Materials Matter:
How biologically inspired alternatives to conventional neural networks improve meta-learning and continual learning

Jeff Clune

@jeffclune

Associate Professor, Computer Science
University of British Columbia

Research Team Leader
OpenAI

Evolution of Structural Organization

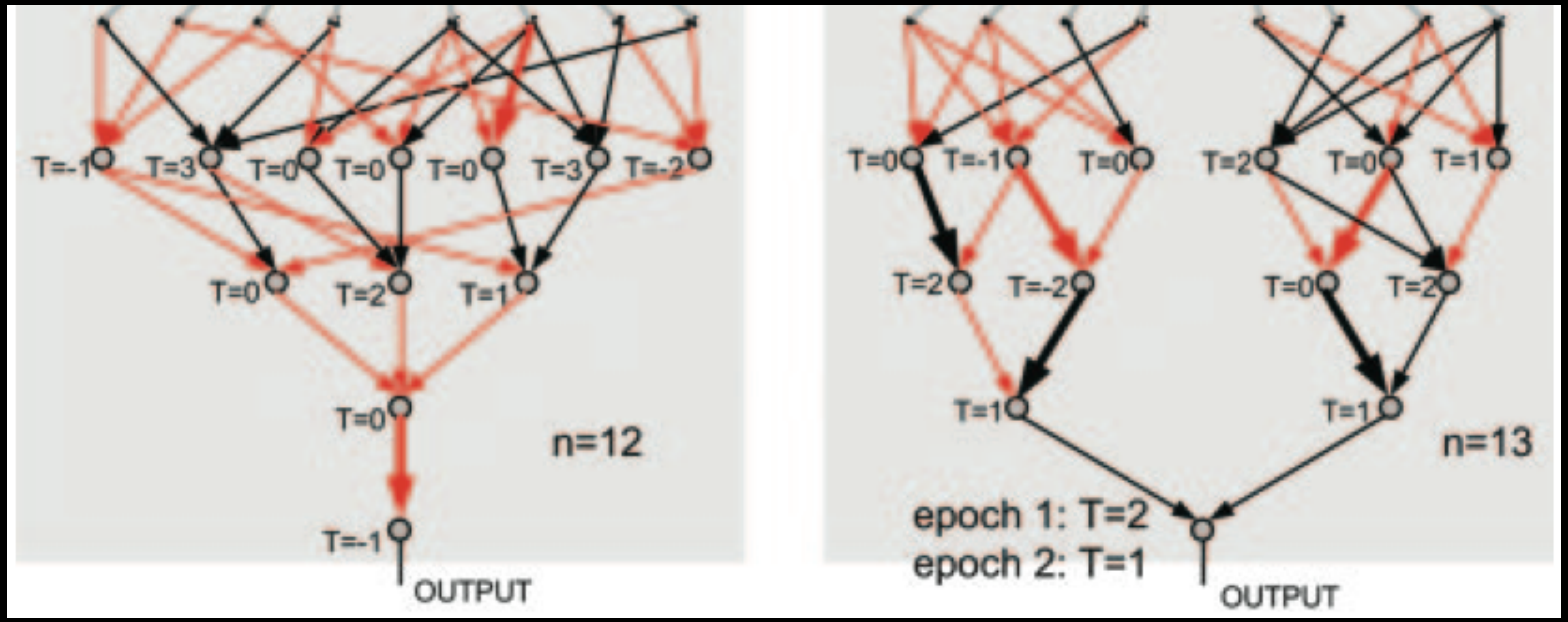
- Modularity
- Hierarchy

Modularity

- Localization of function in an encapsulated unit (Lipson 2007)
  - Car (spark plug, muffler, wheel), bodies (organs), brains, software, etc.
- Enables increased
  - Complexity
  - Adaptability

Modularity

- Rare in previous neuroevolution
  - Suggests selection on performance alone does not produce modularity



Kashtan and Alon 2005

Evolutionary Origins of Modularity

Clune, Mouret, & Lipson, Proc. Royal Society, 2013

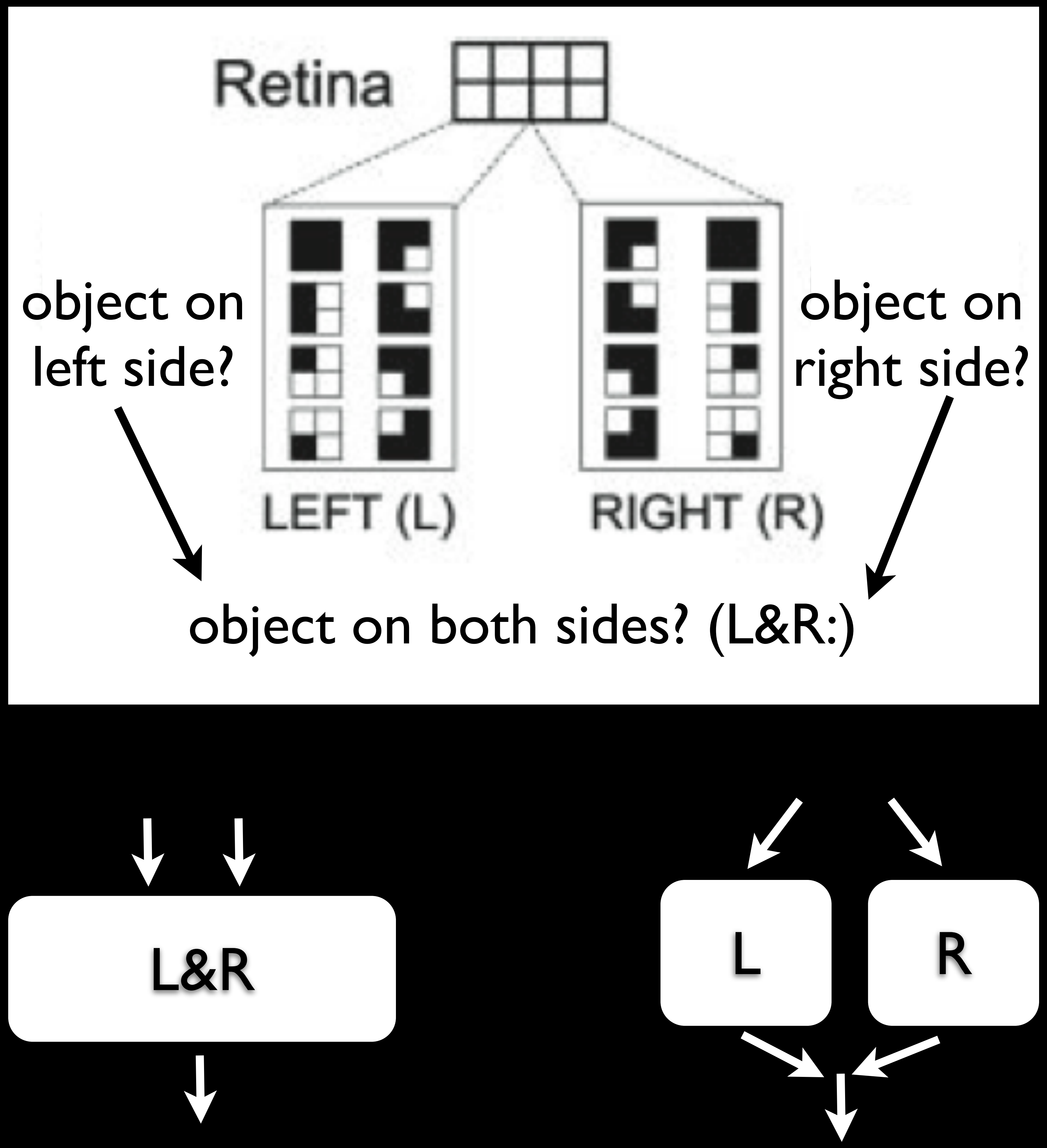
Jeff Clune

Jean-Baptiste Mouret

Hod Lipson

PROCEEDINGS OF THE ROYAL SOCIETY B BIOLOGICAL SCIENCES

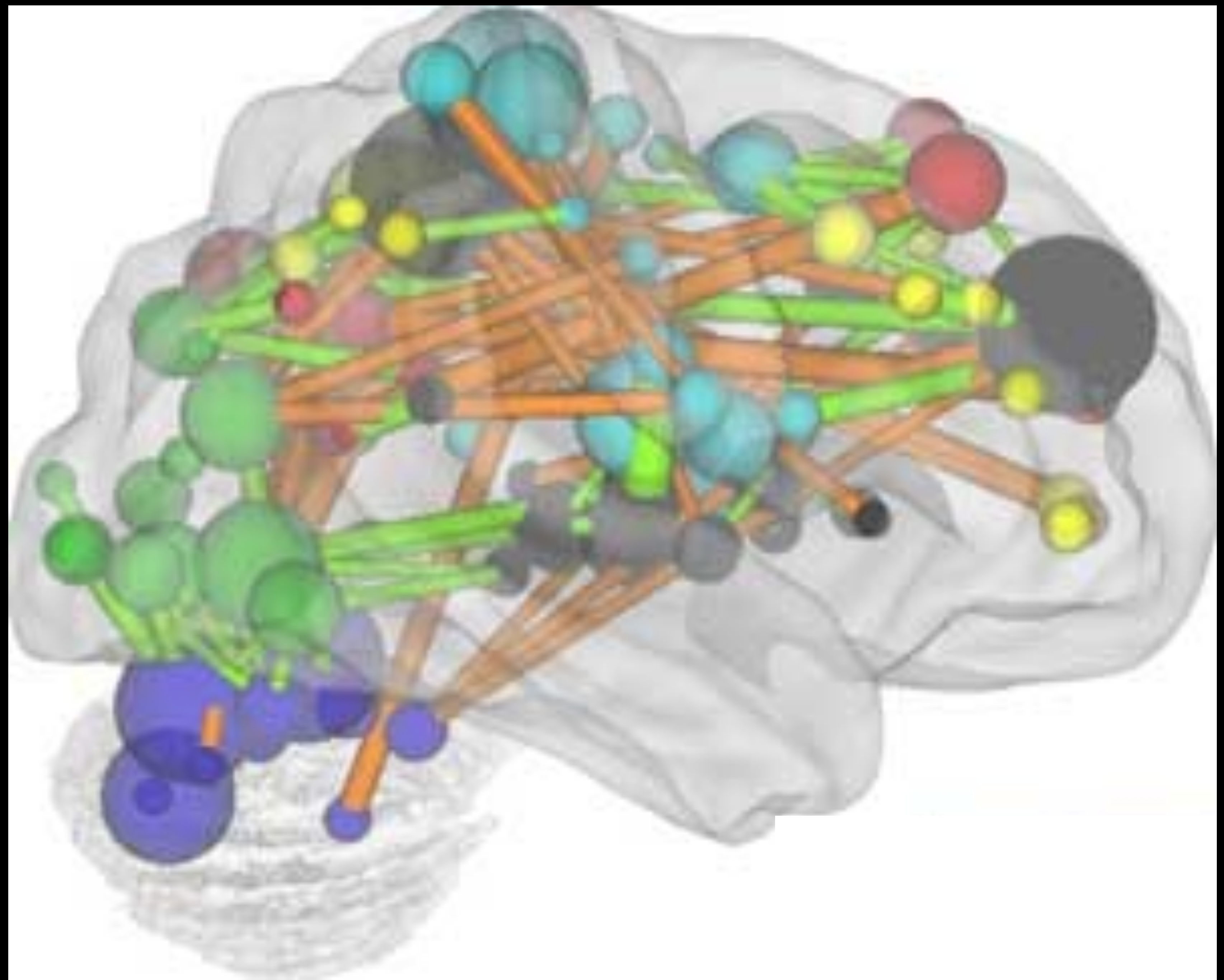
Retina Problem



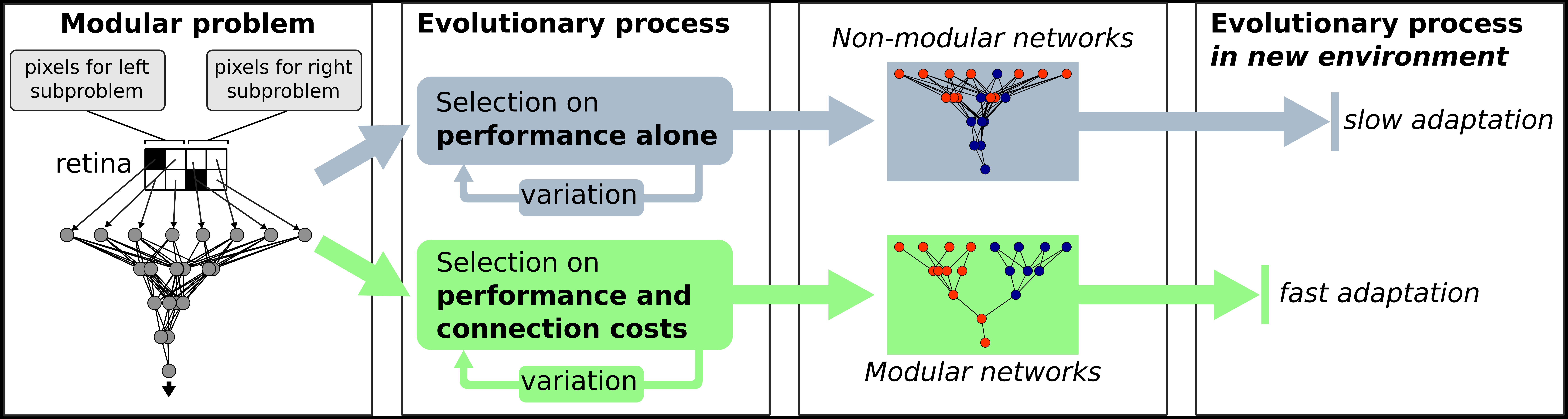
Kashtan and Alon. PNAS. 2005

Why does modularity evolve?

- Hypothesis from founding neuroscientist (Ramón y Cajal 1899)
  - Selection to minimize <u>connection costs</u>

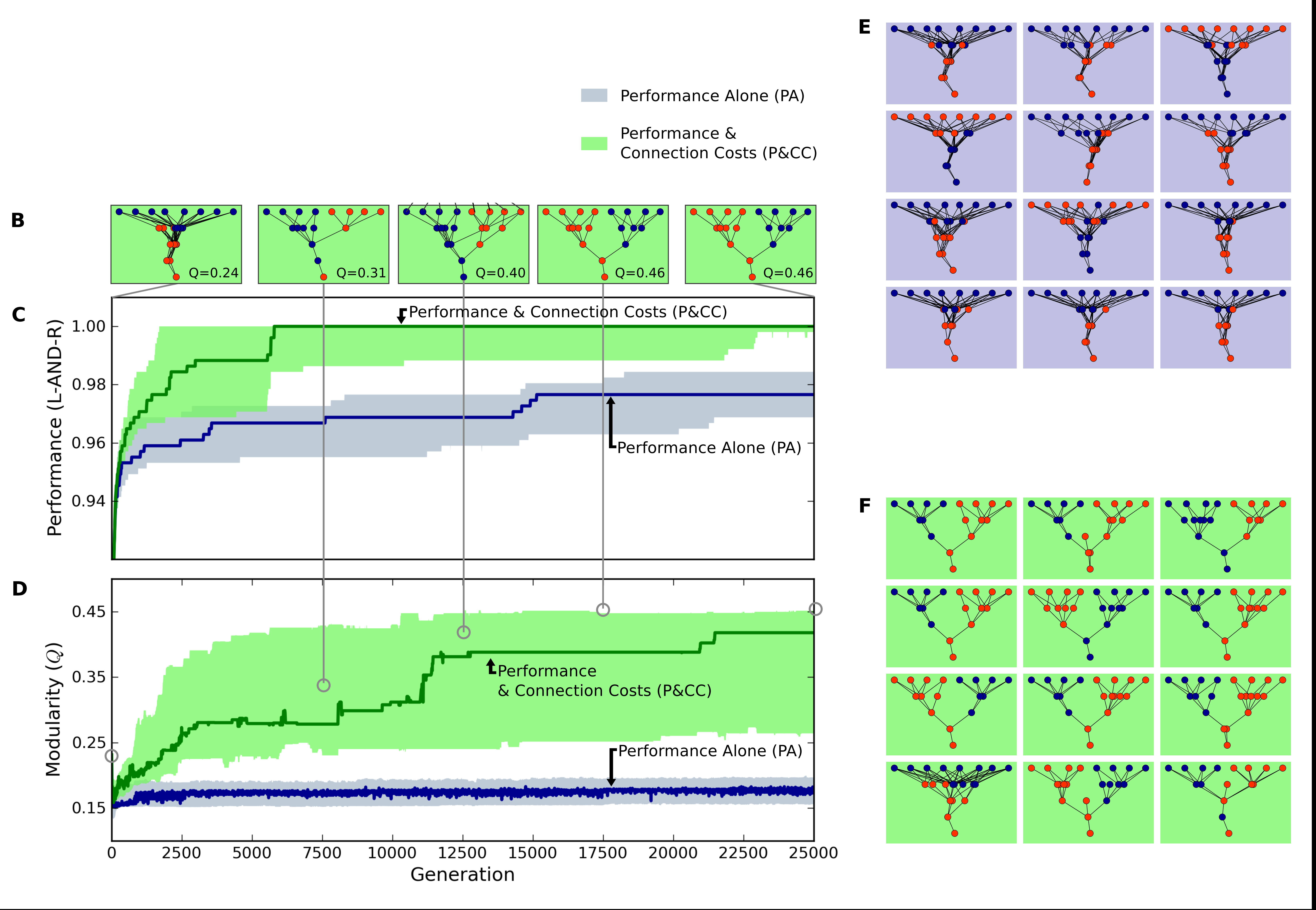


Summary



- Performance Alone (PA)
- Performance & Connection Costs (P&CC)

Clune, Mouret, & Lipson. 2013. Proceedings of the Royal Society



- P&CC significantly more modular, higher-performing ($p < 0.0001$)
- Perfect decomposition in 56% of P&CC, never for PA ($p < 0.0001$)
- Significantly more evolvable ($p < 0.0001$)

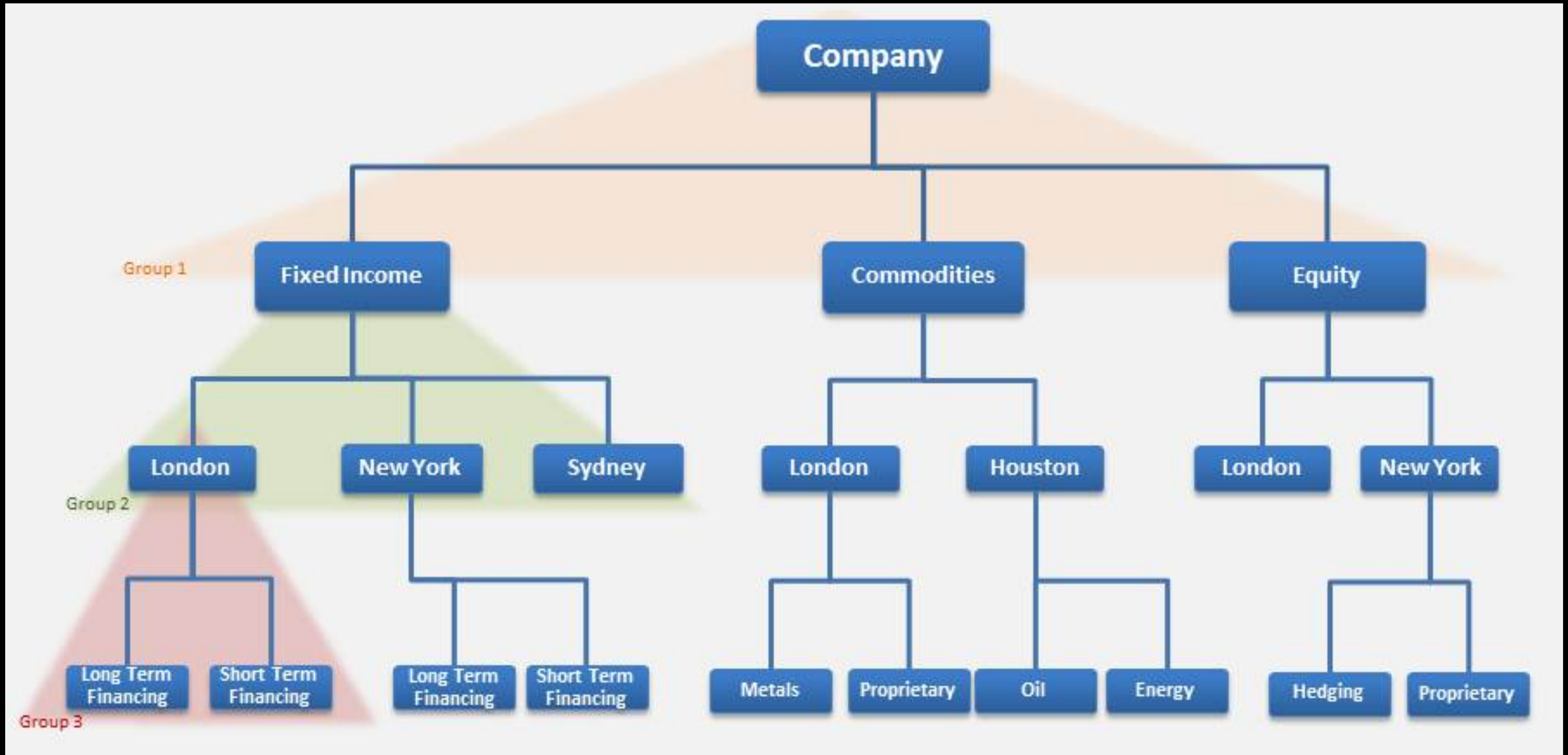
Clune, Mouret, & Lipson. 2013. Proceedings of the Royal Society

Evolution of Structural Organization

- Modularity
- Hierarchy

Hierarchy

- recursive composition of lower-level units (Lipson 2007)
- important principle in brains
- also doesn't occur in evolution by default



Mengistu, Huizinga, Mouret & Clune. 2016. PLoS Comp. Bio.

