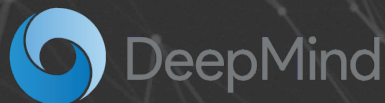


Multiple scales of task and reward-based learning

Jane Wang

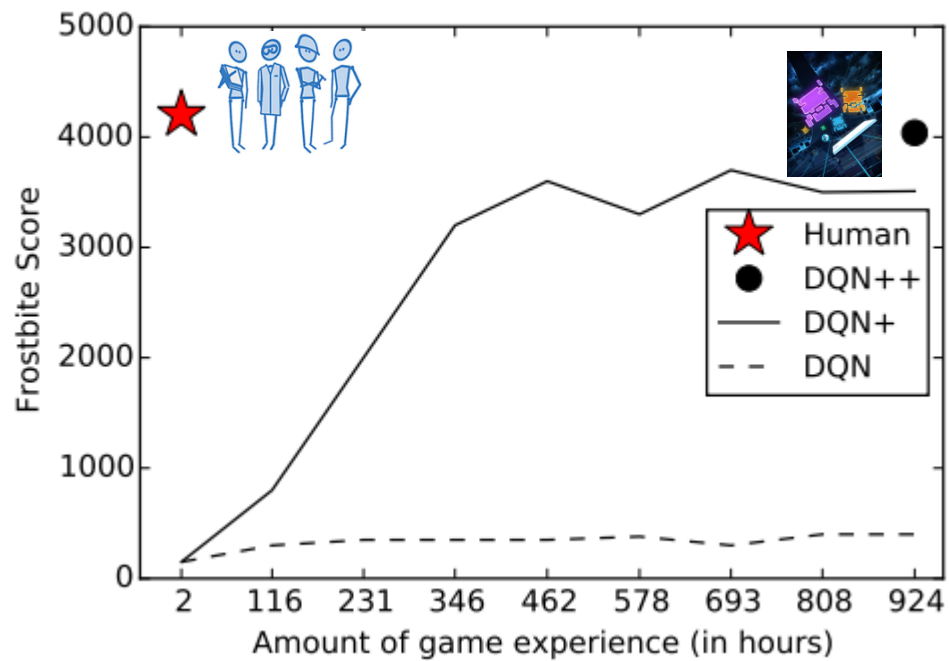
Zeb Kurth-Nelson, Sam Ritter, Hubert Soyer, Remi Munos, Charles Blundell, Joel Leibo, Dhruva Tirumala, Dharshan Kumaran, Matt Botvinick

NIPS 2017 Meta-learning Symposium

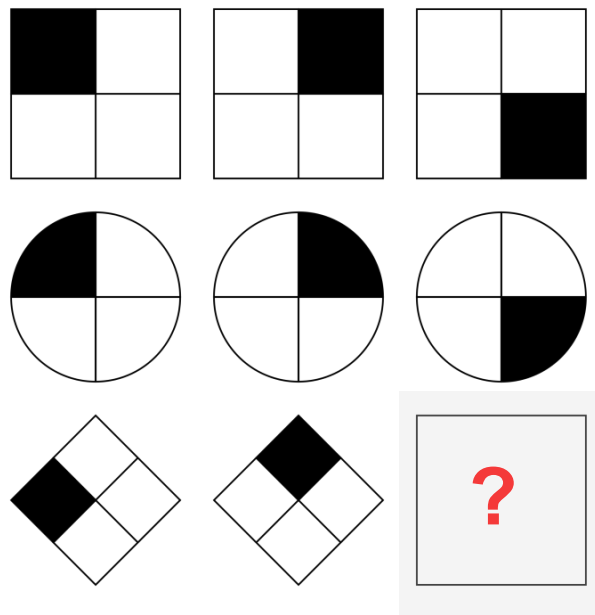


December 7, 2017





Raven's progressive matrices (J. C. Raven, 1936)



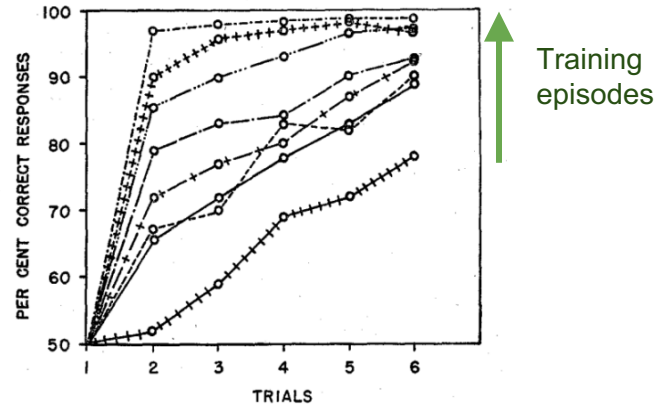
Meta-Learning: Learning inductive biases or priors

Learning faster with more tasks, benefiting from transfer across tasks and learning on related tasks

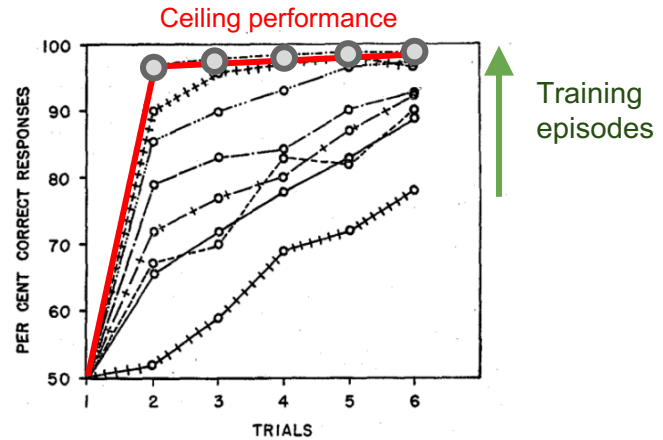
Evolutionary principles in self-referential learning (Schmidhuber, 1987)

Learning to learn (Thrun & Pratt, 1998)

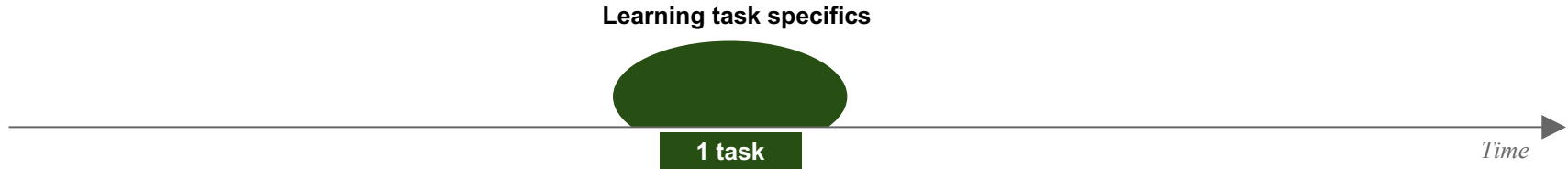
Meta-RL: learning to learn from reward feedback



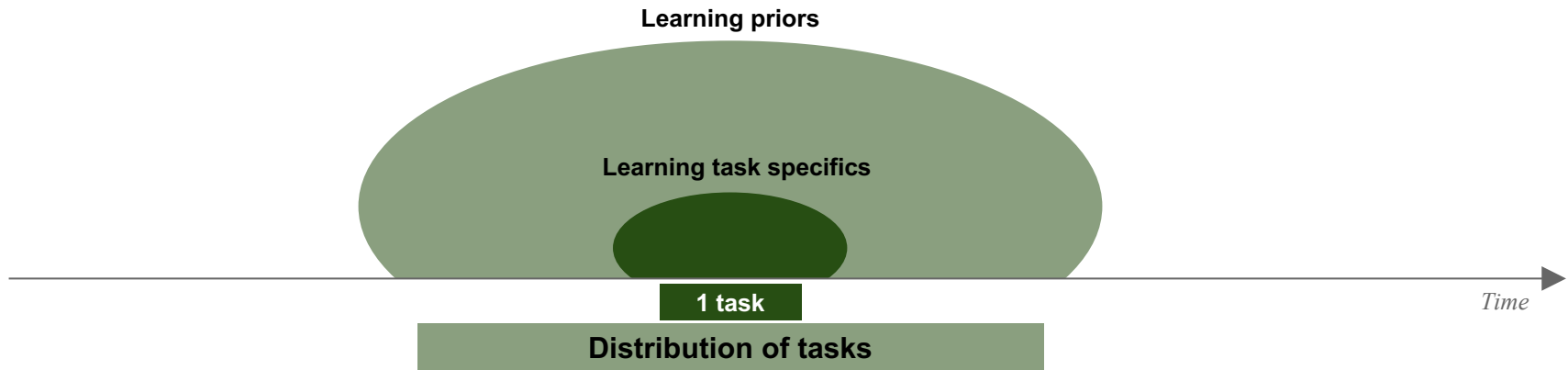
Meta-RL: learning to learn from reward feedback



