Neural Modularity Helps Organisms Evolve to Learn New Skills without Forgetting Old Skills

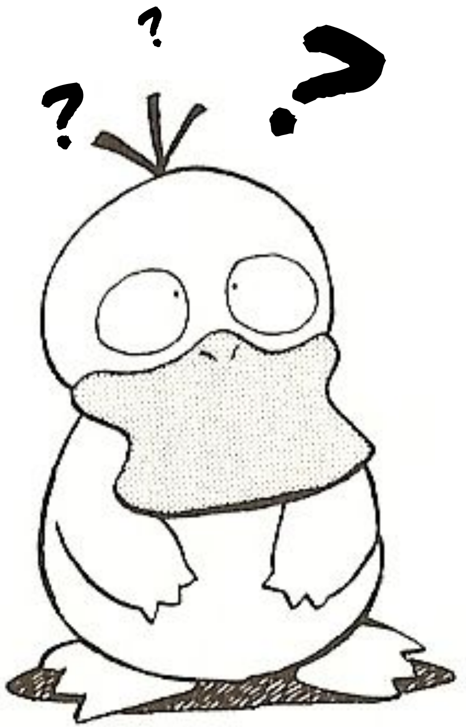
Kai Olav Ellefsen, Jean-Baptiste Mouret, Jeff Clune

MACHINE INTELLIGENCE COMMUNITY

Justin Chen
June 10th, 2018

Motivation

1. Adapt to novel situations over lifetime
2. Address **catastrophic forgetting** by evolving **modular** neural networks
    a. Reduce forgetting between tasks by separating functionality
    b. Selectively adapt modules based on current environmental stimuli



https://bulbapedia.bulbagarden.net/wiki/Psyduck_(Pok%C3%A9mon)

Problems with Previous Modularity Approaches

1. Architecture-specific
2. Unrealistic assumptions
    a. Access to data that may not be available in real environments
3. Memory and computational scalability
    a. As number of tasks/data
    b. As complexity of task increases

Challenges with Modularity

1. **Designing environments** that rapidly change with common subproblems (**modularly varying goals**)
2. How do natural environments change modularly?
3. **Designing problems** with modularly varying goals?

Approaches to Structural Modularity

Structural modularity emerges from sparsity and functionality

1. **Implicit Structural Modularity** (Performance-based)
    a. Sparsity and Functionality
        i. Regularized loss e.g. L1 norm
2. **Explicit Structural Modularity**
    a. Sparsity
        i. Penalizing structural development e.g. Connection Cost
    b. Functionality
        i. Allocating capacity e.g. Evolution or PNNs

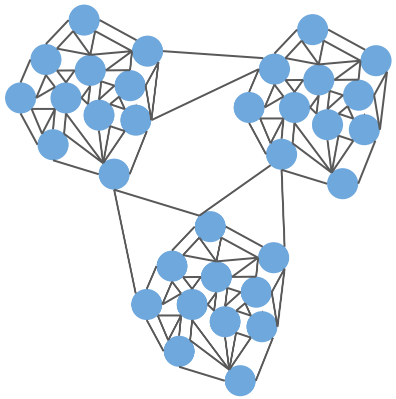
Goals of this Approach

1. Modularity mitigates catastrophic forgetting
   a. **Selectively regulate** learning in modules
2. Modularity improves **skill acquisition**
   a. Skill module
   b. Reward module
3. **Connection cost** encourages emergence of modularity

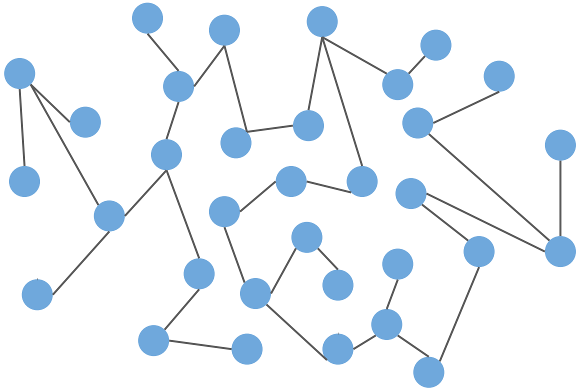
https://www.pinterest.com.au/pin/313070611567311147/

Structural Neural Modularity

- **Modular network** - a graph with dense neuron clusters that are sparsely connected to other clusters
    - **Community structure** in graph theory nomenclature
- **Neuromodulation** - ability to activate specific modules (clusters) conditioned on input (learning a program/ learning how to route/ conditional execution)