Evolutionary Origins of Hierarchy

2016. PLoS Comp. Bio.

Henok Mengistu

Joost Huizinga

Jean-Baptiste Mouret

Jeff Clune

Hypothesis: Connection Costs also Cause Hierarchy

- Hierarchical networks are
  - sparse
  - composed of nested modules

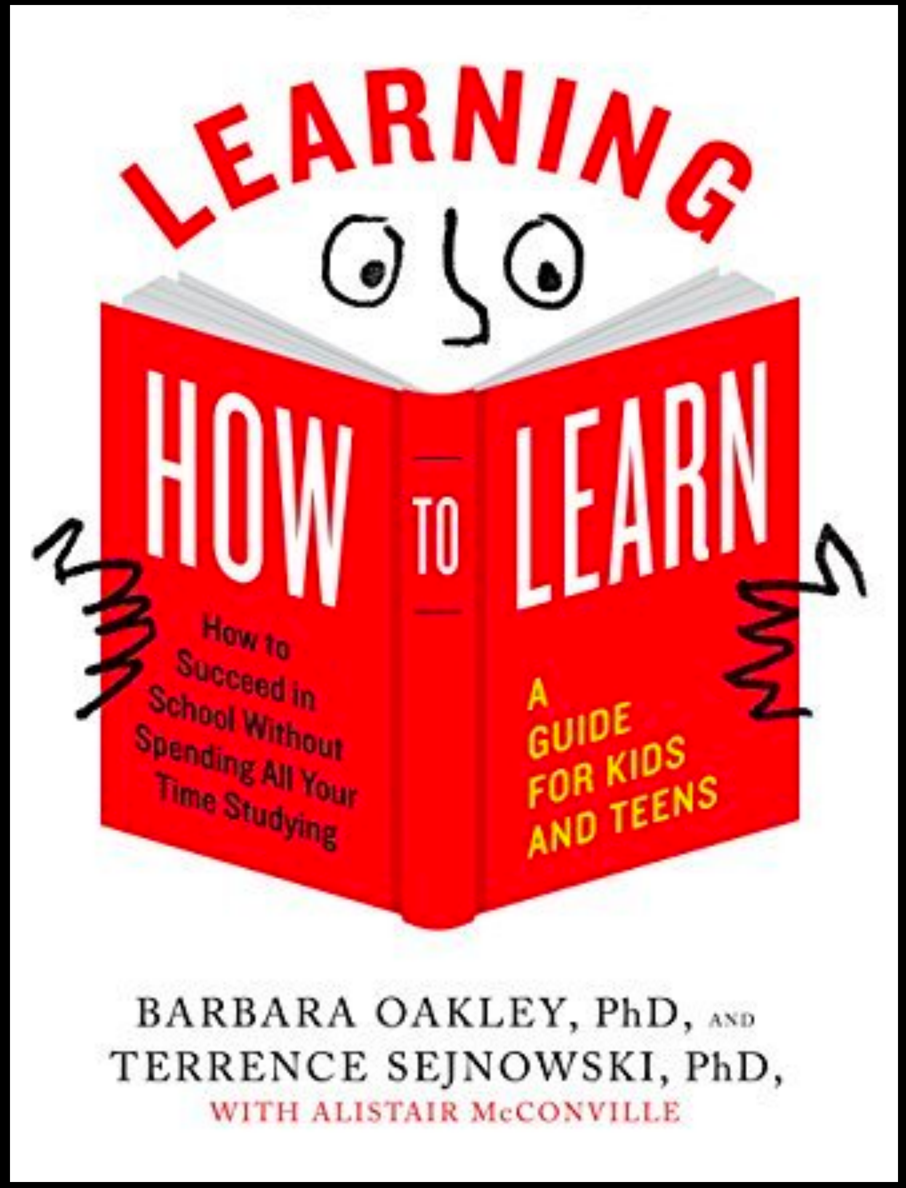
PLoS COMPUTATIONAL BIOLOGY
An official journal of the International Society for Computational Biology
www.ploscompbiol.org

Mengistu, Huizinga, Mouret & Clune. 2016. PLoS Comp. Bio. To appear.

Talk Overview

- Describe alternatives to conventional neural networks loosely inspired by biology
  - that can improve meta-learning, continual learning
- Deep dives
  - Differentiable Hebbian Plasticity
  - Differentiable Neuromodulated Hebbian Plasticity (“backpropamine”)
  - ANML

Meta-Learning Algorithms



- Two major camps
  - Meta-learn good initial weights + SGD
    - e.g. MAML, Finn et al. 2017
  - Meta-learn RNN, which creates its own learning algorithm
    - Learning to Reinforcement Learn, Wang et al. 2016
    - RL^2, Duan et al. 2016

Meta-Learning Algorithms

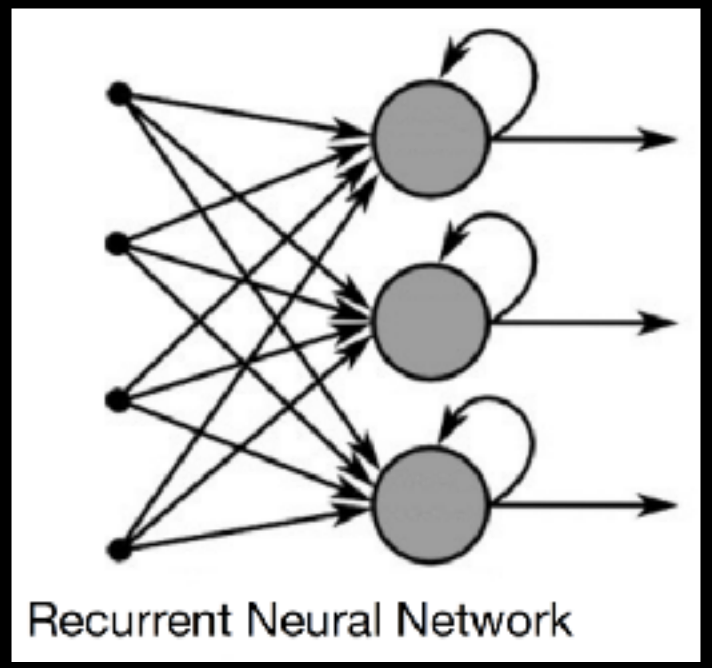
- Two major camps
  - Meta-learn good initial weights + SGD
    - e.g. MAML, Finn et al. 2017
  - Meta-learn RNN, which creates its own learning algorithm
    - Learning to Reinforcement Learn, Wang et al. 2016
    - RL^2, Duan et al. 2016

LRL

- Outer loop: optimize RNN with parameters θ for “lifetime” performance
  - Inner loop: run θ (with reward as input)

S_t

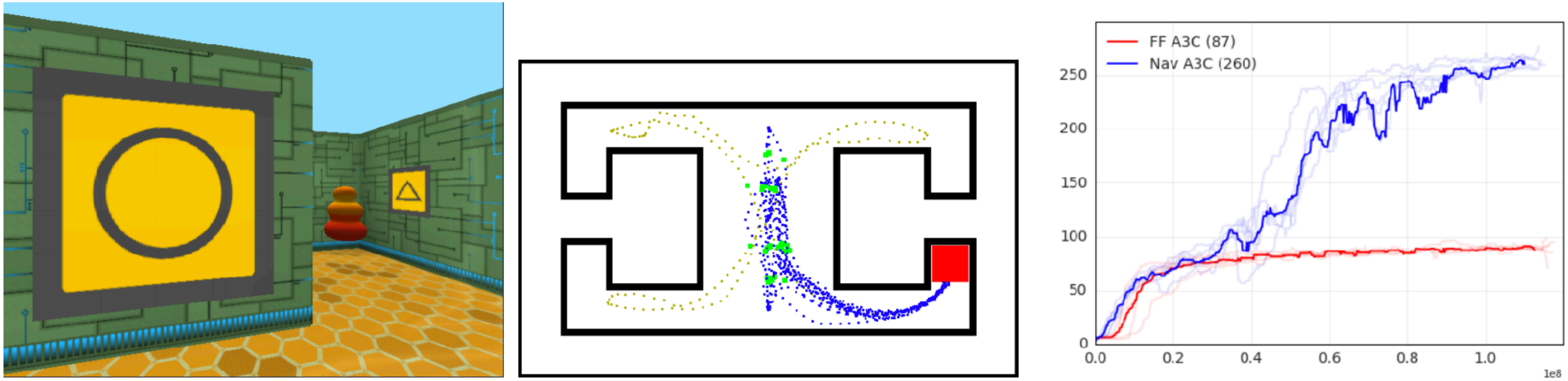
R_{t-1}



A_t

- Et voila!
  - It learns an entire RL algorithm
  - Theoretically can learn any RL algorithm

LRL



(a) Labryinth I-maze

(b) Illustrative Episode

(c) Performance

RNN

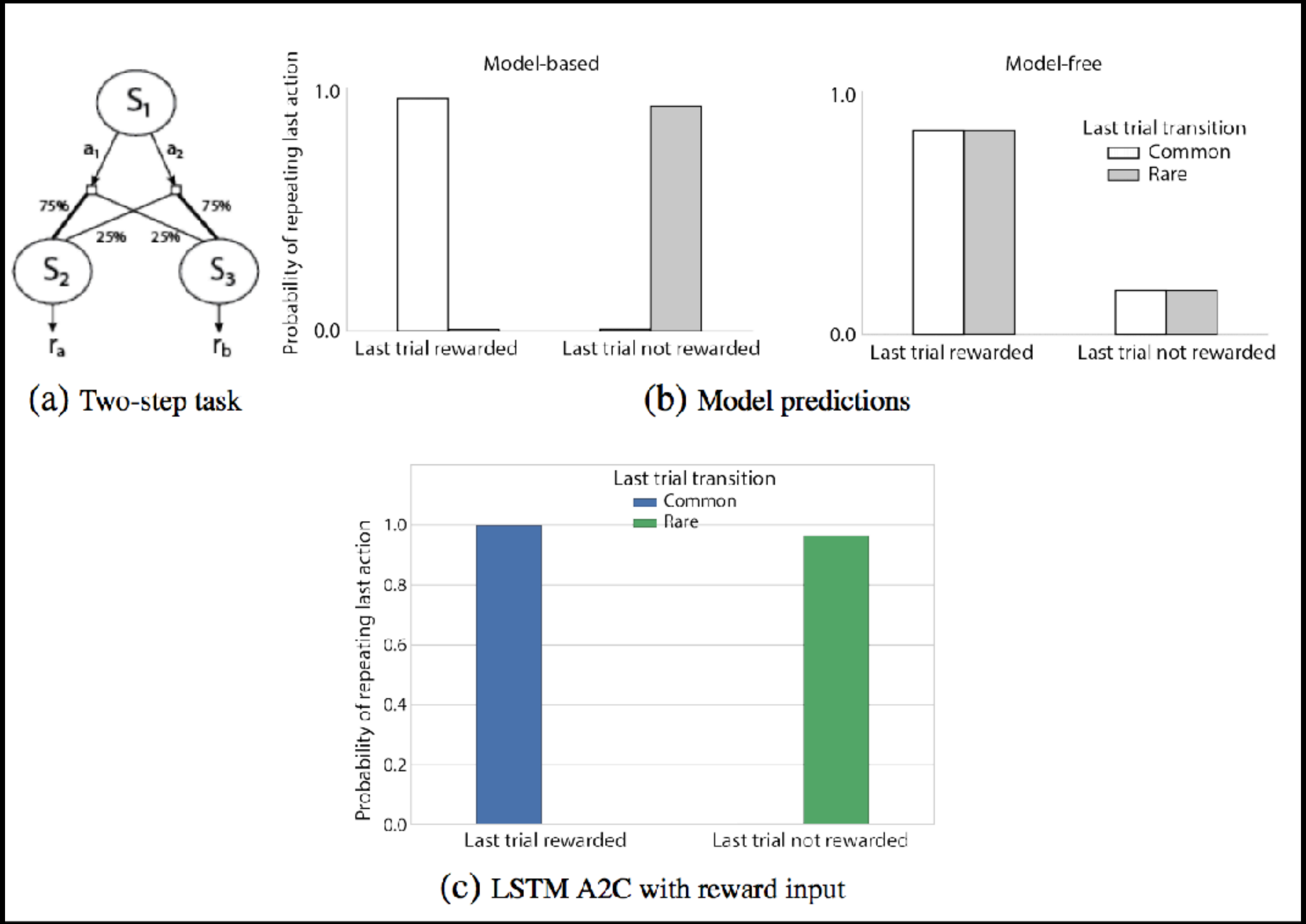
Feedforwar
NN

Mirowski et al. 2016, Wang et al. 2016

Learns to

- explore
- exploit
- all on its own!

LRL



(a) Two-step task

(b) Model predictions

(c) LSTM A2C with reward input

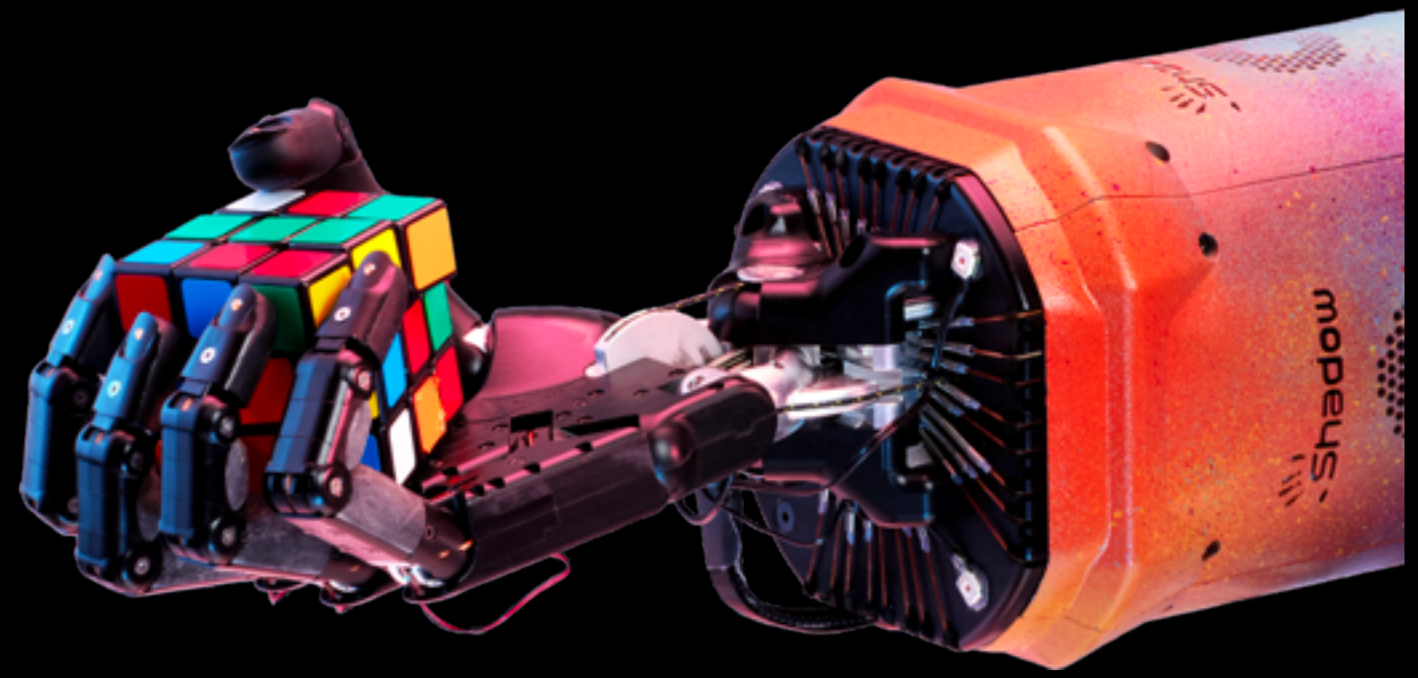
Wang et al. 2016

Learns to

- build a model
- plan
- all on its own!

Rubik's Cube

- Identifies properties of the world
  - friction, mass of cube, size of cube, etc.
- Exploits that information



OpenAI et al. 2019

Materials Matter

- Still have to decide the materials of the network
- RNNs forced to do all lifetime learning with activations
  - may be unstable
  - proposal: store information in weights too

Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

Thomas Miconi

Jeff Clune

Ken Stanley

UBER AI Labs

Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

- Can store info in weights (in addition to activations)
- Hebbian learning (trained via SGD)

Hebbian Learning

- neurons that fire together, wire together

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta x_i^t x_j^t$$

- many capabilities
  - unsupervised learning (e.g. PCA)
  - associative recall
  - …

Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

- Recurrent, Hebbian network
  - inner loop: network updates with no SGD
  - outer loop: differentiate through episode, update trainable parameters via SGD

fixed part

plastic part

$$y_j = \tanh\left\{ \sum_{i \in inputs} \left(w_{i,j} + \alpha_{i,j} \mathrm{H}_{i,j}(t) \right) y_i \right\}$$

$$\mathrm{H}_{i,j}(t+1) = \eta \, y_i \, y_j + (1 - \eta) \, \mathrm{H}_{i,j}(t)$$

$w_{i,j}$ $\alpha_{i,j}$

Trainable parameters, optimized by SGD to maximize lifetime/ episode reward

$H_{i,j}$

Lifetime quantity (init=0)

Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

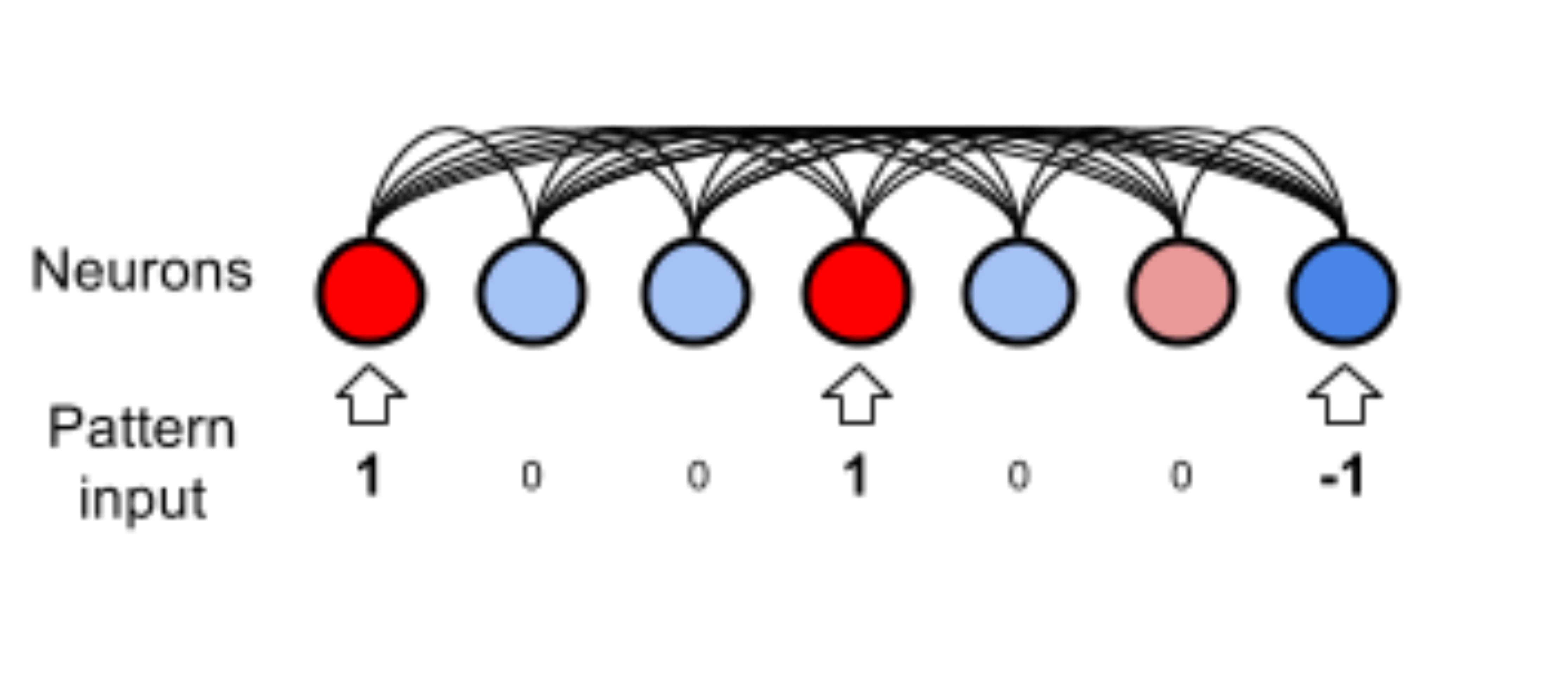
- Near then-SOTA on Omniglot

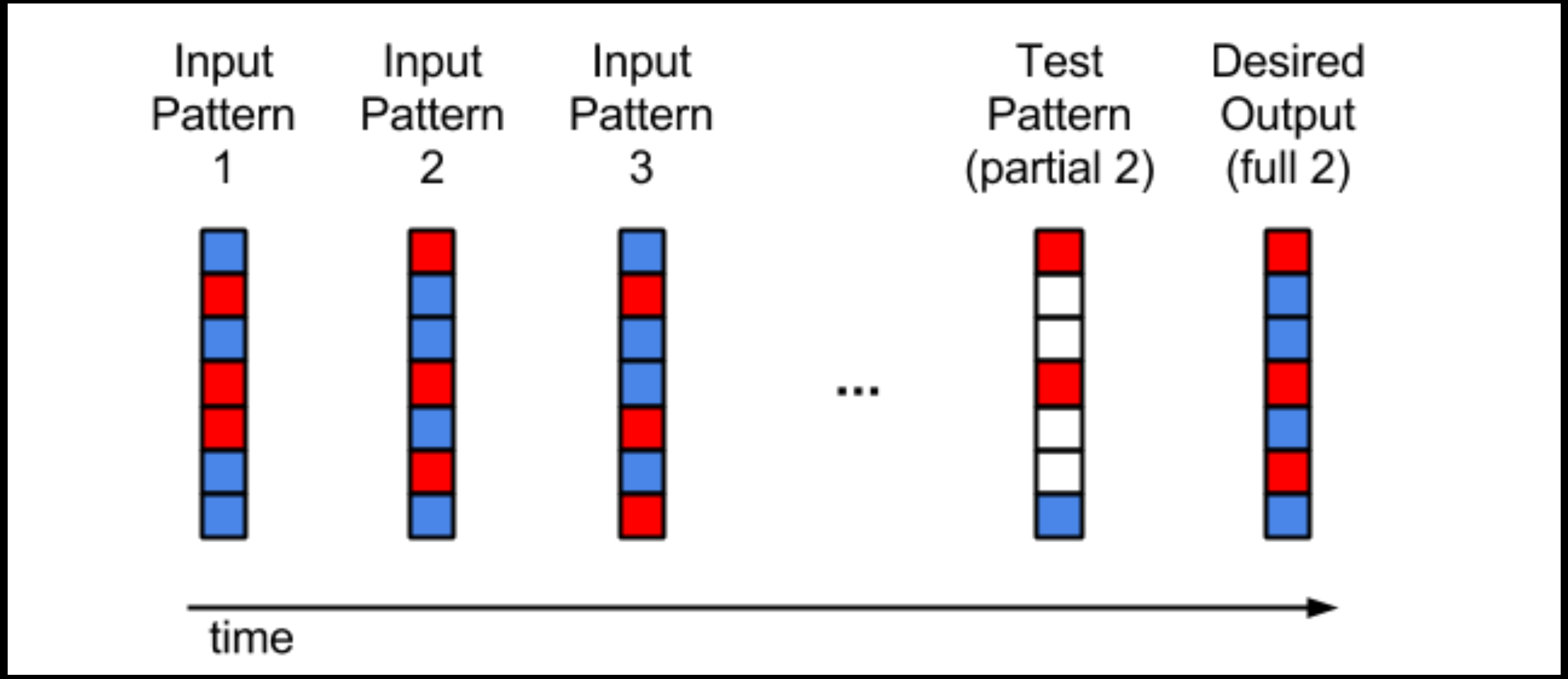
Table 1: Results for the 5-way, 1-shot omniglot tasks, including recent reported results and the new differentiable plasticity (DP) result (± indicates 95% CI). Note that these reports describe widely varying approaches and model sizes (see text).