Multiple scales of reward-based learning

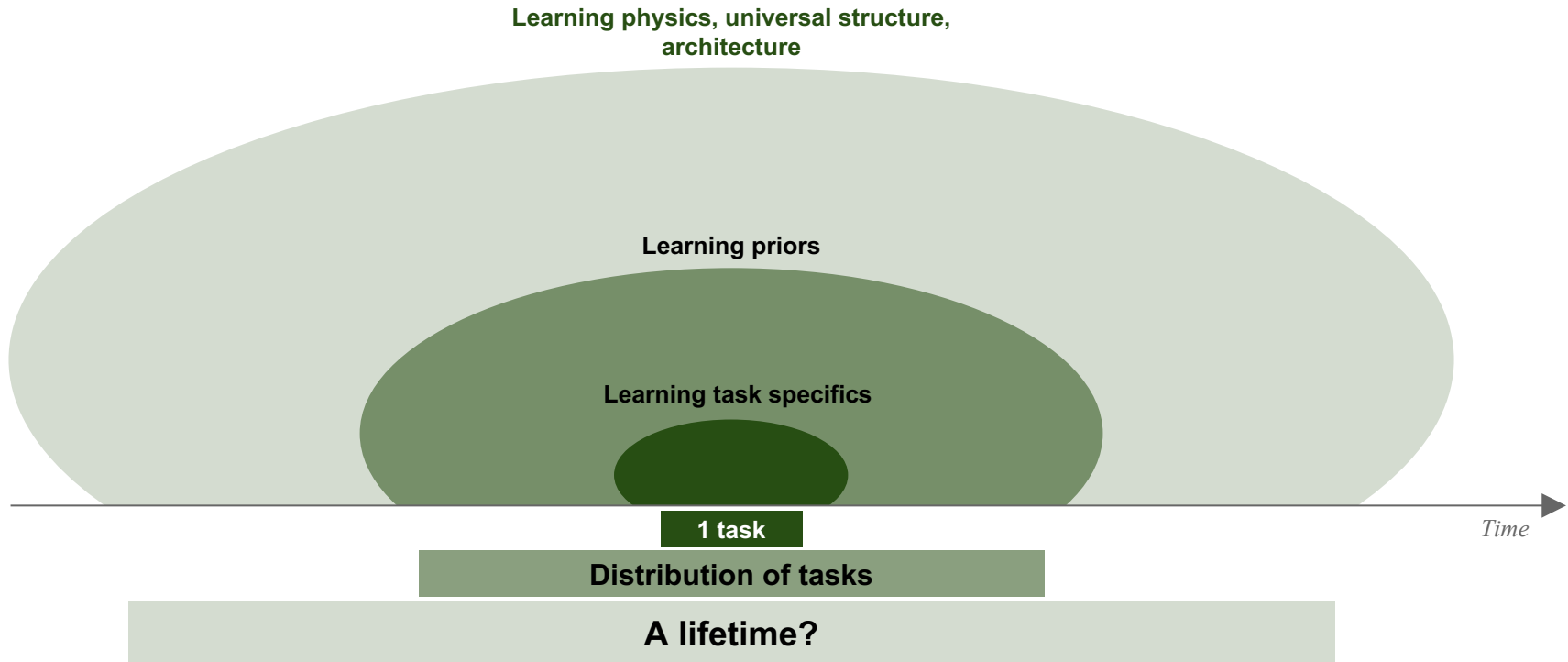


Multiple scales of reward-based learning

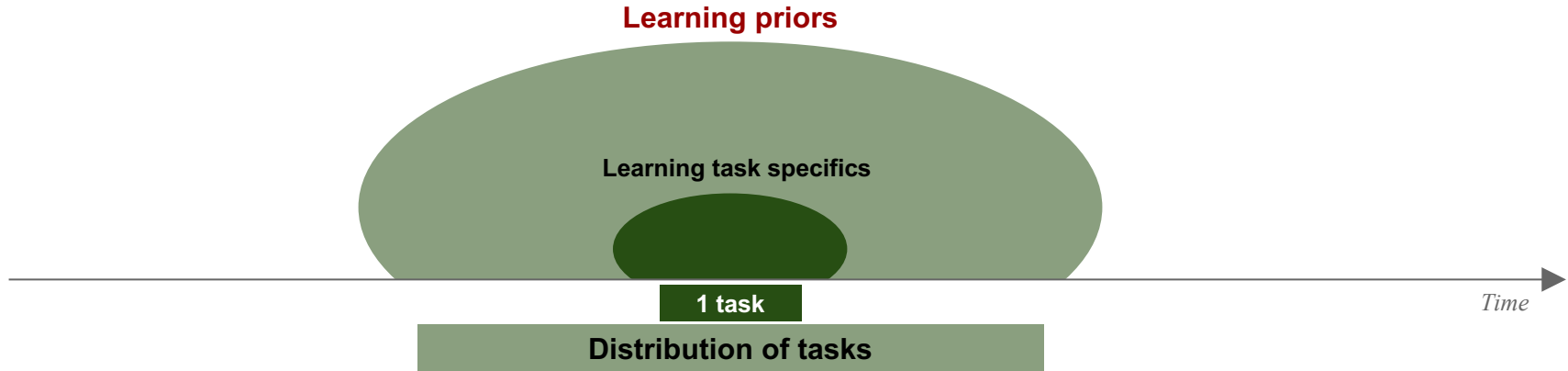


Nested RL algorithms happening in parallel, on **different timescales**

Multiple scales of reward-based learning



Multiple scales of reward-based learning



Different ways of building priors

Handcrafted features, expert knowledge, teaching signals

Learning good initialization

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks (Finn et al, 2017 ICML)

Learning a meta-optimizer

Learning to learn by gradient descent by gradient descent (Andrychowicz et al, 2016)

Learning an embedding function

Matching networks for one shot learning (Vinyals et al, 2016)

Bayesian program learning

Human-level concept learning through probabilistic program induction (Lake et al, 2015)

Implicitly learned via recurrent neural networks/external memory

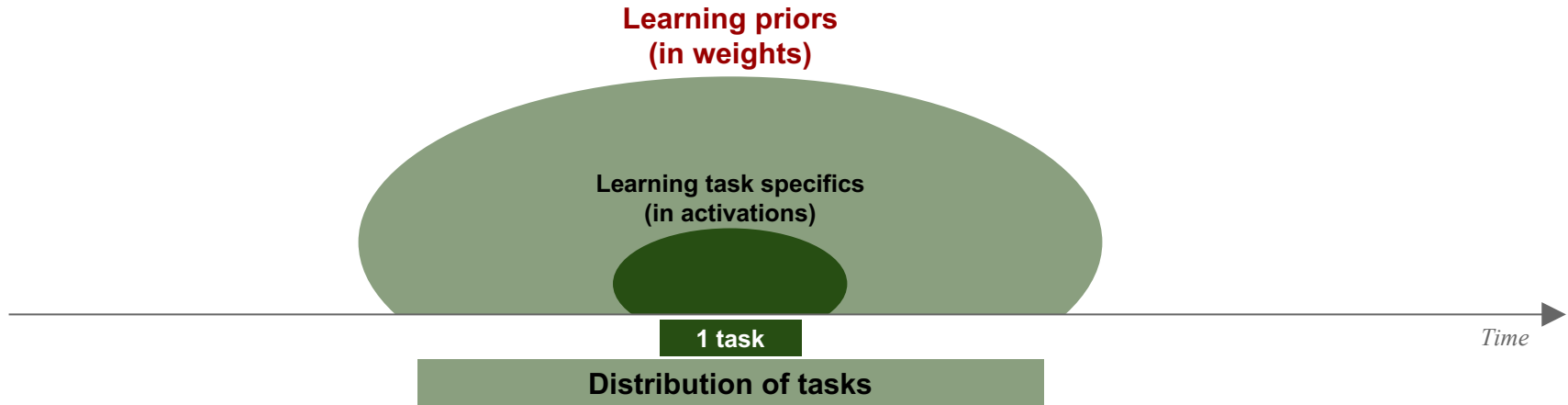
Meta-learning with memory-augmented neural networks (Santoro et al, 2016)

...

What all these have in common is a way to build in assumptions that **constrain the space of hypotheses** to search over

RNNs + distribution of tasks to learn prior implicitly

Use activations of a recurrent neural network (RNN) to implement RL in dynamics, shaped by priors learned in the weights

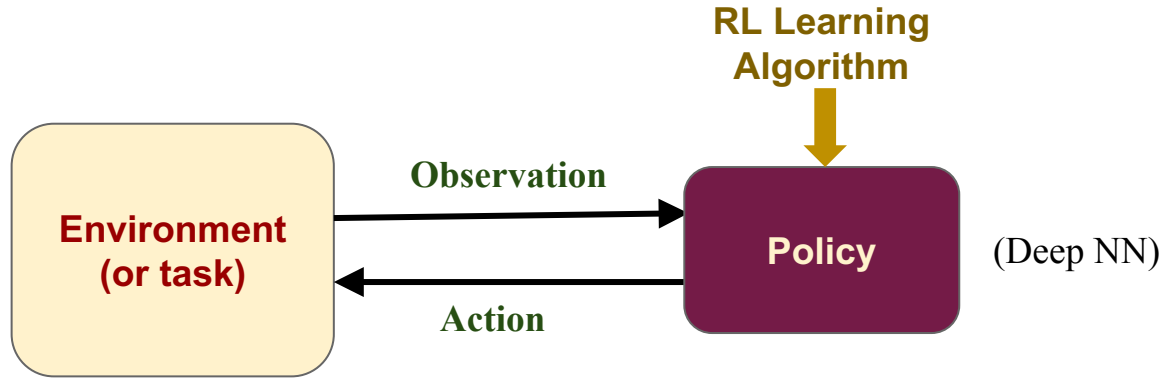


Constrain hypothesis space with task distribution, correlated in the prior we want to learn, but different in ways we want to abstract over (ie specific image, reward contingency)

Prefrontal cortex and flexible cognitive control: Rules without symbols (Rougier et al, 2005)

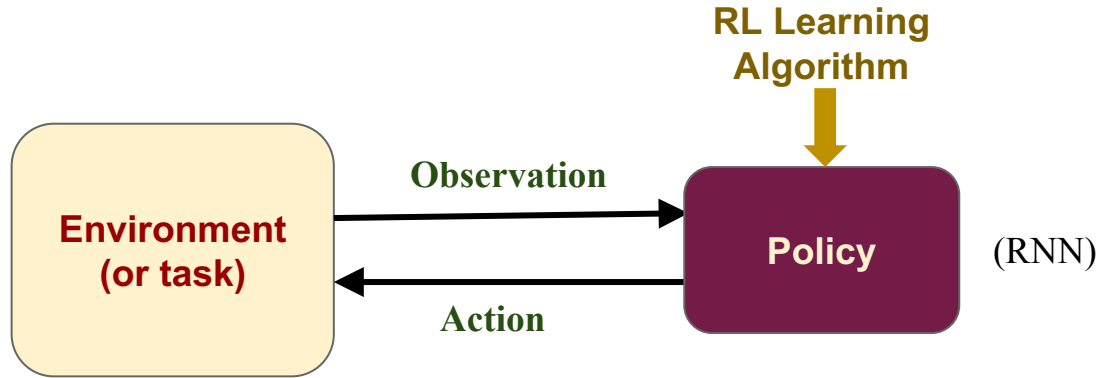
Domain randomization for transferring deep neural networks from simulation to the real world (Tobin et al, 2017)

