

Jane Wang  
(Deep Mind)

Fida Mohamed  
(Columbia)

Jacqueline  
Geffels  
(Columbia)

# NeurIPS 2019

## - Day 5 -

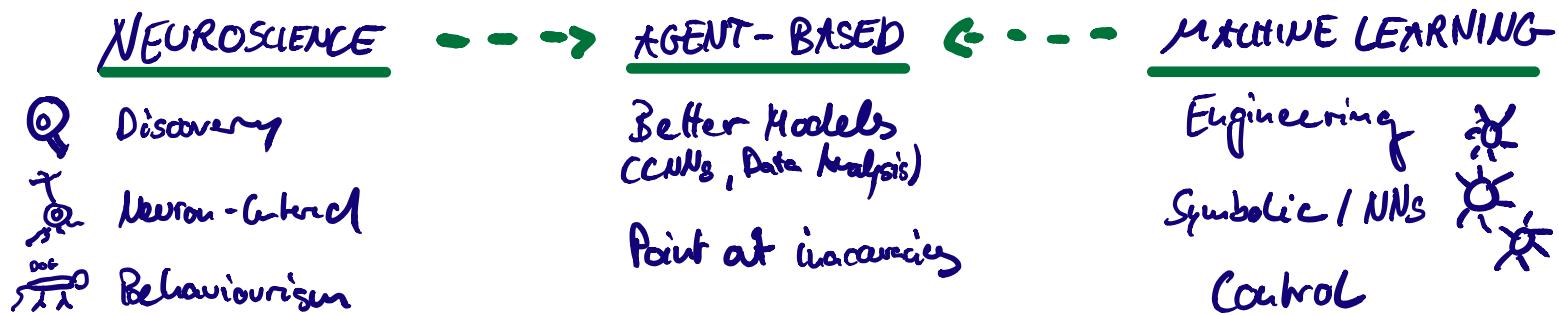
### WORKSHOPS I

Raia  
Hadsell  
(Deep Mind)

David Abel  
(Duke)

Rich Suttner  
(Uo Alberta)

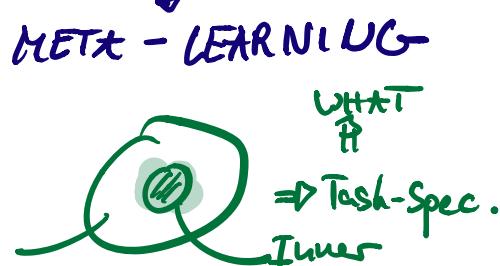
# Jake Wang (DeepMind) · From Brains to Agents & Back



## TWO TYPES OF LEARNING

- ① INNATE BEHAVIOURS → Slow
- ② LEARNED BEHAVIOURS → fast

↓  
BALDWINIAN EVOLUTION  
↳ Learning traits ⇒ Rest on!



## HARLOW TASK (Way et al. 18')

- |    |    |                                  |                   |
|----|----|----------------------------------|-------------------|
| 01 | 02 | ① Learn task structure<br>↳ SLOW | ② Fast adaptation |
|----|----|----------------------------------|-------------------|

↳ ImageNet RL Experiments  
⇒ Need many different images  
⇒ GENERALIZATION BETTER W.  
MORE TRAINING TASKS  
↳ PRIOR MISMATCH

## COOKIE-CUTTER ANXIETY

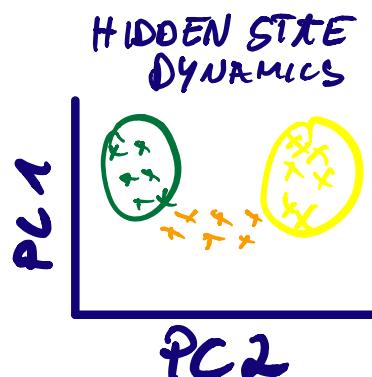
\* How to learn?  
\* What to learn?  
→ Zador 18': Place fields  
Propensity ↔ Inherent  
learned