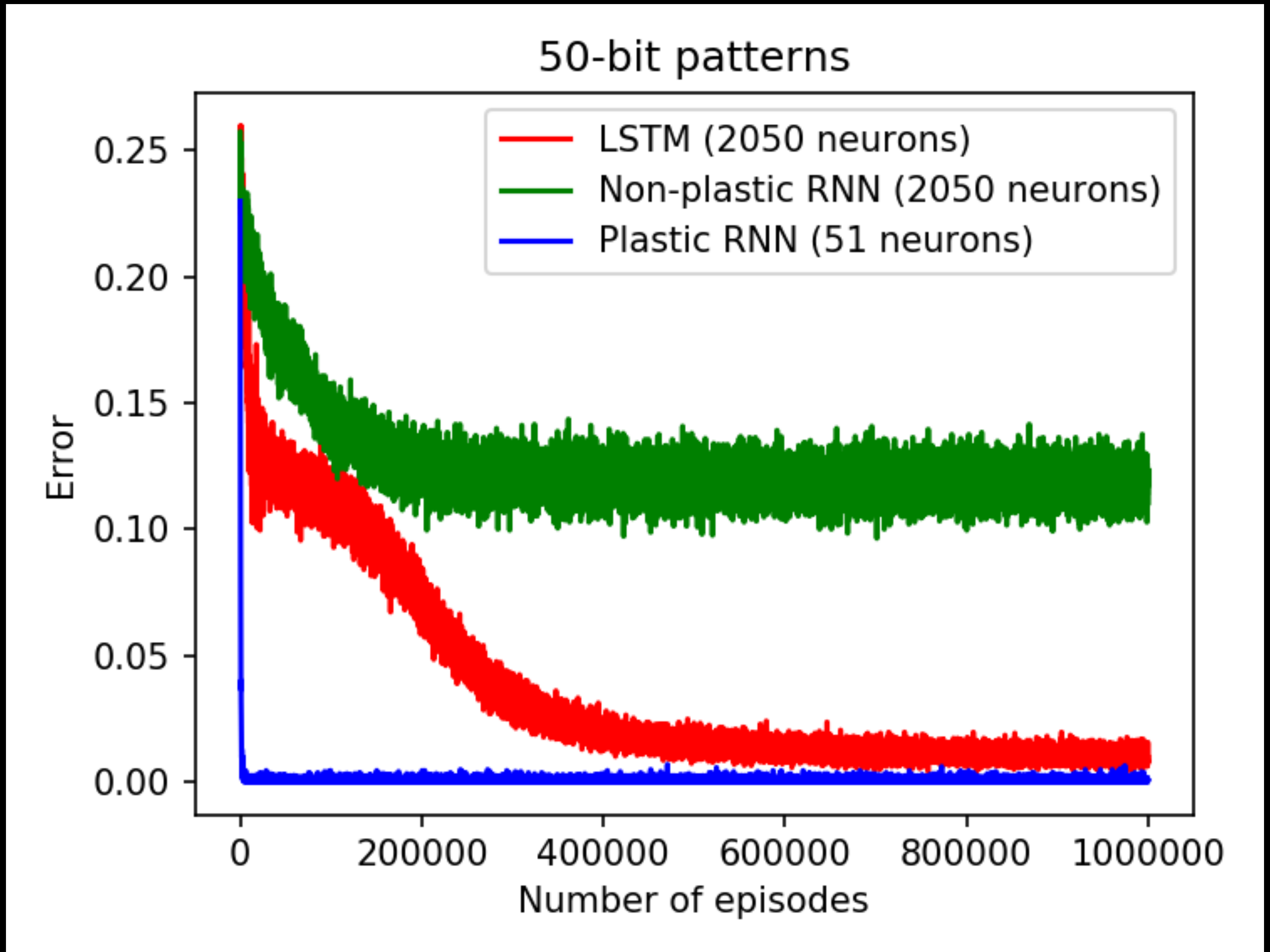
VINYALS ET AL. (MATCHING NETWORKS) (VINYALS ET AL., 2016)	SNELL ET AL. (PROTONETS) (SNELL ET AL., 2017)	FINN ET AL. (MAML) (FINN ET AL., 2017)	MISHRA ET AL. (SNAIL) (MISHRA ET AL., 2017)	DP (OURS)
98.1 %	97.4%	98.7% ± 0.4%	99.07% ± 0.16	98.5% ± 0.57

Differentiable Hebbian Plasticity

Miconi, Clune, Stanley, ICML 2018







Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

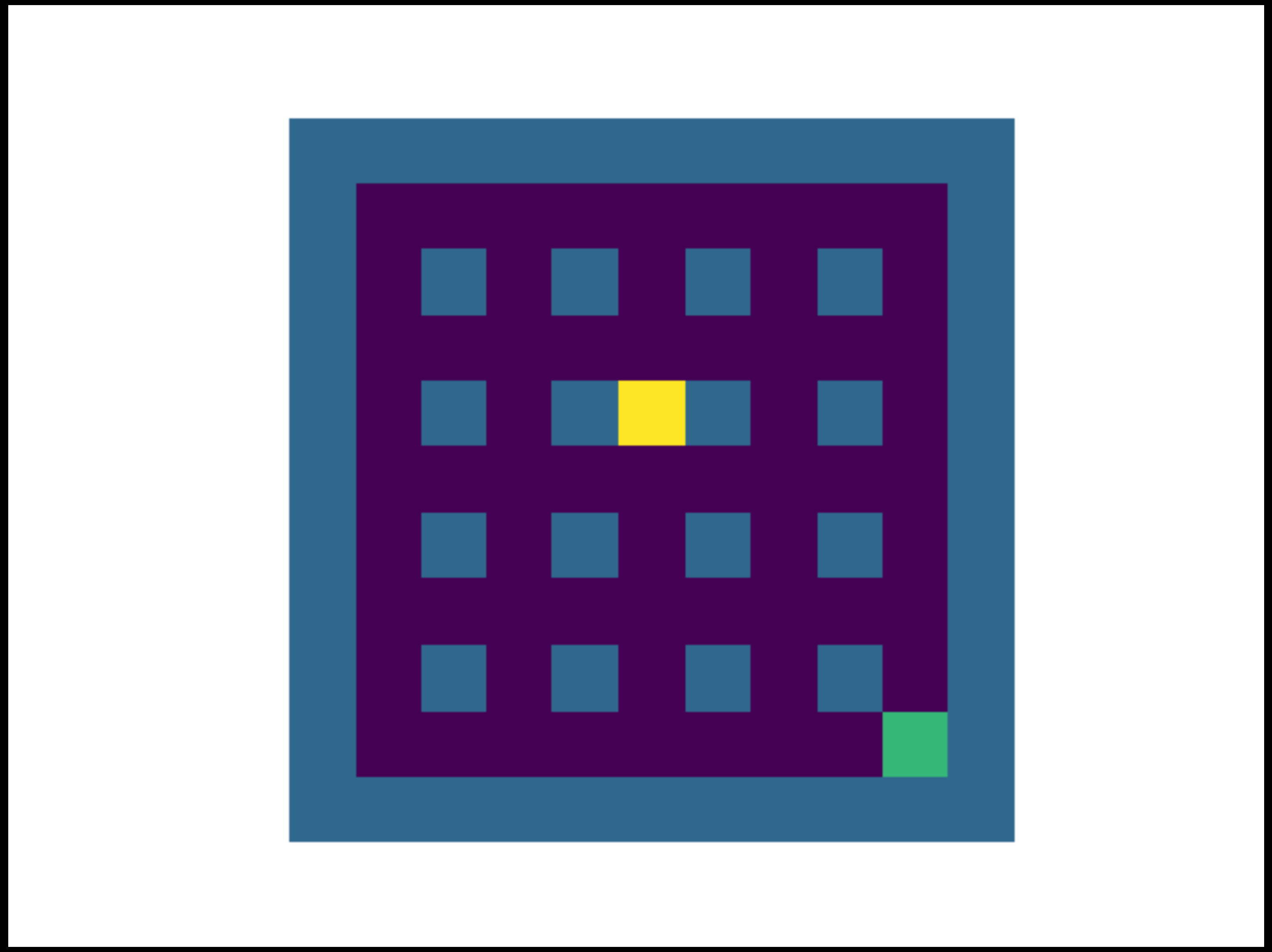
- Image reconstruction: learn (memorize) an image, reconstruct it
- 2M+ parameters

LSTMs cannot solve this

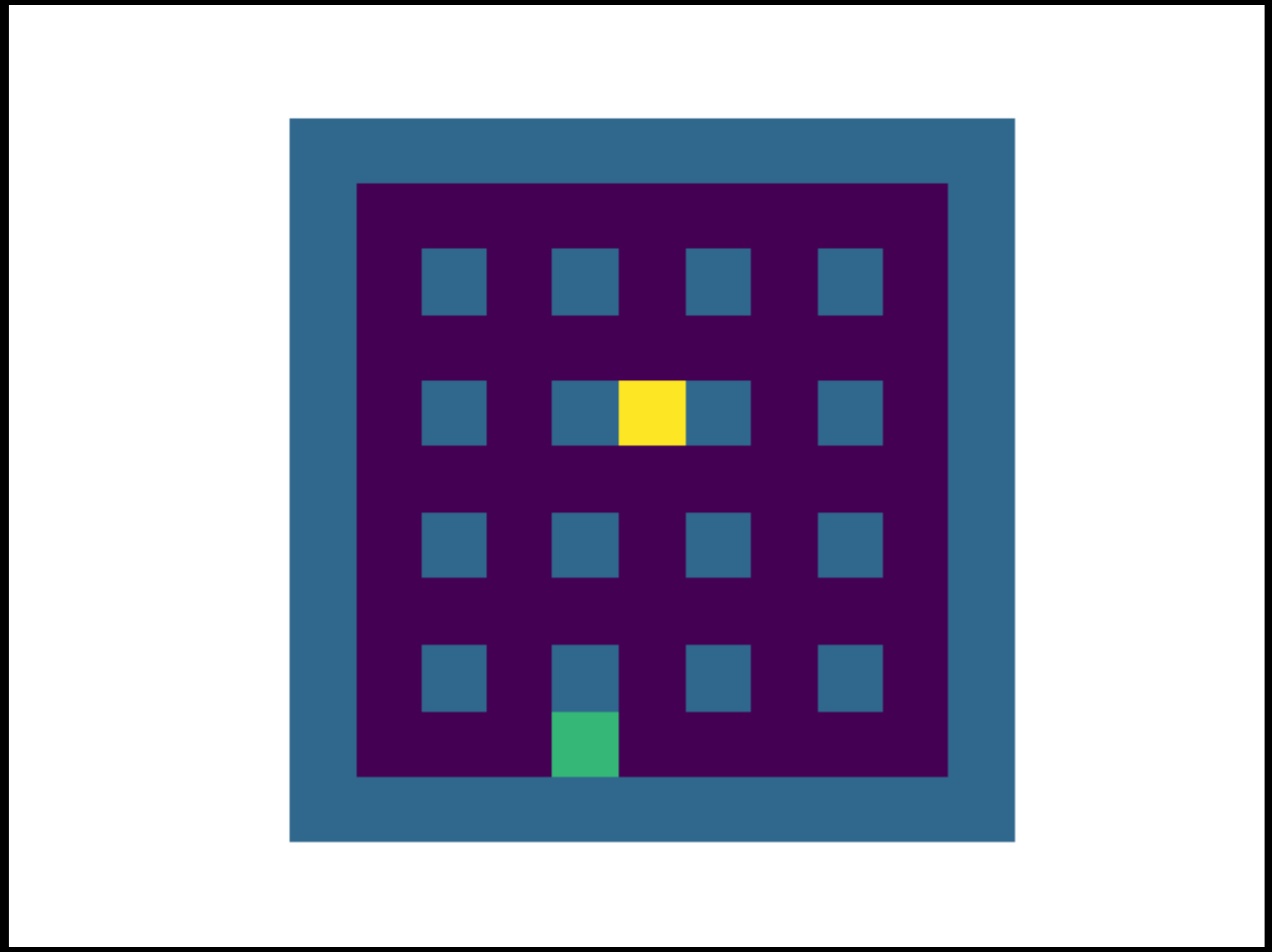
Differentiable Hebbian Learning

Differentiable plasticity: training plastic neural networks with backpropagation
Miconi, Clune, Stanley. ICML. 2018

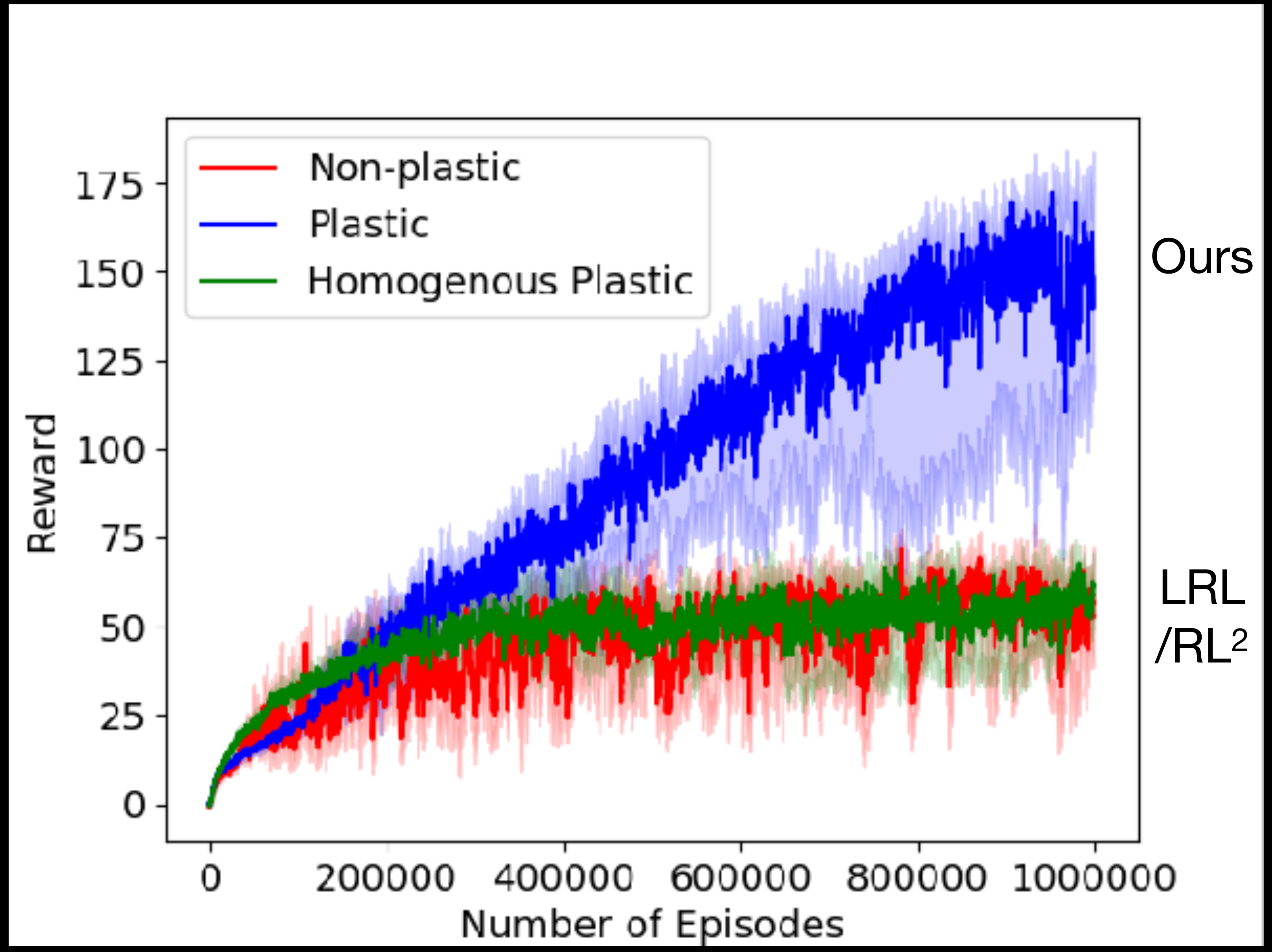
Maze Navigation



Episode 0



Episode 500,000

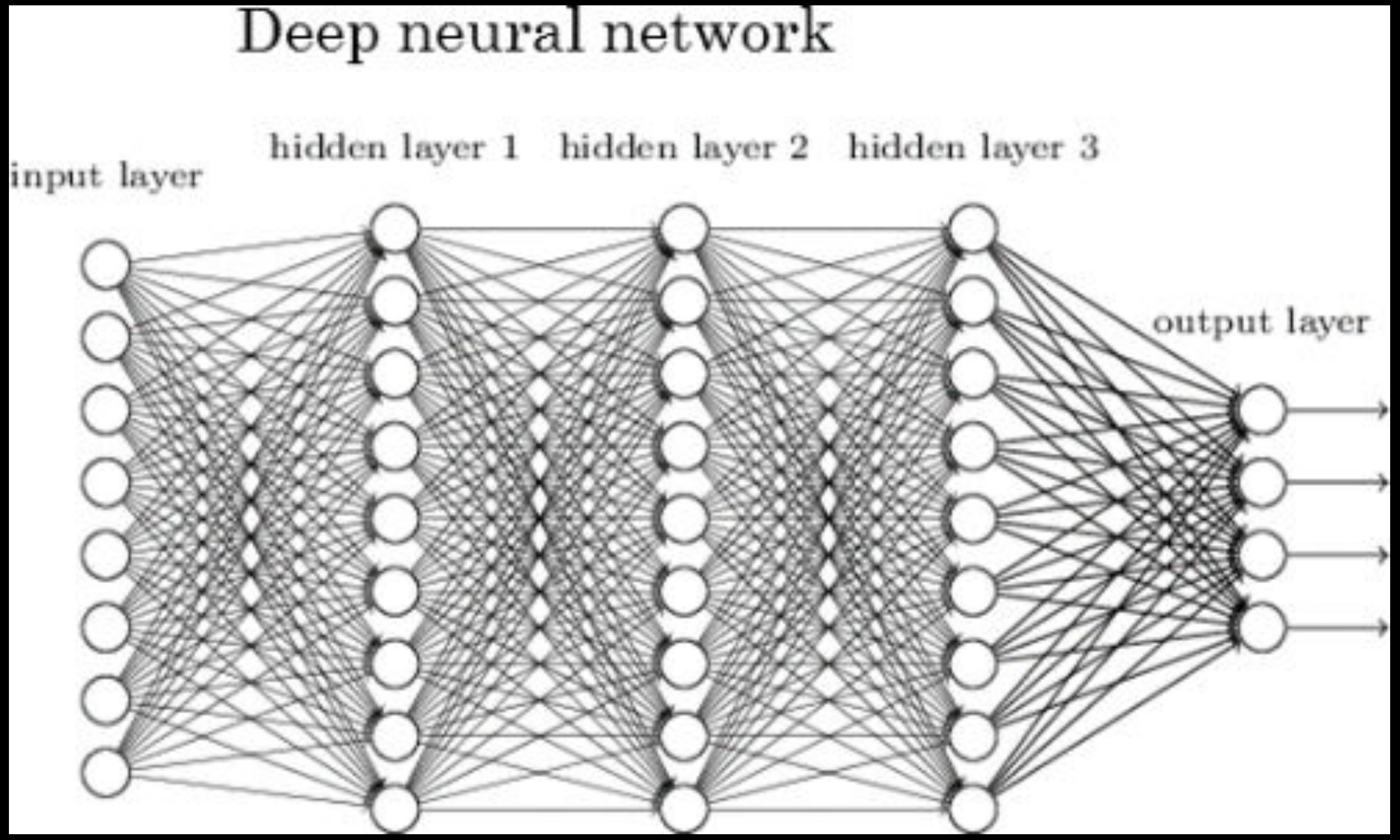


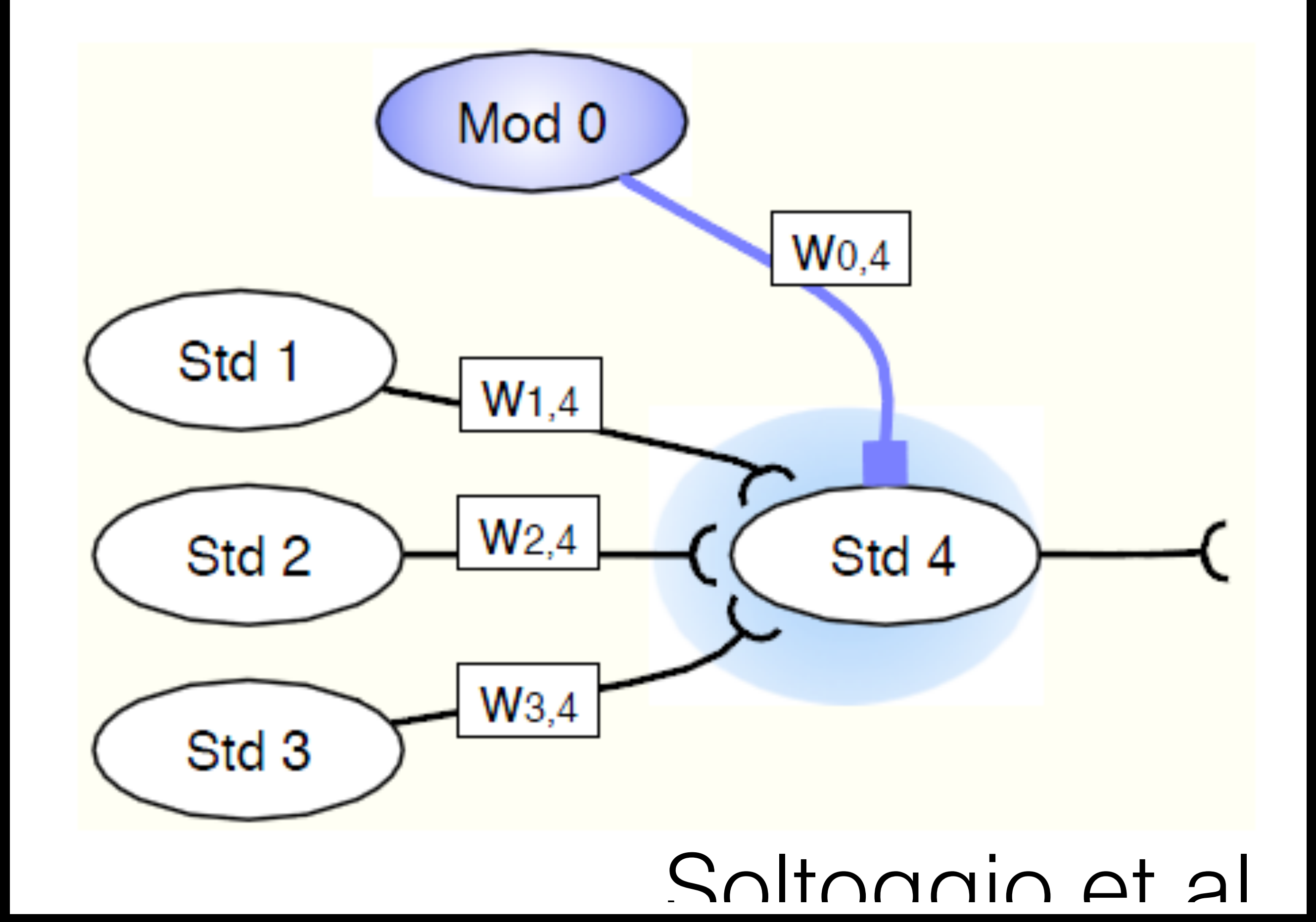
Learned to Explore & Exploit (Better)

Differentiable Neuromodulated Plasticity

"Backpropamine": Miconi, Rawal, Clune, Stanley, ICLR, 2018

- Hebbian learning is local (hard optimization problem)
- Better: turn learning on in *some* weights *only* in certain contexts
  - e.g. if I am playing chess AND I just won, THEN:
    - increase learning in only chess playing parts of the brain





Differentiable Neuromodulated Plasticity

“Backpropamine”: Miconi, Rawal, Clune, Stanley, 2018

Hebbian Learning

$$x_j(t) = \sigma\{\sum_{i \in \text{inputs to } j} (w_{i,j} + \alpha_{i,j}\text{Hebb}_{i,j}(t))x_i(t-1)\}$$
$$\text{Hebb}_{i,j}(t+1) = \text{Clip}(\text{Hebb}_{i,j}(t) + \eta x_i(t-1)x_j(t)),$$