Learning the correct policy



Map observations to actions in order to maximize reward for environment

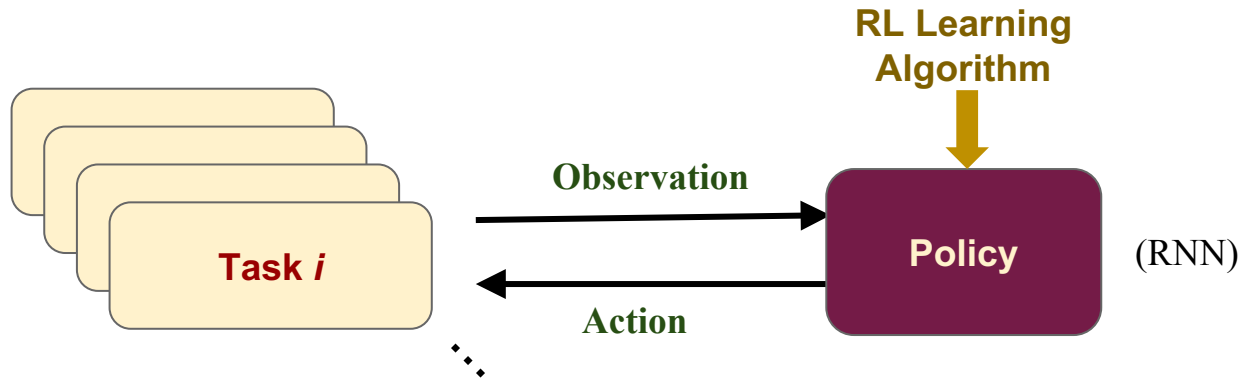
Learning the correct policy with an RNN



Map history of observations and states to future actions in order to maximize reward for a sequential task

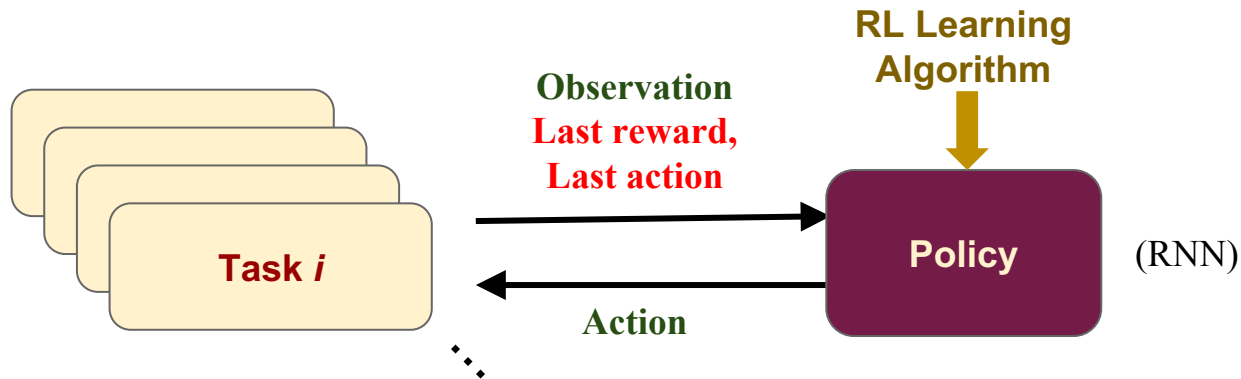
Song et al, 2017 eLife; Miconi et al, 2017 eLife; Barak, 2017 Curr Opin Neurobiol

Learning to learn the correct policy: meta-RL



Map **history** of observations and past rewards/actions to future actions in order to maximize reward for a **distribution** of tasks

Learning to learn the correct policy: meta-RL

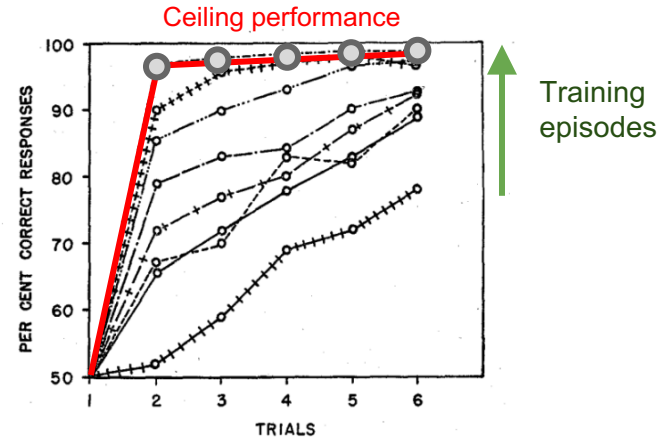


Map **history** of observations and past rewards/actions to future actions in order to maximize reward for a **distribution of tasks**

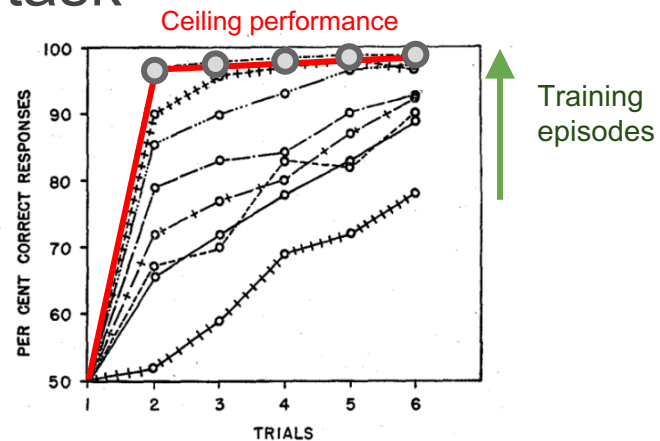
Wang et al, 2016. Learning to reinforcement learn. arXiv:1611.05763

Duan et al, 2016. RL²: Fast reinforcement learning via slow reinforcement learning. arXiv:1611.02779

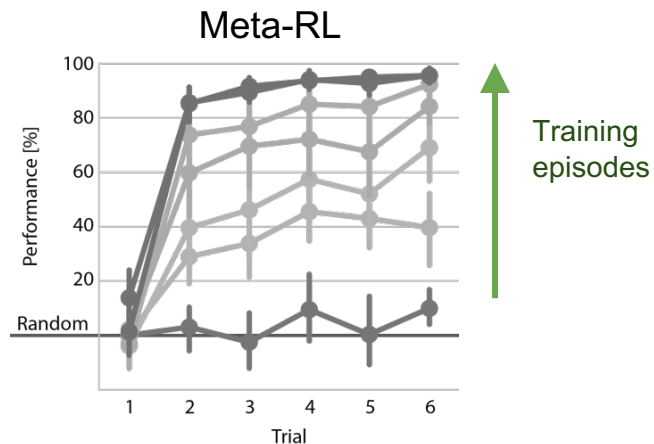
Harlow task



Meta-RL in the Harlow task

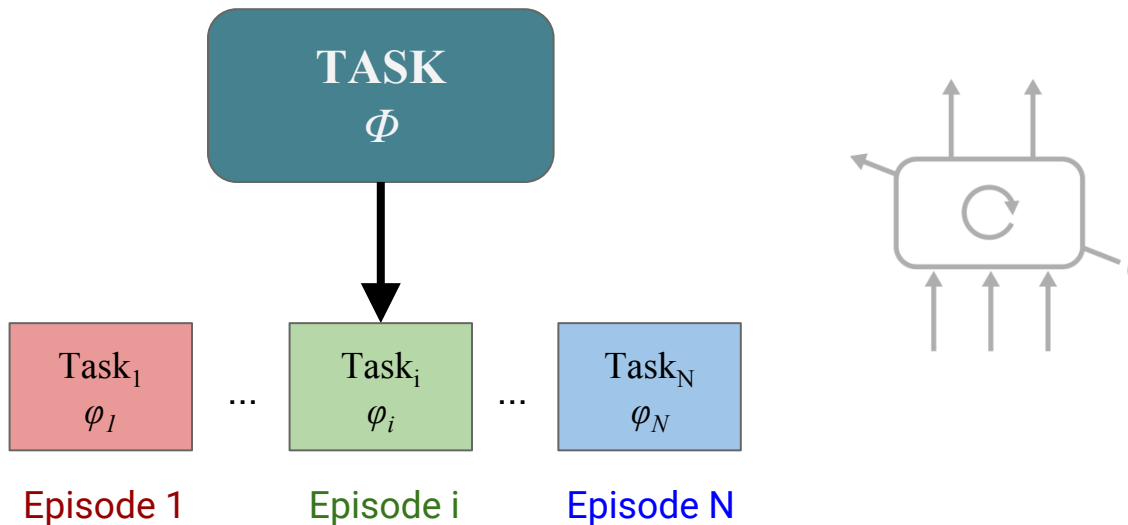


Training episodes



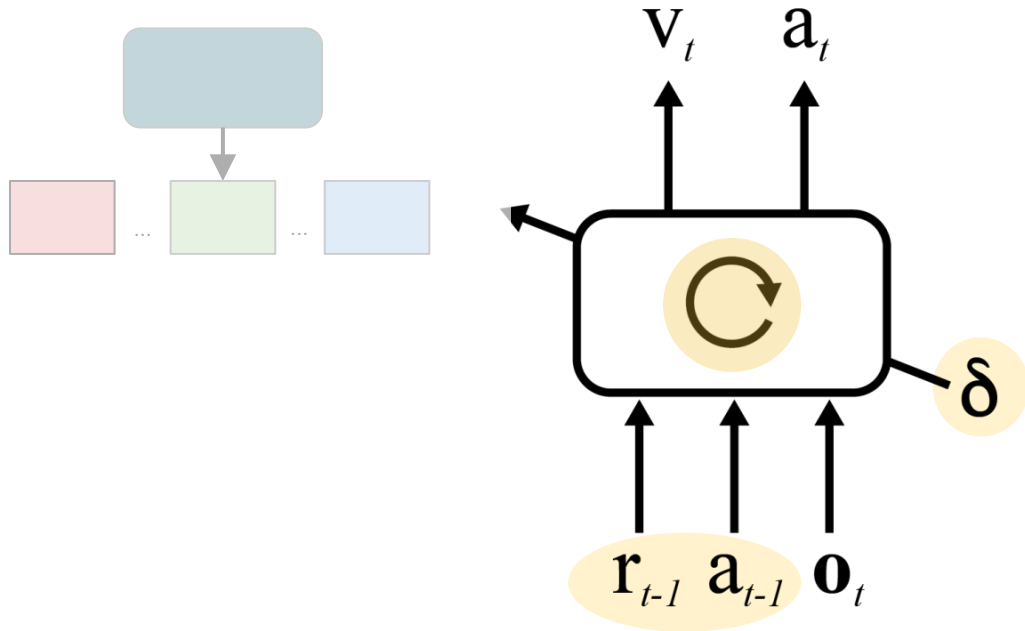
Training episodes

Ingredients: Environment



- **Distribution** of RL tasks with **structure**

Ingredients: Architecture

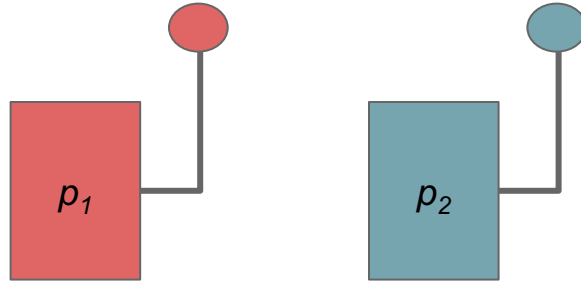


- **Primary** RL algorithm to train weights: Advantage actor-critic (*Mnih et al 2016*)
 - Turned off during test
- Auxiliary inputs in addition to observation: reward and action
- Recurrence (LSTM) to integrate history
- Emergence of **secondary** RL algorithm implemented in **recurrent activity dynamics**
 - Operates in absence of weight changes
 - **With potentially radically different properties**