## META-LEARNING IN RL

- ① Memory-based agent (LSTM)
- ② Task Distribution → Overlap

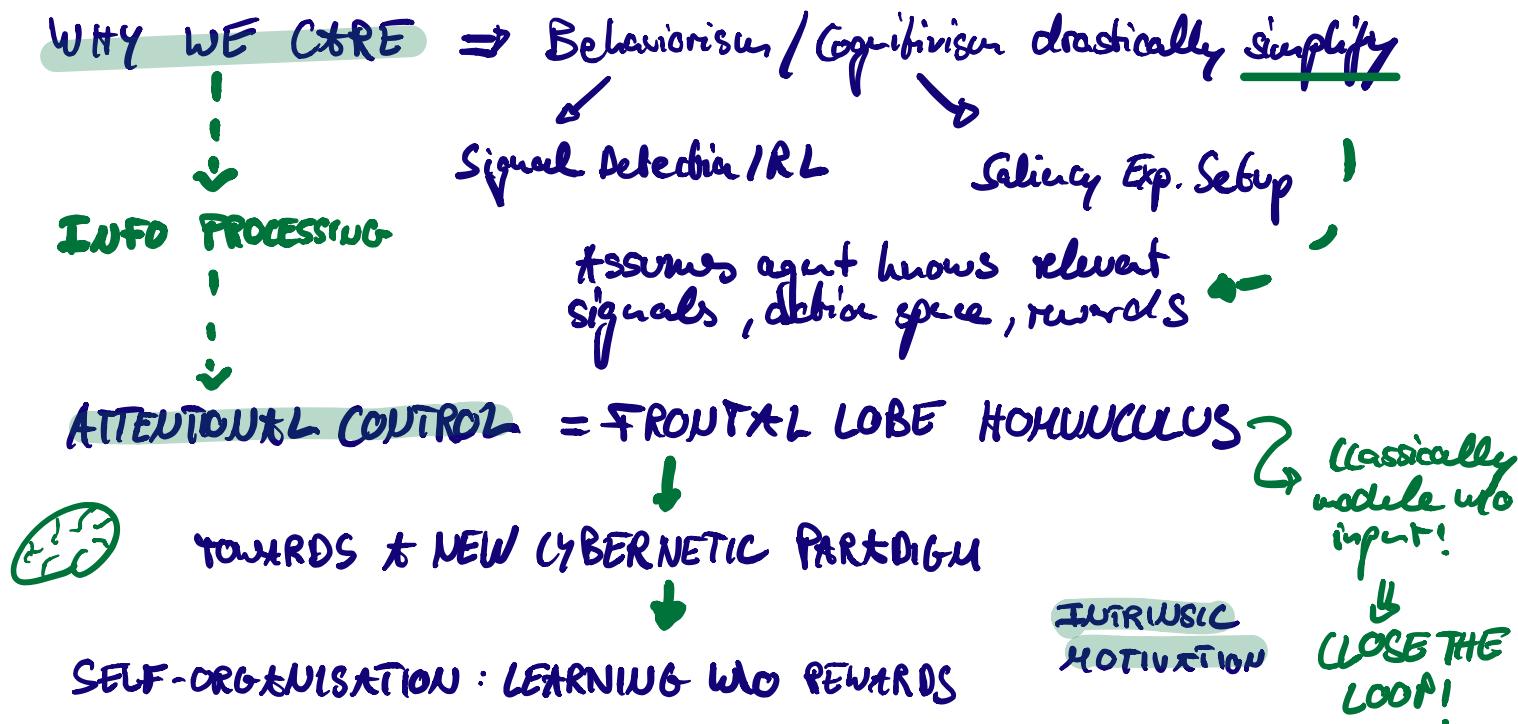
	<u>BIOLOGICAL</u>	<u>ARTIFICIAL</u>
<u>INNATE</u>	Ecological Evolved	Learned Struct. Priors
<u>LEARNED</u>	Lab Train	Env Train

