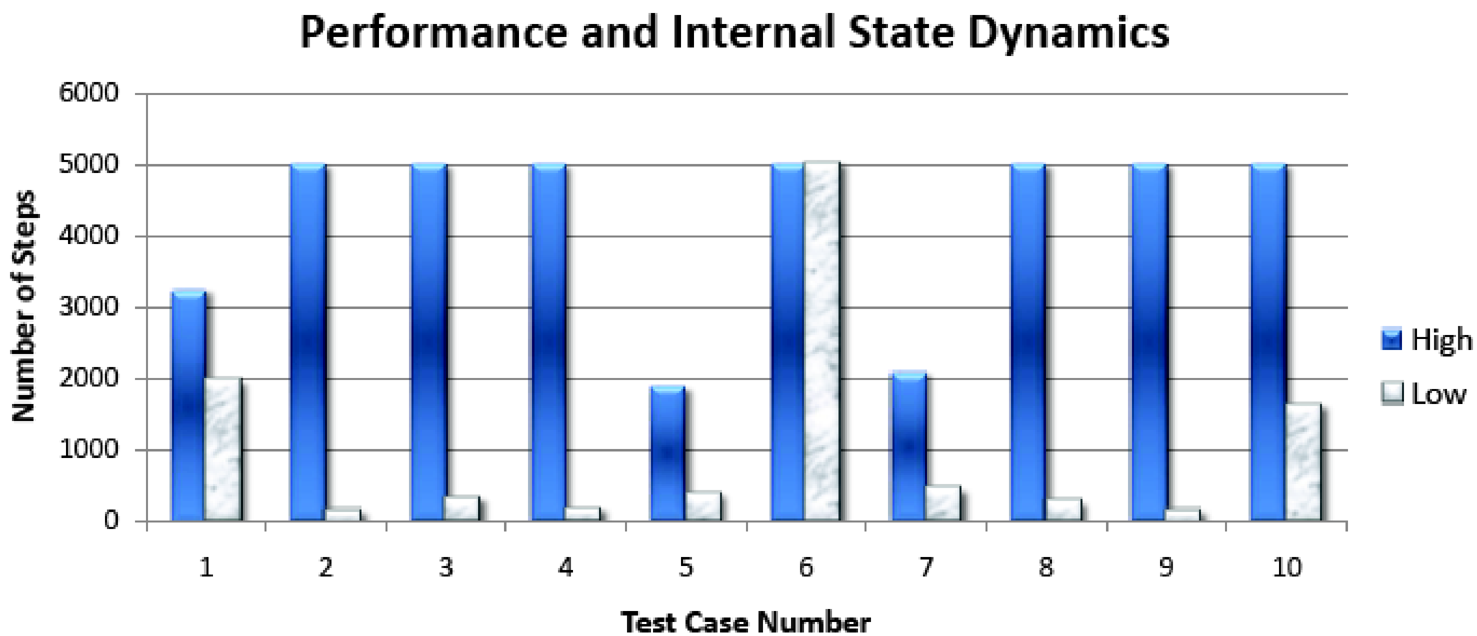
- Measure predictability of internal state dynamics.
- Compare internal dynamics of equally sucessful ones.

Predictable vs. Unpredictable Internal Dyn.



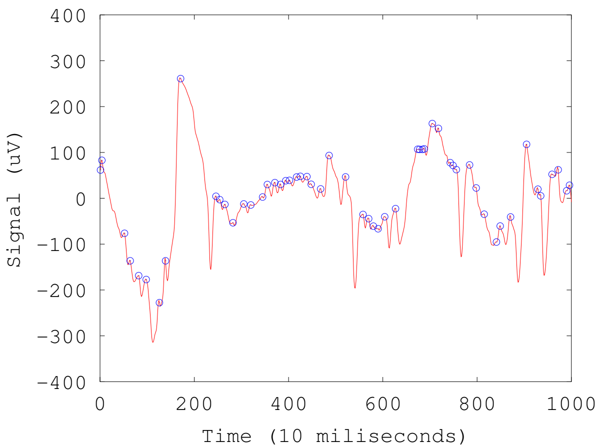
- Internal dynamics of a simple pole-balancing controller neural network (Kwon and Choe 2008)

Predictable vs. Unpredictable Internal Dyn.

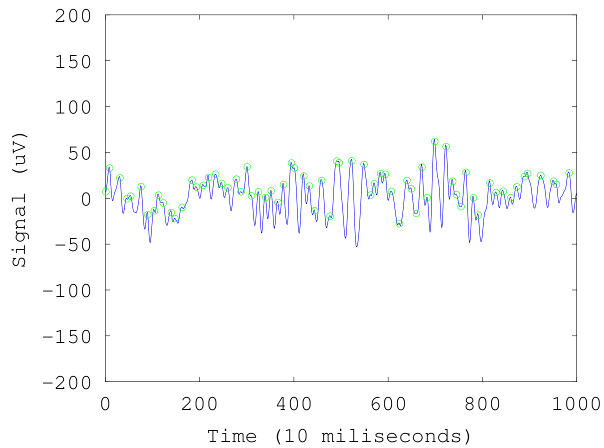


- Performance in controllers with high vs. low internal state predictability (Kwon and Choe 2008)
- Controllers with high ISP better fit in changing environment: Necessary condition can evolve!

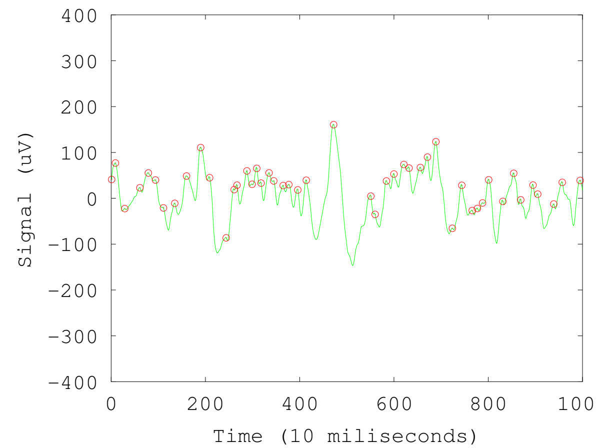
Analysis of Real EEG Data



Awake



REM Sleep

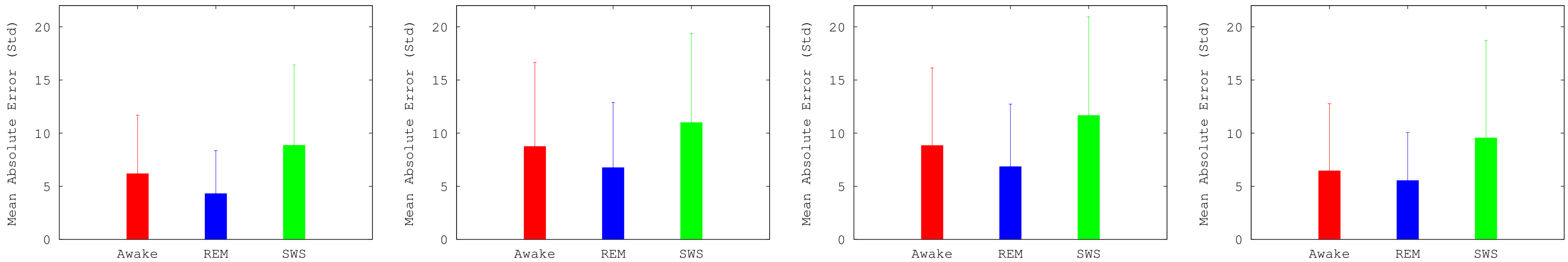


Slow Wave Sleep

- Awake, REM sleep, and Slow-wave sleep EEG data.
- Inter-Peak Interval (IPI) predictability.

Yoo et al. *Frontiers in Neurorobotics* 2013.

Real EEG Data: Prediction Error



Subject 1 Subject 2 Subject 3 Subject 4

- Awake and REM more predictable than SWS.
- All differences were significant ($p < 10^{-6}$) except for subject 4, Awake vs. REM.

Yoo et al. *Frontiers in Neurorobotics* 2013.

Summary: Consciousness

- Internal dynamics of neural networks can relate to subjective phenomena.
- Predictable internal dynamics may be the precursor of consciousness.
- Such predictable dynamics can facilitate intrinsic understanding within the neural network.
- Implications on deep learing:
  - Need to look at internal neural dynamics.
  - Need to explore predictive properties.

Part 2.3. Open-Ended Improvement

DL Can't Improve Open-Endedly

- Current DL excels only in very specific tasks.
  - Tasks and (kind of) data are fixed.
  - What it can learn is limited by the task itself.
- Current DL is confined to its brain
  - Neural network weights
  - Optionally external memory, but strongly integrated with the neural network.

Open-Ended Improvement

Possible directions:

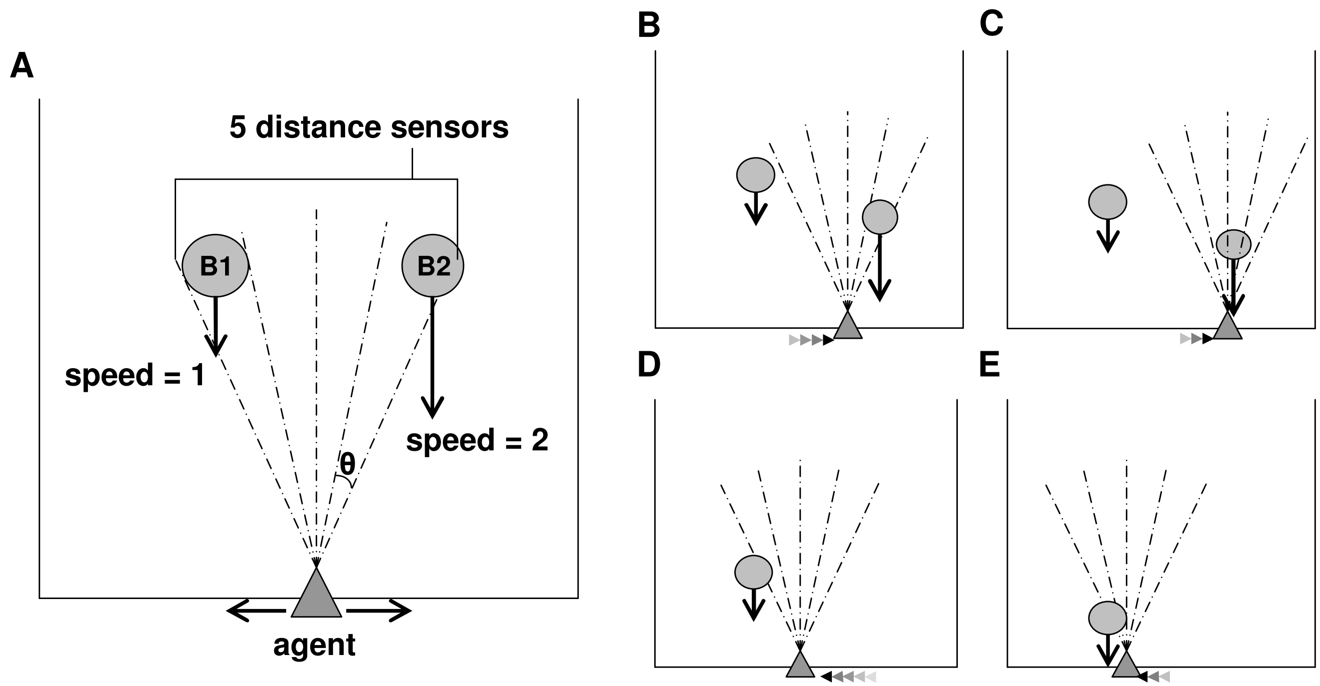
- Use of external medium, beyond the bounds of the brain
  - Stigmergy
- Co-evolution of brain and tools
  - New tools enable new problem definitions.