Independent bandits

2-armed bandits
independently drawn from
uniform Bernoulli distribution

Held constant for 100 trials
=1 episode

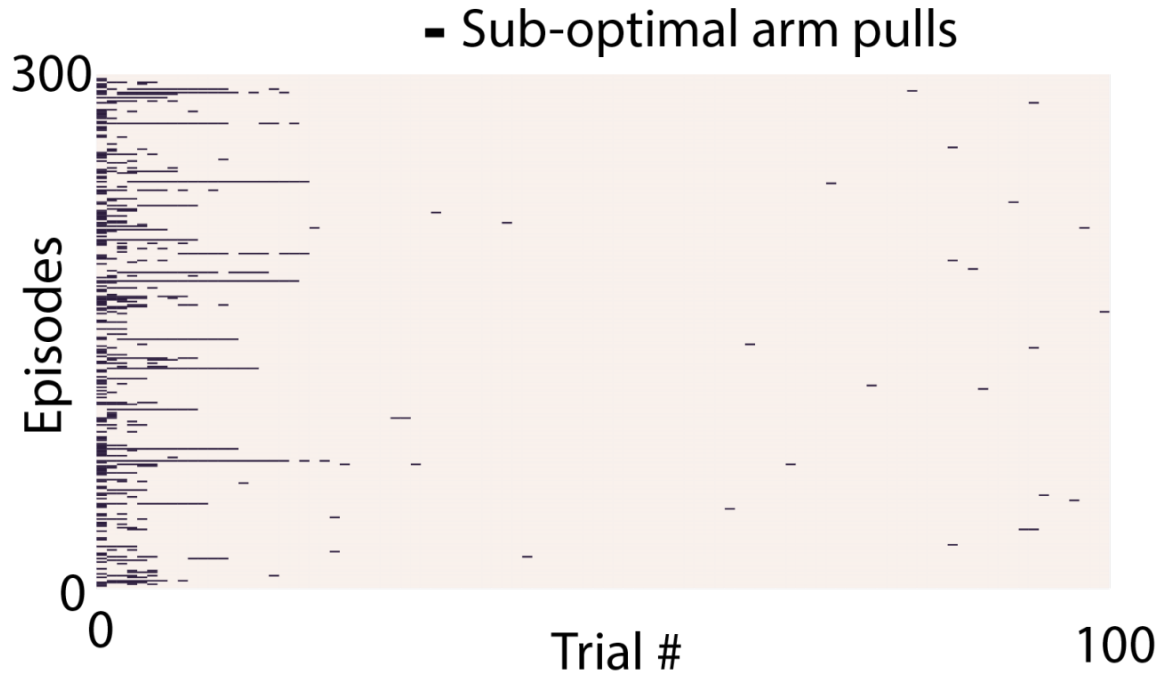


p_i = probability of payout,
drawn uniformly from $[0, 1]$,

Independent bandits

2-armed bandits
independently drawn from
uniform Bernoulli distribution

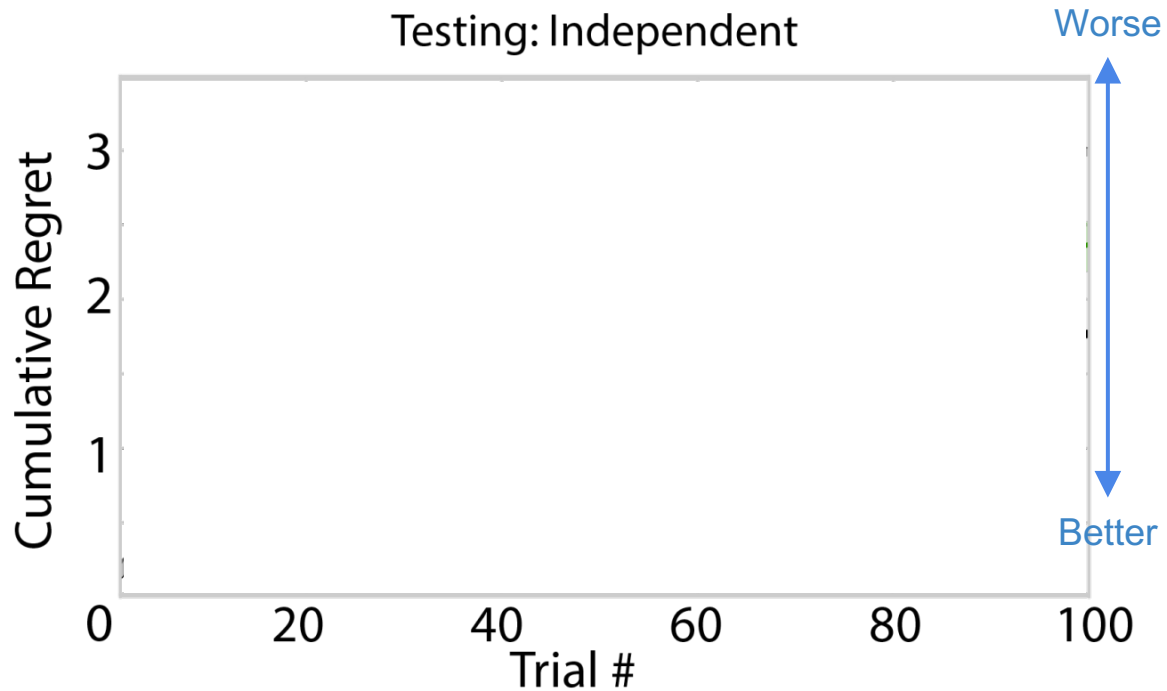
Tested with fixed weights



Independent bandits

2-armed bandits
independently drawn from
uniform Bernoulli distribution

Tested with fixed weights



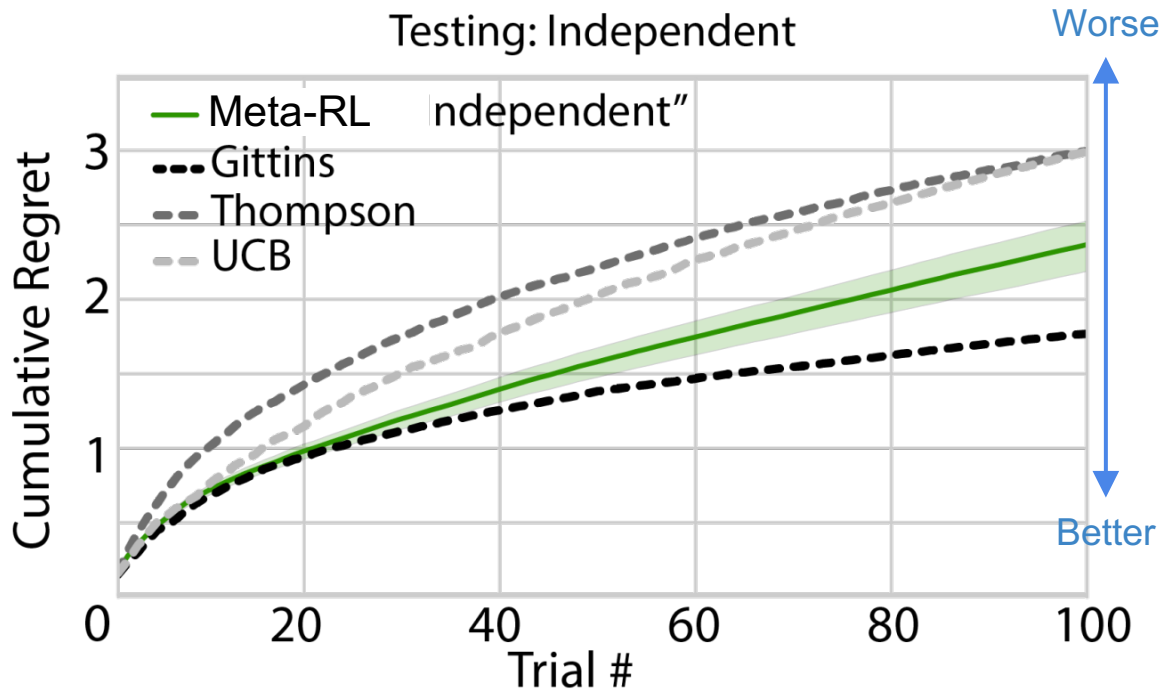
Independent bandits