Neuromodulated Hebbian Learning

new
part

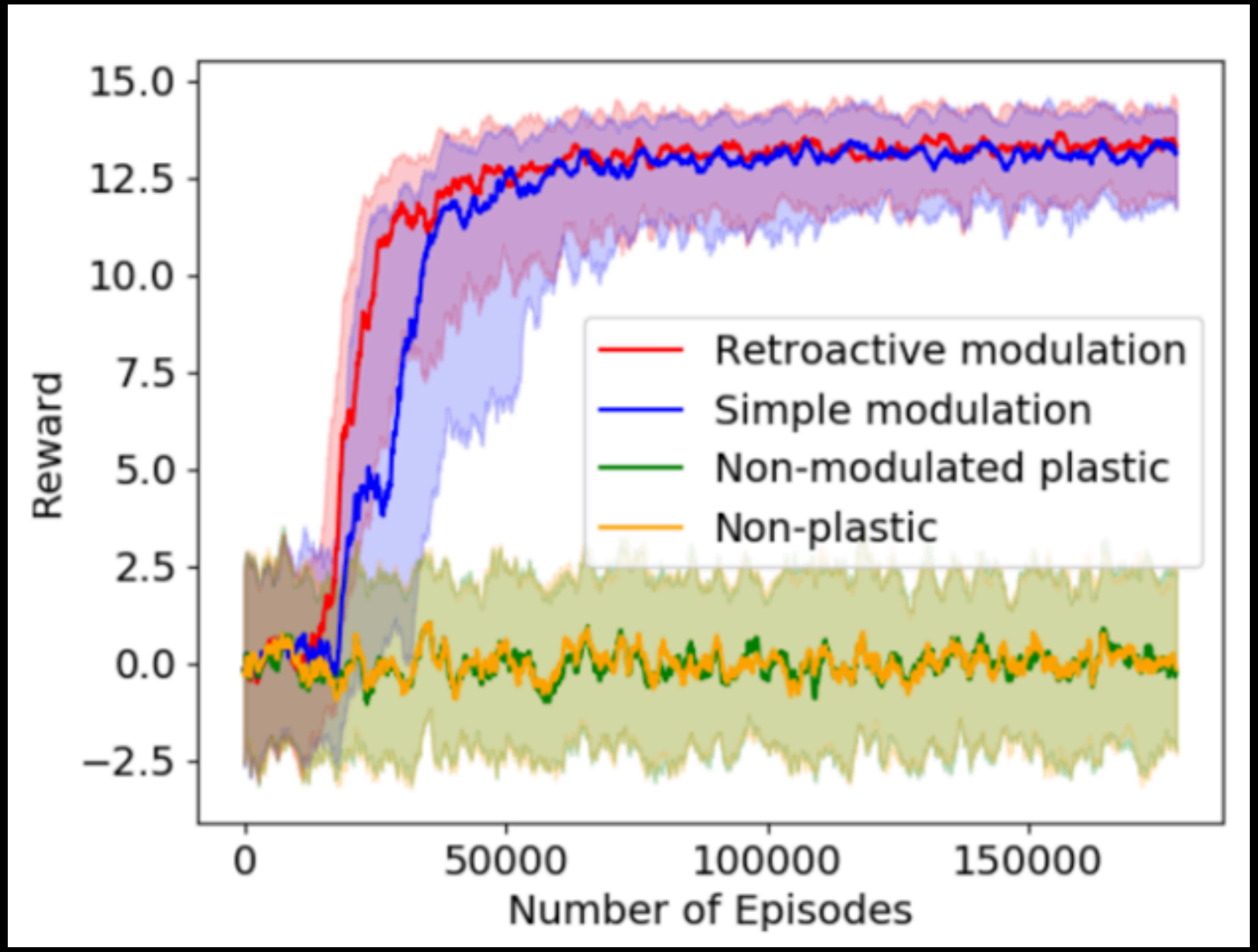
$$\text{Hebb}_{i,j}(t+1) = Clip(\text{Hebb}_{i,j}(t) + M(t)x_i(t-1)x_j(t))$$

$$\text{Hebb}_{i,j}(t+1) = Clip(\text{Hebb}_{i,j}(t) + M(t)E_{i,j}(t))$$
$$E_{i,j}(t+1) = (1-\eta)E_{i,j}(t) + \eta x_i(t-1)x_j(t)).$$

Eligibility Trace Version

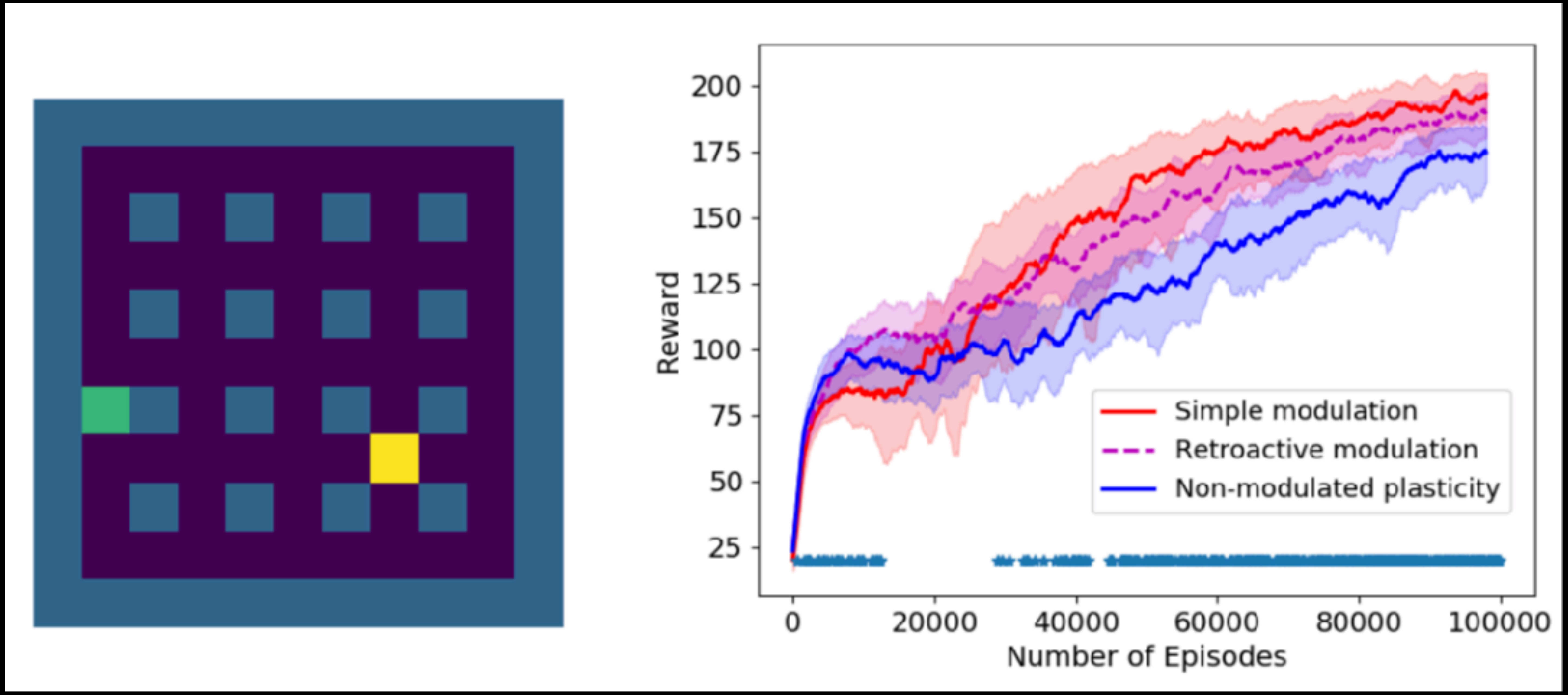
Differentiable Neuromodulated Plasticity

"Backpropamine": Miconi, Rawal, Clune, Stanley, 2018



Simple Task

network says if one of the symbols just shown is the secret symbol



$p < 0.05$
NM vs. Non

Model	Test Perplexity
Baseline LSTM (Zaremba et al., 2014)	104.26 ± 0.22
LSTM with Differential Plasticity	103.80 ± 0.25
LSTM with Simple Neuromodulation	102.65 ± 0.30
LSTM with Retroactive Neuromodulation	102.48 ± 0.28

Word prediction, Penn-Tree Bank

Learning to Continually Learn

Shawn Beaulieu

Lapo Frati

Joel Lehman

Thomas Miconi

Ken Stanley

Jeff Clune*

Nick Cheney*

*co-senior authors

ECAI 2020

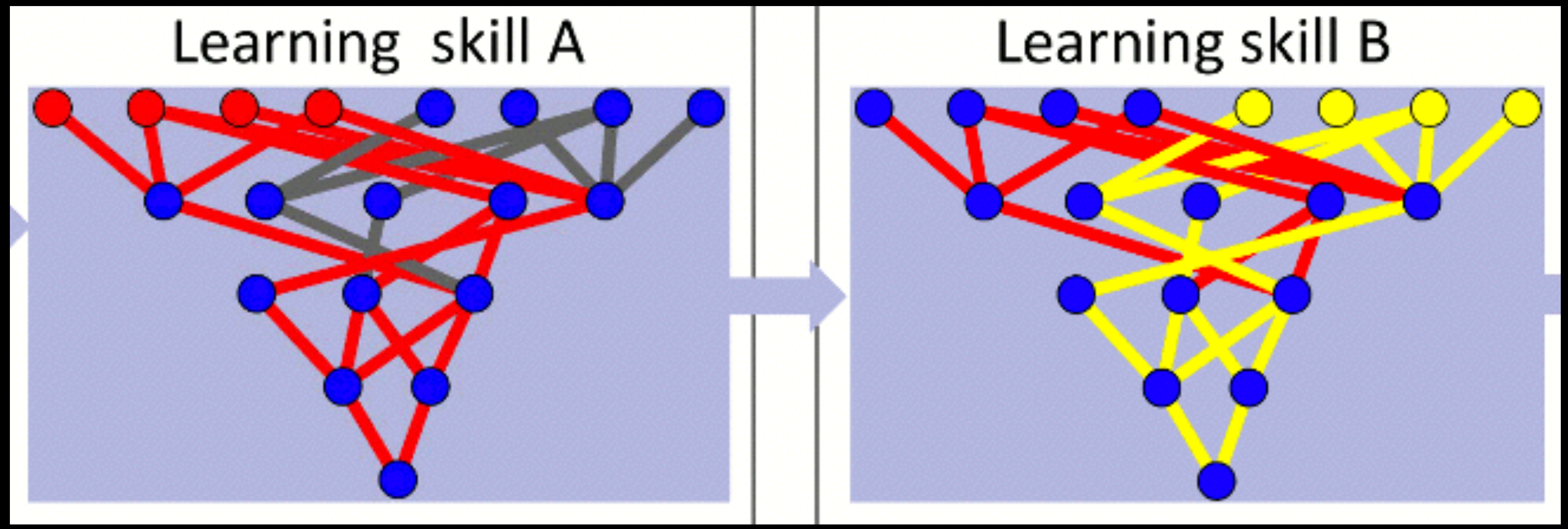
The University of Vermont

DARPA

UBER AI Labs

Catastrophic Forgetting

- Achilles Heel of machine learning
- In sequential learning
  - Learn task A, then learn task B
  - ML overwrites A when learning B
    - forgets catastrophically
  - Animals, including humans
    - pick up where we left off
    - forget gradually
- Must solve catastrophic forgetting to continually learn



Many Proposed Solutions: All Manual

- Rehearsal techniques
- Pseudo-patterns
- Activation sharpening
- Sparse representations
- Progressive networks
- Elastic weight consolidation
- PathNet
- Intelligent synapses
- Experience replay
- Generative replay
- Progress & Compress
- etc.

Many Proposed Solutions: All Manual

- Rehearsal techniques
- Pseudo-patterns
- Activation sharpening
- Sparse representations
- Progressive networks
- Elastic weight consolidation
- PathNet
- Intelligent synapses
- Experience replay
- Generative replay
- Progress & Compress
- etc.

Frequent Manual Path Philosophy

- Optimize for one thing and hope for in other
  - e.g. optimize for sparse representations, hope for decreased catastrophic forgetting

Meta-Learning Philosophy

- Don’t optimize for one thing and hope for another
- Optimize for what you want

Hypothesis

- There's a good chance humans are not smart enough to manually build systems that continually learn well

Proposal: Use meta-learning to learn to continually learn

- Optimize for we what
  - Learn a sequence of tasks
  - Be good on all of them at the end

Meta-Learning Algorithms

- Two major camps
  - Meta-learn good initial weights + SGD
    - e.g. MAML, Finn et al. 2017
  - Meta-learn RNN, which creates its own learning algorithm
    - Learning to Reinforcement Learn, Wang et al. 2016
    - RL^2, Duan et al. 2016
    - OpenAI et al. 2019, Rubik’s Cube

Meta-Learning Algorithms

- Two major camps
  - Meta-learn good initial weights + SGD
    - e.g. MAML, Finn et al. 2017
  - Meta-learn RNN, which creates its own learning algorithm
    - Learning to Reinforcement Learn, Wang et al. 2016
    - RL^2, Duan et al. 2016
    - OpenAI et al. 2019, Rubik's Cube

θ + SGD (e.g. MAML)

“meta-training”

"meta-training"

meta-train training | meta-train testing

meta-training (outer-loop learning)

θ_1 → inner-loop learning: $\theta_1^1 \quad \theta_1^2 \quad \cdots \quad \theta_1^n$ → $\mathscr{L}_{meta}(\theta_1^n)$

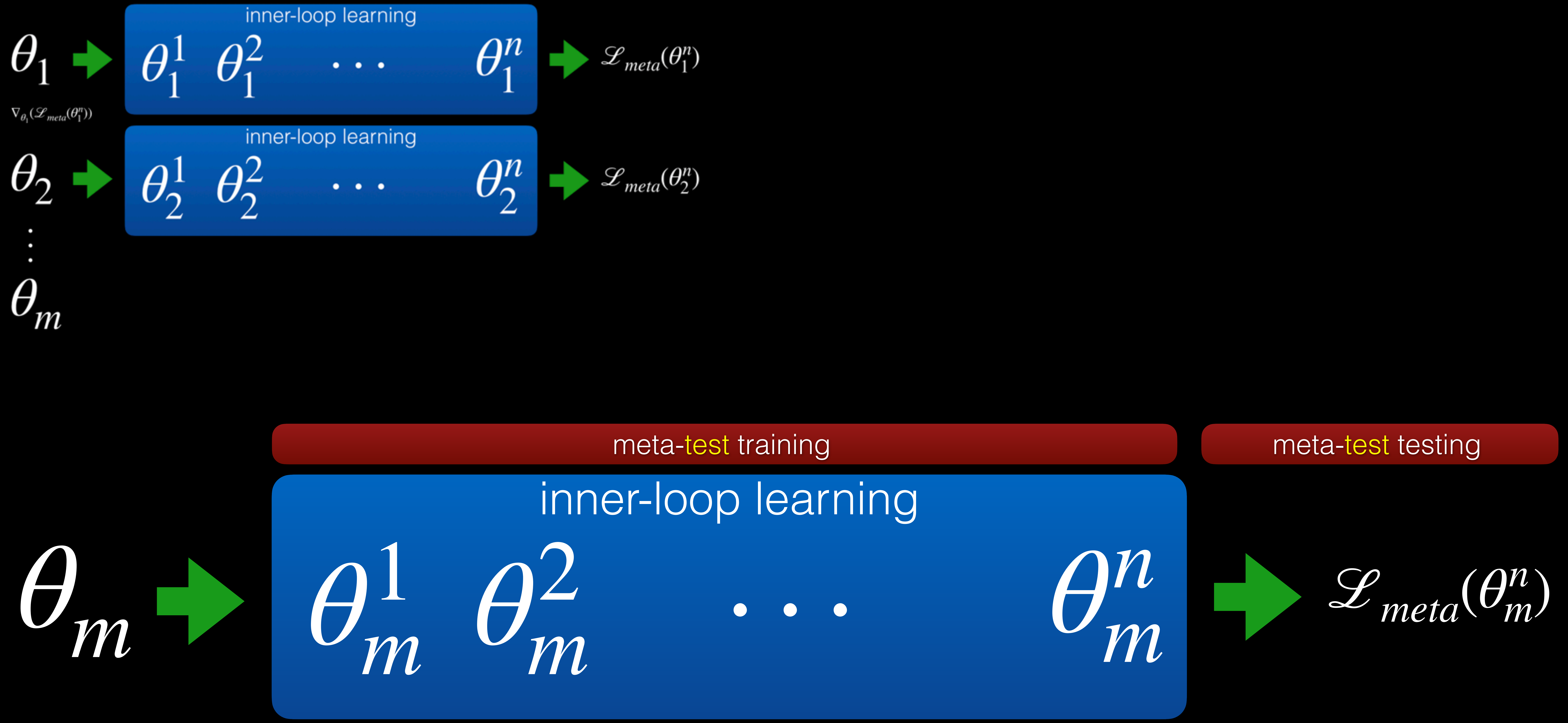
$\nabla_{\theta_1}(\mathscr{L}_{meta}(\theta_1^n))$

θ_2 → inner-loop learning: $\theta_2^1 \quad \theta_2^2 \quad \cdots \quad \theta_2^n$ → $\mathscr{L}_{meta}(\theta_2^n)$

$\vdots$

θ_m

“meta-testing”



meta-learning for continual, multi-task learning

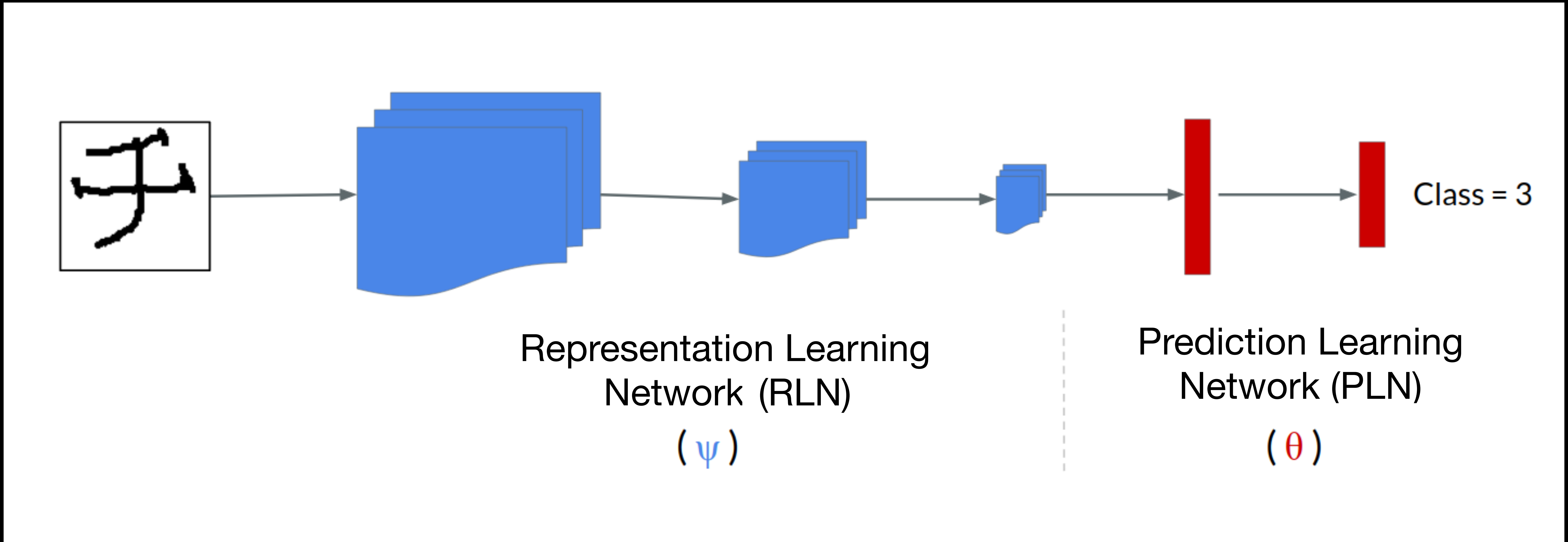
Online-aware Meta-Learning (OML)

Javed & White, NeurIPS, 2019

- validates the vision of meta-learning solutions to continual learning
- we were
  - inspired by it
  - compare to it

OML

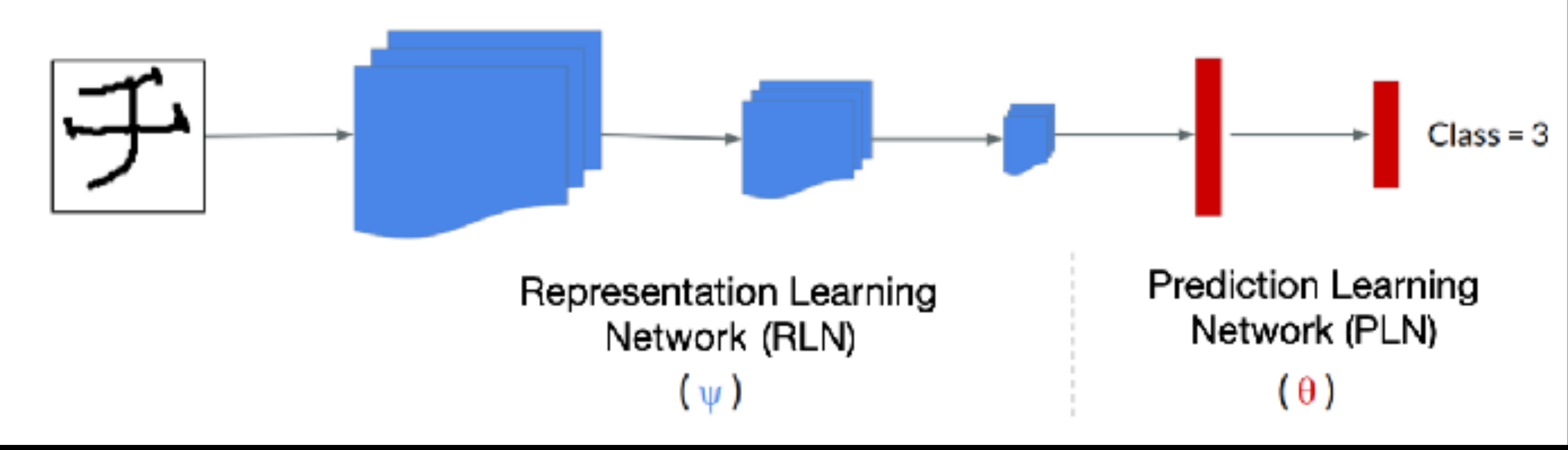
Javed & White 2019



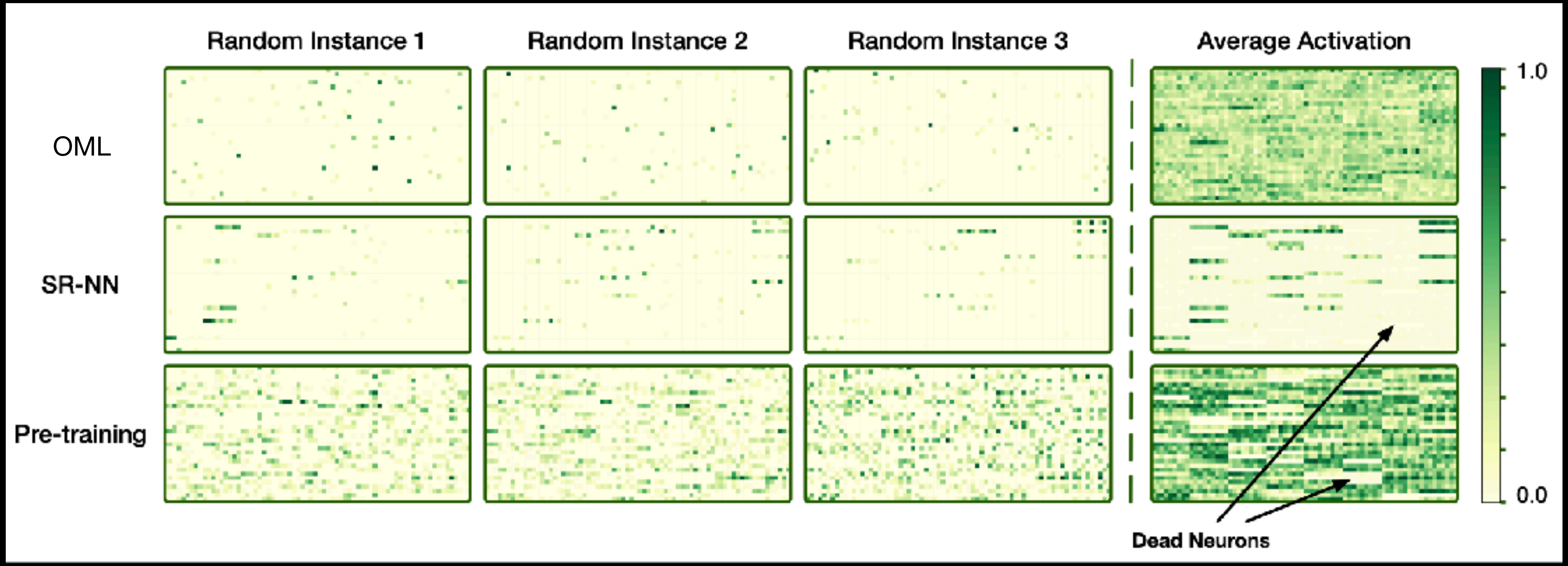
meta-learn then freeze **representation, SGD for PLN**