Using the External World as Memory

Is it possible for a feedforward network to show memory capacity?

- What would be a minimal augmentation?
- **Idea:** allow **material interaction**, dropping and detecting of external markers.

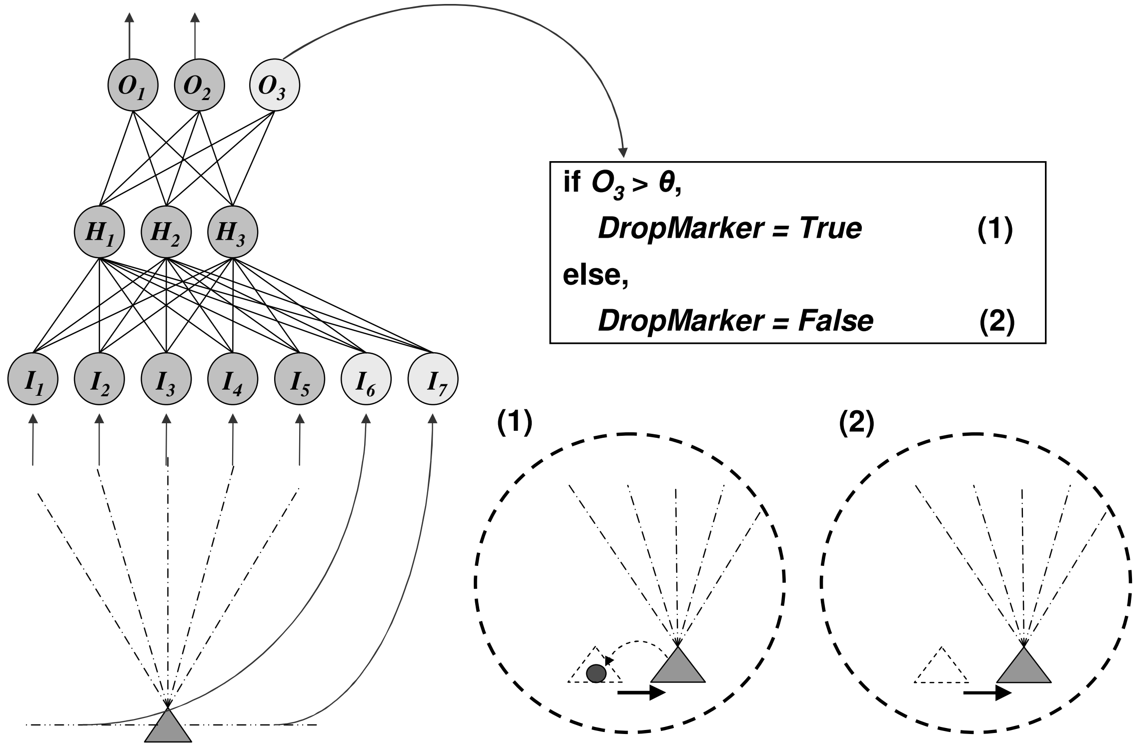
Memory Task: Catch the Balls



cf. Beer (2000); **?**

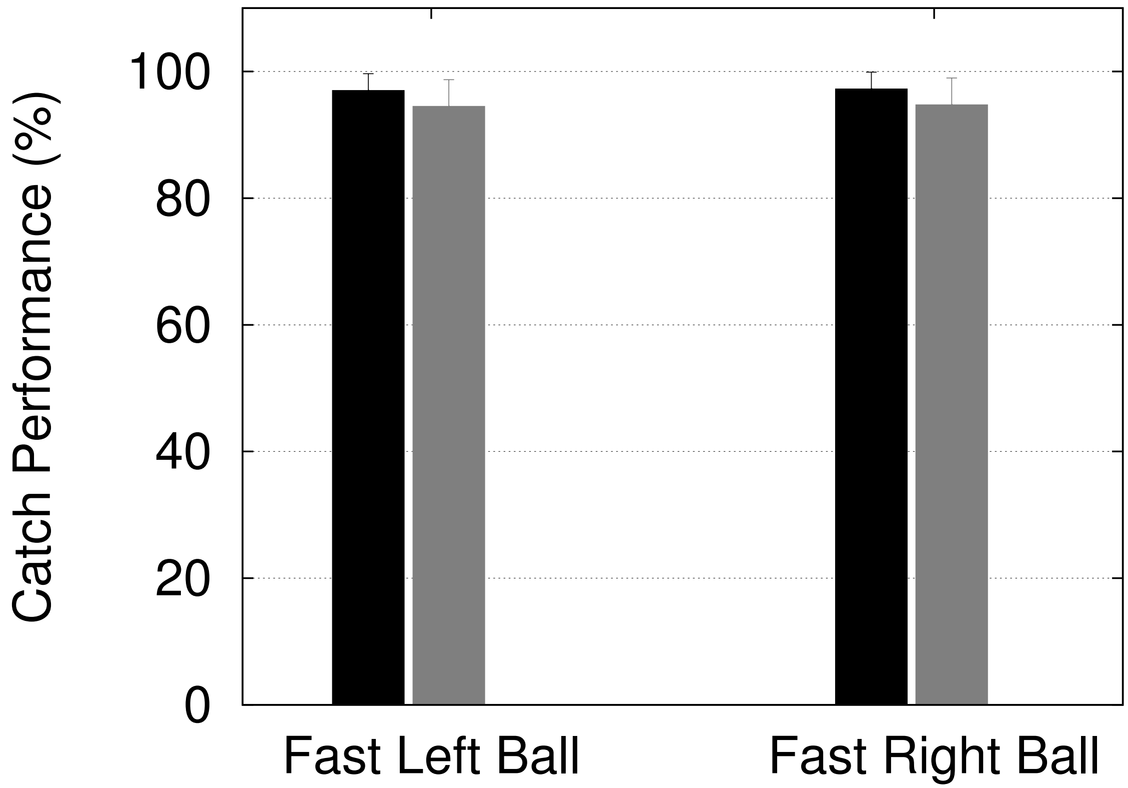
- Agent with range sensors move left/right.
- Must catch both falling balls.
- Memory needed when ball goes out of view.

Feedforward Net + Dropper/Detector



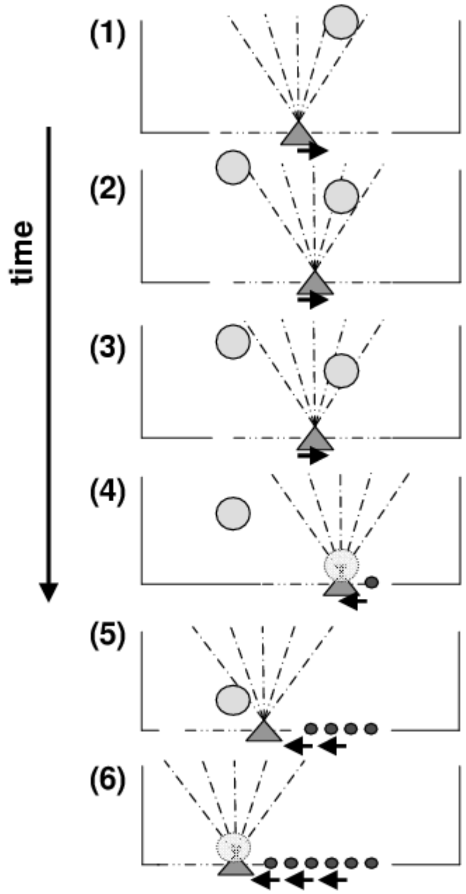
- Feedforward network plus:
  - Extra output to **drop** markers.
  - Extra sensors to **detect** the markers.
- Neuroevolution used for training the weights.