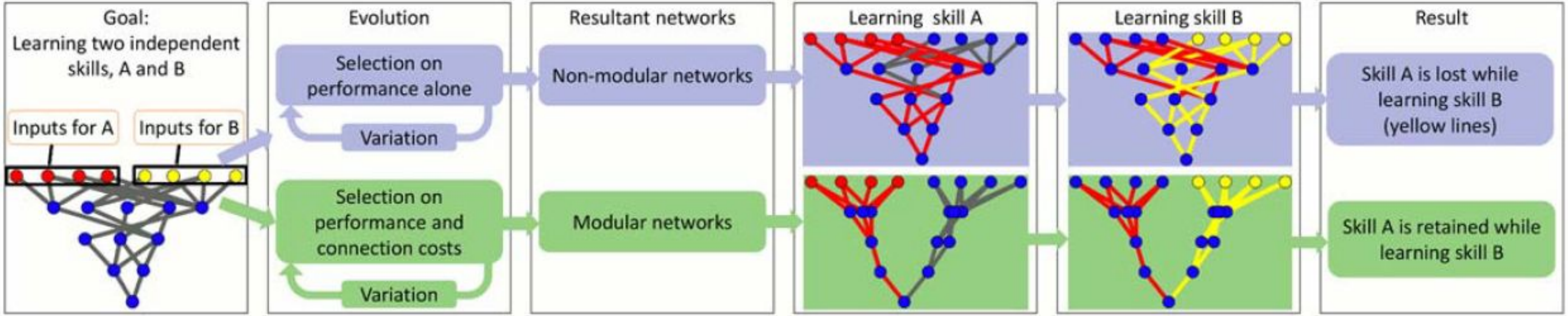


Community structure

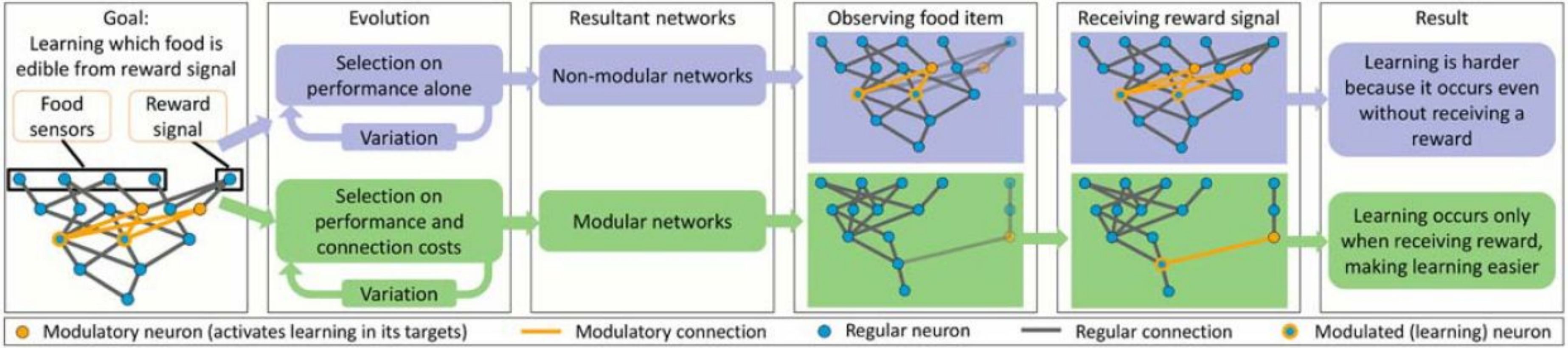


Sparse structure

Hypothesis 1:
Modularity Mitigates Catastrophic Forgetting



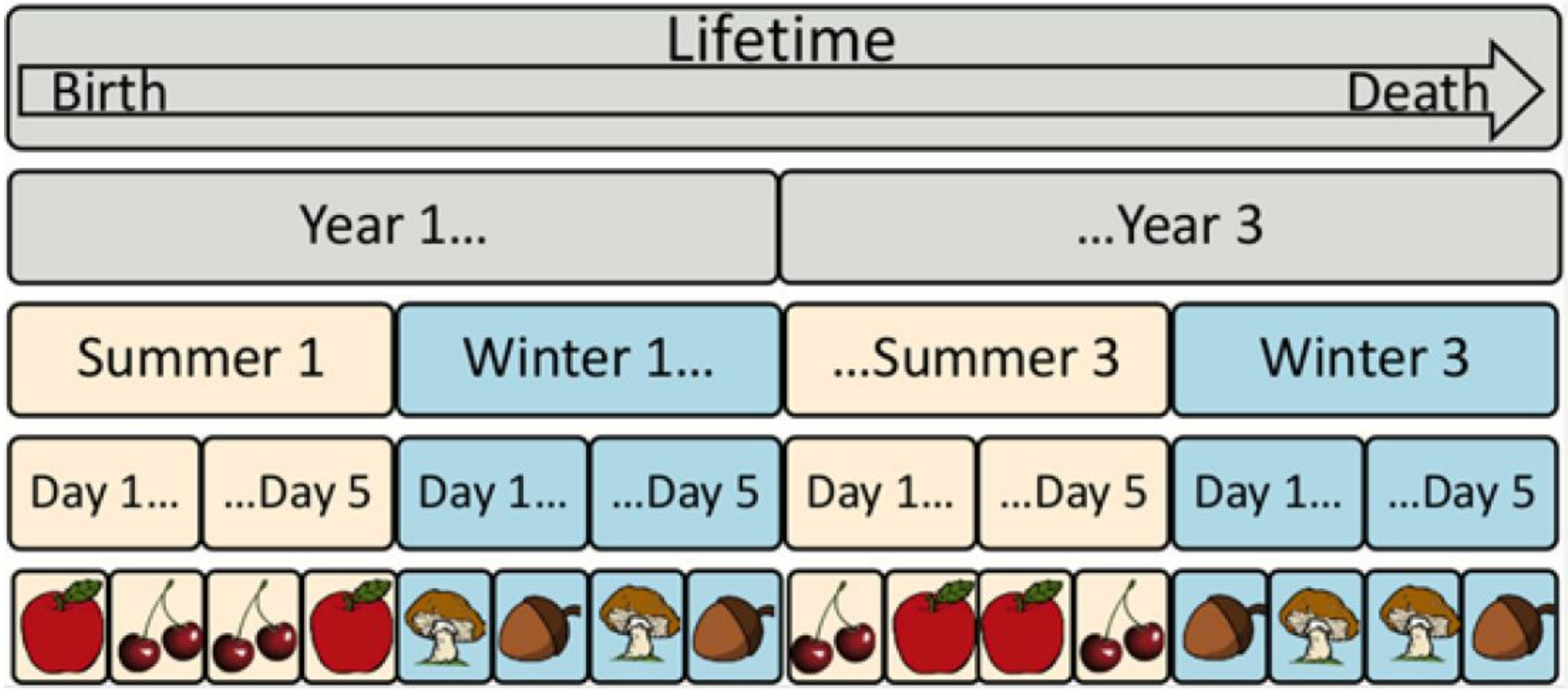
Hypothesis 2:
Skill and Reward Modules Improve Skill Acquisition