## UNDERSTANDING DYNAMIC



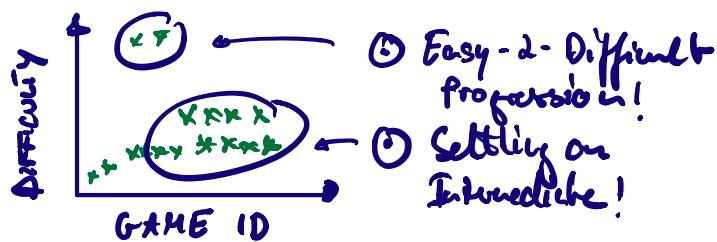
Convergence to  
Task Repr.  
Clusters!

# Jacqueline Gottlieb (Columbia): Info Demand - Simple & Complex

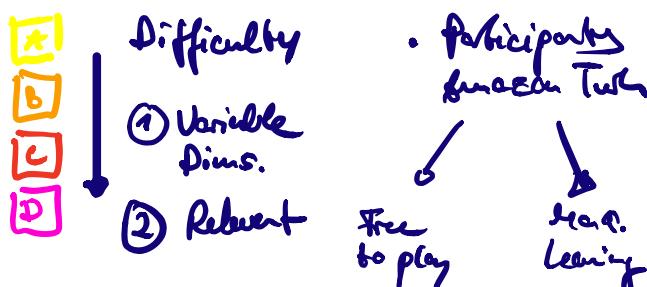


- HUMAN INFO DEMAND  $\rightarrow$  Complex setting  $\Rightarrow$  What to investigate?
- $\rightarrow$  Gottlieb & Oudeyer: LEARNING PROGRESS  
 $\hookrightarrow$  Reducible uncertainty
- $\rightarrow$  Optimal Student Paradigm: Many complications  $\Rightarrow$  Realisation of learning through uncertainty

$\rightarrow$  Baranes et al. 14': 'Guitar Hero'



$\rightarrow$  Endogenous Task Selection: 2-TFC



$\Rightarrow$  BEST LEARNERS CHOOSE INTERMEDIATE SELF-CHALLENGE LEVEL!

- ① RL Model  $\rightarrow$  Softmax Task Selection  
 $\hookrightarrow$  Selection sensitivity best model fit  
 $\Rightarrow$  Coupling reward nearer + profits!

TASK SELECTION DIFFERENCES

- $\rightarrow$  Learn Mar. game challenges itself more, Free game more chill!  
 $\hookrightarrow$  Individual Variation: Some participants in free game are highly motivated to explore fresh space

# Icha Momennejad (Columbia): Predictive Cognitive Maps

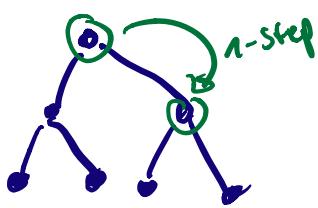
## HOW DO BRAINS BUILD COGNITIVE MAPS FOR LEARN./PLANNING?

↳ Tolman: Cognitive Maps in Hippocampus → Behrens et al. 18'

### KIND OF REPRESENTATIONS

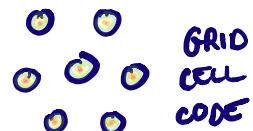
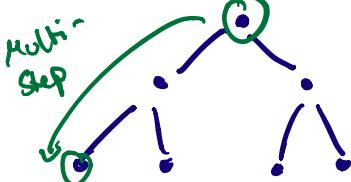
1-Step Transitions