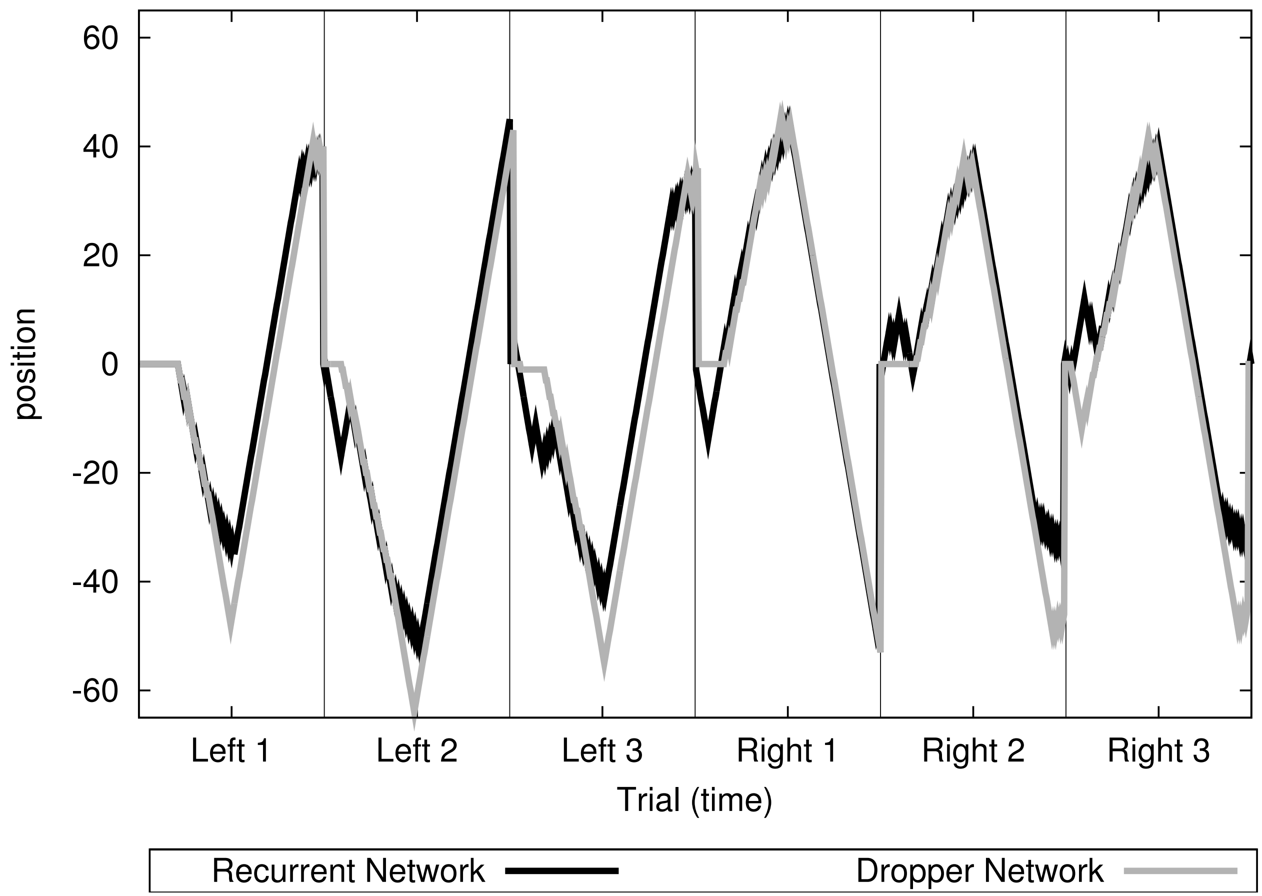
Results (vs. Recurrent Networks)



- No difference in performance between dropper/detector net (gray) and recurrent network (black).

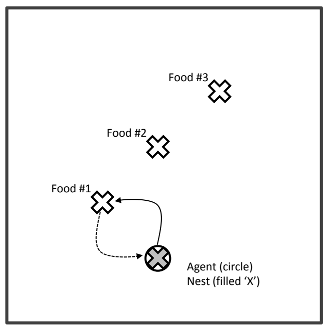
Behavior



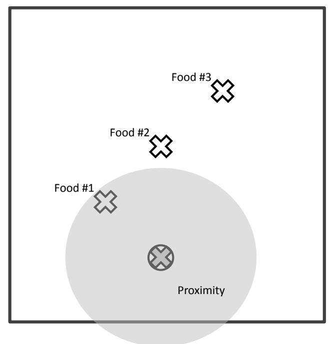


- Slight overshoot and drop the marker.
- Subsequent move **repelled** away from the marker.

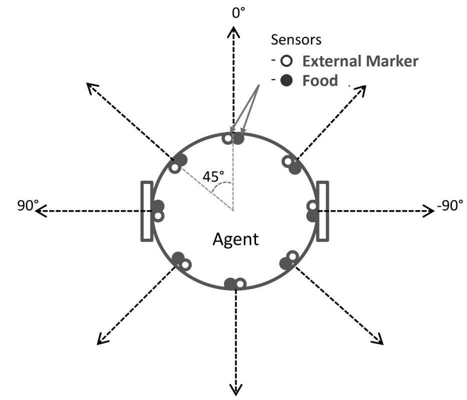
Task 2: Foraging in 2D



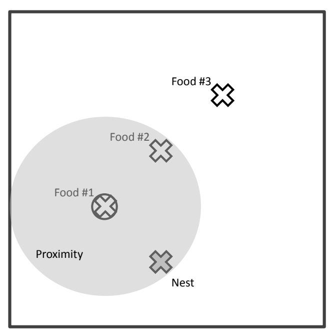
A. Task Setup



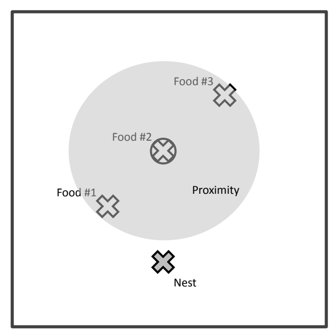
B. Agent at Initial Location



Agent



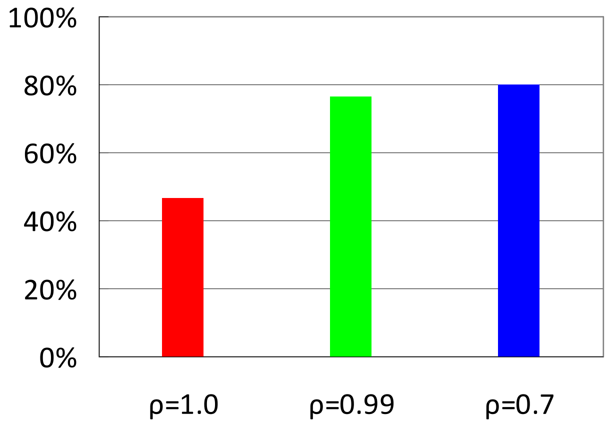
C. Agent Getting Food♯1



D. Agent Getting Food♯2

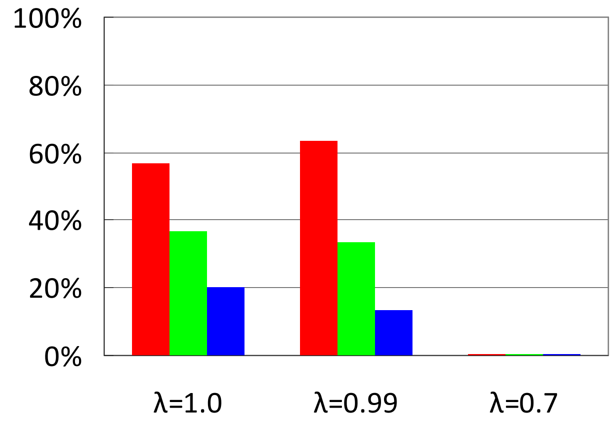
- 2D foraging task requiring memory.
- Agent w/ directional food/nest sensor (limited range).

Foraging: Results



FFW+Dropper

$\rho =$ evaporation rate



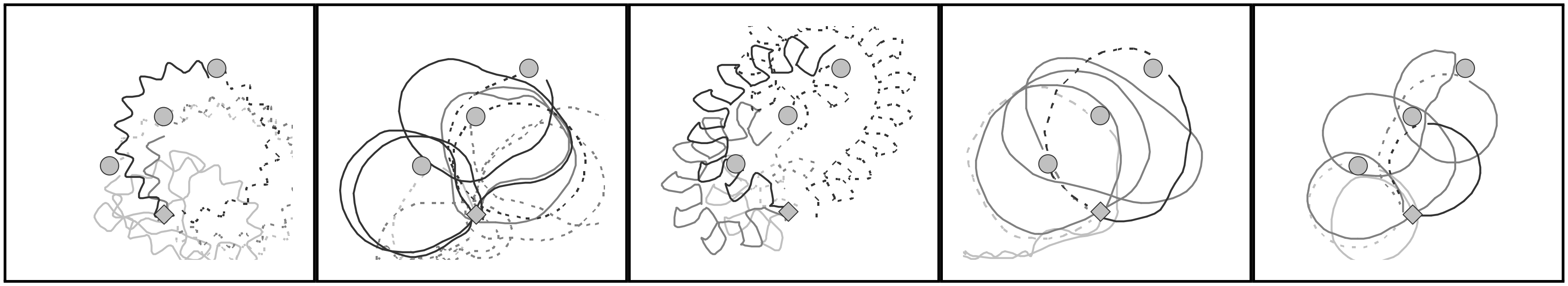
RNN

$\lambda =$ discount factor#

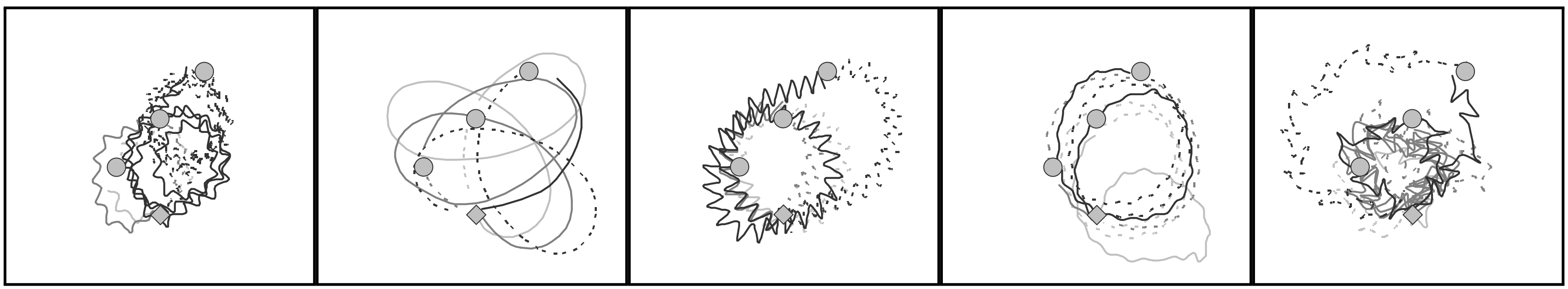
stack height: red=5, green=10, blue=10

- Comparison of FFW-net+Dropper vs. RNN (Elman tower) success rate.