Experiment Setup

- Optimized via evolution
- 20,000 generations
- Fitness Evaluations:
  - Performance Alone (PA)
    - One objective
  - Maximizing performance and minimizing connection costs (P&CC)
    - Two objectives
- Connection cost = total number of connections

Environment for an Individual's Lifetime



Network Architecture

Input Features

$$\begin{array}{rl} \text{Summer food} \in \{0,1\} & \left\{ \begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \end{array} \right. \\ \text{Winter food} \in \{0,1\} & \left\{ \begin{array}{c} x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \right. \\ \text{Nutritious} & \{\, x_8 \\ \text{Poisonous} & \{\, x_9 \end{array}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}$$

Output Space

- Logistic Regression
- 1 = Eat food
- 0 = Ignore food

Layers

- 10 input neurons
- 3 hidden layers (10, 4, 2)
- 1 output neuron
- Sigmoid activation for all layers

Weight Updates

$$\underbrace{m_i}_{\text{Modulatory signal}} = \varphi\left(\sum_{j \in C_m} w_{ij} a_j\right) \in \mathbb{R}$$

$$\forall j \in C_n : \Delta w_{ij} = \underbrace{\eta}_{\text{Learning rate}} \cdot m_i \cdot \underbrace{a_i}_{\text{Post-synaptic activity}} \cdot \underbrace{a_j}_{\text{Pre-synaptic activity}}$$

Notice ith index refers to post-synaptic and jth refers to pre-synaptic - paper poorly explained this point

Also product of post and pre-synaptic activity is a scalar

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

Neuromodulated Architecture

Neuromodulatory vector is a column vector computed per layer

Each element modulates each neuron in this layer

$$m = \varphi\left(\begin{bmatrix} w_{11} & \cdots & w_{1C_m} \\ \vdots & \ddots & \vdots \\ w_{r1} & \cdots & w_{rC_m} \end{bmatrix} \begin{bmatrix} a_1^j \\ \vdots \\ a_{C_m}^j \end{bmatrix}\right) \in \mathbb{R}^r$$

Pre-synaptic activation matrix maintained per layer for training and inference b/c Hebbian

Each row corresponds to incoming activation from previous neurons

$$\begin{bmatrix} a_{11} & \cdots & a_{1C_m} \\ \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rC_m} \end{bmatrix}$$

NSGA-II Evolutionary Optimization

- Non-dominated Sorting Genetic Algorithm (NSGA-II) [3]
- Does not consider importance of multi-objective tasks
- Modifications:
  - Stochastic Pareto dominance [2]
  - Increased diversity - select for different outputs
    - Diversity of output calculated as normalized bitwise XOR or output between two individuals

NSGA-II Mutation Operators

- Add connection
- Remove connection
- Change connection strengths
- Move connections
- Change type of neurons (modulatory or non-modulatory)