OML

Javed & White 2019

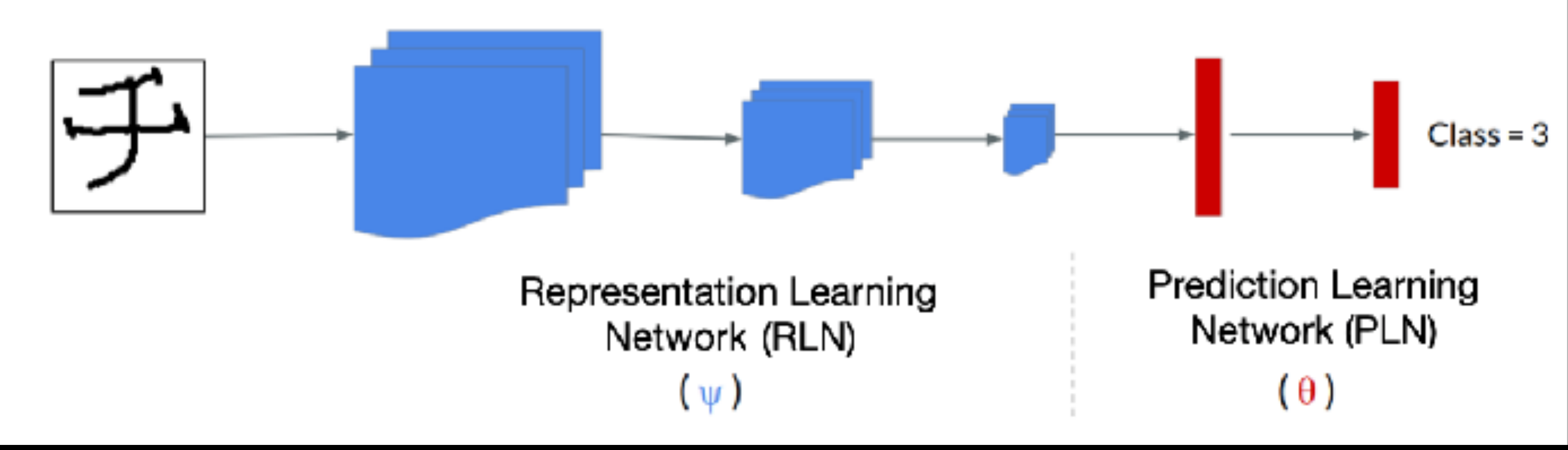


- Performs well
  - After sequentially training on 150 classes of Omniglot
    - 97% on meta-test training set (near-perfect memorization)
    - ~63% on meta-test test set (worse at generalizing, but still impressive)
- Learns a sparse representation



OML

Javed & White 2019



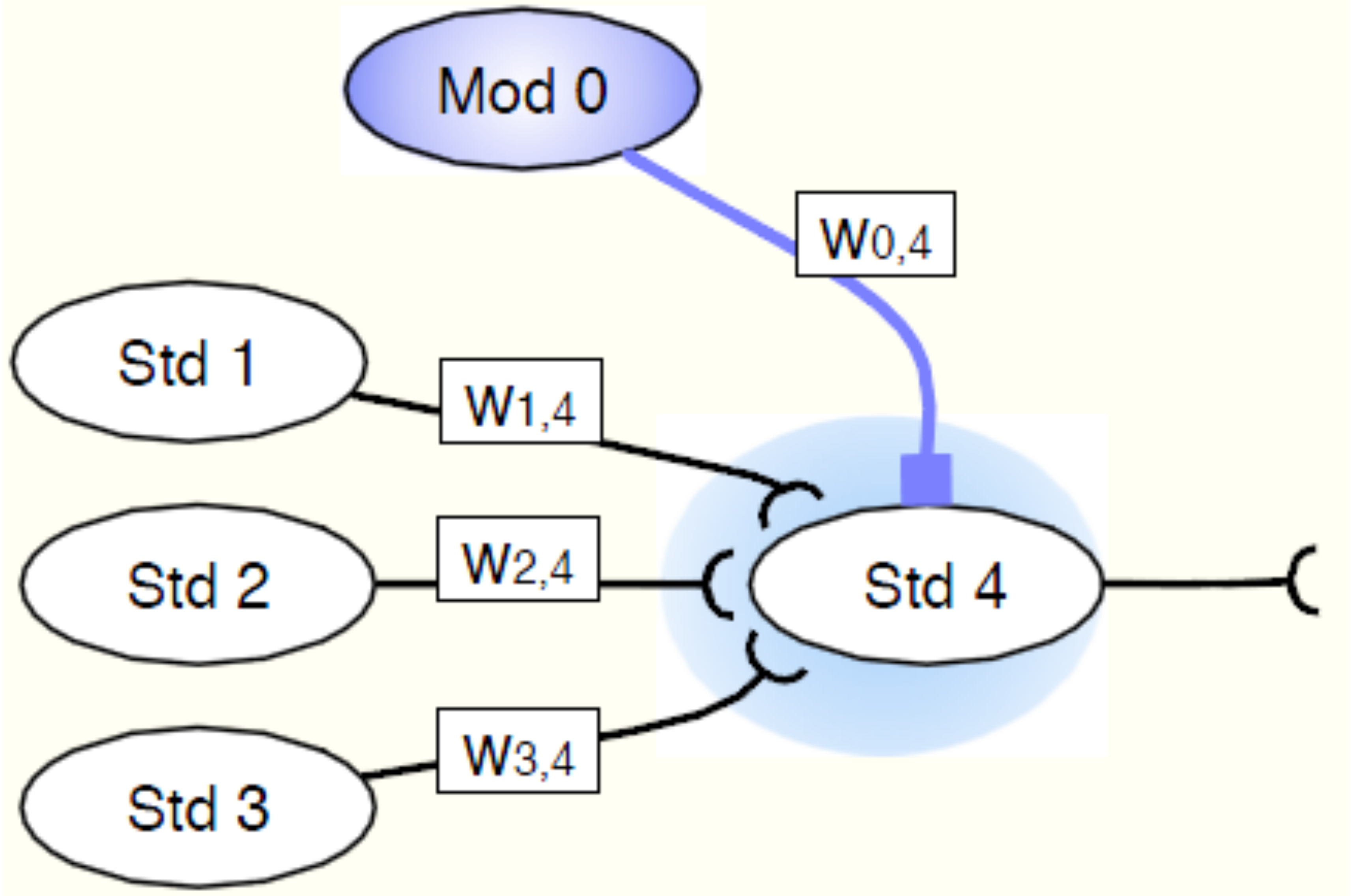
- Gets a lot right
- But is still ultimately subject to SGD
  - which was not optimized for continual learning
  - has to find a representation that avoids CF when SGD is applied

Can we do better?

- We propose: allowing control over SGD via neuromodulation

Traditional Neuromodulation

- NM neurons change learning rates in other neurons
- Enables data-dependent, thus task-specific, learning



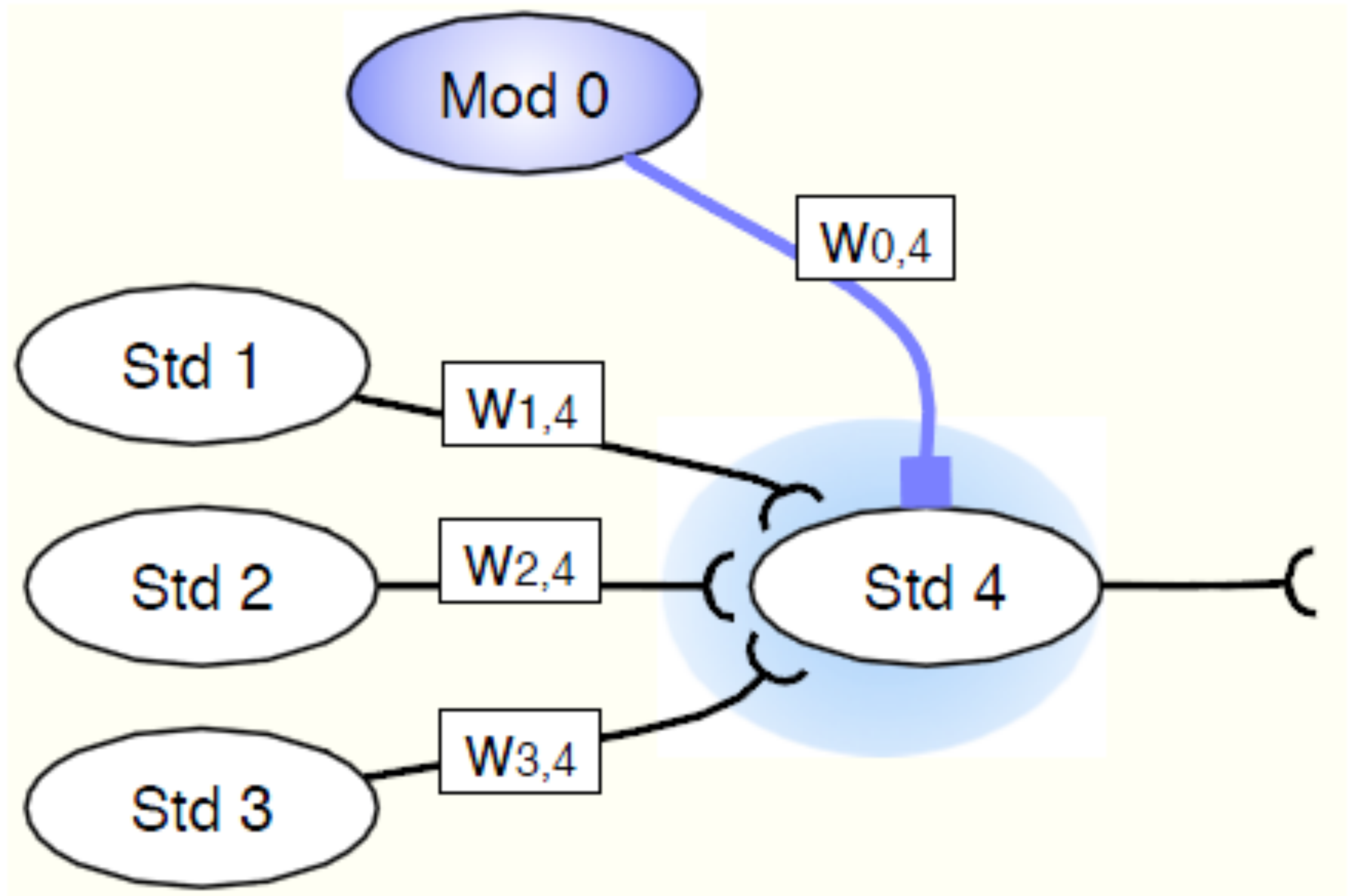
Soltoggio et al. (2008)

Neuromodulation Solves CF on Simple Networks & Problems

- Ellefsen KO, Mouret JB, Clune J. 2015. PLoS Computational Biology
- Velez & Clune. 2017. PLoS One

Scaling Traditional Neuromodulation

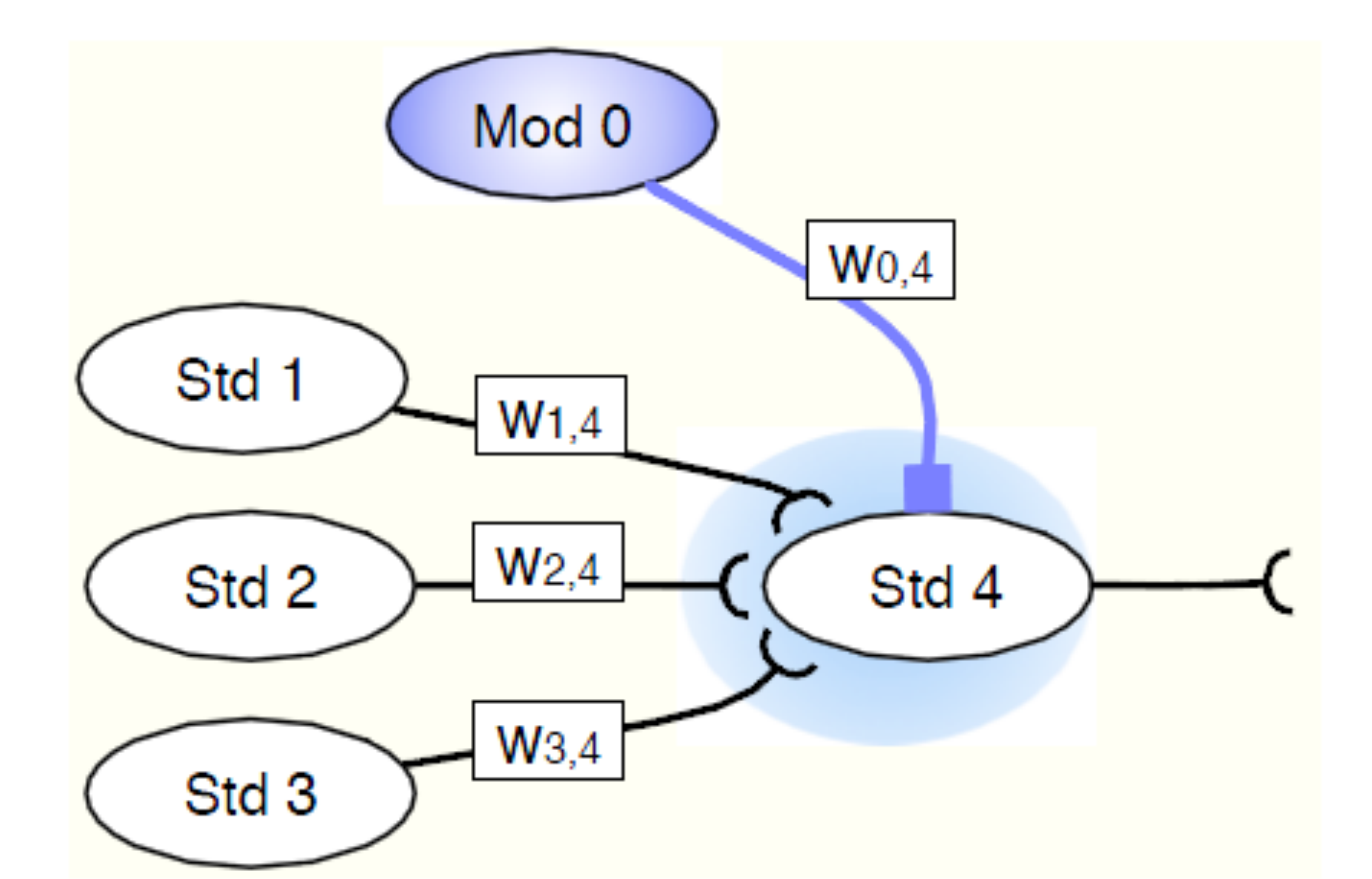
- Struggled to scale it up
- Insight (Shawn Beaulieu)
    - maybe it is because the forward pass is not affected
    - thus forward-pass interference still exists



Soltoggio et al. (2008)

Activation-Based Neuromodulation

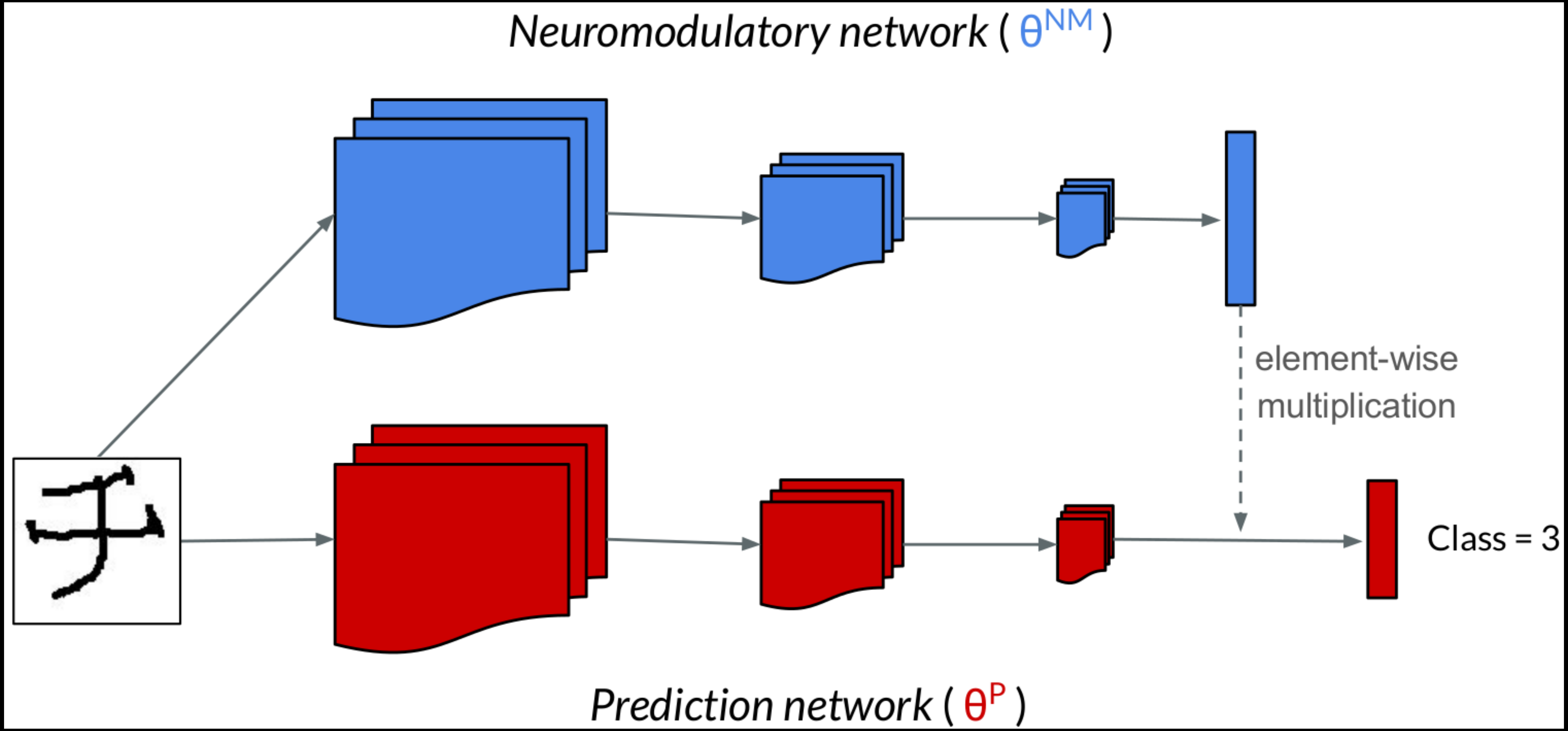
- Neuromodulation that
  - directly modulates activations: selective activation
  - indirectly modulates learning: selective plasticity



Soltoggio et al.

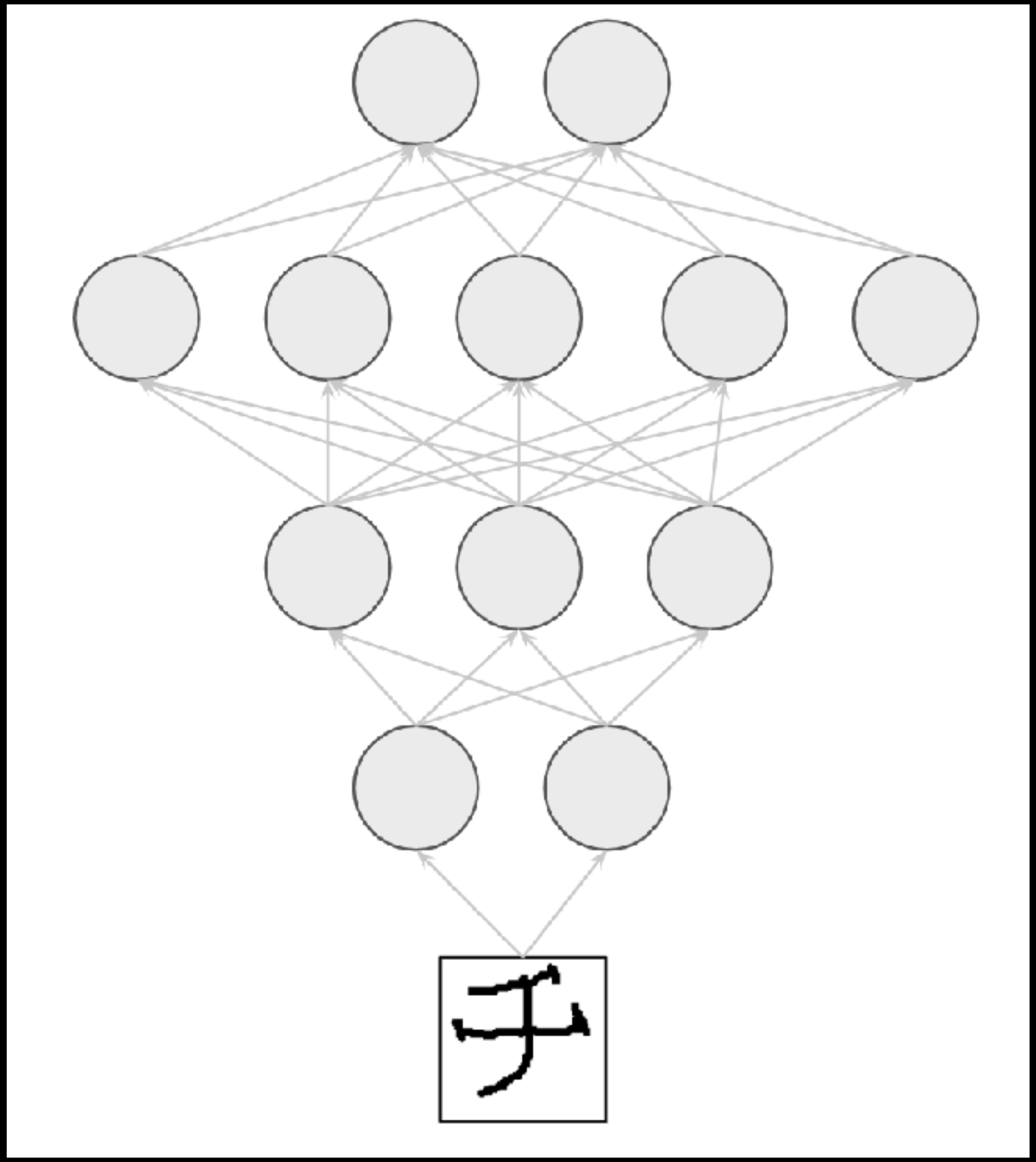
A Neuromodulated Meta-Learning algorithm (ANML)