2-armed bandits

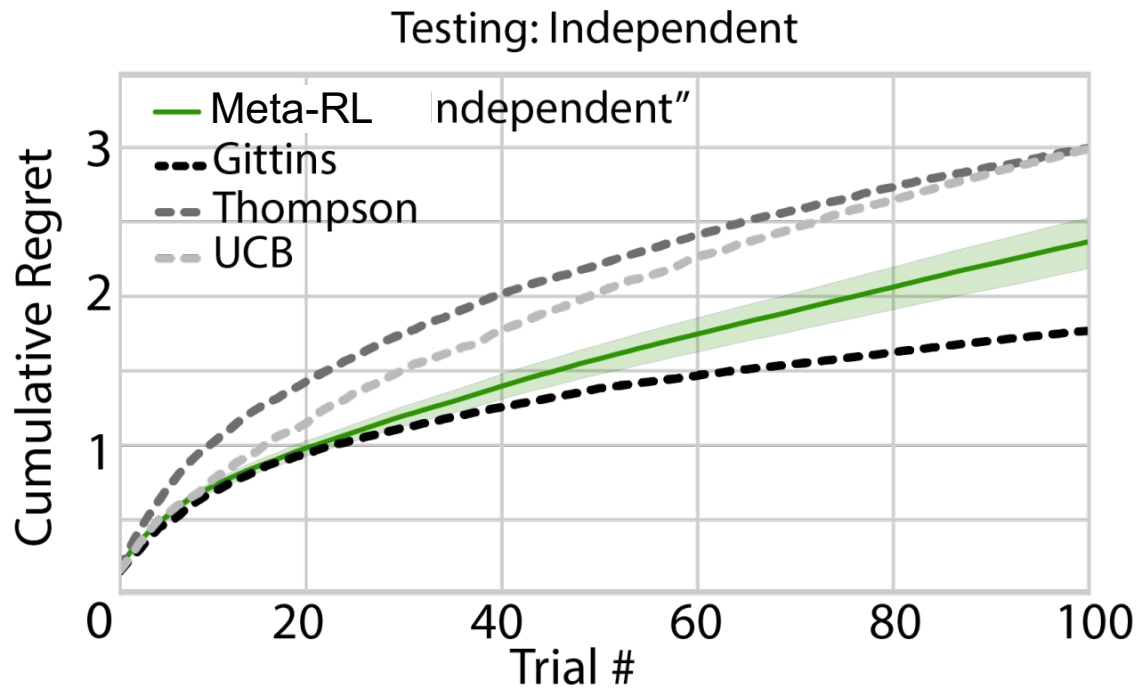
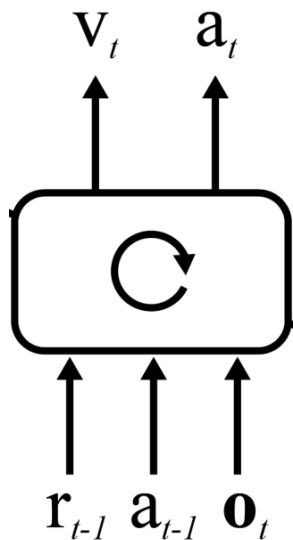
independently drawn from uniform Bernoulli distribution

Tested with fixed weights

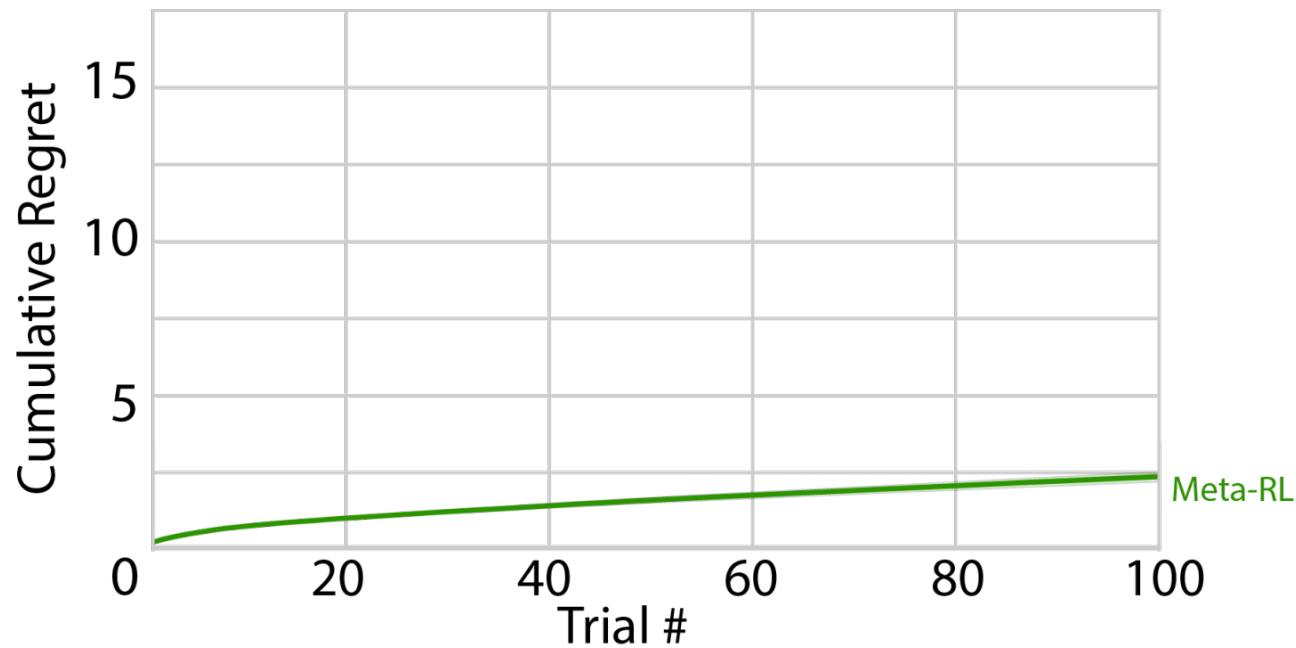
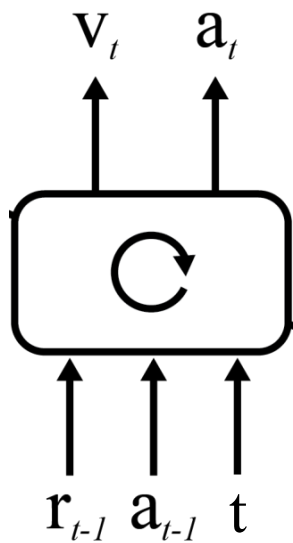
Performance comparable to standard bandit algorithms



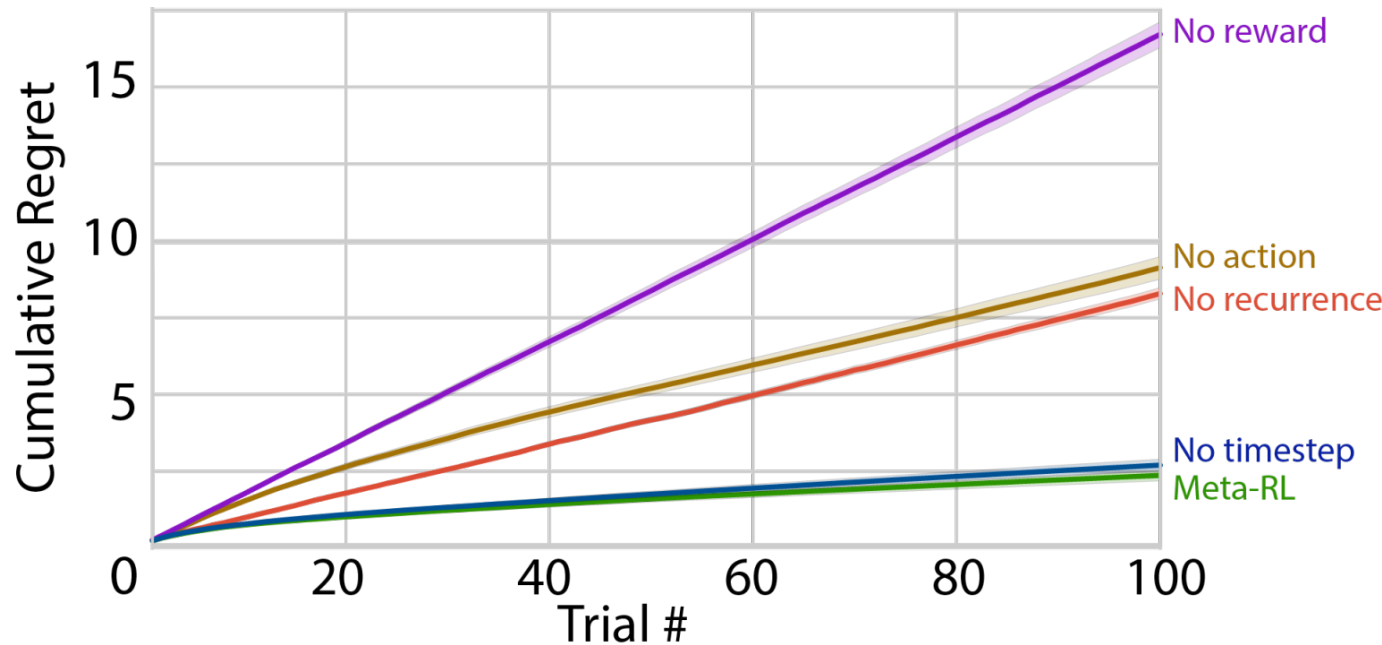
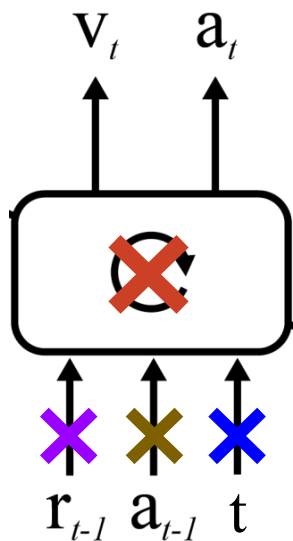
Ablation Experiments



Ablation Experiments



Ablation Experiments

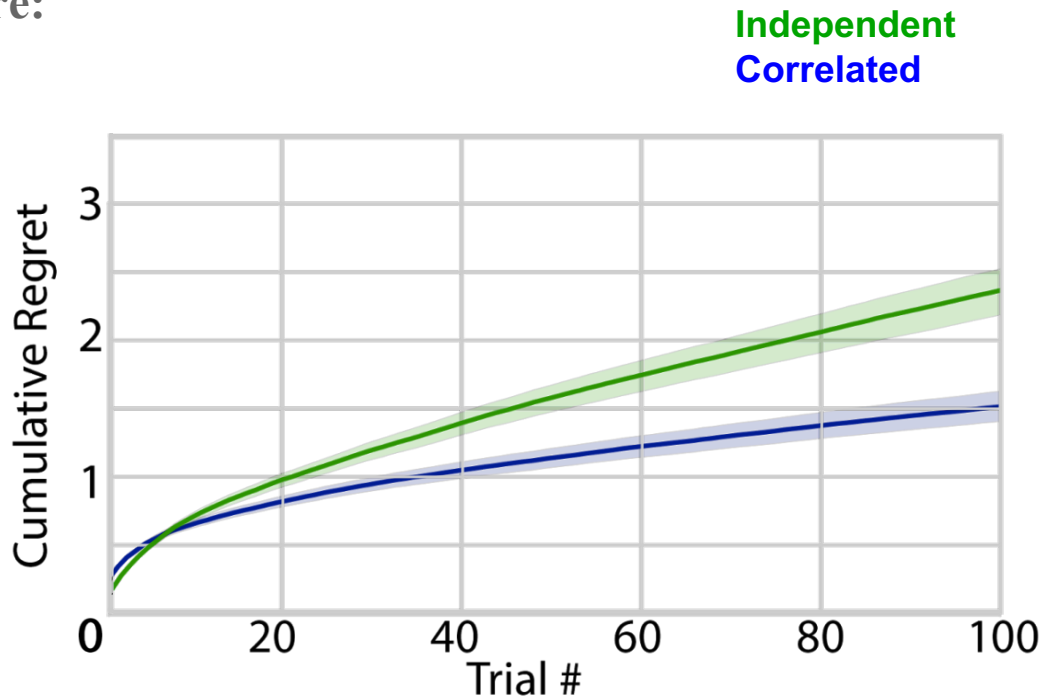


Structured bandits

Bandits with **correlational structure**:

$$\{p_L, p_R\} = \{\mu, 1-\mu\}$$

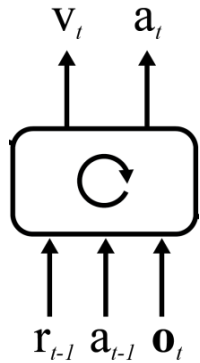
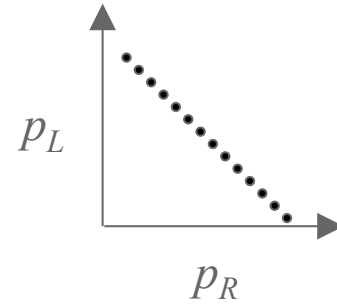
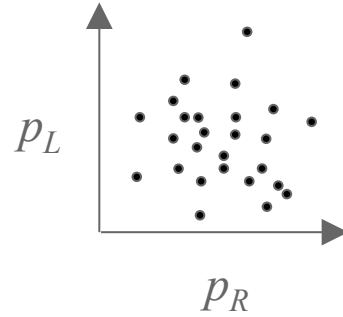
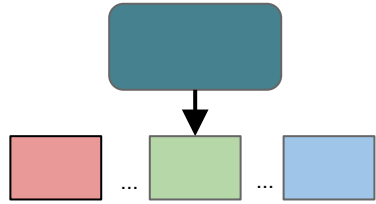
Meta-RL learns to exploit structure in the environment



LSTM hidden states internalize structure

Independent

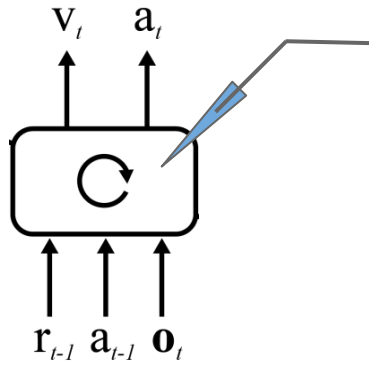
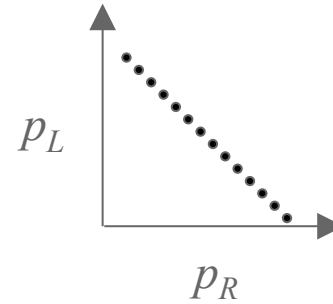
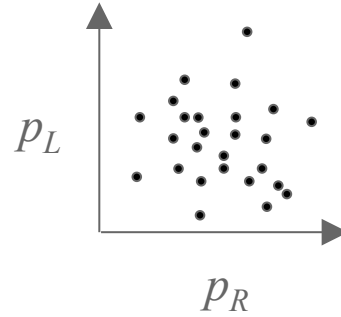
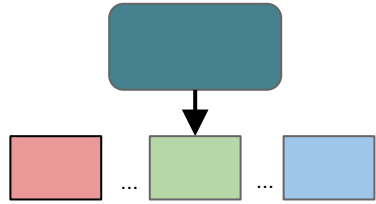
Correlated



LSTM hidden states internalize structure

Independent

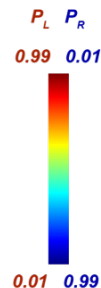
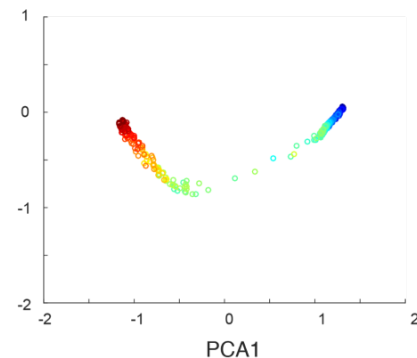
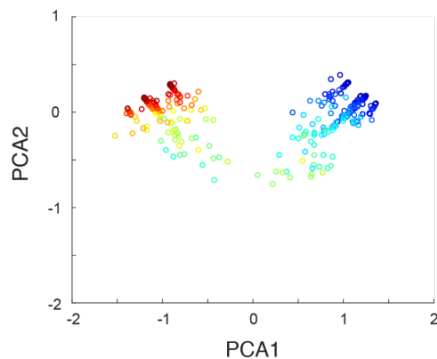
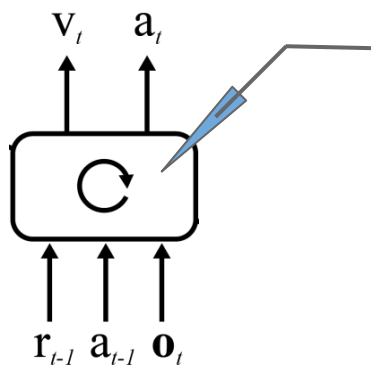
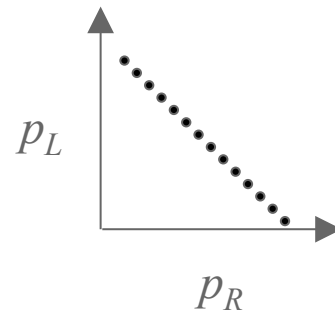
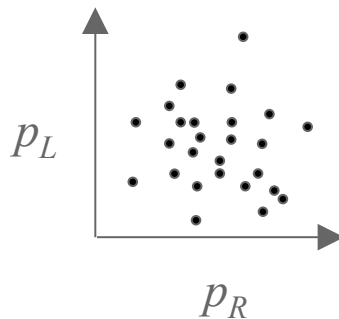
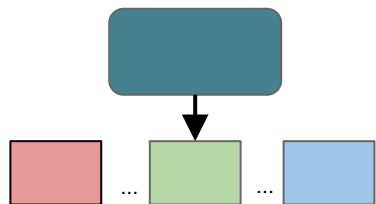
Correlated



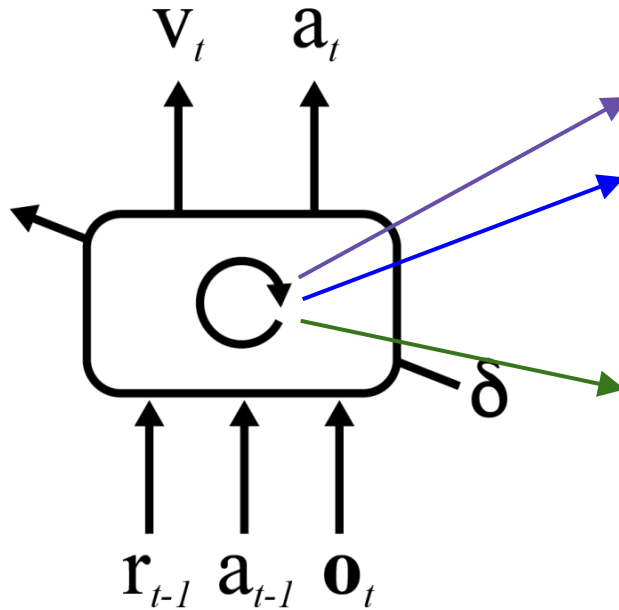
LSTM hidden states internalize structure