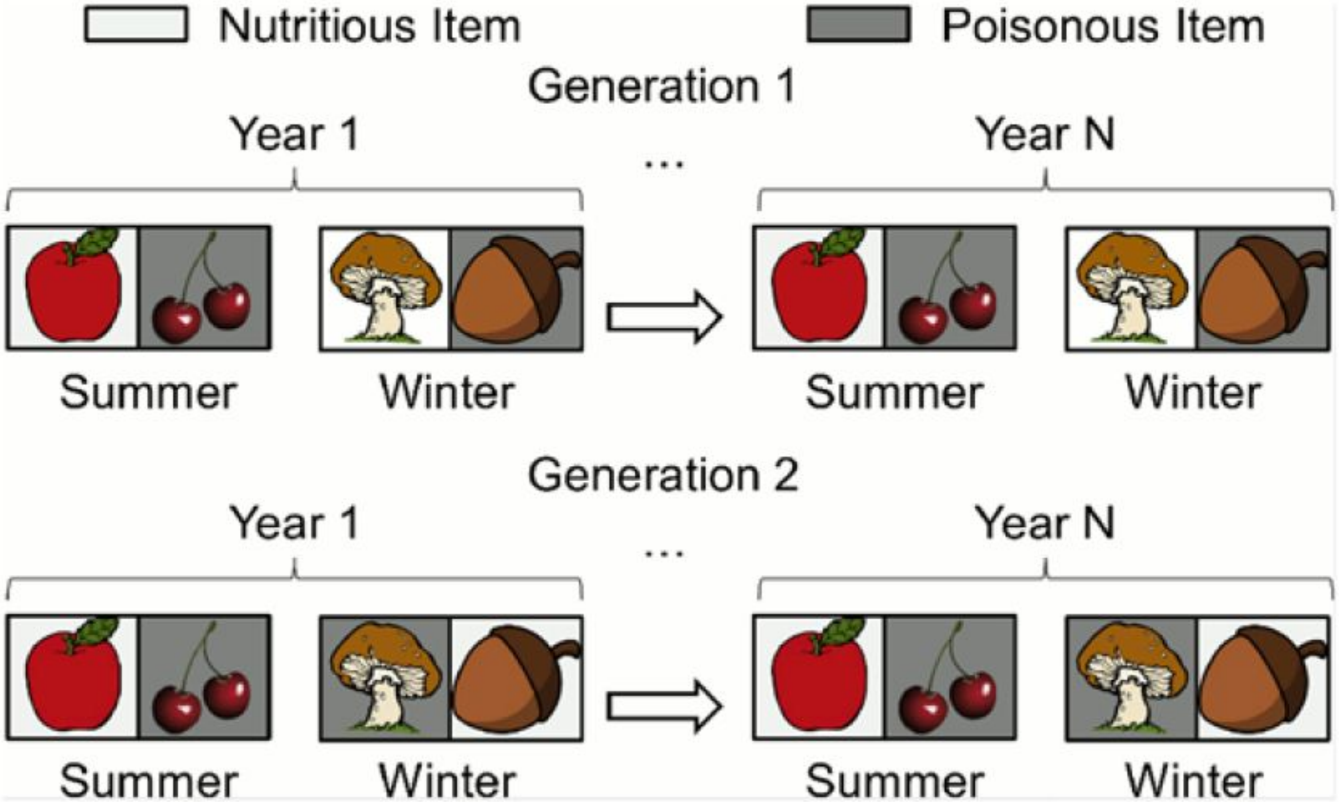
Diversity to Escape Local Minima

$$d^{(i)} = \frac{1}{N} \sum_{j \neq i}^{N} (|\underbrace{h_\theta^{(i)}}_{\text{Output of } i^{th} \text{ agent}} - \underbrace{h_\theta^{(j)}}_{\text{Output of } j^{th} \text{ agent}}|)$$

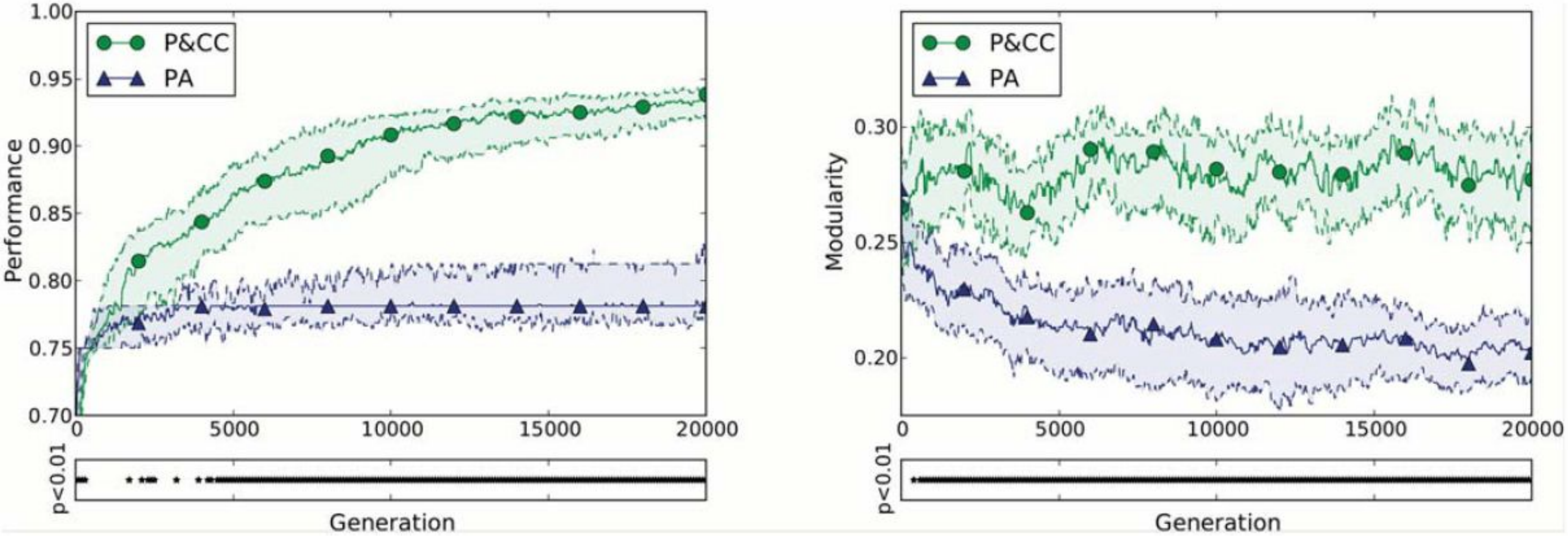
Computed for each agent for each input

Score stored for each agent

Randomized Food Associations Each Generation



Q-Modularity Score



Q-Modularity Score

$$Q = \frac{1}{m}\sum_{ij}\left[A_{ij} - \frac{k_i^{in}k_j^{out}}{m}\right]\delta_{ci,cj}$$

Total number of edges in network

Connectivity matrix

Probability that network has a connection between node i and j

Binary-valued function indicating, 1, if i and j belong to the same module, and 0 otherwise.

Note: Directly maximizing Q is NP-Hard, so it must be approximated. This work used the *Spectral Optimization Method*.

Testing Hypothesis 2

Did the networks evolve modular structures?