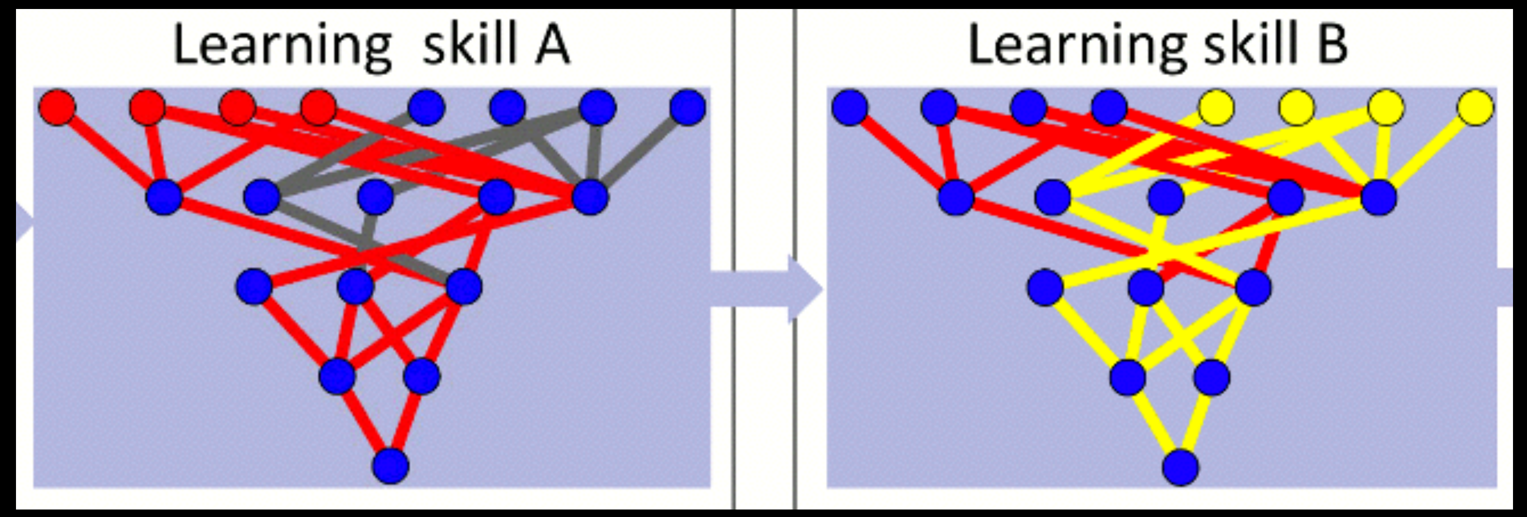
A Neuromodulated Meta-Learning algorithm (ANML)



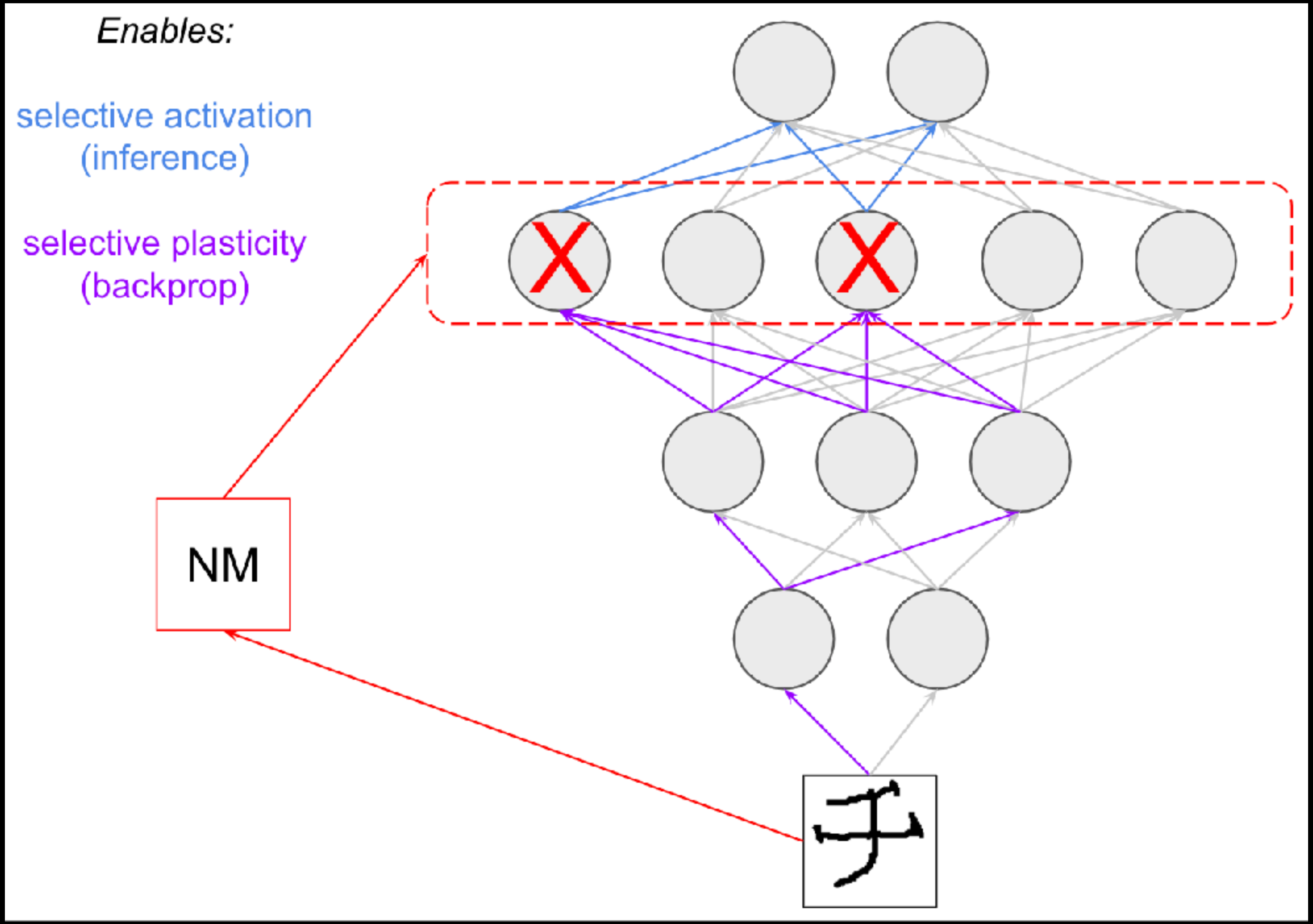
Normal Deep Learning

Inference everywhere
Learning everywhere





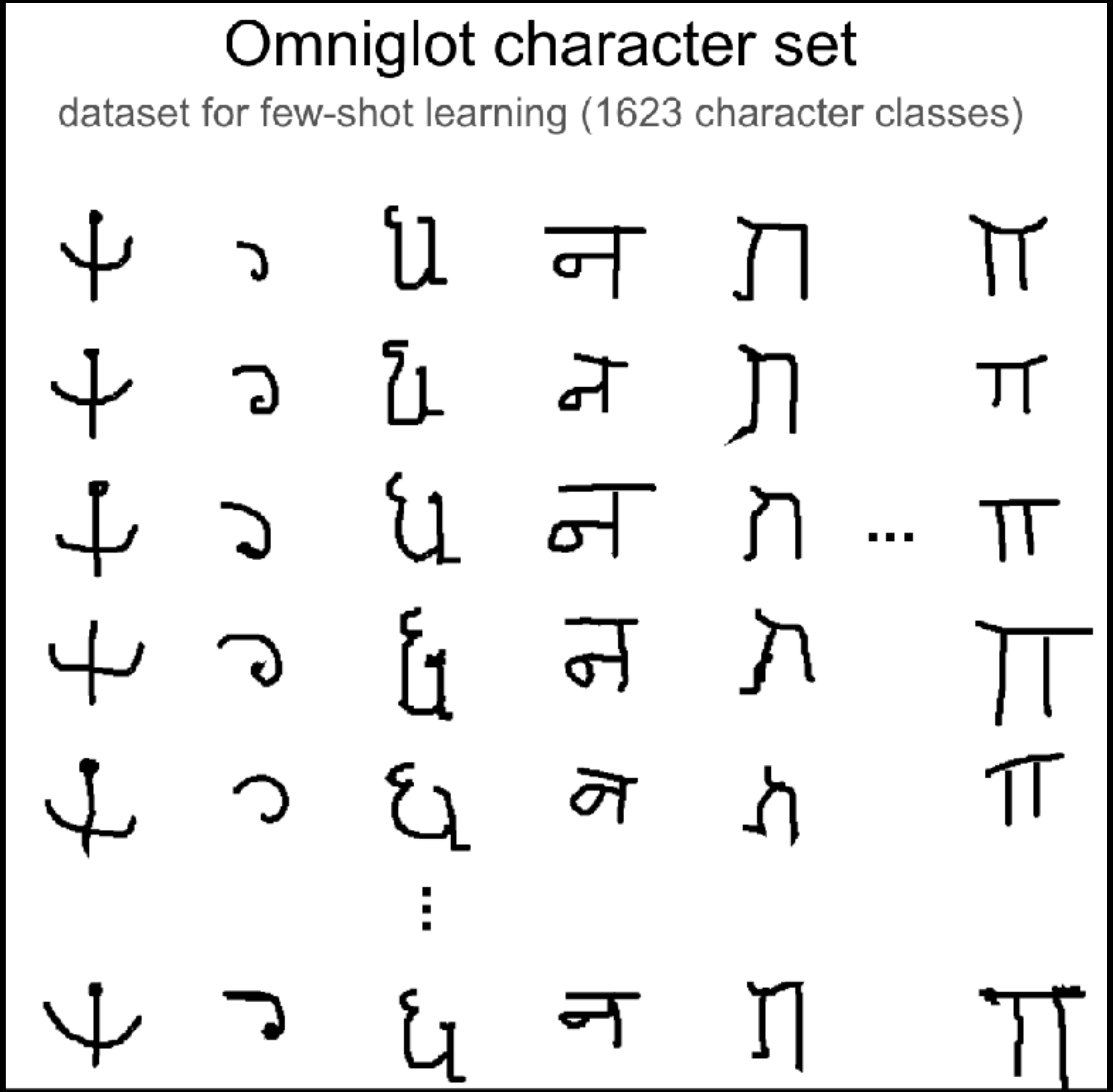
ANML



Enables:
selective activation
(inference)
selective plasticity
(backprop)
NM

Domain

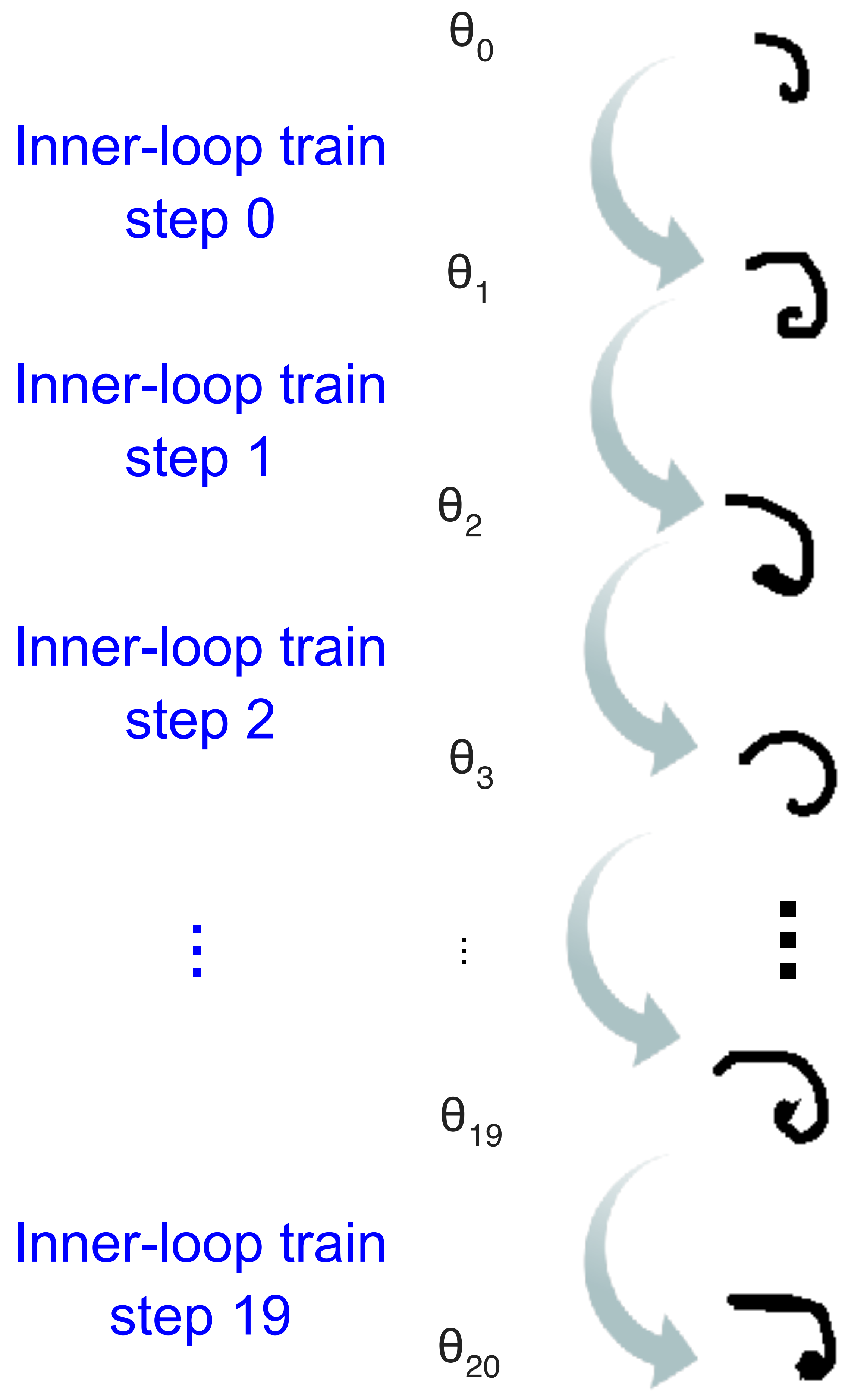
- Omniglot, following OML
  - each character type is a class/task



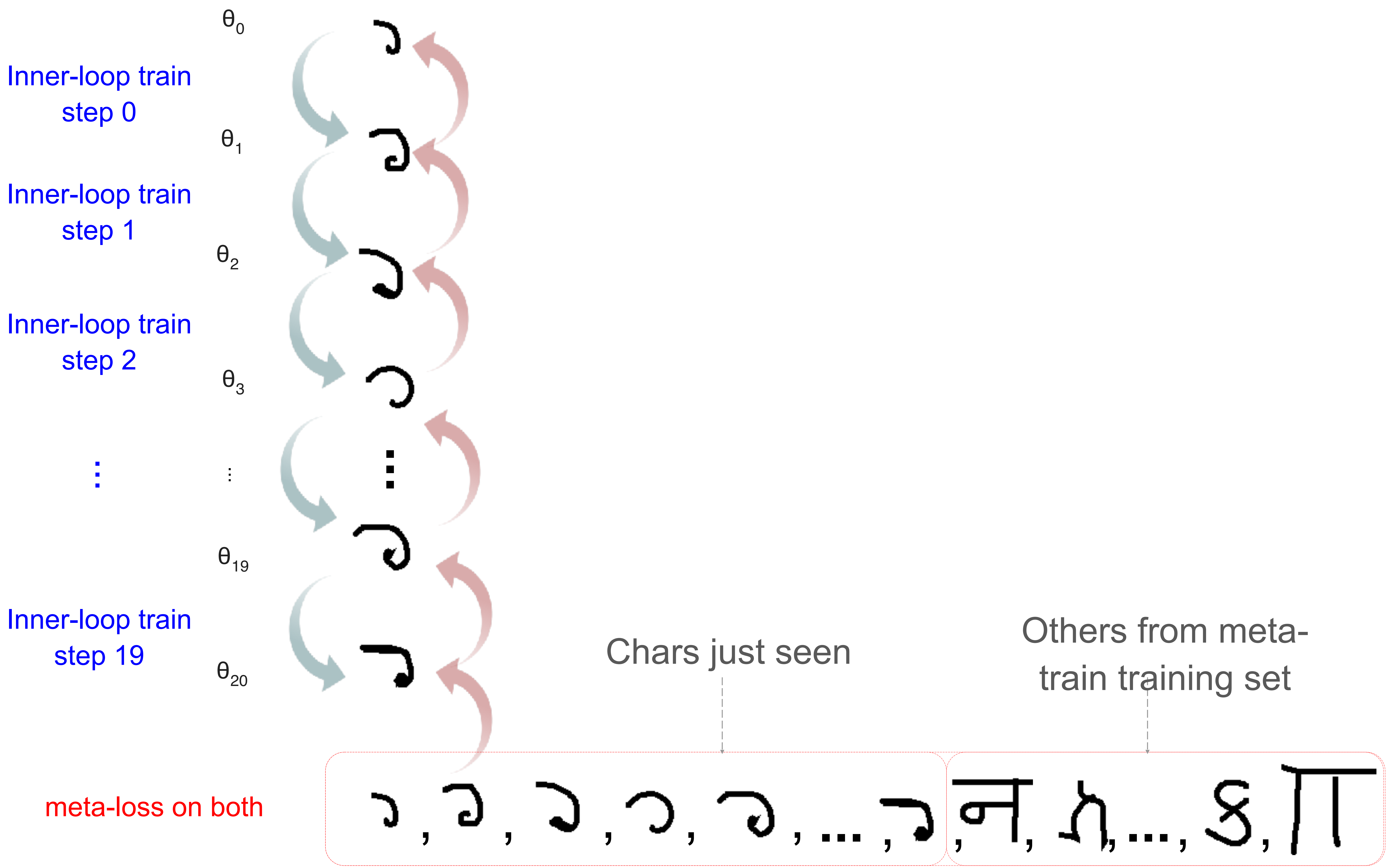
Ideally, differentiate through 600 tasks

Approximation: train on task t+1
validate on t+1 & some previous tasks

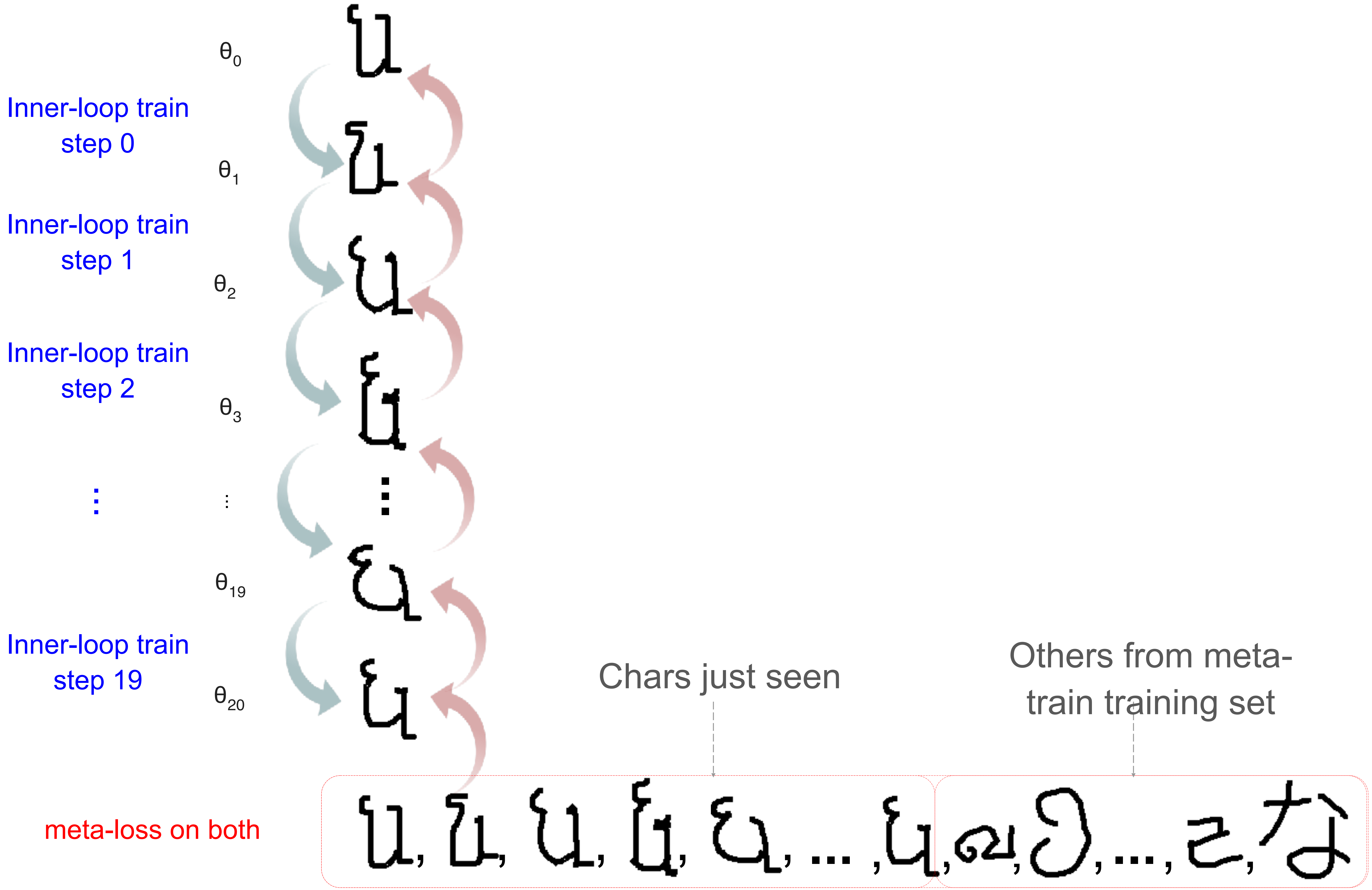
Learn sequentially on one class in the inner-loop



Backpropagate through the SGD steps



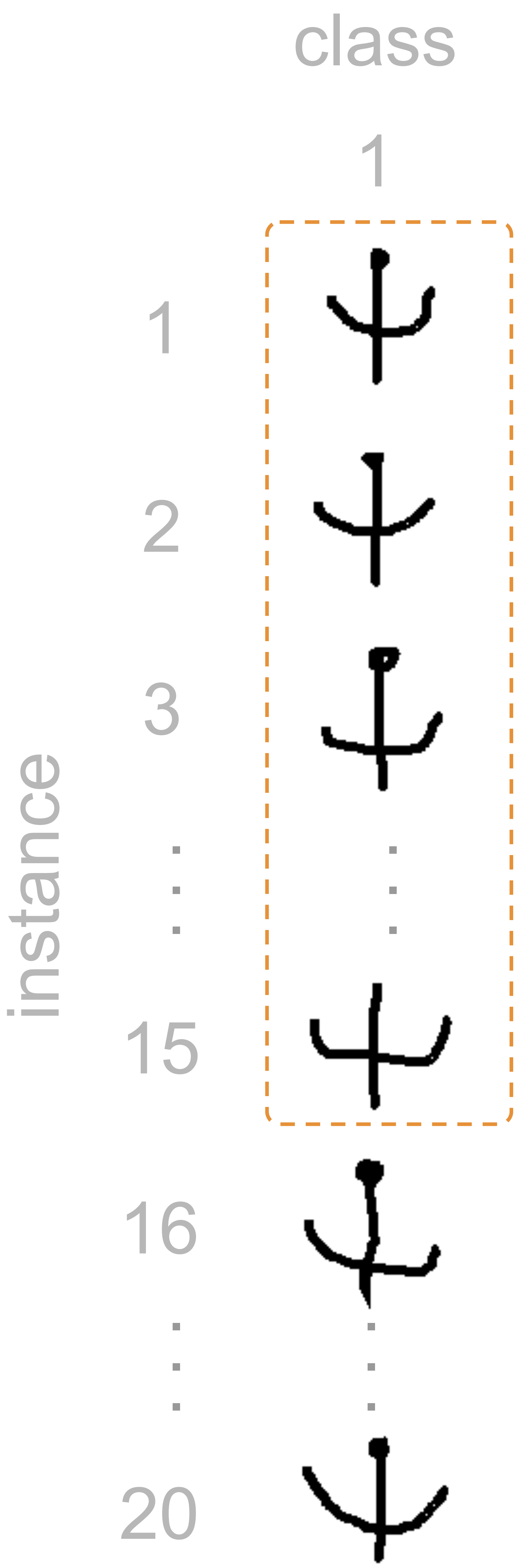
Backpropagate through the SGD steps



META-TESTING

Meta-test-training and meta-test-testing

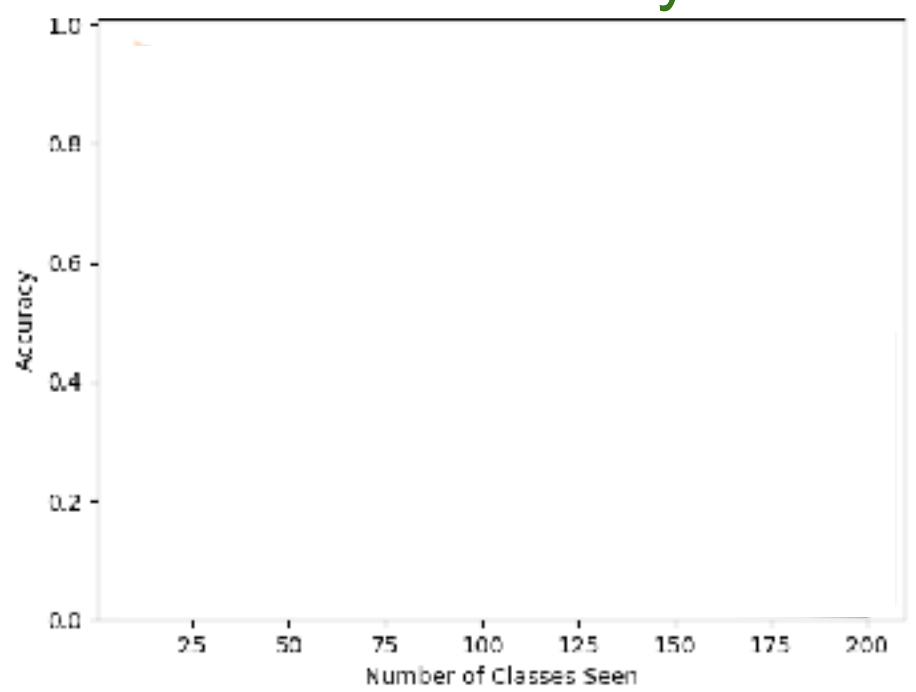
meta-testing post-training on a held-out set of 200 classes