Independent

Correlated

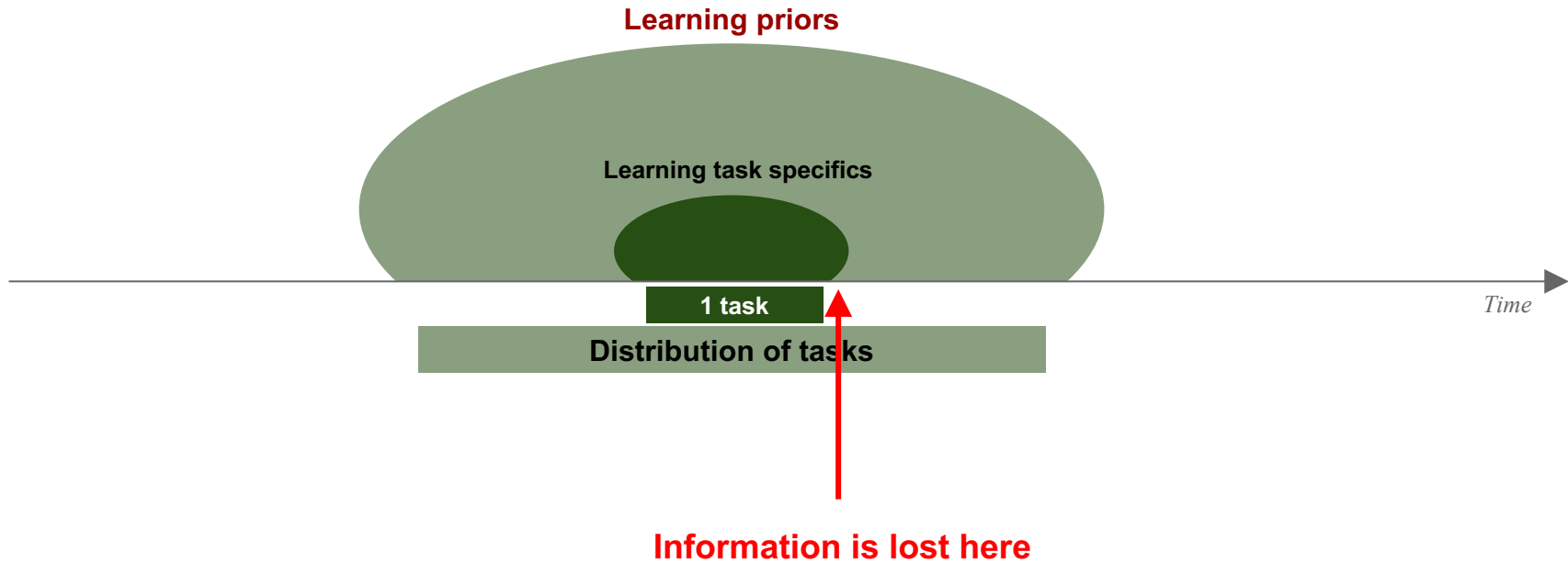


Emergent RL algorithm is capable of conforming to wide variety of task structure

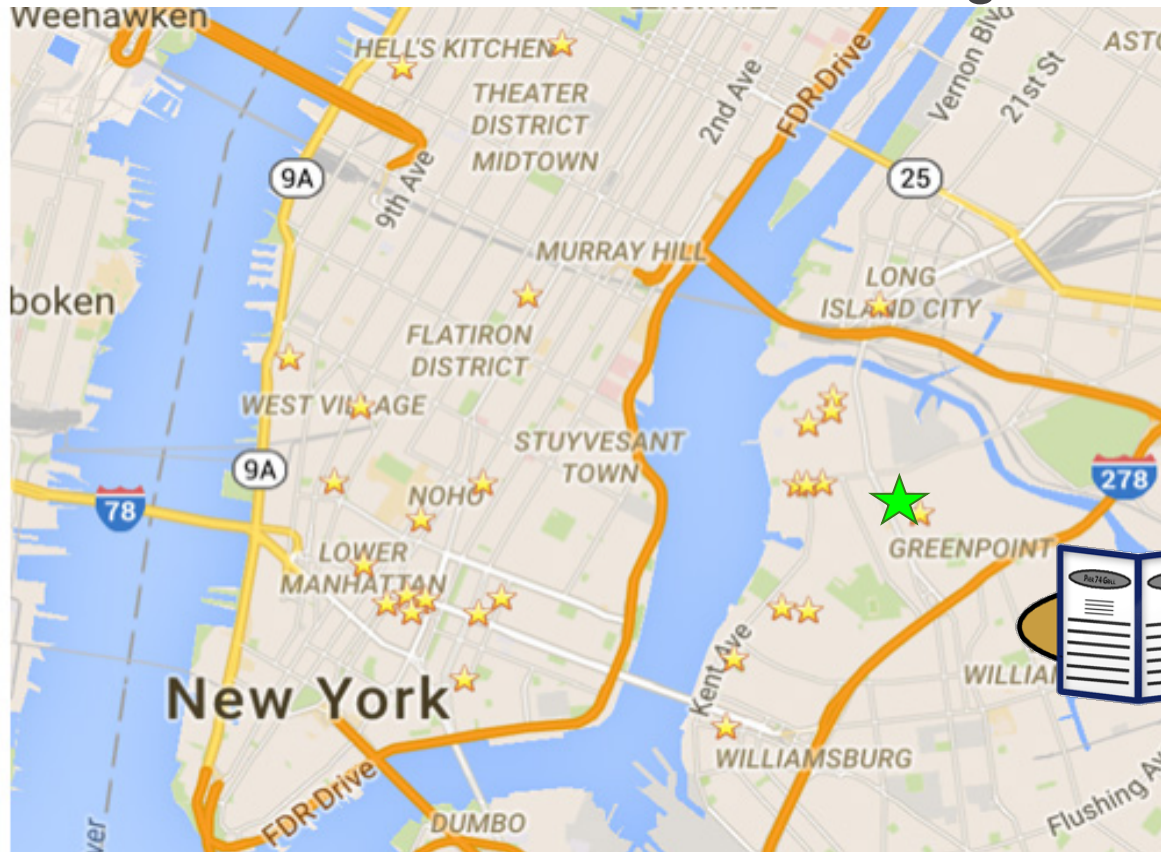


- Negotiate **exploration-exploitation** tradeoff
- **Leverage task structure** (correlations in environment, informative choices, abstractions, etc.)
- Display **different effective hyperparameters** (e.g., learning rate)
- ...

Drawbacks to using RNNs

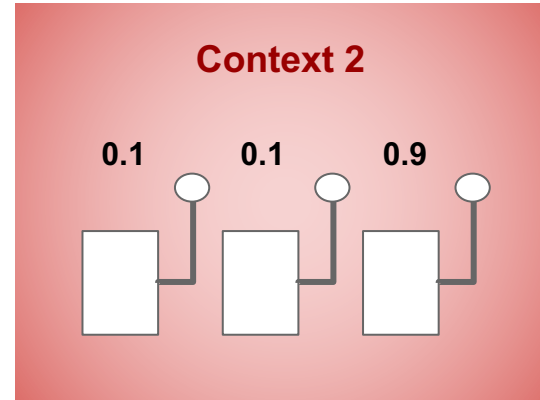
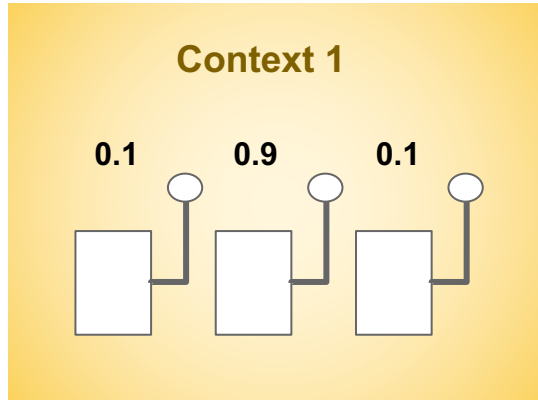


Using memory of specific past experiences to influence decision-making

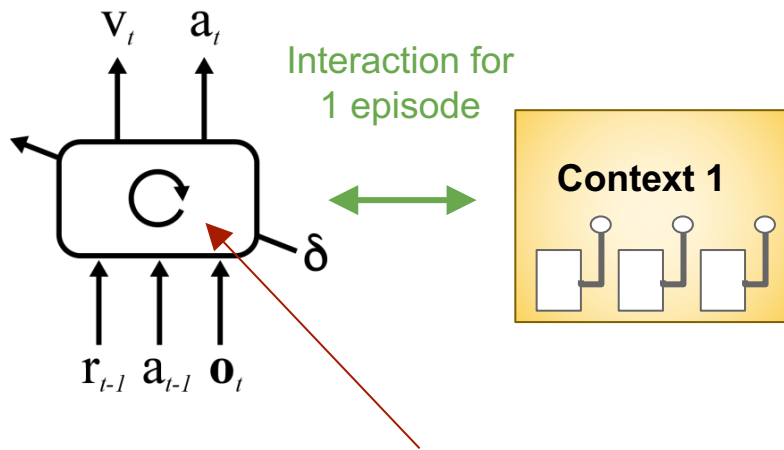


Contextual bandits

$p_r =$



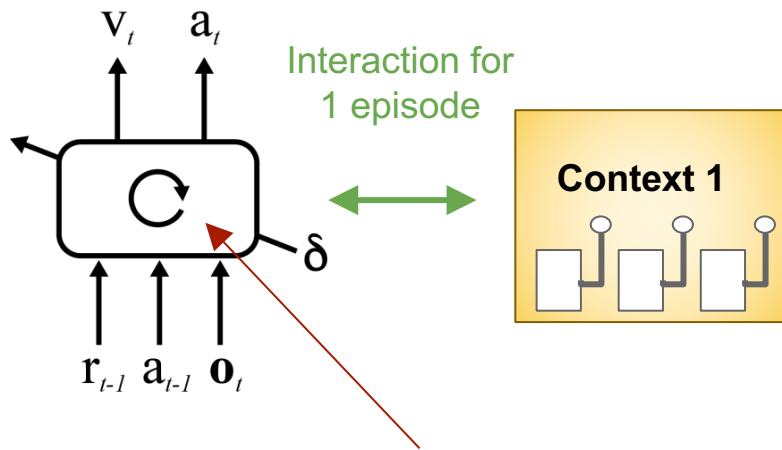
Using memory of past exploration



\mathbf{A}_1 : Hidden state at end of episode; contains critical task-related information

KEY	VALUE

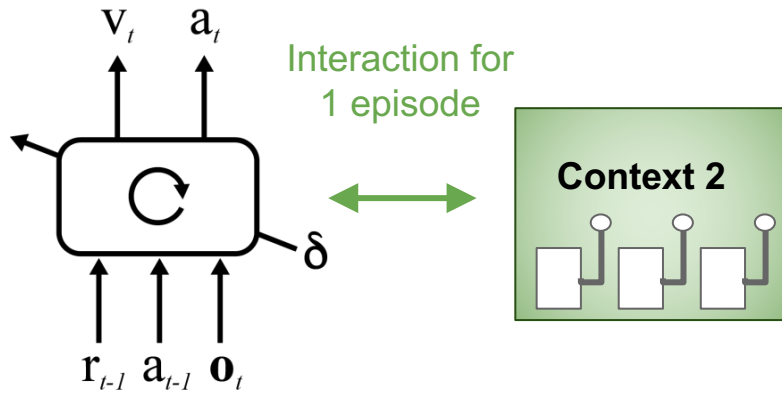
Using memory of past exploration



\mathbf{A}_1 : Hidden state at end of episode; contains critical task-related information