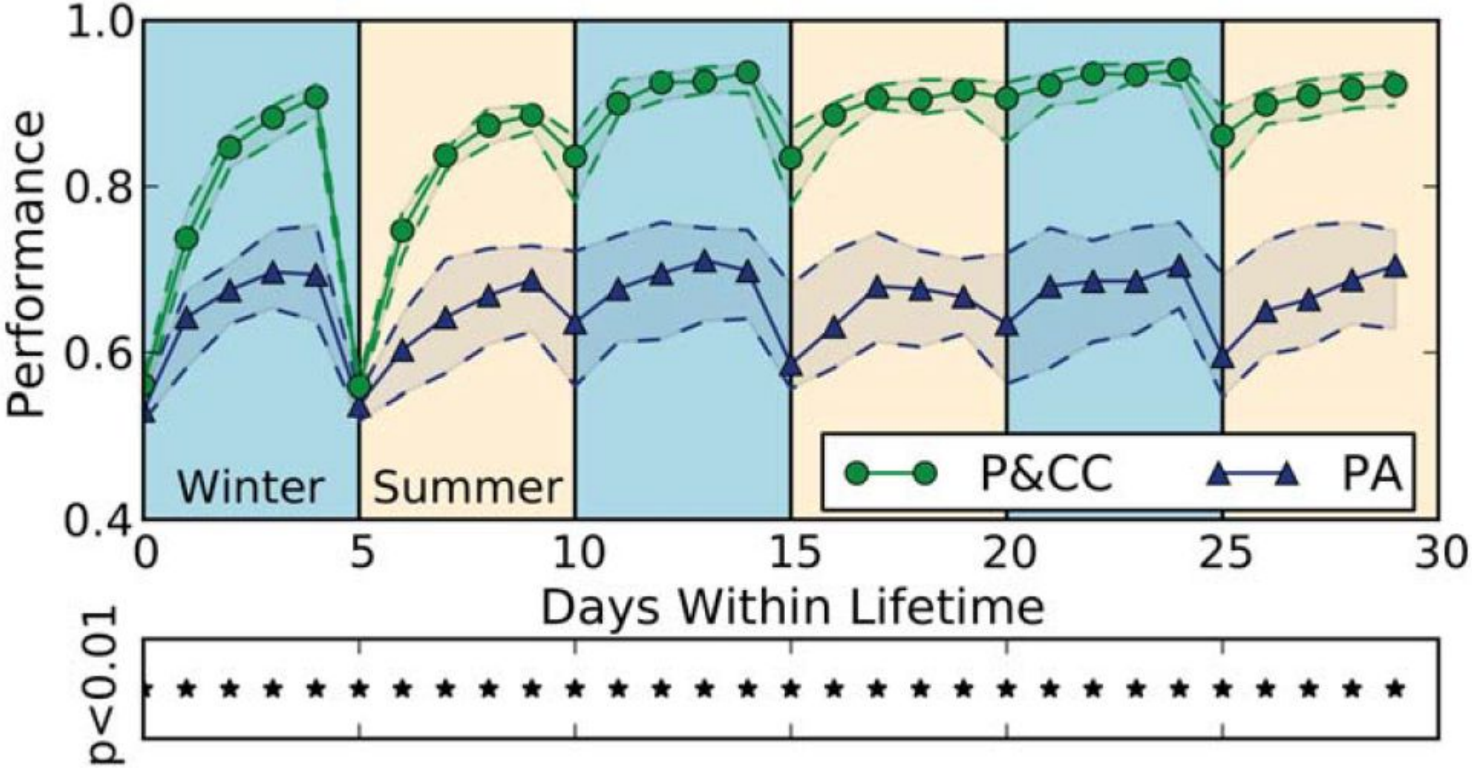
1. Decomposed network in modules according to modularity Q-score
2. Measured frequency of RL signals in each module

31% of evolutionary trials, **P&CC networks** develop separate modules

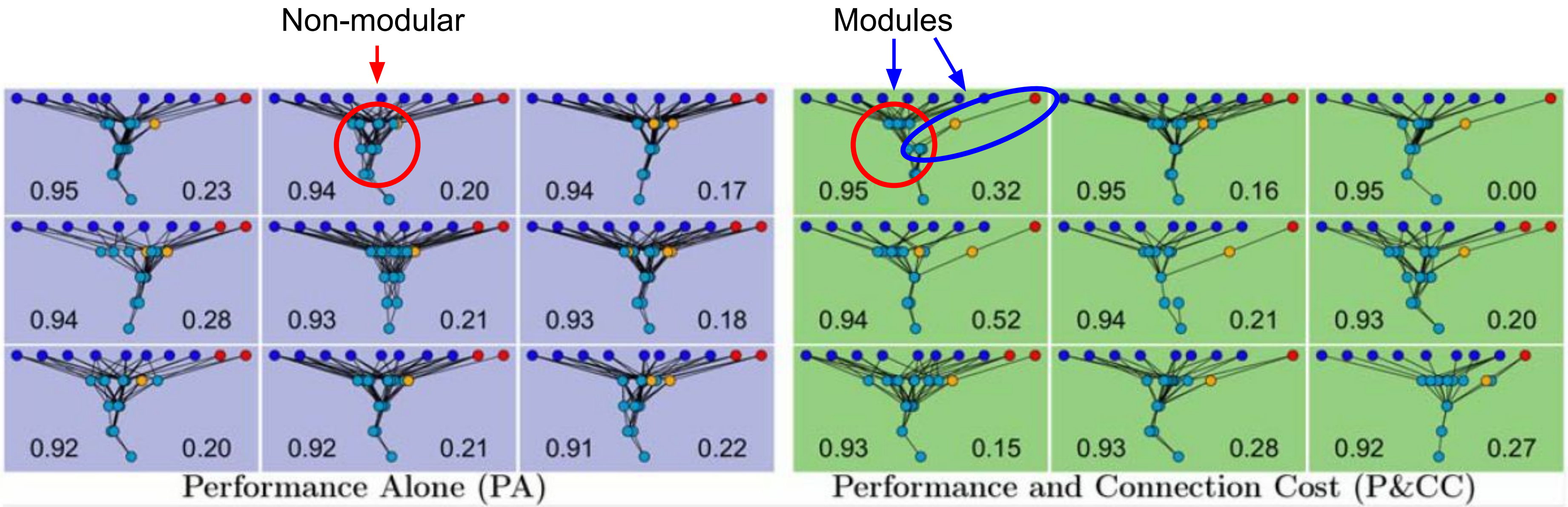
4% of evolutionary trials, **PA networks** develop separate modules