Meta-test
training

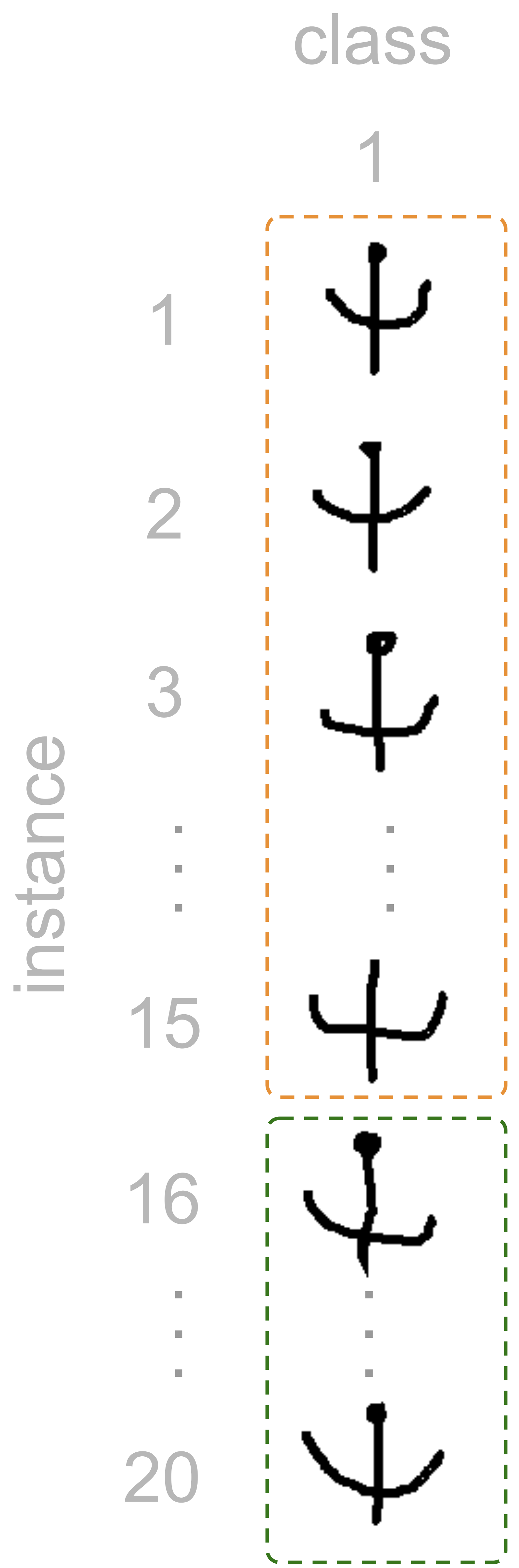
train on first class
(15 instances)

Meta-test
test accuracy



Meta-test-training and meta-test-testing

meta-testing post-training on a held-out set of 200 classes

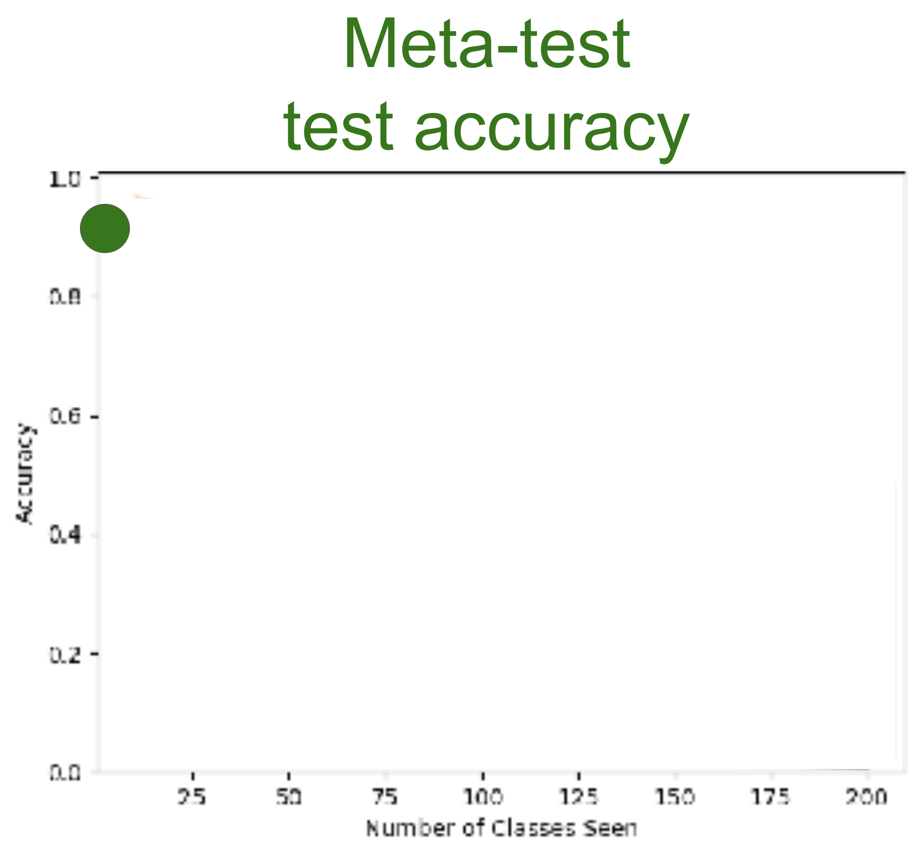


Meta-test training

train on first class (15 instances)

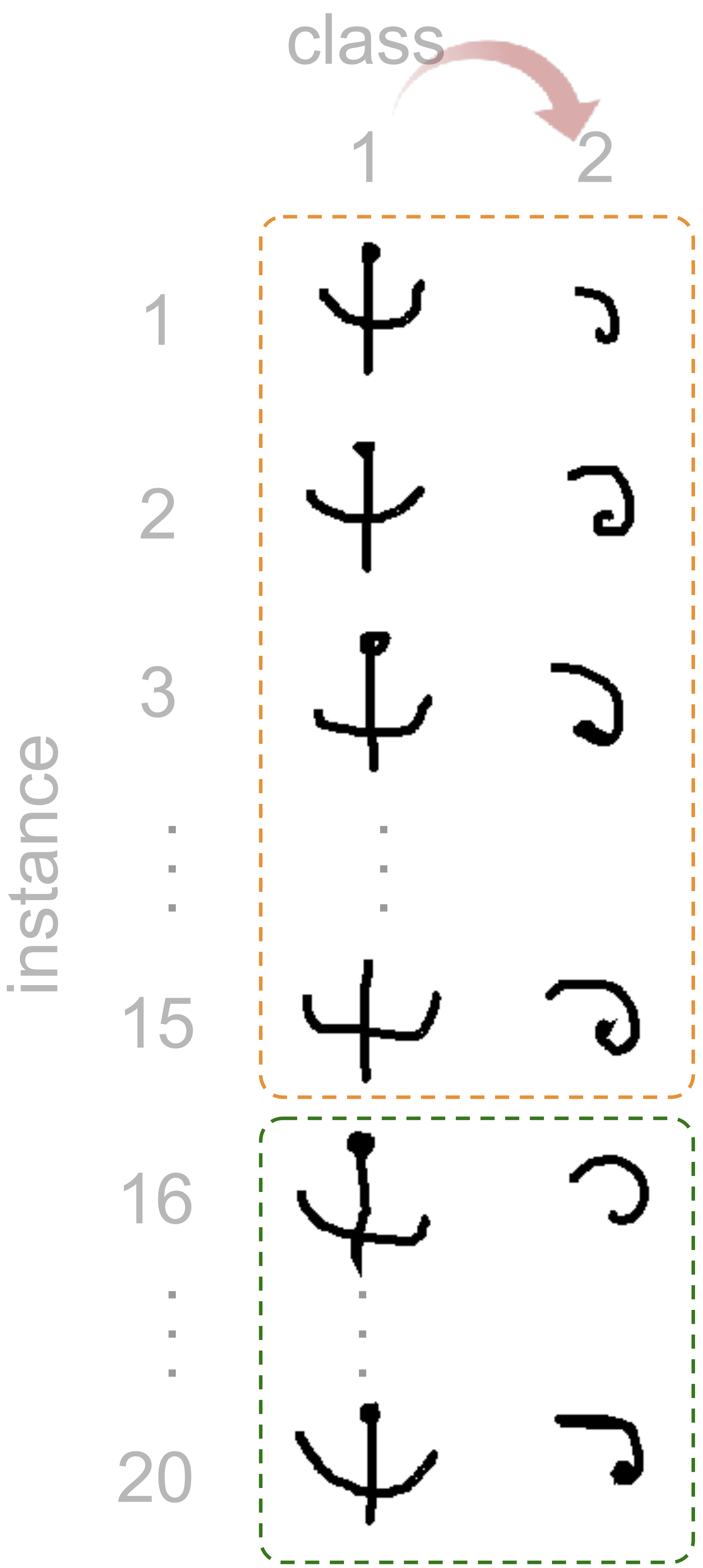
Meta-test testing

test on remaining 5 instances of first class



Meta-test-training and meta-test-testing

meta-testing post-training on a held-out set of 200 classes



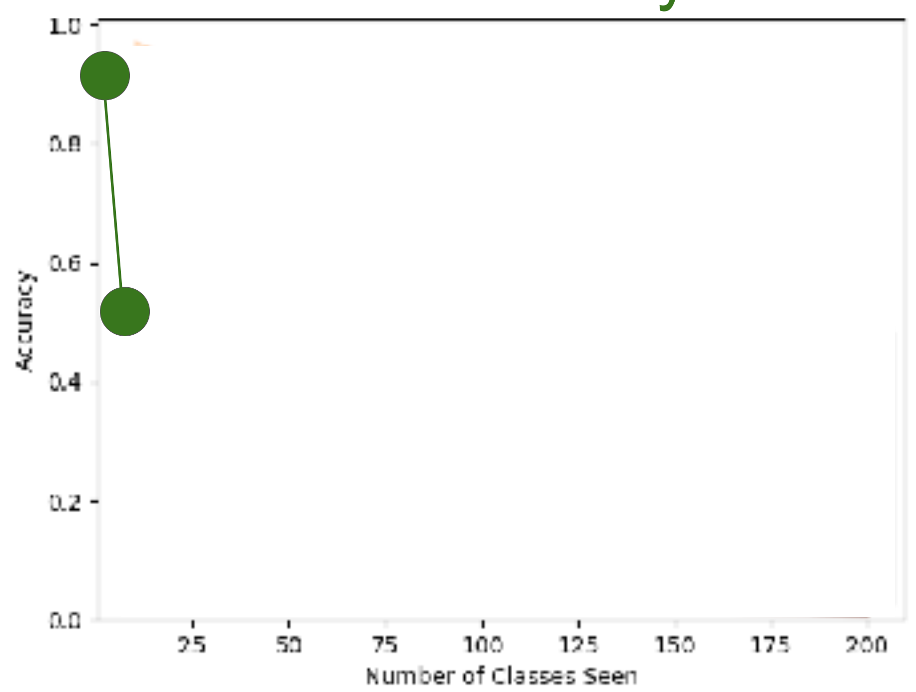
Meta-test training

sequentially train on 2 classes (15 instances each)

Meta-test testing

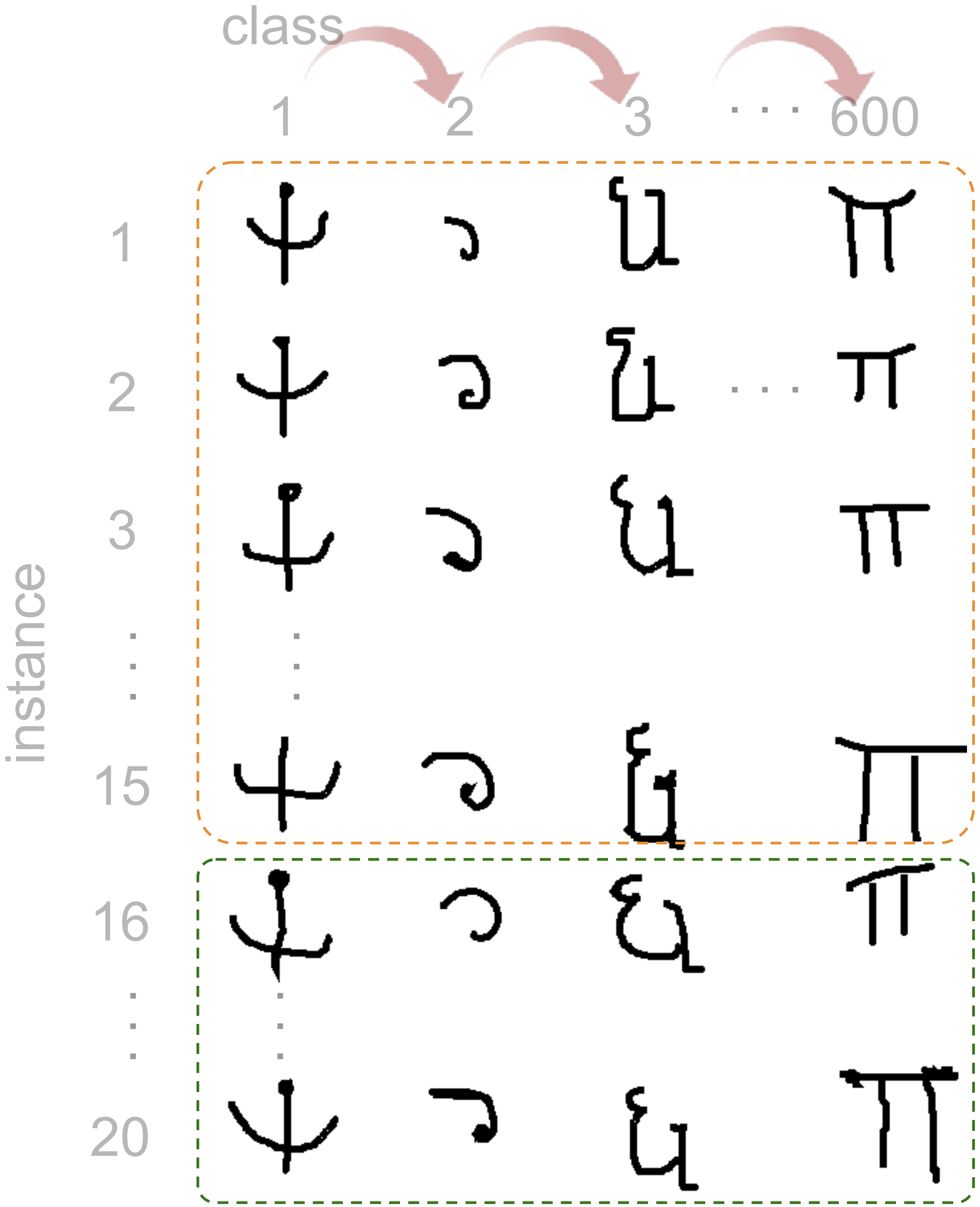
test on remaining 5 instances of 2 classes

Meta-test test accuracy

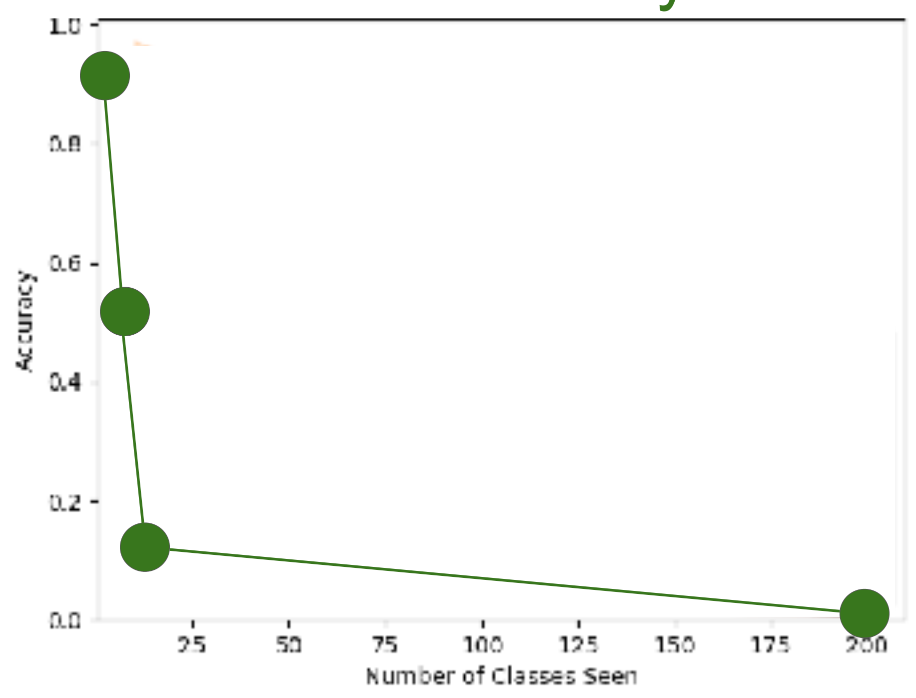


Meta-test-training and meta-test-testing

meta-testing post-training on a held-out set of 600 classes



Meta-test
test accuracy



Meta-test
training

sequentially
train on 600 classes
(15 instances each)

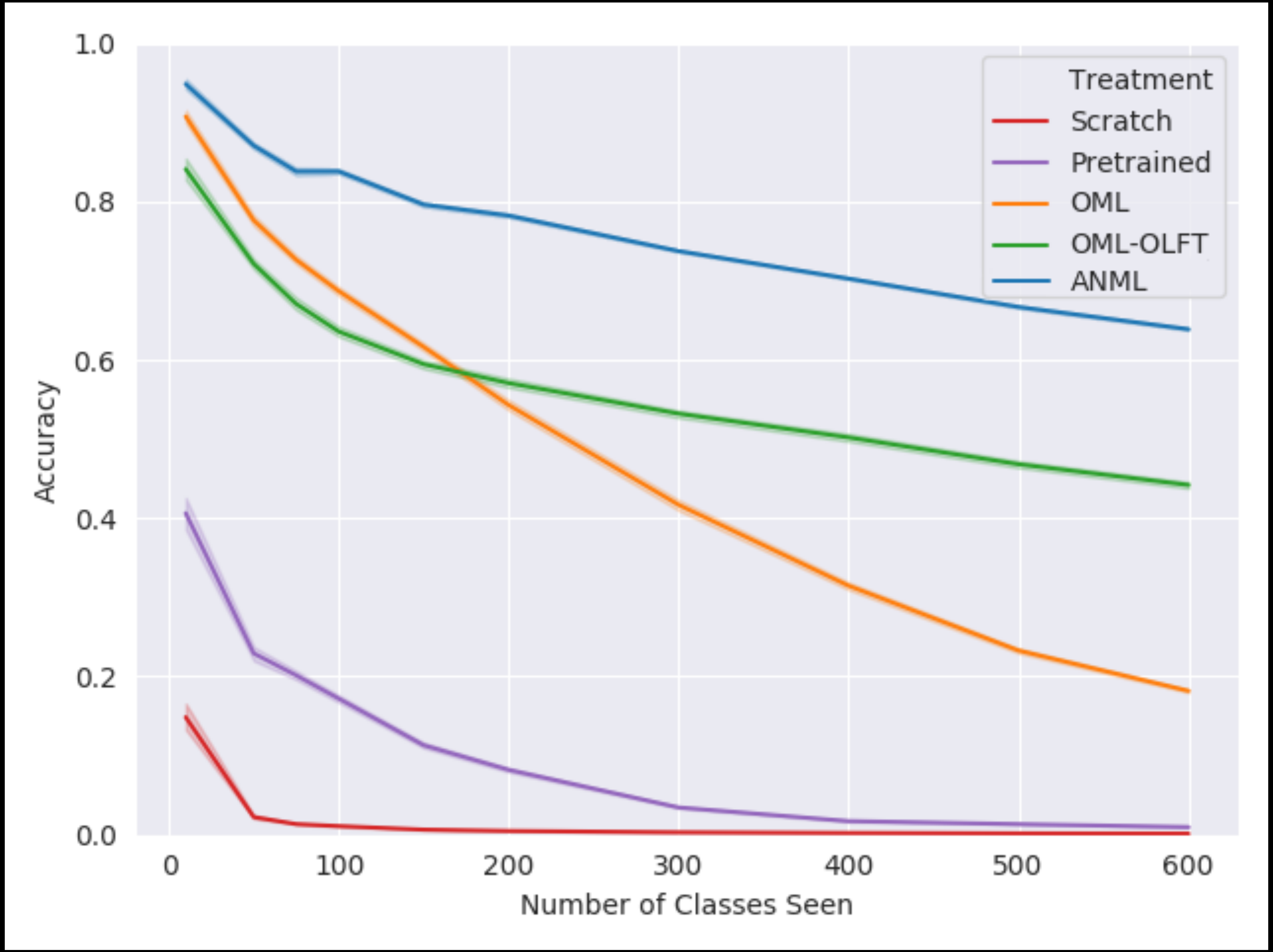
Meta-test
testing

test on remaining
5 instances
of all 600 classes

Reminder: Continual Learning is Hard

- Normal Deep Learning
  - IID sampling (no catastrophic forgetting)
  - Multiple passes through data
- Sequential Learning
  - Catastrophic Forgetting
  - One pass through data