Foraging Behavior: RNN

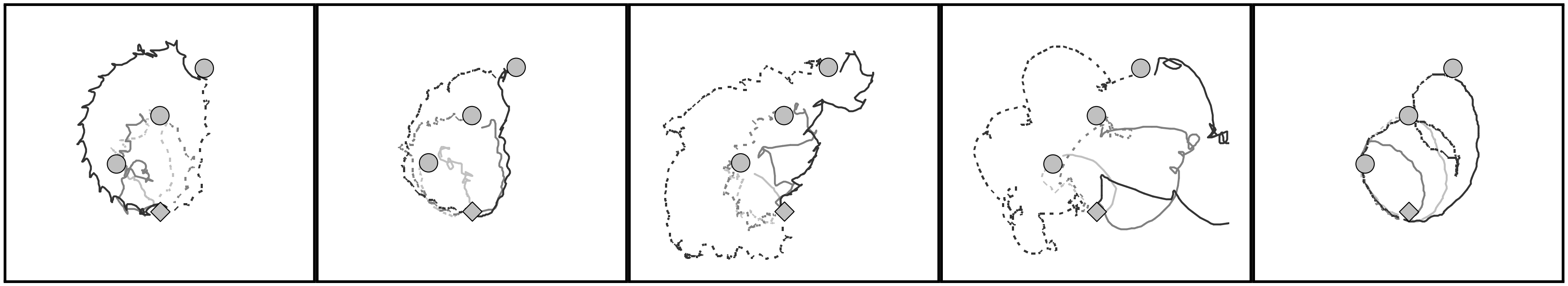


A. Trajectories of Successful Recurrent Agents with $\lambda = 1.0$

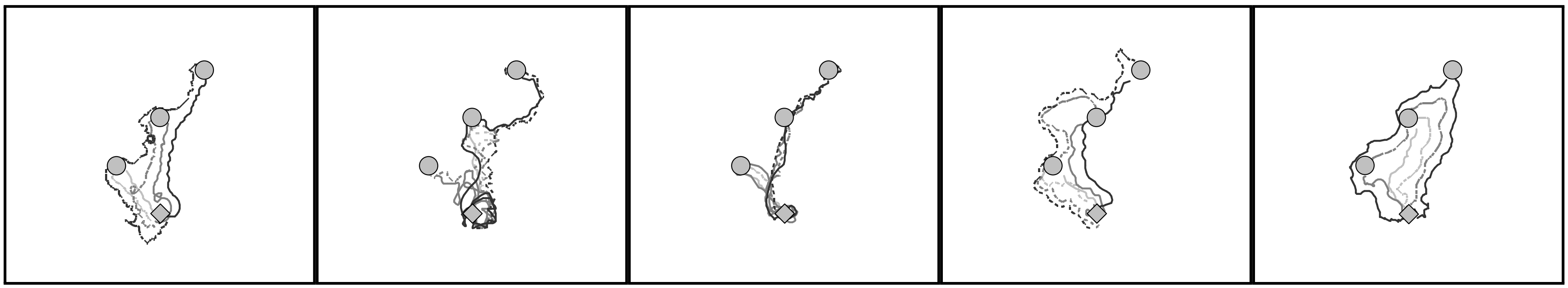


B. Trajectories of Successful Recurrent Agents with $\lambda = 0.99$

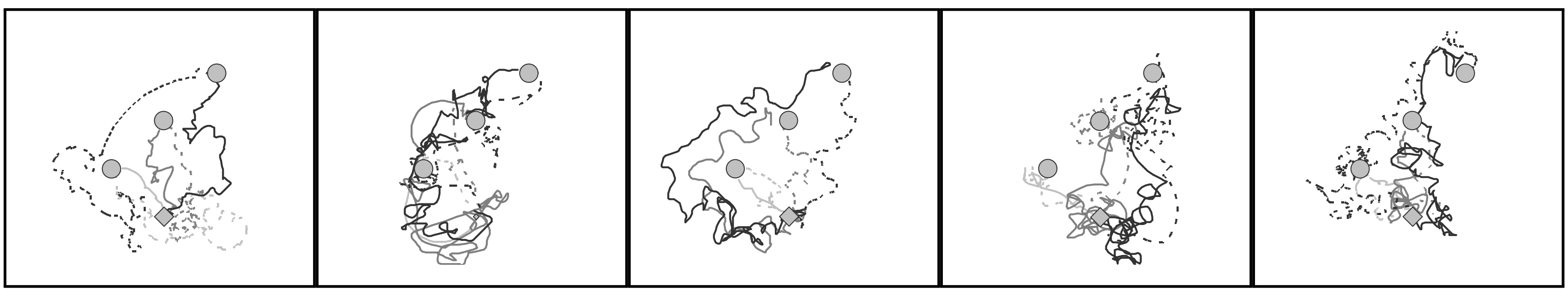
Foraging Behavior: FFW+Dropper



A. Trajectories of Successful Dropper Agents with $\rho = 1.0$



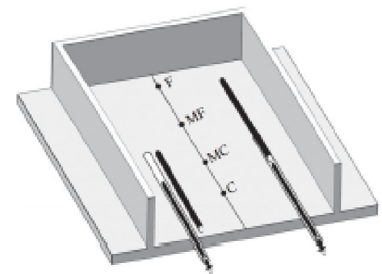
B. Trajectories of Successful Dropper Agents with $\rho = 0.99$



C. Trajectories of Successful Dropper Agents with $\rho = 0.7$

Tool Construction and Use

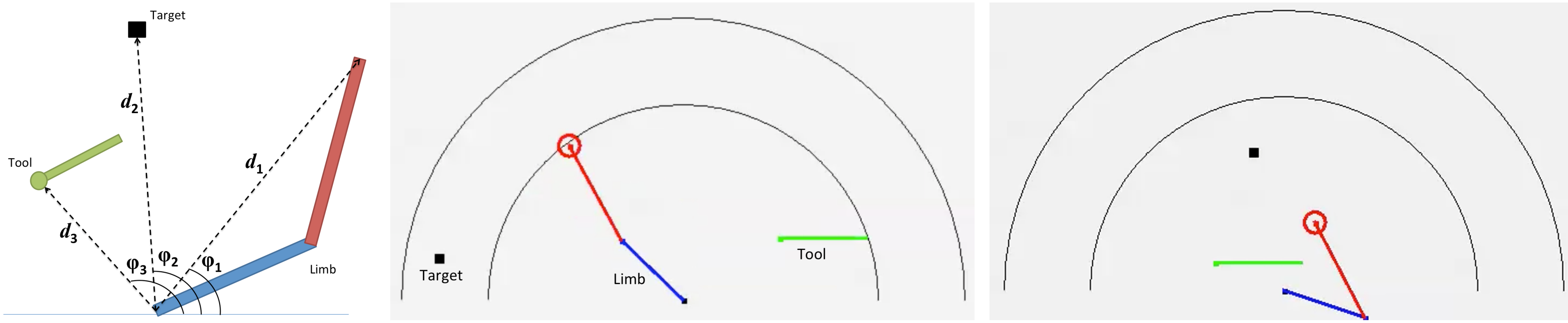
(a) Tool construction behavior



(b) Composite tool

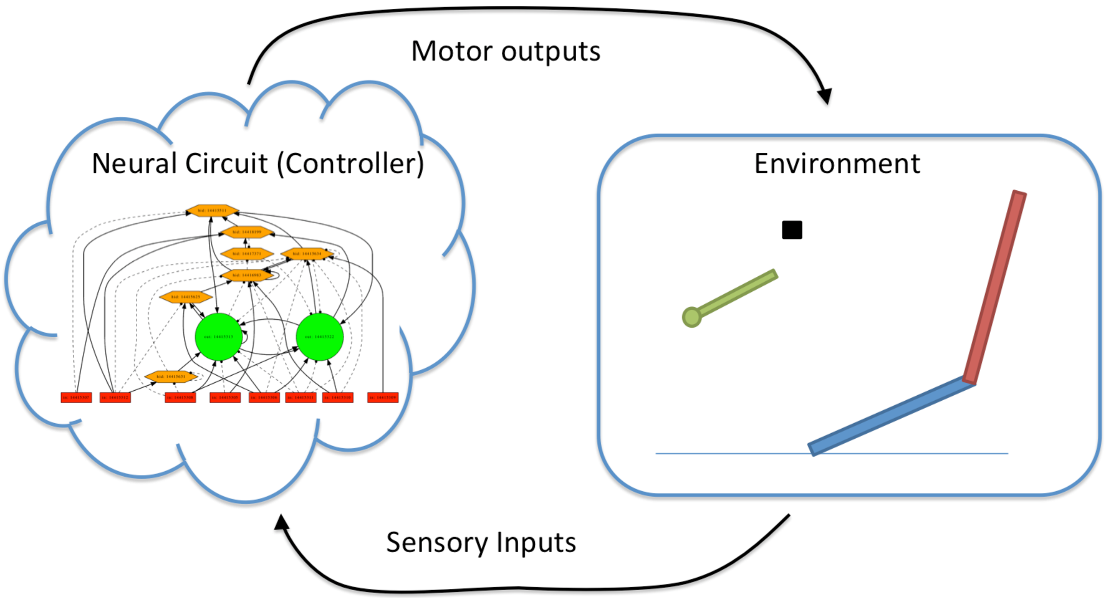
- Animals have shown limited tool construction capability in lab environment.
- Why care for tool construction?
  - Tool construction and use as a measure of intelligence (St. Amant and Wood 2005; Choe et al. 2015).
  - Agent-tool co-evolution (only observed in humans!).

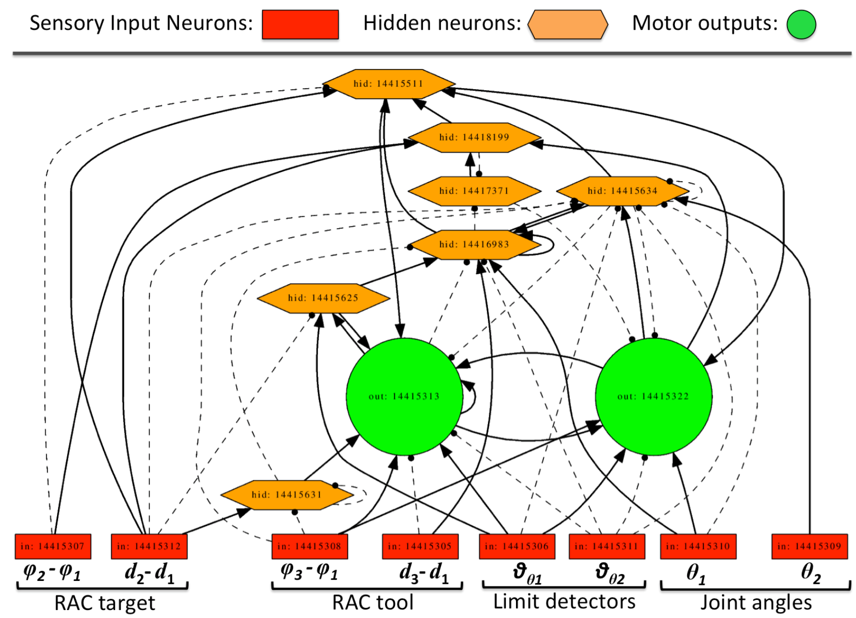
Task: Reaching Close/Far Targets



- Sensors: Joint angles/limits, angle/distance to target/tool.
- Motor: Control joint angle to reach target or tool (stick).
- Targets could be within/beyond reach.
- Reaching tool extends limb (automatic).