Modular networks from either setting have higher median performance

Median Performance, 100 Organisms, 80 Envs

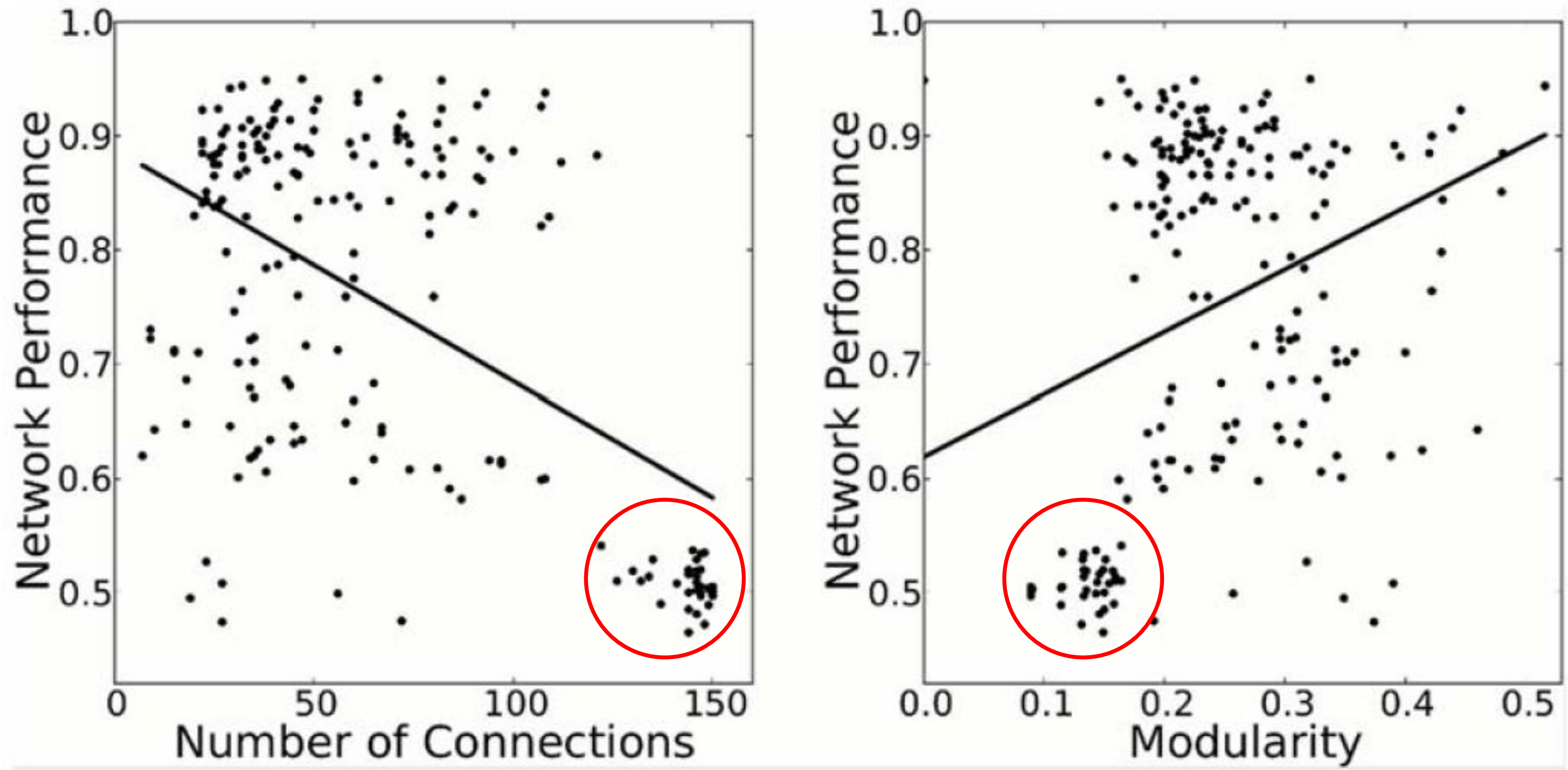


Evolved Network Structures

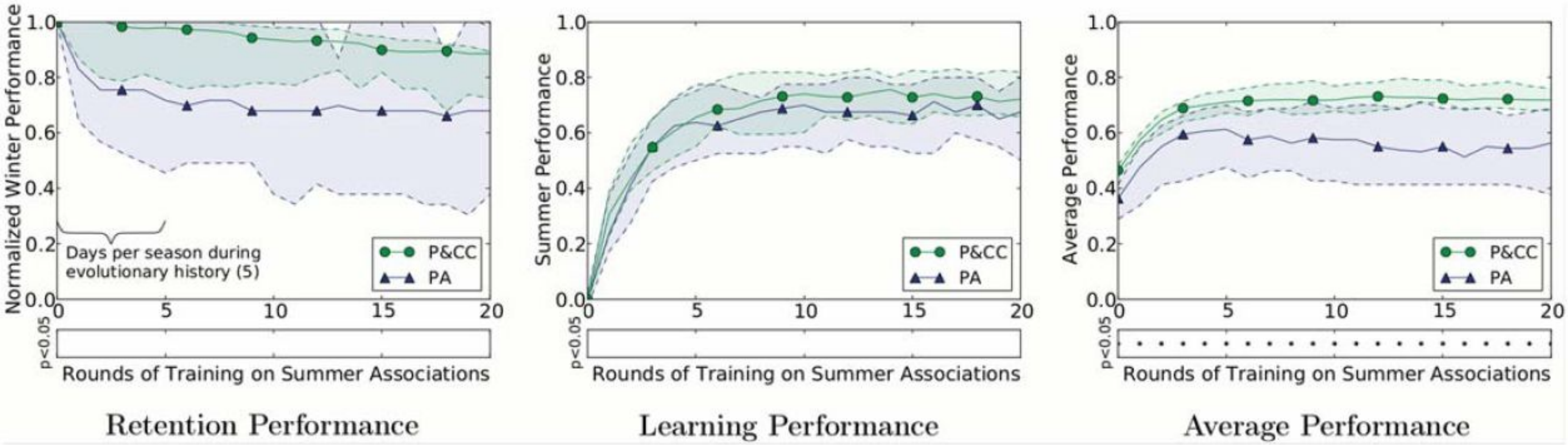


- Input node for food
- Nutrition or Poison
- Internal non-modulator neurons
- Neuromodulatory neurons

Testing Hypothesis 1



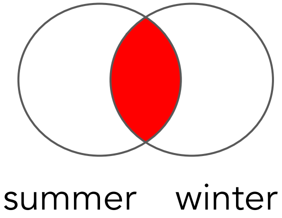