MB Learner



Successor Repr. (Dayan, 1998)

$$V(s) = \sum_{s'} M(s, s') R(s')$$



MULTI-SCALE  
⊕ MULTI-STEP

### HOW TO LEARN?

$$M_{\pi_t, p}(s, s') = E_p \left[ \sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s') \mid s_0 = s \right]$$

$$\hat{M}_{\text{true}}(s_t, s') \leftarrow \hat{M}_b(s_t, s')$$

$$+ \gamma [\mathbb{1}(s_t = s') + \gamma \hat{M}_b(s_{t+1}, s') - \hat{M}_b(s_t, s')]$$

SUCCESSOR PRED. ERROR

① Kim Stachenfeld Studies → place & Grid Cells

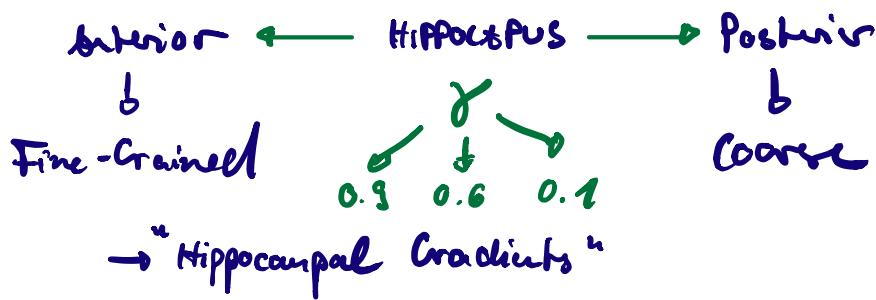
② Momennejad et al. 17' ⇒ Reward vs. Transition Revaluation  
⊕ Russell\*



↳ SUCCESSOR REPRESENTATION DYN → Best at explaining human  
⊕ rodent behavior!  
↳ also for policy recall.

↳ EPILEPSY SINGLE-CELL RECORDINGS → Place Cells ⇒ Room 2 Room Sw.

↳ LARGE-SCALE TORONTO MAP NAVIGATION → Levels of abstraction



↳ Familiar vs. GPS condition

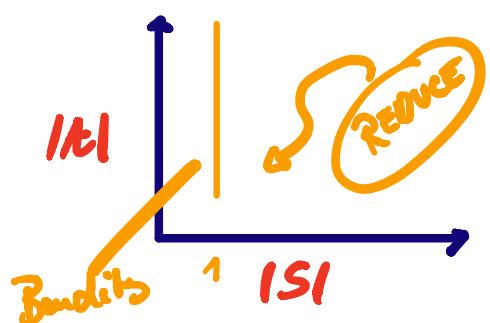
↳ DYN VALIDATION

Surprise ⇒ Replay + Replay  
Momennejad et al. 18'  
↳ prioritization 10%

⇒ PREDICTIVE HORIZON CONTROLLED BY DISCOUNT

# David Abel (Duke) : Abstraction & Meta RL

## SPACE OF MDPs



LEARN ABSTRACTIONS THAT MAKE RL EASIER

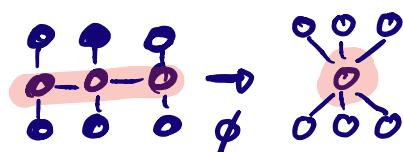
STATE ABSTR.

ACTION ABSTR.

$$\phi: S \rightarrow S_\phi$$

Option framework  $O \rightarrow \pi_o$   
(Sutton et al., 1999)

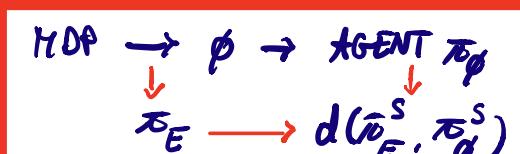
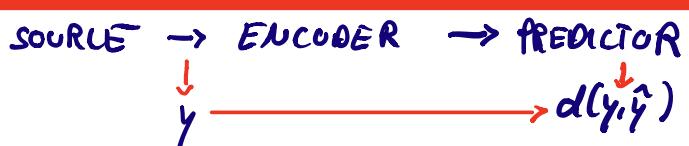
STATE ABSTRACT ACTION ABSTRACT



## STATE ABSTRACTIONS : COMPRESSION VS. VALUE PRESERVATION

→ How small can  $|S_\phi|$  be while still preserving 'good' solutions!