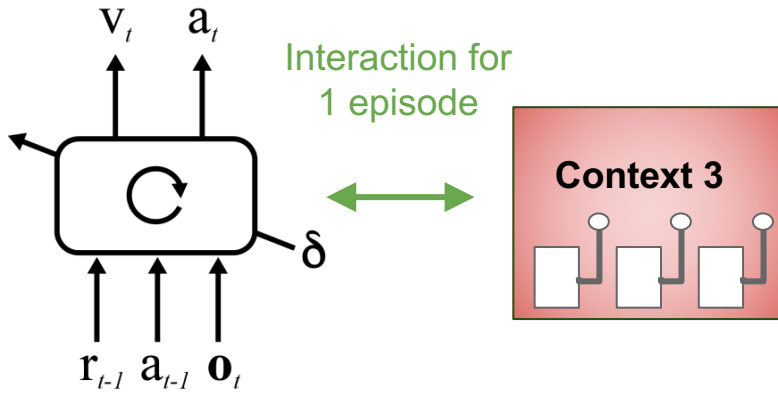
KEY	VALUE
Context 1	\mathbf{A}_1

Using memory of past exploration



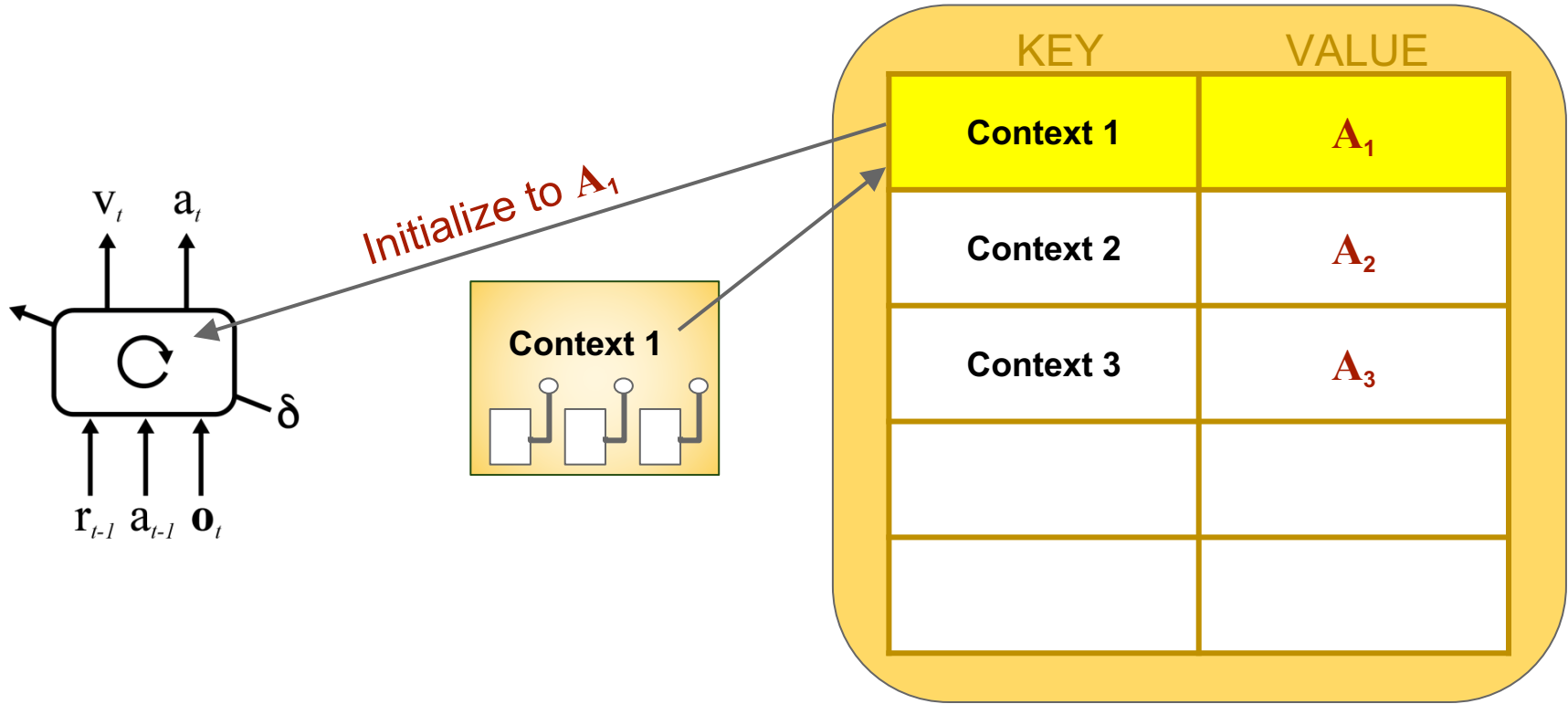
KEY	VALUE
Context 1	A_1
Context 2	A_2

Using memory of past exploration



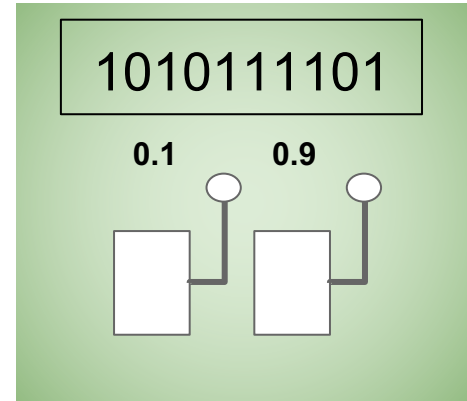
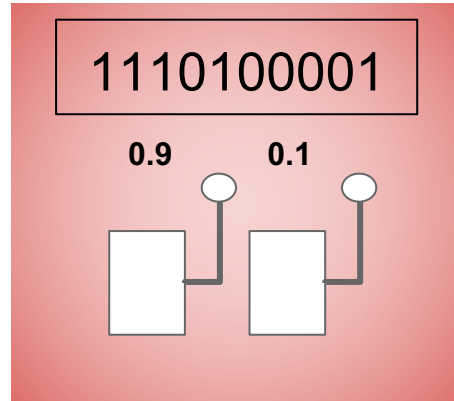
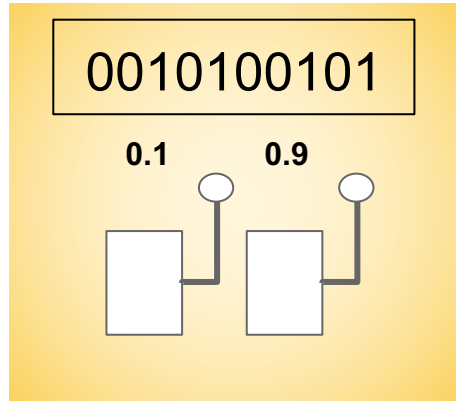
KEY	VALUE
Context 1	A_1
Context 2	A_2
Context 3	A_3

Using memory of past exploration



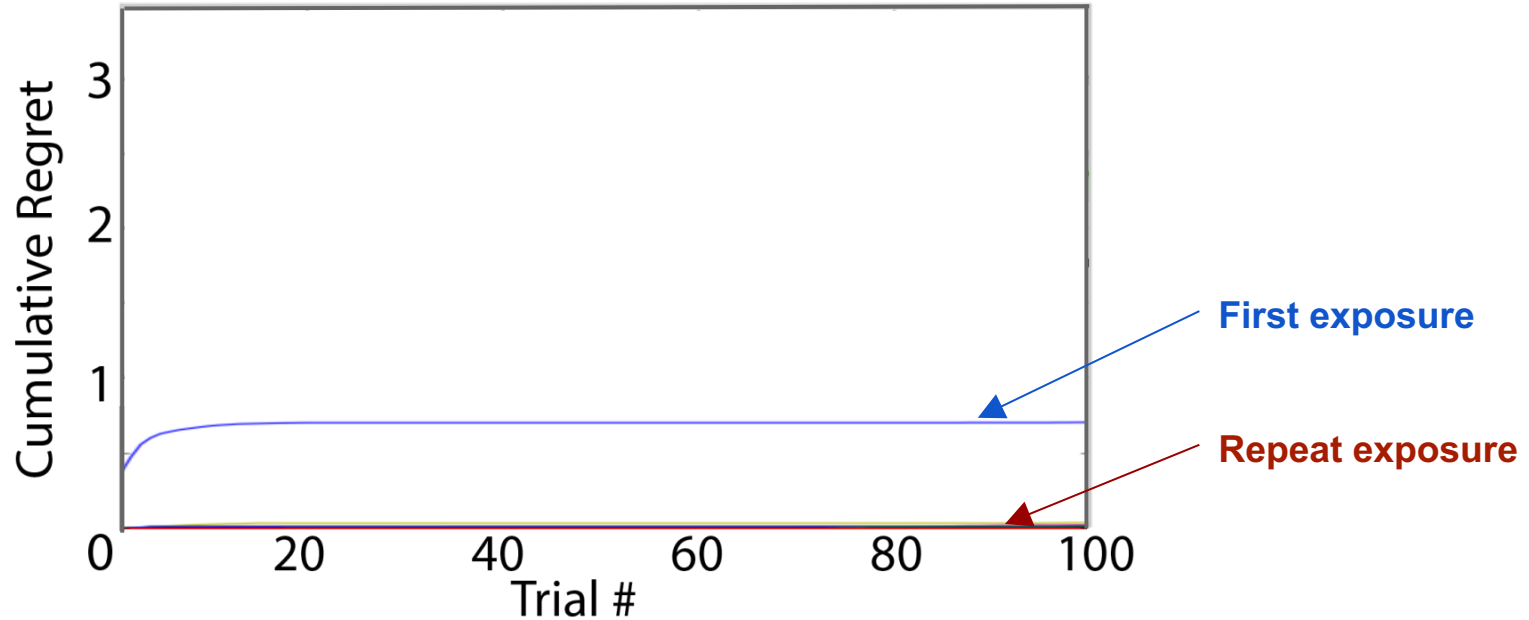
Contextual bandits: Barcodes

$p_r =$



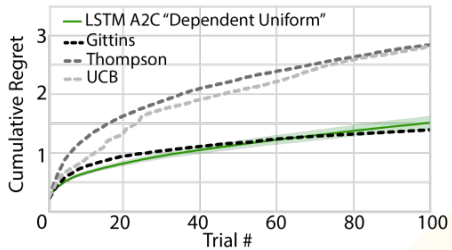
...

Contextual bandits: Barcodes



Meta-reinforcement learning

- Key requirements:
 - Recurrent dynamics integrating past reward, history, and observations
 - Primary error-based RL algorithm that uses reward prediction error to adjust weights
 - Distribution of interrelated tasks with shared structure
- Resultant effects
 - Structure of tasks is absorbed into the weights as priors, leading to faster learning with more tasks
 - Learned RL algorithm is implemented in recurrent activation, not weights, with potential to be drastically different from base algorithm, matched to task structure



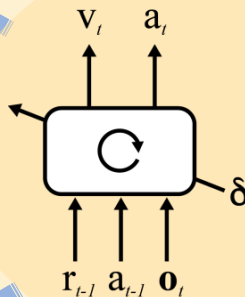
Exploration-exploitation



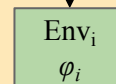
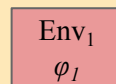
Complex task structure

Recurrent network with history input

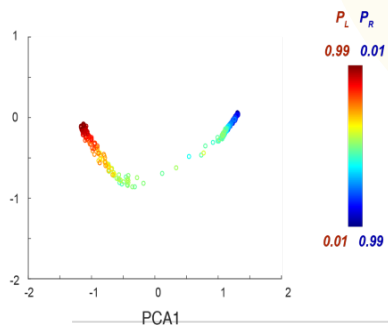
Trained on a set of interrelated RL tasks



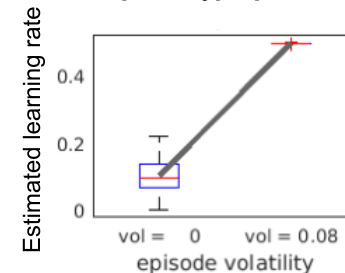
META-RL



Internalized task structure



Adaptive hyperparameters



Thank you!

Matt Botvinick
Zeb Kurth-Nelson
Sam Ritter
Dharshan Kumaran
Chris Summerfield
Hubert Soyer
Joel Leibo
Dhruva Tirumala
Remi Munos
Charles Blundell
Demis Hassabis
...and many others at DeepMind

All of you



DeepMind

joinus@deepmind.com