PA and P&CC Retention and Forgetting



P&CC Learning and Retention

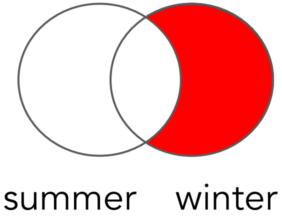
Perfect

Measures number of seasons agent knew both associations



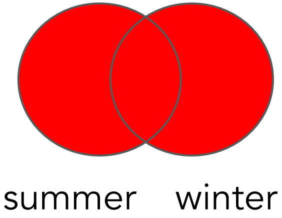
Forgotten

Total number of seasons only one association was completely known



Known

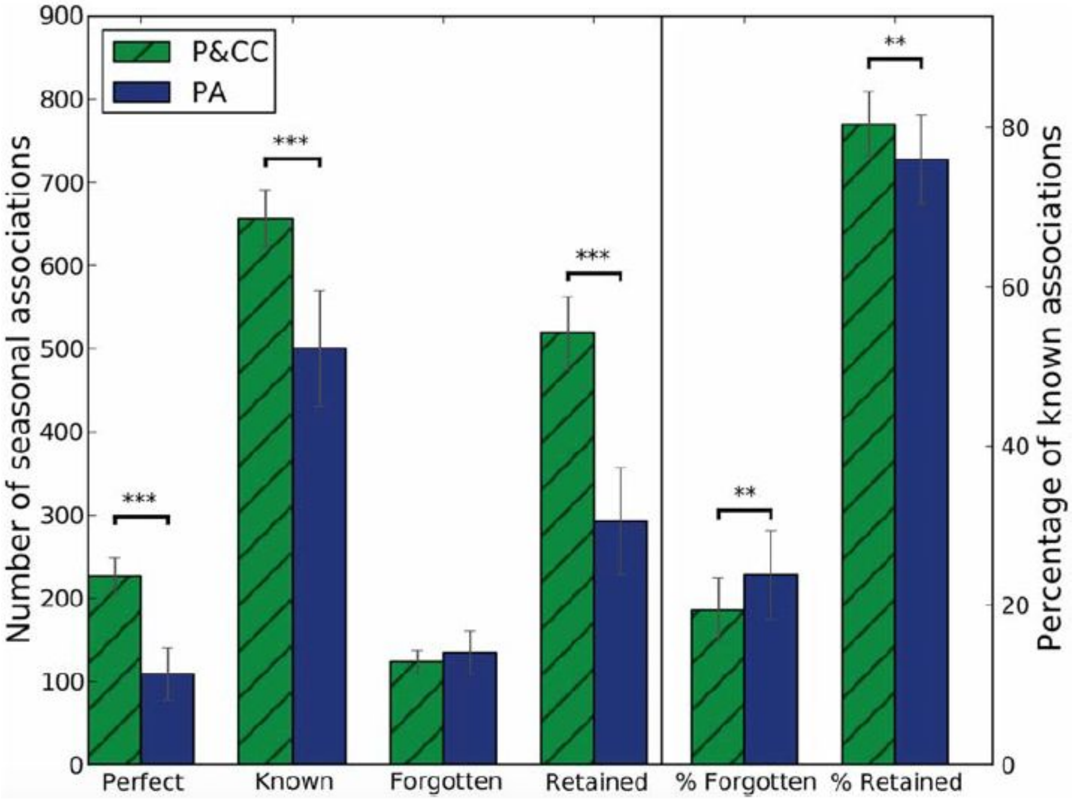
Total number of seasons that both associations were known