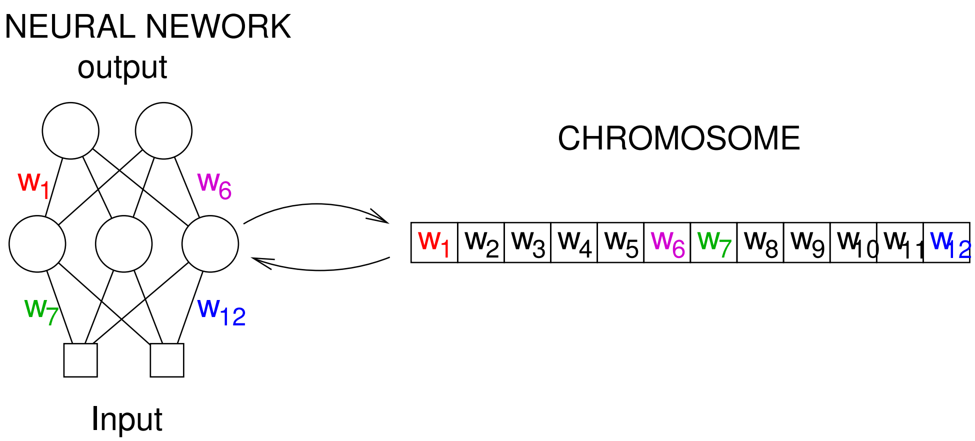
Task: Reaching Close/Far Targets

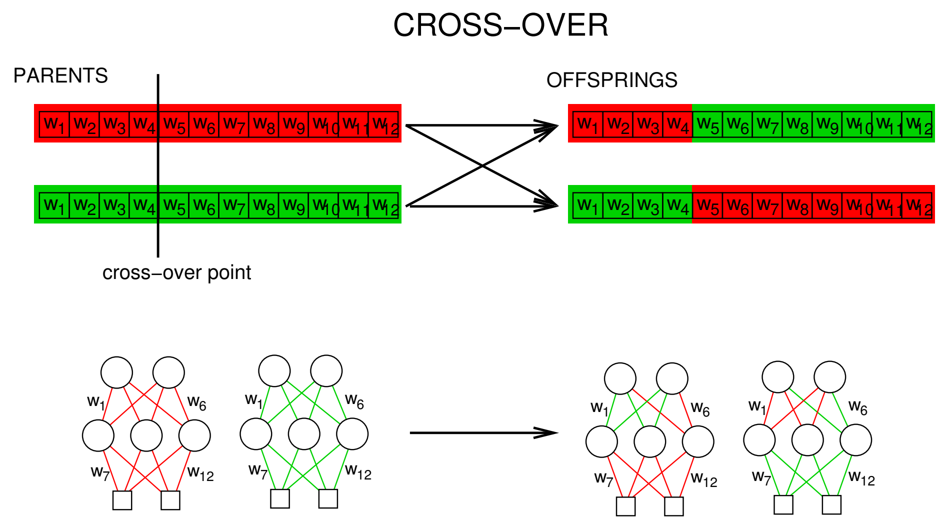




- Sensors: Joint angles/limits, angle/distance to target/tool.
- Motor: Control joint angle to reach target or tool (stick).

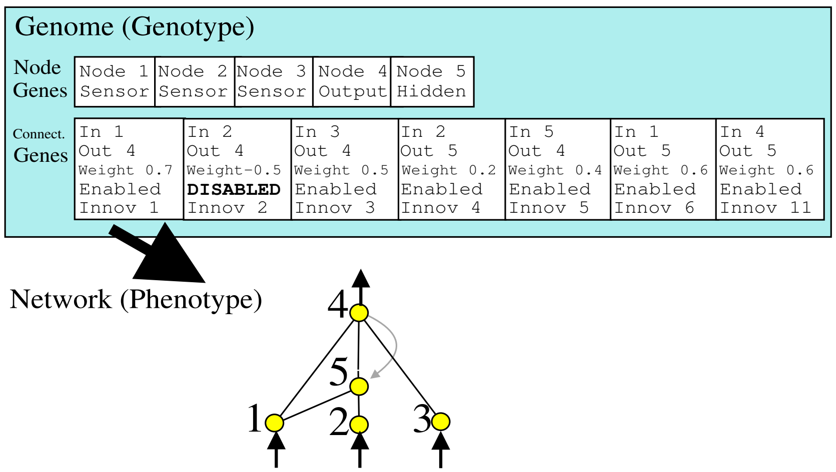
Evolving Neural Network Controllers

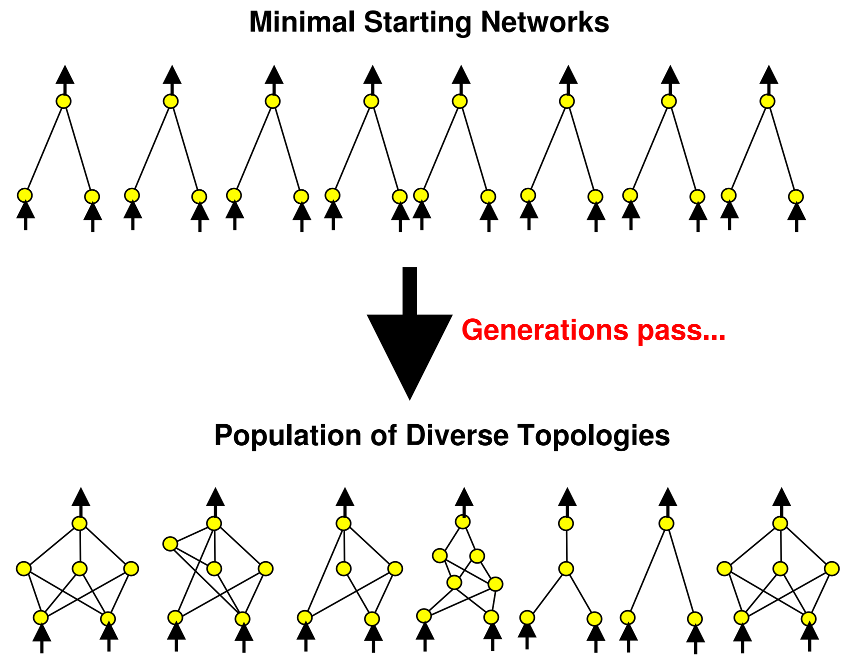




- Above: vanilla neuroevolution (mutation not shown).
  - Genotype $\rightarrow$ phenotype, then run in the environment
  - Fitness evaluation and selection
  - Mating and reproduction

Evolving Neural Network Controllers





- We used NeuroEvolution of Augmenting Topologies (NEAT) algorithm by Stanley and Miikkulainen (2002).
- Networks of arbitrarily complex topologies can be evolved, leading to increasingly complex behavior.

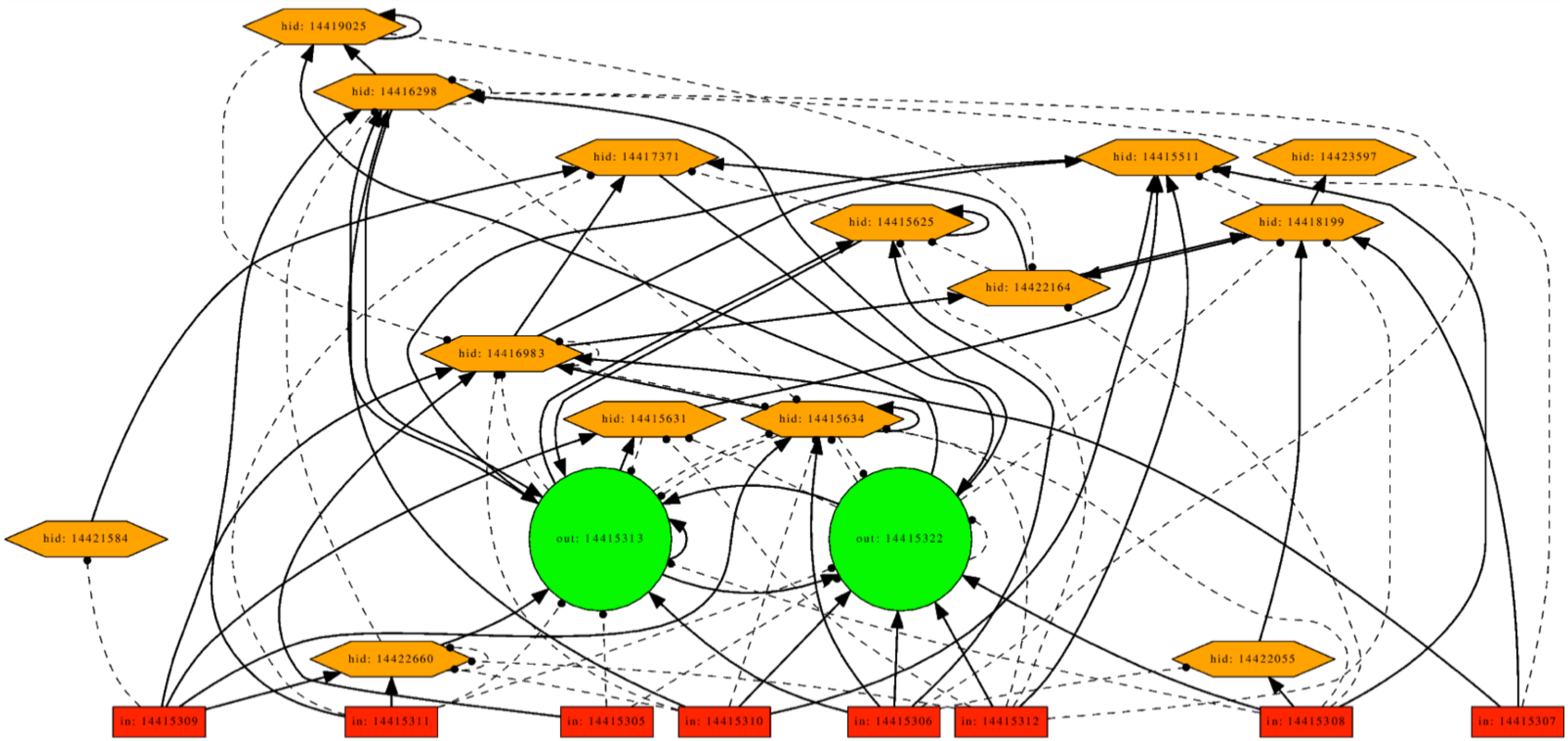
Fitness Evaluation

- D: final distance to target
- S: number of steps to reach target
- T: number of times tool picked up
- ... : DS, DT, DST, etc. (multiplied combination)