→ Info Bottleneck (Tishby et al., 99')  $\Rightarrow$  RL abstraction Perspective:



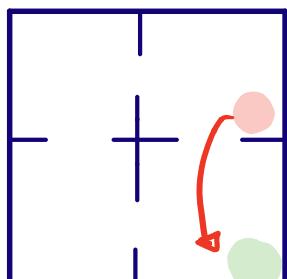
RL Formulation :  $\min_{\phi} |S_\phi| + \beta [E[V^{\pi_E}(s)] - V^{\pi_\phi}(s)] \leq \text{DIB Objective}$

SAMPLE EFFICIENCY  $\xrightarrow{\phi: \text{Only compression}} \xrightarrow{\beta: \text{Trade-off}} \xrightarrow{\text{4 Rooms Problem}} \xrightarrow{\text{Lunar Lander}}$

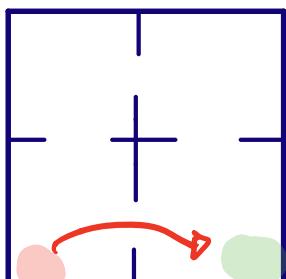
## ACTION ABSTRACTIONS : OPTIONS FOR VALUE PLANNING

$V=1$

OPT.



PPROK.



① Allow for better value propagation!

② NP-Hard Problem

↳ Not clear on qualitative differences

# Raia Hadsell (DeepMind) : Challenges for Continual & Meta RL

① Learning from i.i.d. data



"1-page" per topic in "batch" analogy



INSTEAD: PROLONGED INSPECTION / CONSIDERATION ON ONE TOPIC

② How FAST / SLOW should learning algo be? → Adaptation/Evolution  
WHAT WE WANT

NOT TOO-FAST

NON-STATIONARY

NOT TOO-SLOW

NON-IID

META-LEARN

② MEMORY

OPTIMIZATION ①

COMPARE

TASK INFER.

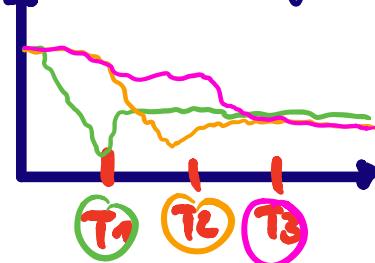
① WRAPPED GRADIENT DESCENT (WarpGrad) → S. FLENNER HTG

↓  
Learn geometry does not depend  
on idit but on search space

↓  
Explicit precond. & our space

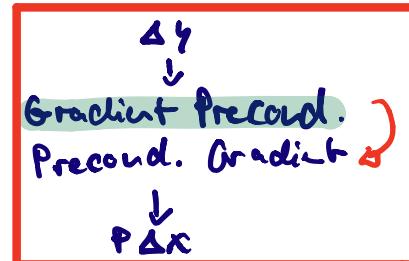
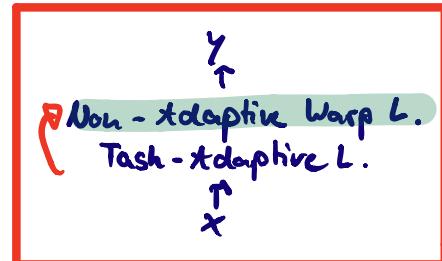
↓  
Couline w. HTML-type init!

→ Continual learning application (Siu Reg)



③ Get faster  
at each new  
task!