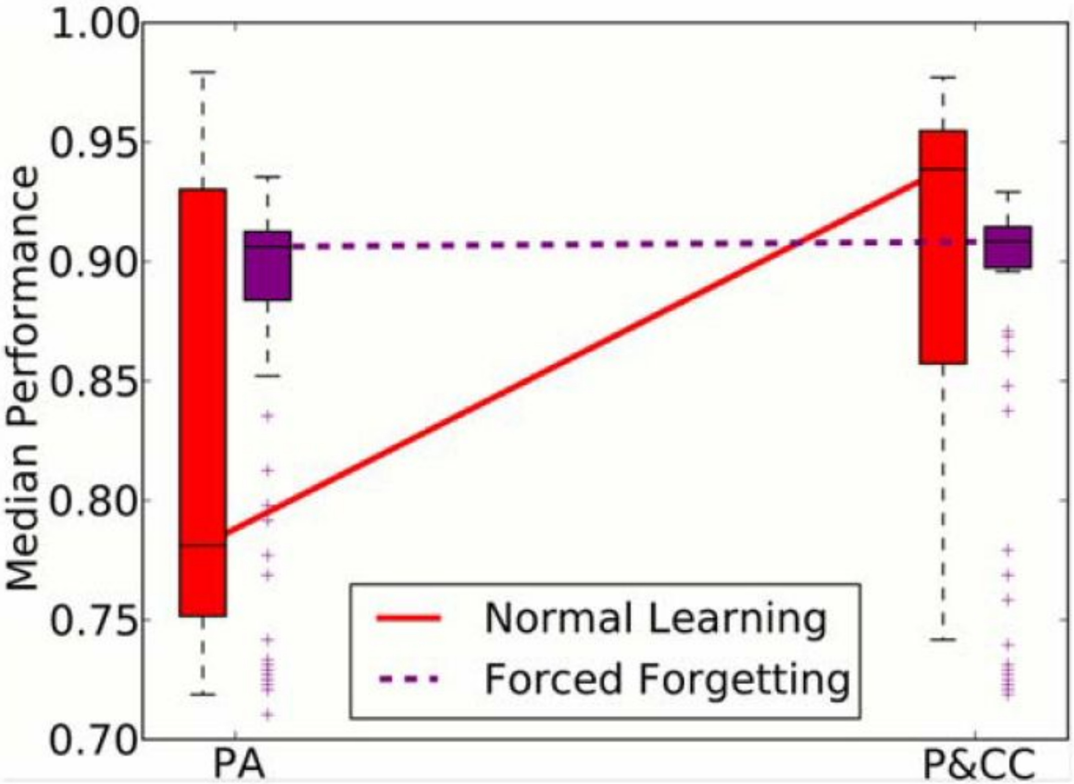
Retained

Total number of seasons an association was known in consecutive seasons

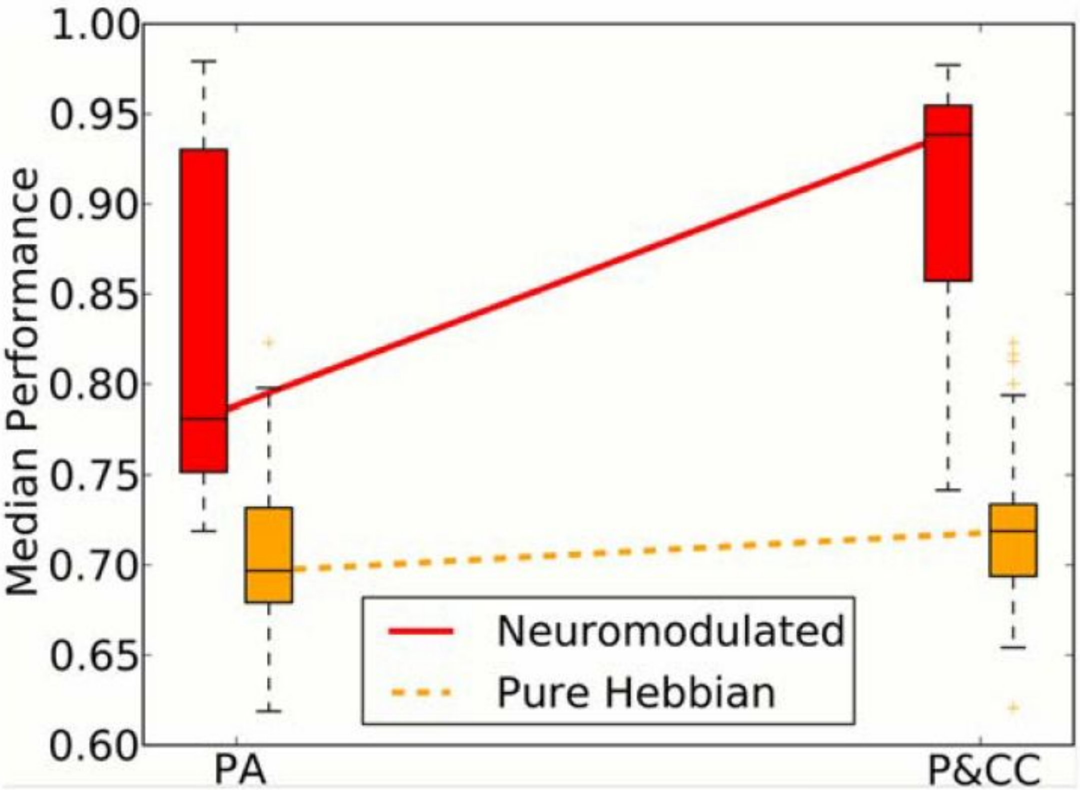
P&CC Learning and Retention



Forced Forgetting Counteracts Modularity



Neuromodulation, CC, and CF



Open Questions

1. Can CC also mitigate CF when inputs are shared between skills that require multiple modules?
2. Can CC also mitigate CF on more complex tasks that can't be learned if forgetting must occur between episodes?
3. What are optimal hyperparameters?
4. What evolutionary pressures cause learning dynamics that give rise to modularity?

Citations and Further Reading

1. Ellefsen, Kai Olav, Jean-Baptiste Mouret, and Jeff Clune. "**Neural modularity helps organisms evolve to learn new skills without forgetting old skills.**" PLoS computational biology 11.4 (2015): e1004128.
2. Clune, Jeff, Jean-Baptiste Mouret, and Hod Lipson. "The evolutionary origins of modularity." Proc. R. Soc. B 280.1755 (2013): 20122863.
3. Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation 6.2 (2002): 182-197.