Task: 50% within reach, 50% beyond reach targets

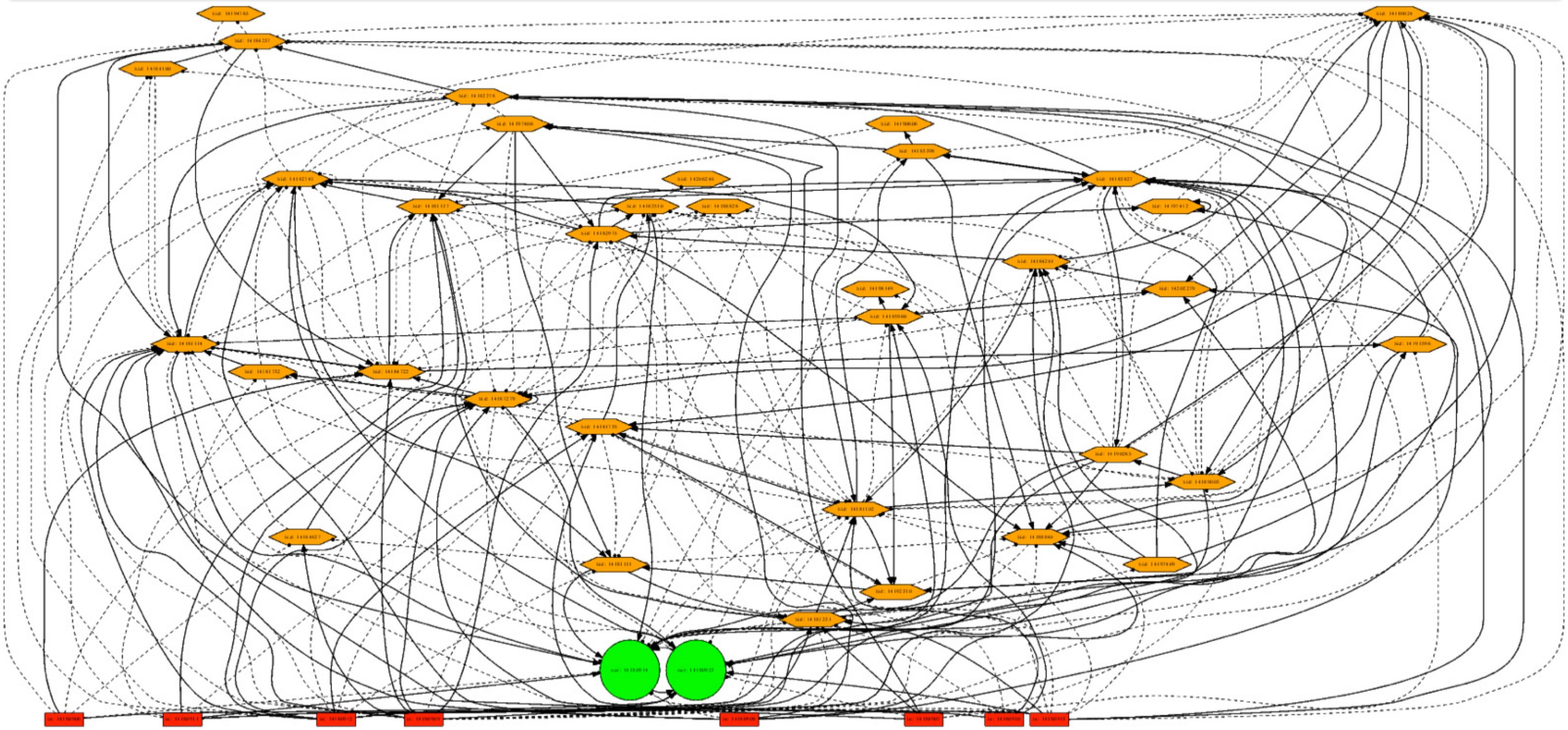
Evolved Neural Networks 1

Sensory Input Neurons: Hidden neurons: Motor outputs:



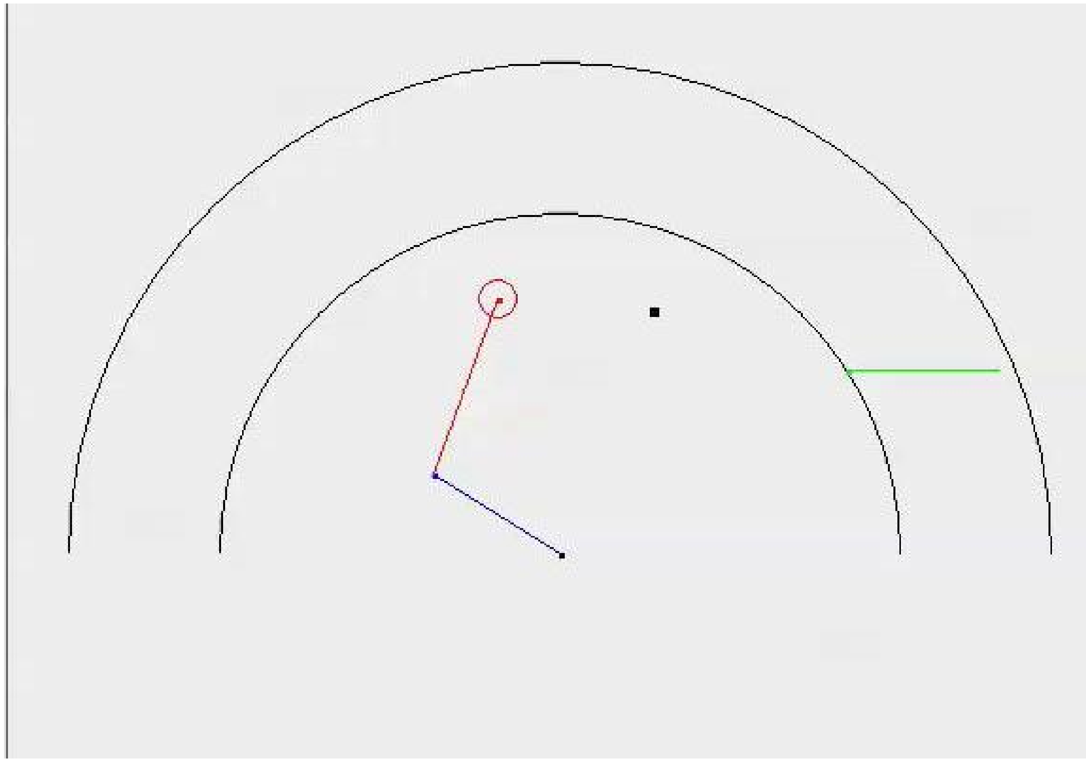
Fitness = S^2T

Evolved Neural Networks 2



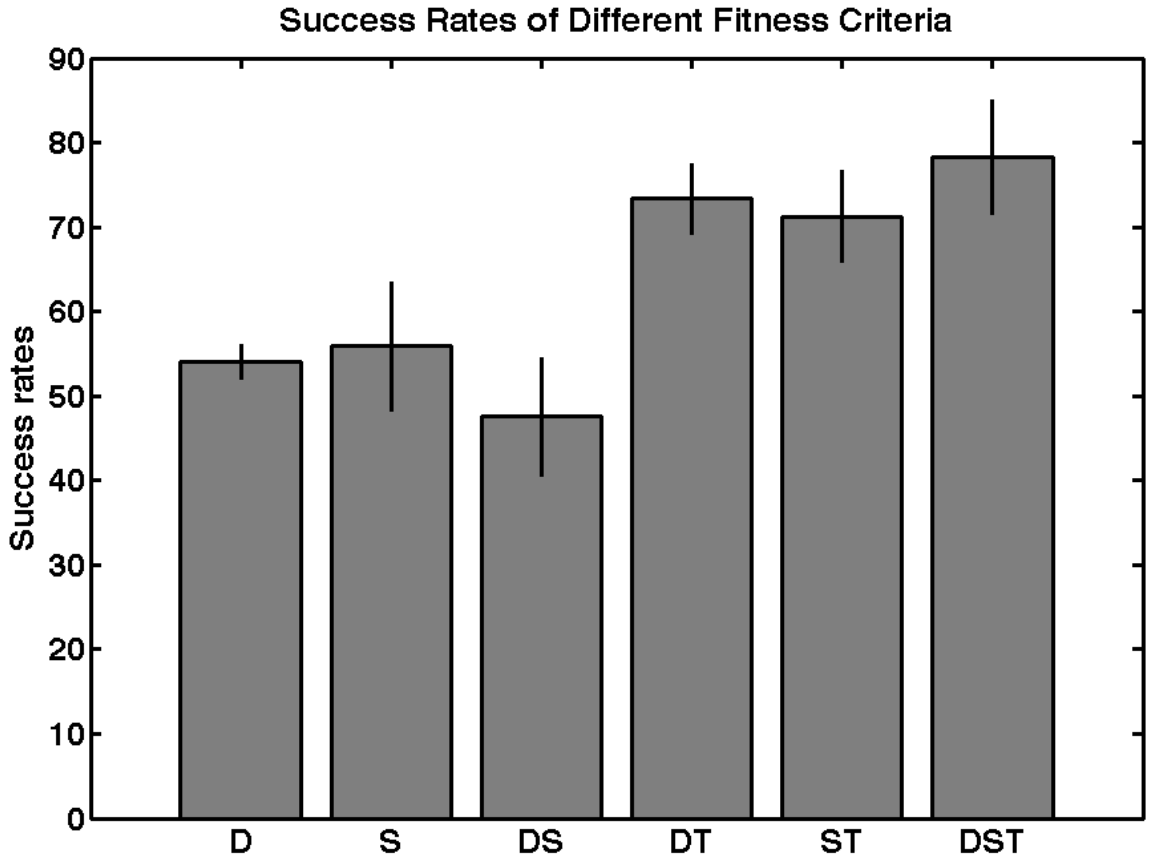
Fitness = DS

Tool Use Behavior



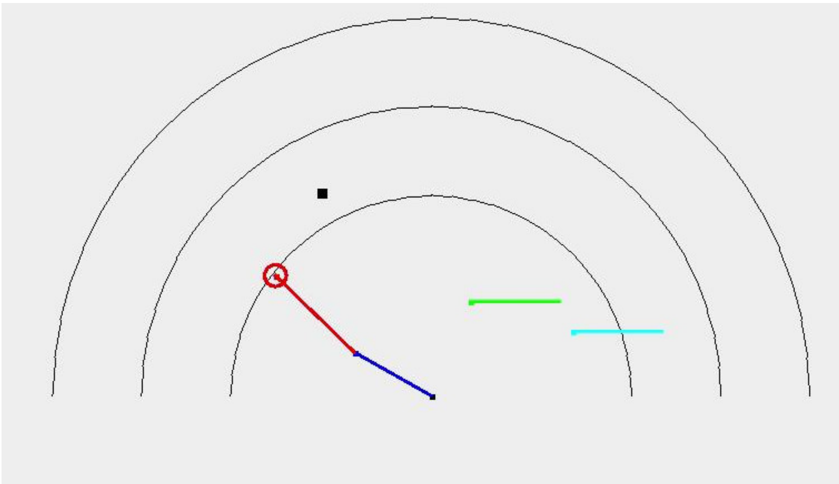
- Articulated arm.
- Tool (green bar) pick up and reach goal.

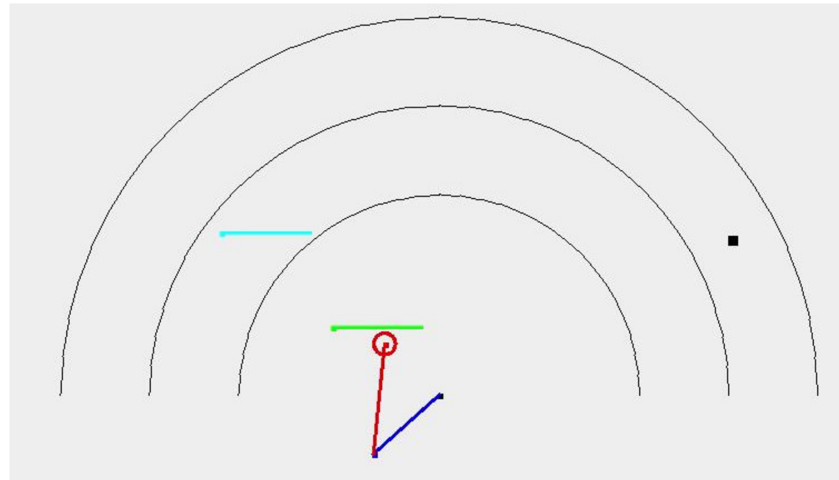
Target Reaching Performance



- Fitness criterion T helps, but not necessary in evolving tool use behavior (avg/std shown; $n = 4$ sets, each with 1,000 trials).

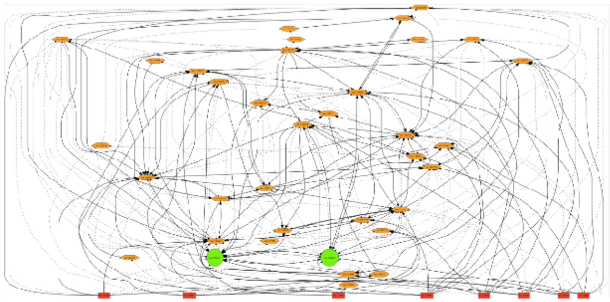
Simple Tool Construction Task



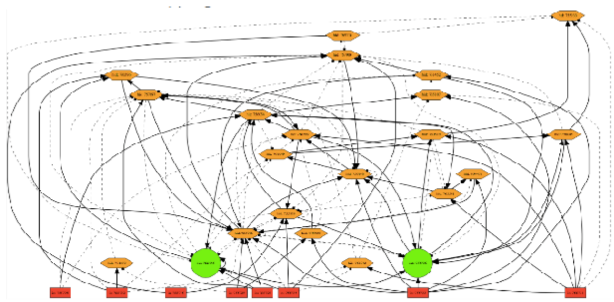


- Combine two sticks to reach out-of-reach targets.
- Some targets reachable without a stick.
- Some reachable with one stick.
- Some reachable with two sticks.

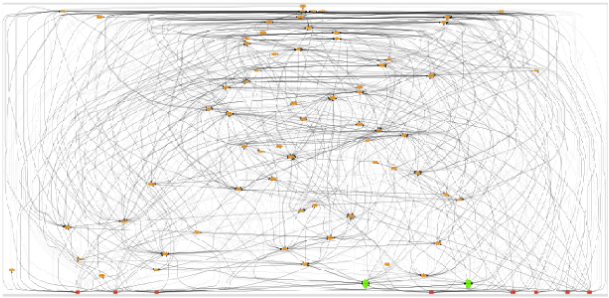
Results: Example Evolved Networks



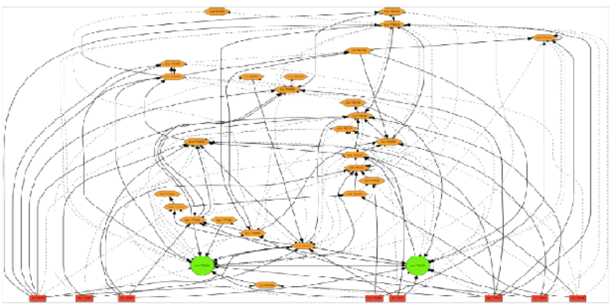
(a) Highest Fitness Network Evolved on *C1*



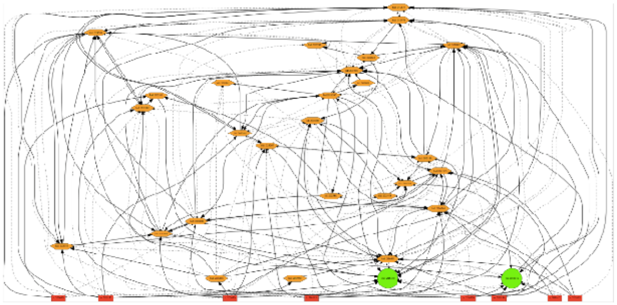
(b) Highest Fitness Network Evolved on *C2*



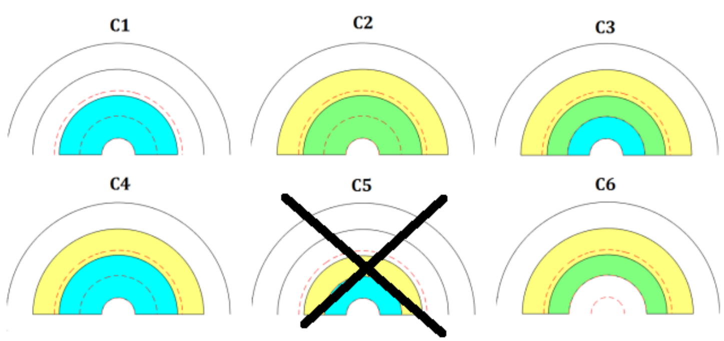
(c) Highest Fitness Network Evolved on *C3*



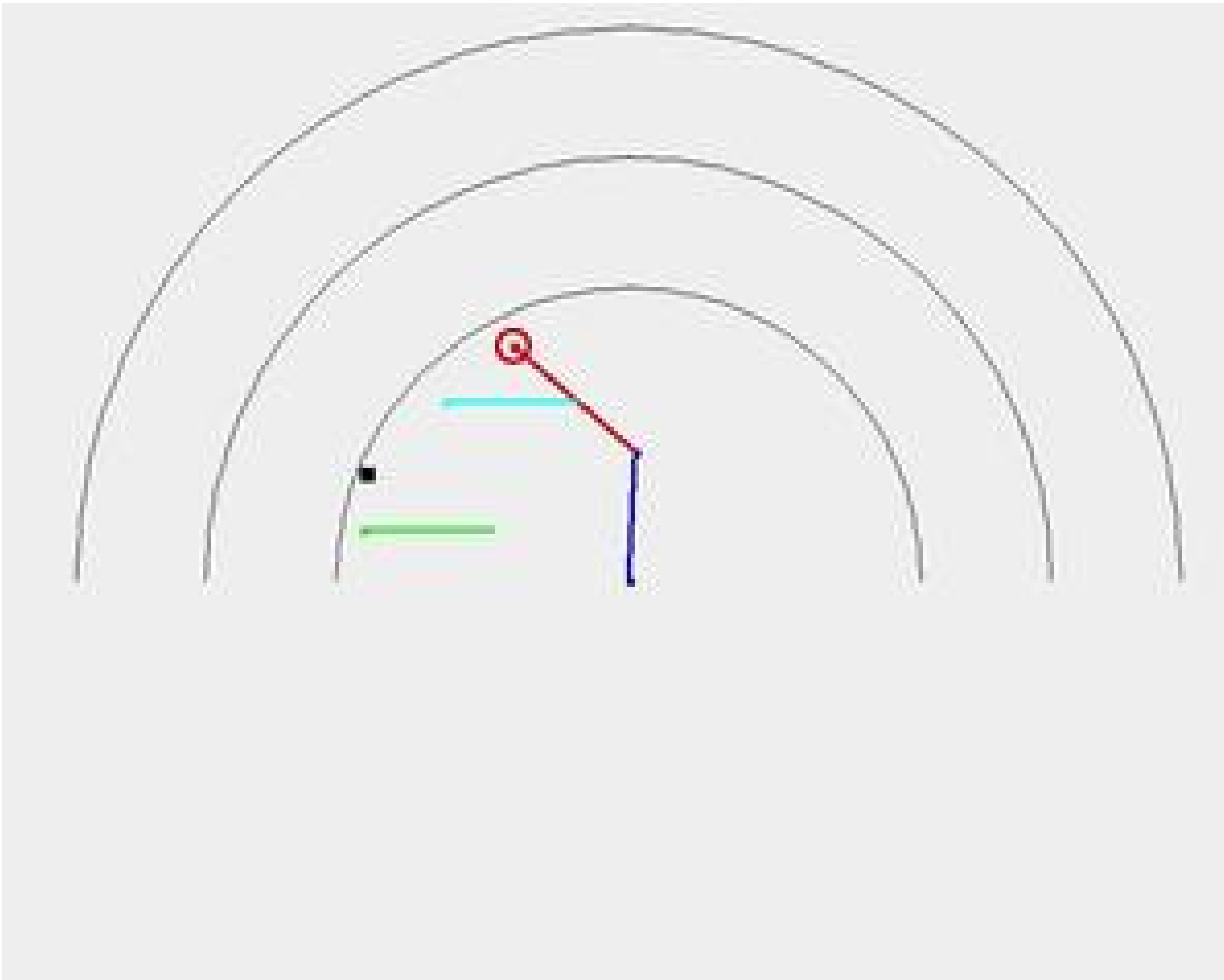
(d) Highest Fitness Network Evolved on *C4*