Results



sequential learning, one epoch

vs. IID Oracles, Relative Performance Drop

- Oracles eliminate CF
- Oracle - Sequential
  - isolates performance drop due to CF

vs. IID Oracles, Relative Performance Drop

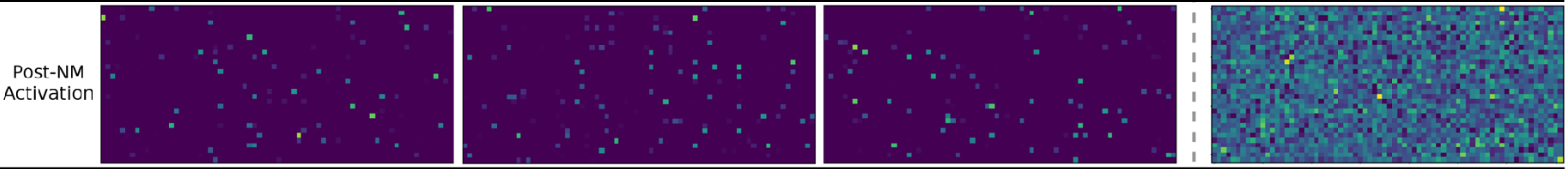
after one pass through 600 classes

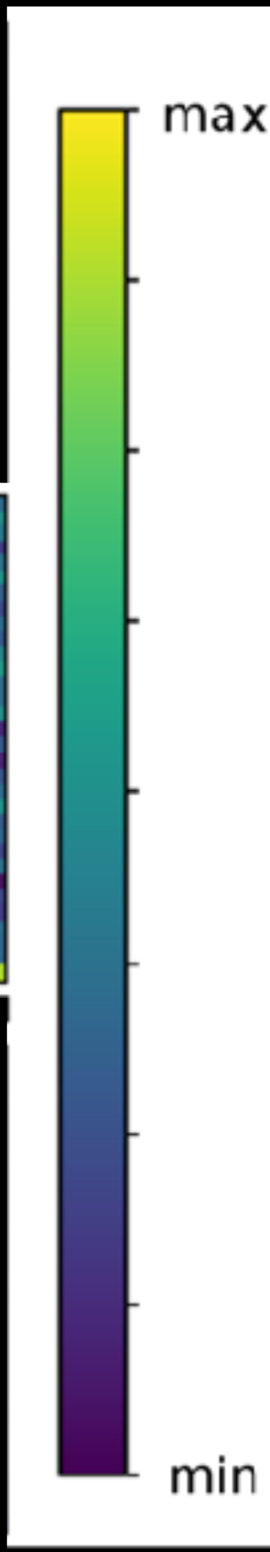
	Scratch	Pretrain & Transfer	OML	ANML
Performance Drop	99%	67%	47%	8%

Suggests ANML has mostly solved CF is in this problem

Learned Sparsity

Random Image 1 | Random Image 2 | Random Image 3 | Mean Over Dataset





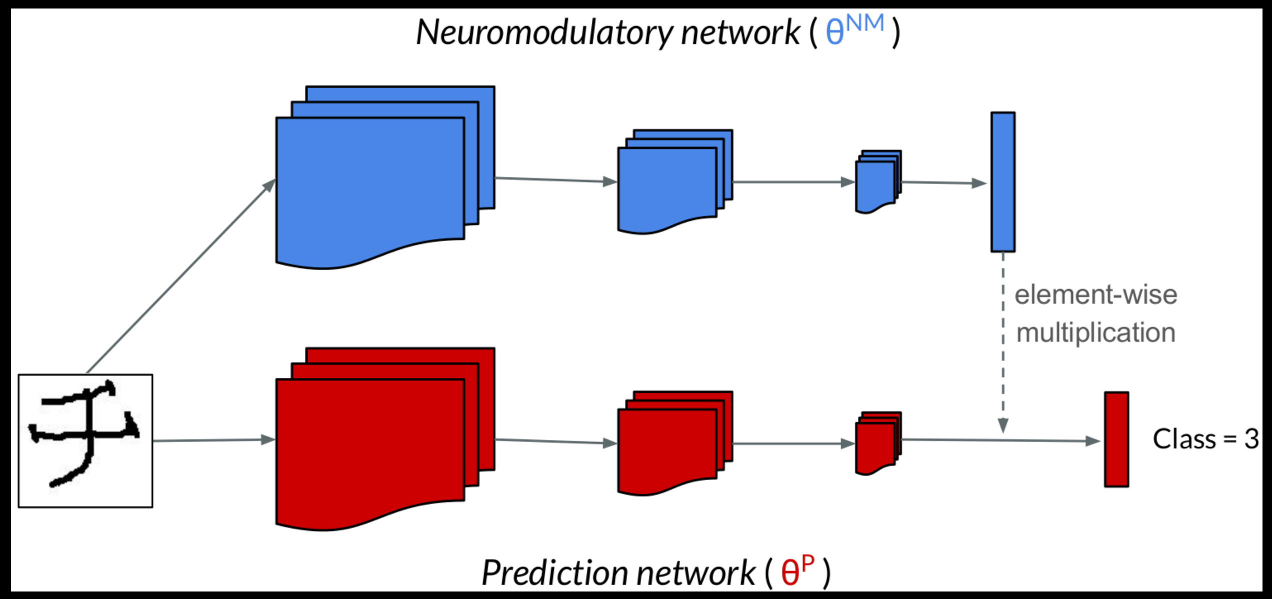
Sparse on each instance

Efficiently used across dataset

Both OML and ANML: No dead neurons!
vs. ~14% with sparsity auxiliary loss

(Javed & White 2019)

Update



- Sara Pelivani et al. at UCL / Evolution.ai found results are ~just as good without the NM network (she will share more soon)
- We had controlled num params, so made red smaller
  - Turns out being smaller is the key driver of improved performance
- We are still investigating
  - why smaller models do better
  - where neuromodulation helps
    - e.g. for domain transfer: https://arxiv.org/abs/2108.12056

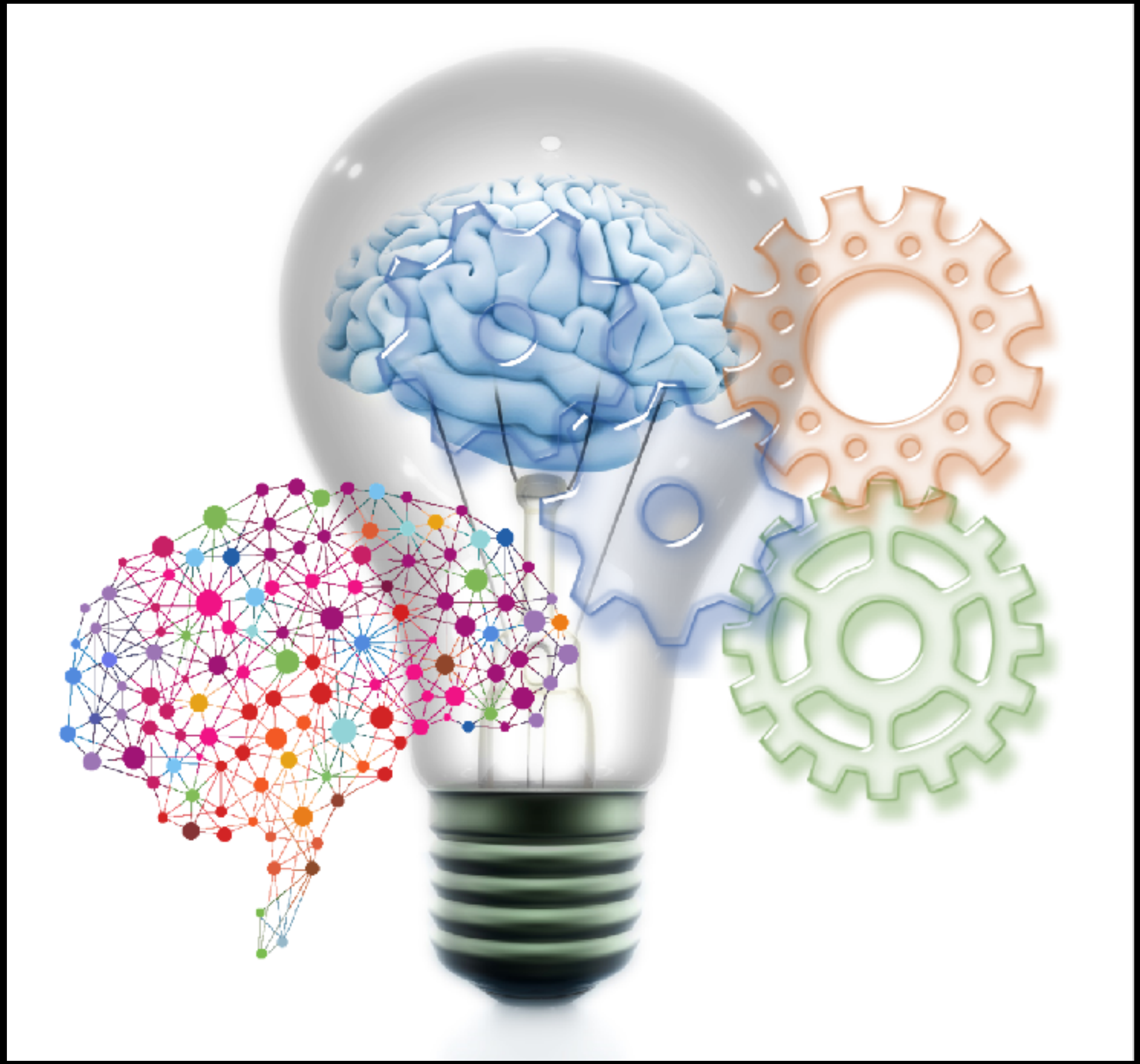
ANML Conclusions

- OML/ANML can learn 600 sequential tasks, and still perform pretty well on all on average
- Learns to produce sparse representations
  - and likely many other things to solve CF
- Future work:
  - more and harder domains
  - other flavors of meta-learning (e.g. RNNs)

Artificial General Intelligence (AGI)

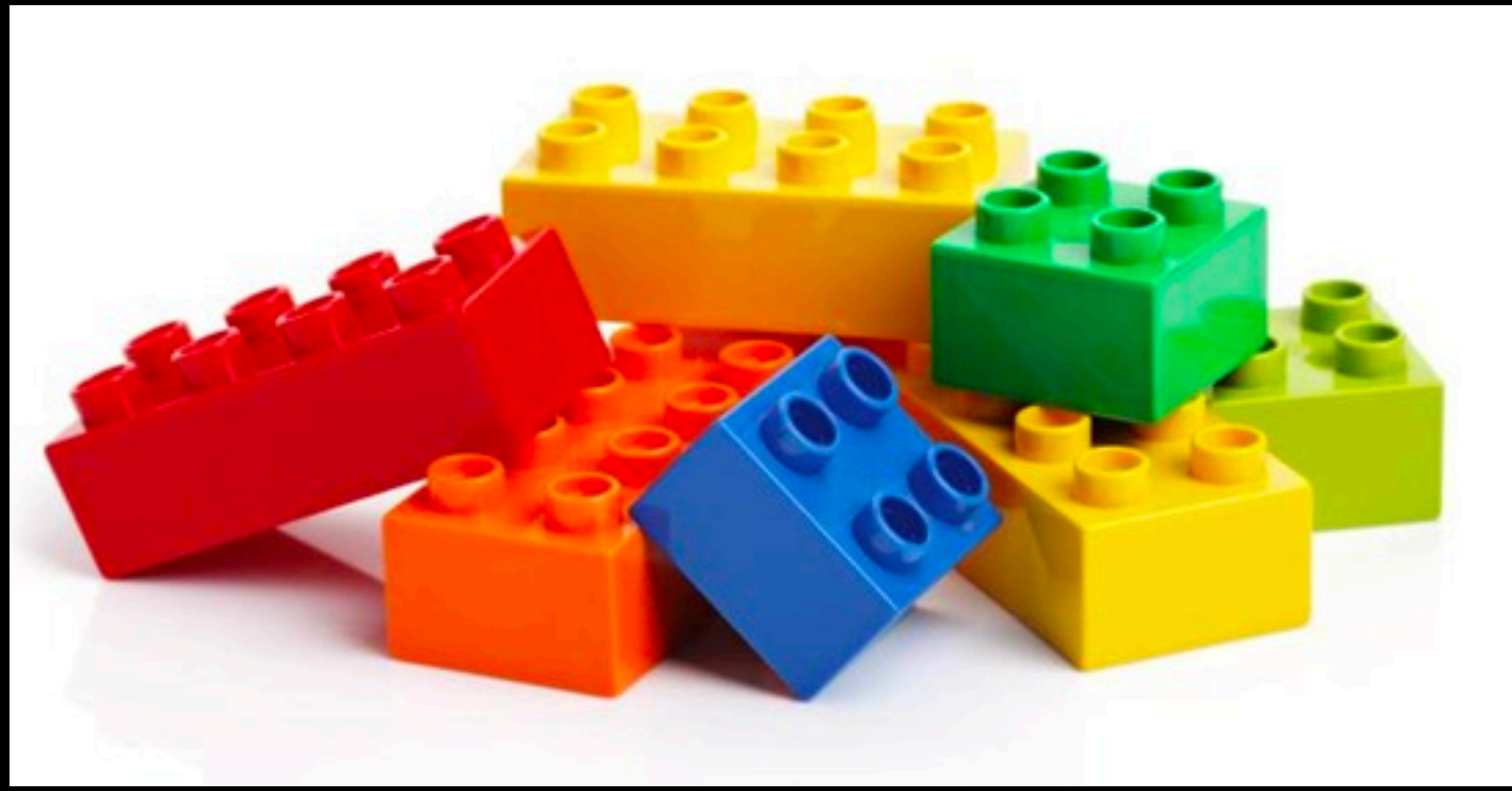
or Human-level AI, if you prefer

- Long way to go
- How will we get there?



Manual Path to AI

- Dominant paradigm in ML
- Phase 1: Identify key building blocks



Key Building Blocks?

how many more?
hundreds? thousands?

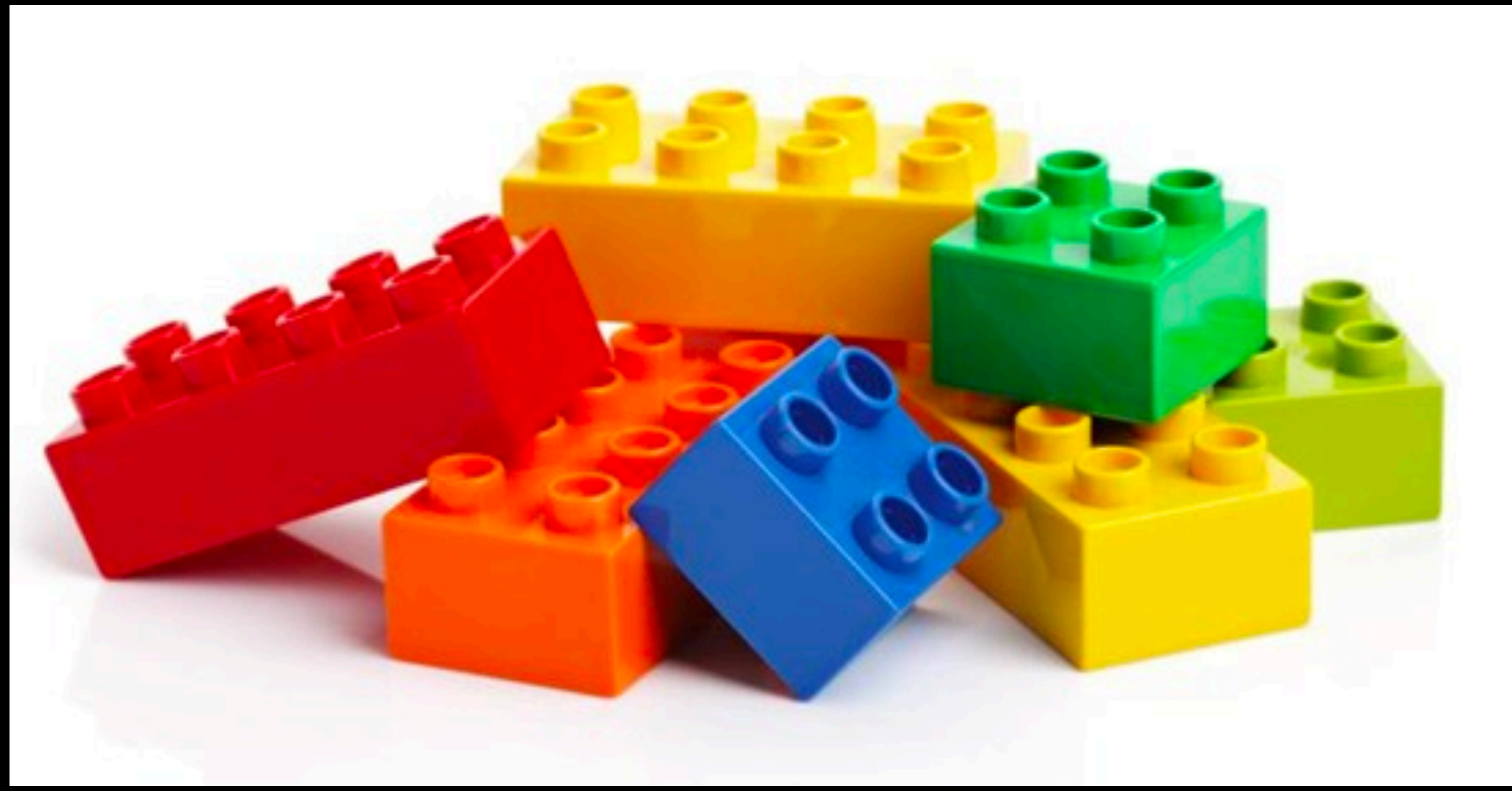
can we find them all?

- convolution
- attention mechanisms
- spatial tranformers
- batch/layer norm
- a learned loss (e.g. evolved policy gradients)
- hierarchical RL, options
- structural organization (regularity, modularity, hierarchy)
- intrinsic motivation (many different flavors)
- auxiliary tasks (predictions, autoencoding, predicting rewards, etc.)
- good initializations (Xavier, MAML, etc.)
- catastrophic forgetting solutions
- universal value functions
- hindsight experience replay
- LSTM cell machinery variants
- complex optimizers (Adam, RMSprop, etc.)

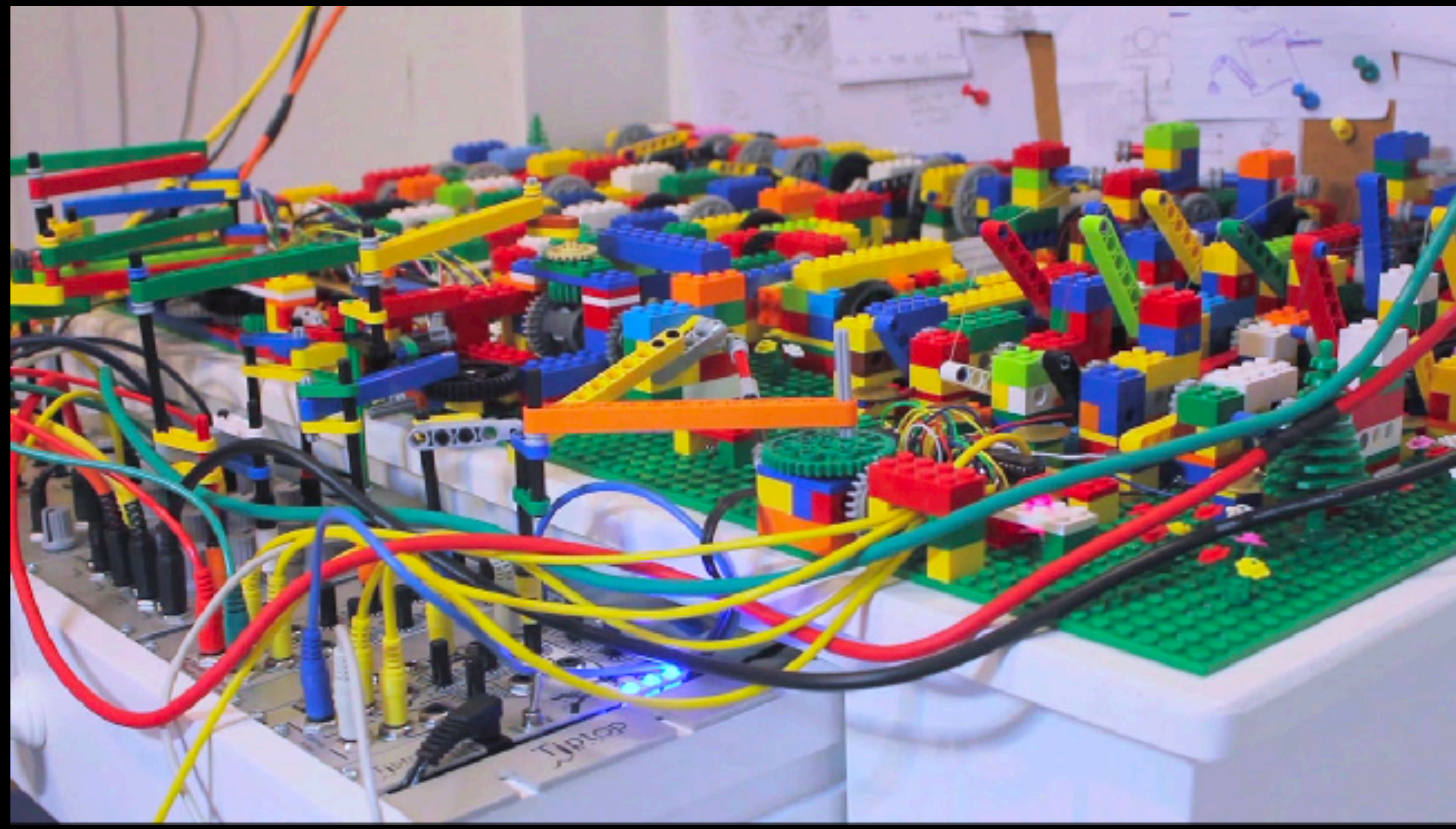
- Dyna
- variance reduction techniques
- activation functions
- good hyperparameters
- capsules
- gradient-friendly architectures (skip connections, highway networks)
- value functions, state-value functions, advantage functions
- recurrence (where?)
- multi-modal fusion
- trust regions
- Bayesian methods
- Active learning
- Probabilistic models
- Distance metrics (latent codes)
- etc.

Manual Path to AI

- Dominant paradigm in ML
- Phase 1: Identify key building blocks



- Phase 2: Combine building blocks into complex thinking machine
  - Herculean task
    - complex, non-linear interactions
    - debugging, optimizing would be a nightmare
    - massive team required (e.g. CERN, Apollo)



Clear Machine Learning Trend:
Hand-designed pipelines are ultimately outperformed by learned solutions

hand designed $\longrightarrow$ learned

- Features
- Architectures
- Hyperparameters & data augmentation
- RL algorithms

suggests alternate path

AI-Generating Algorithms

Clune 2019

- Learn as much as possible
- Bootstrap from simple to AGI
- Expensive outer loop
  - produces a sample-efficient, intelligent agent
- Existence proof
  - Earth

AI-Generating Algorithms

Clune 2019

Three Pillars

1. Meta-learn architectures
2. Meta-learn learning algorithms
3. Generate effective learning environments

Handcrafting each is slow, limited by our intelligence/time

Better to learn them. Let ML+compute do the heavy lifting

AI-Generating Algorithms

Clune 2019

Three Pillars

1. Meta-learn architectures
2. Meta-learn learning algorithms
3. Generate effective learning environments

AI-Generating Algorithms

Clune 2019

Three Pillars

1. Meta-learn architectures
    - Evolved NAS Real et al. 2017
    - Generative Teaching Networks Such et al. ICML 2020.
    - Synthetic Petri Dish. Rawal et al. 2020

AI-Generating Algorithms

Clune 2019

Three Pillars

1. Meta-learn architectures
2. Meta-learn learning algorithms
3. Generate effective learning environments

CORL Keynote, see jeffclune.com/videos.html

Overall Conclusions

- Described
  - Differentiable Hebbian plasticity
  - Differentiable neuromodulated Hebbian Plasticity
  - ANML: Learning to continually learn via neuromodulation
  - AI-Generating Algorithms
- In all, materials matter
  - Hebbian plasticity vs. normal RNN
  - Neuromodulation
- What other materials should we be building with?
- Might we be able to search for them?

Thanks!

Main collaborators

- Thomas Miconi
- Shawn Beaulieu
- Ken Stanley
- Nick Cheney
- Joel Lehman
- Lapo Frati

DARPA

NSF CAREER Award

UBER AI Labs

Join us at U. British Columbia!