FORWARD



② STABILIZING TRANSFORMERS FOR RL (GTrXL) → E. Perisotto

↓  
Over time usage! Problem:  
Hard to optimize

↓  
DM Lab application w.  
MERLIN (Wayne et al. 18)

Reactive  
Levels

Memory  
Levels

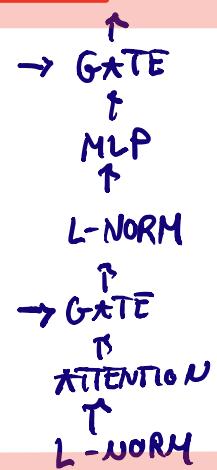
① Differences to TrXL

→ Move layer norm inside  
↳ better gradient flow

→ GRU gating (was best!)

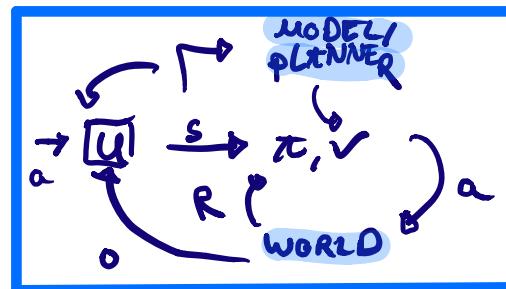
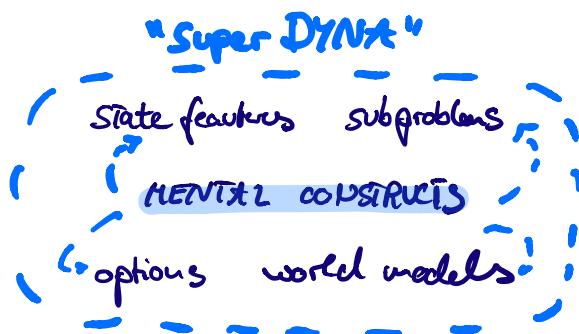
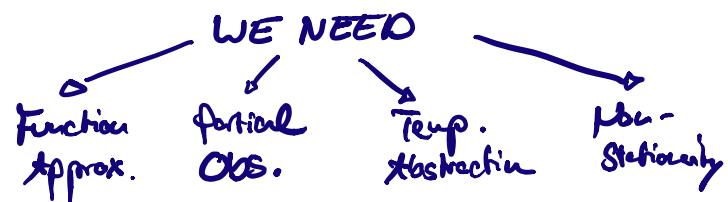
→ Param pruning ≈ 22M.

→ Merged Sequences



# Rich Sutton (U of Alberta): Towards a General-AI architecture

WTW: GENERAL PRINCIPLES  
TO LEARN TO...  
PREDICT → CONTROL



## STATE FEATURE CONSTRUCTION

- ① Linear GVF
- ② Non-Linear Update
- ③ SGD + Search
- ④ Recurrent Process

## DYNA (1990) PLANNING

- ① Learn model T, R
- ② DP-Style Planning  
↳ 'look-ahead'  
⇒ Output 2 feature

Distribution Model  
Sampling Model  
Expectation Model  
Nothing lost if linear!

## VALUE ITERATION WITH DISTRIBUTION/EXPECTATION MODEL

$$u_w(s) = w^T x(s) = \max_a [\hat{r}(s, a) + \sum_{s'} \hat{p}(s'|s, a) u_w(s')] = \max_a [\hat{r}(s, a) + w^T \hat{x}(s, a)]$$

- ① Linearity is key! ⇒ Nothing lost → Read out linear in learned features

SUBPROBLEMS ⇒ SHAPES REPRESENT.  
BEHAVIOR + HIGH-L. PLANNING

- ② What should problems be?  
⇒ Each subproblem char. one state feature
- ③ Where do subproblems come from?  
⇒ Come from state features!
- ④ How do subs help work?  
⇒ Solution to sub is option over. func  
↳ Plan on top!

NON-STATIONARITY  
⇒ MEMORY

Permanent	Transient
$\alpha, w_0$	$\tilde{\alpha}, w_0 + \tilde{w}_0$
$\alpha$ small	$\tilde{\alpha} \gg \alpha$

Not Required if Vexact

## "SUPER-DYNA"