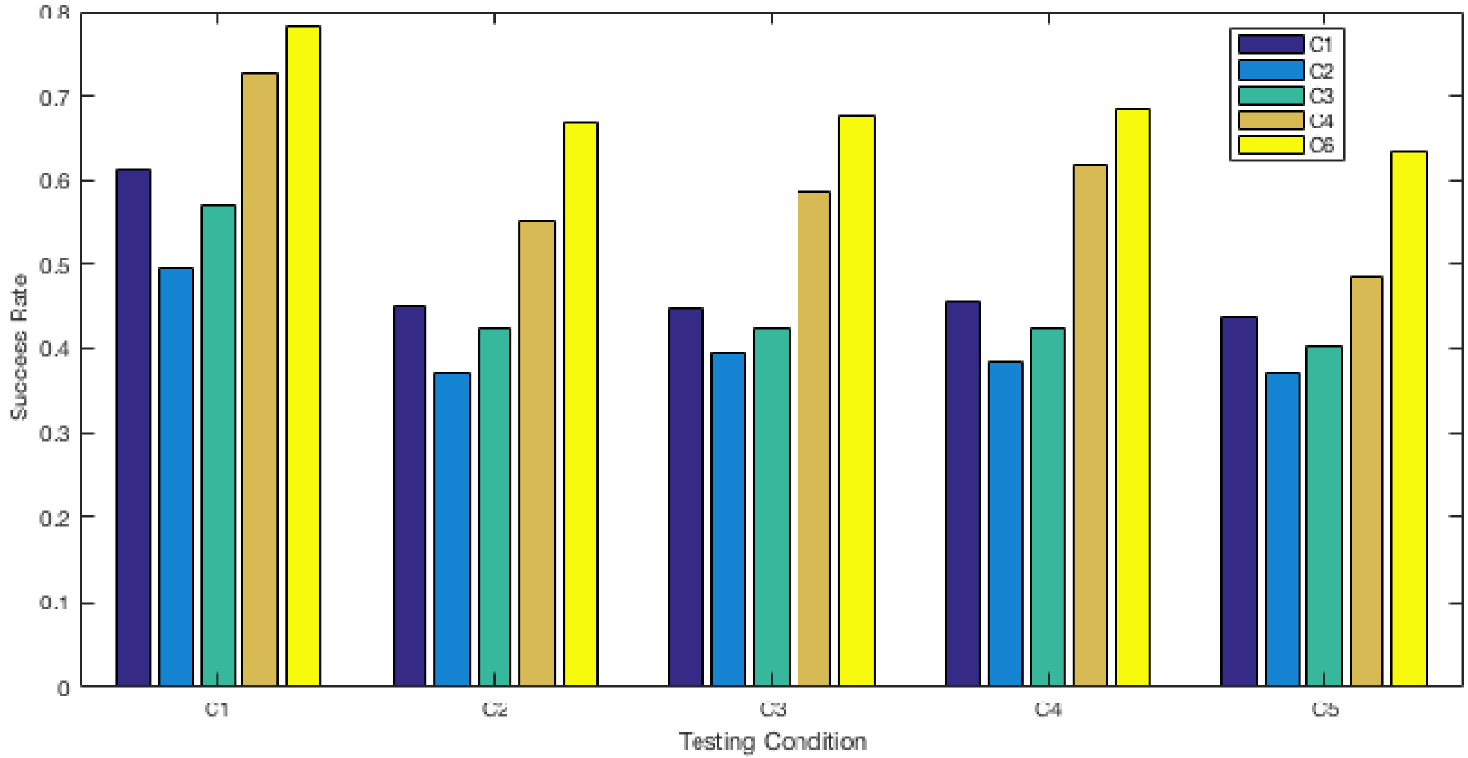
(e) Highest Fitness Network Evolved on *C6*



Results: Demo



Results: Average Performance



- On average (top), training on simple (one tool) or ambiguous tasks leads to lower performance during testing.

More Demo

- End-to-end Tool Use demo

Summary

- Going beyond the confines of the brain (network weights, integrated external memory).
- Using the environment as a canvass empowers neural networks, even very simple feedforward networks.
- Tool use and tool construction can have a synergistic effect: co-evolution of tool and intelligence.
- Implications on deep learning:
  - Both of the above can enable the definition of new tasks previously unavailable to the agent.
  - Potential for open-ended improvement, not limited to the immediate task.

Wrap Up

Conclusion

- There are multiple practical and fundamental limitations of deep learning.
- Practial limits already have potential solutions.
- Investigating fundamental limits allows us to go beyond deep learning.
  - Meaning through action
  - Consciousness through predictive dynamics
  - Open-ended improvement through stigmergy and tool construction/use.

Acknowledgments

- Neural coding: Bhamidipati (2004); Choe and Bhamidipati (2004); Choe and Smith (2006); Choe et al. (2007); Choe (2011); Choe et al. (2008)
- Internal dynamics: Kwon and Choe (2008); Choe et al. (2012); Chung et al. (2012); Yoo et al. (2014)
- Stigmergy: Chung and Choe (2009, 2011)
- Tool use and construction: Li et al. (2015); Reams and Choe (2017). Kinetic demo: Yoo (PhD thesis 2018).

References

Beer, R. D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, 4:91–99.

Bhamidipati, S. K. (2004). *Sensory invariance driven action (SIDA) framework for understanding the meaning of neural spikes.*. Master's thesis, Department of Computer Science, Texas A&M University.

Borst, A., and Egelhaaf, M. (1989). Principles of visual motion detection. *Trends in neurosciences*, 12(8):297–306.

Choe, Y. (2011). Action-based autonomous grounding. In Modayil, J., Precup, D., and Singh, S., editors, *AAAI-11 Workshop on Lifelong Learning from Sensorimotor Experience*, 56–57. Palo Alto, CA: AAAI Press. AAAI Workshop Technical Report WS-11-15.

Choe, Y., and Bhamidipati, S. K. (2004). Autonomous acquisition of the meaning of sensory states through sensory-invariance driven action. In Ijspeert, A. J., Murata, M., and Wakamiya, N., editors, *Biologically Inspired Approaches to Advanced Information Technology*, Lecture Notes in Computer Science 3141, 176–188. Berlin: Springer.

Choe, Y., Kwon, J., and Chung, J. R. (2012). Time, consciousness, and mind uploading. *International Journal on Machine Consciousness*, 4:257–274.

Choe, Y., and Smith, N. H. (2006). Motion-based autonomous grounding: Inferring external world properties from internal sensory states alone. In Gil, Y., and Mooney, R., editors, *Proceedings of the 21st National Conference on Artificial Intelligence(AAAI 2006)*, 936–941.

Choe, Y., Yang, H.-F., and Eng, D. C.-Y. (2007). Autonomous learning of the semantics of internal sensory states based on motor exploration. *International Journal of Humanoid Robotics*, 4:211–243.

Choe, Y., Yang, H.-F., and Misra, N. (2008). Motor system's role in grounding, receptive field development, and shape recognition. In *Proceedings of the Seventh International Conference on Development and Learning*, 67–72. IEEE.

Choe, Y., Yoo, J., and Li, Q. (2015). Tool construction and use challenge: Tooling test rebooted. In *AAAI-15 Workshop on Beyond the Turing Test*. 2 pages.

Chung, J. R., and Choe, Y. (2009). Emergence of memory-like behavior in reactive agents using external markers. In *Proceedings of the 21st International Conference on Tools with Artificial Intelligence, 2009. ICTAI '09*, 404–408.

Chung, J. R., and Choe, Y. (2011). Emergence of memory in reactive agents equipped with environmental markers. *IEEE Transactions on Autonomous Mental Development*, 3:257–271.

Chung, J. R., Kwon, J., Mann, T. A., and Choe, Y. (2012). Evolution of time in neural networks: From the present to the past, and forward to the future. In Rao, A. R., and Cecchi, G. A., editors, *The Relevance of the Time Domain to Neural Network Models, Springer Series in Cognitive and Neural Systems 3*, 99–116. New York: Springer.

Krapp, H. G. (2000). Neuronal matched filters for optic flow processing in flying insects. *International review of neurobiology*, 44:93–120.

Kwon, J., and Choe, Y. (2008). Internal state predictability as an evolutionary precursor of self-awareness and agency. In *Proceedings of the Seventh International Conference on Development and Learning*, 109–114. IEEE.

Li, Q., Yoo, J., and Choe, Y. (2015). Emergence of tool use in an articulated limb controlled by evolved neural circuits. In *Proceedings of the International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN.2015.7280564.

Lucas, B. D., and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*, vol. 81, 674–679.

Parulkar, A., and Choe, Y. (2016). Motor-based autonomous grounding in a model of the fly optic flow system. In *Proceedings of the International Joint Conference on Neural Networks*. In press.

Reams, R., and Choe, Y. (2017). Emergence of tool construction in an articulated limb controlled by evolved neural circuits. In *Proceedings of the International Joint Conference on Neural Networks*. In press.

Rieke, F., Warland, D., de Ruter van Steveninck, R., and Bialek, W. (1997). *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press. First edition.

St. Amant, R., and Wood, A. B. (2005). Tool use for autonomous agents. In *Twenteth AAAI Conference on Artificial Intelligence*, 184–189.

Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

Taylor, G. K., and Krapp, H. G. (2007). Sensory systems and flight stability: what do insects measure and why? *Advances in insect physiology*, 34:231–316.

Yoo, J., Kwon, J., and Choe, Y. (2014). Predictable internal brain dynamics in EEG and its relation to conscious states. *Frontiers in Neurorobotics*, 8(00018).