# Texts for the course

Christopher Bishop
Pattern Recognition and Machine Learning

Stephen Marsland
Machine Learning *An algorithmic perspective*

| | |
|---|---|
| Kevin Murphy | Detailed |
| Trevor Hastie | Useful on some topics |
| Mehryar Mori | Theoretical |
| Simon Hakin | Useful but somewhat dated |
| Ethem Alpaydin | ?? |

# Value Iteration

initialize $V(s)$ arbitrarily

loop until policy good enough

loop for $s \in S$

loop for $a \in A$

$$Q(s,a) := R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V(s')$$

$$V(s) := \max_a Q(s,a)$$

end loop

end loop

# Policy Iteration

choose an arbitrary policy $\pi'$

loop

$\pi := \pi'$

compute the value function of policy $\pi$ :

solve the linear equations

$$V_\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_\pi(s')$$

improve the policy at each state:

$$\pi'(s) := \arg\max_a \left( R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_\pi(s') \right)$$

until $\pi = \pi'$

# Temporal Difference Learning

**The successive values on paths through the network have to be related by the discount factor**

$$\bar{V}_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

where $0 \leq \gamma < 1$. This formula can be expanded

$$\bar{V}_t = r_t + \sum_{i=1}^{\infty} \gamma^i r_{t+i}$$

by changing the index of i to start from 0.

$$\bar{V}_t = r_t + \sum_{i=0}^{\infty} \gamma^{i+1} r_{t+i+1}$$

$$\bar{V}_t = r_t + \gamma \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}$$

$$\bar{V}_t = r_t + \gamma \bar{V}_{t+1}$$

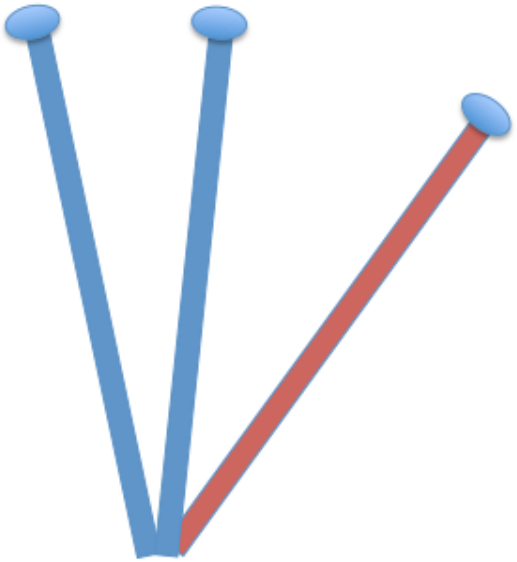
Thus, the reinforcement is the difference between the ideal prediction and the current prediction.

$$r_t = \bar{V}_t - \gamma \bar{V}_{t+1}$$

# Model-free learning has no initial T

## have to learn T by experimentation. Which action to try?

This setting is known as the *multi-arm bandit problem* after Las Vegas slot machines



pulling an arm results in a random reward for that arm.

Which arm has the best average reward?

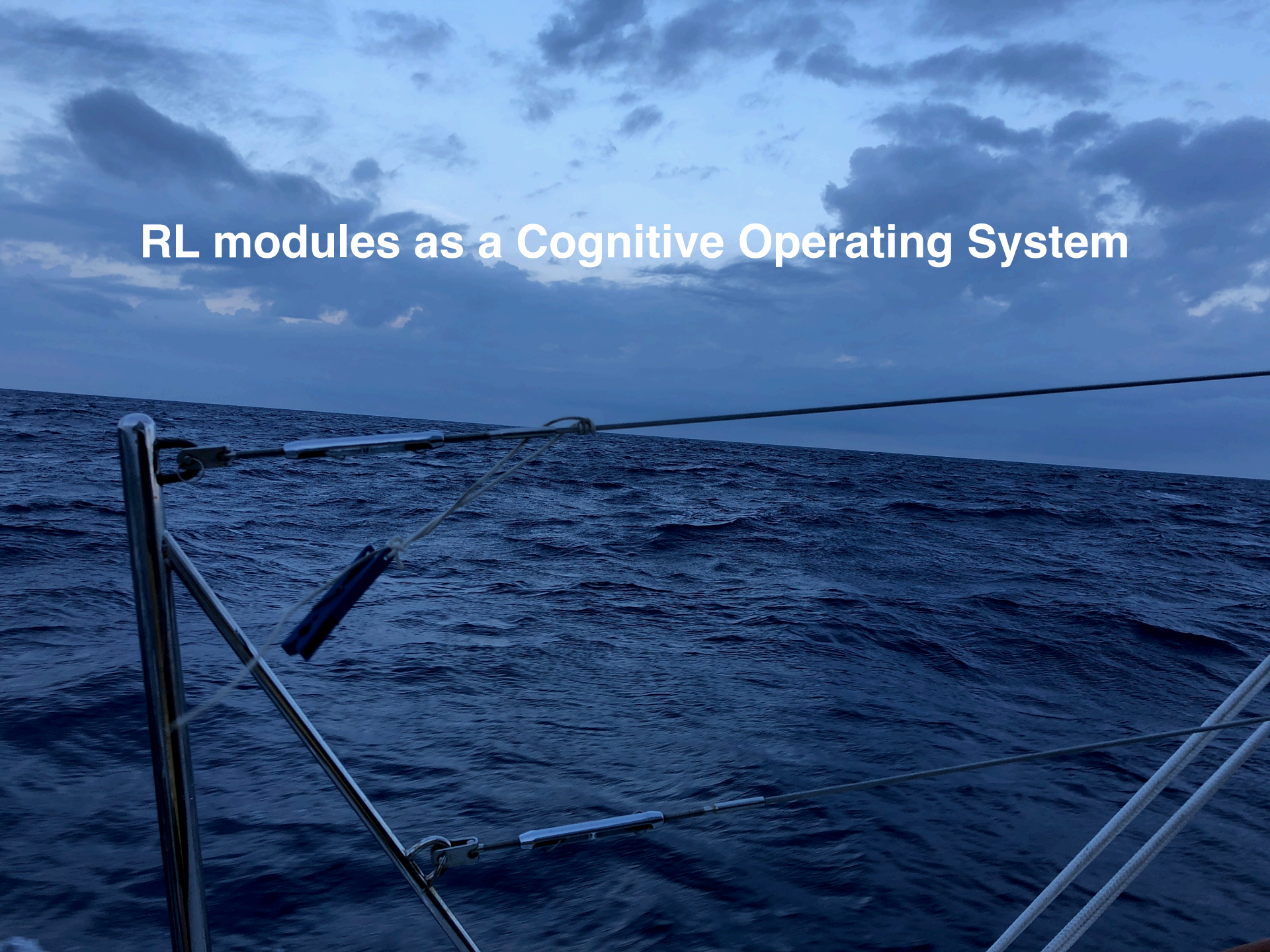
### Epsilon-greedy algorithm

Pull the best arm with probability 1- $\epsilon$
Pull the other arms, including the best, with probability $\frac{\epsilon}{3}$

A value for $\epsilon$ might be 10%

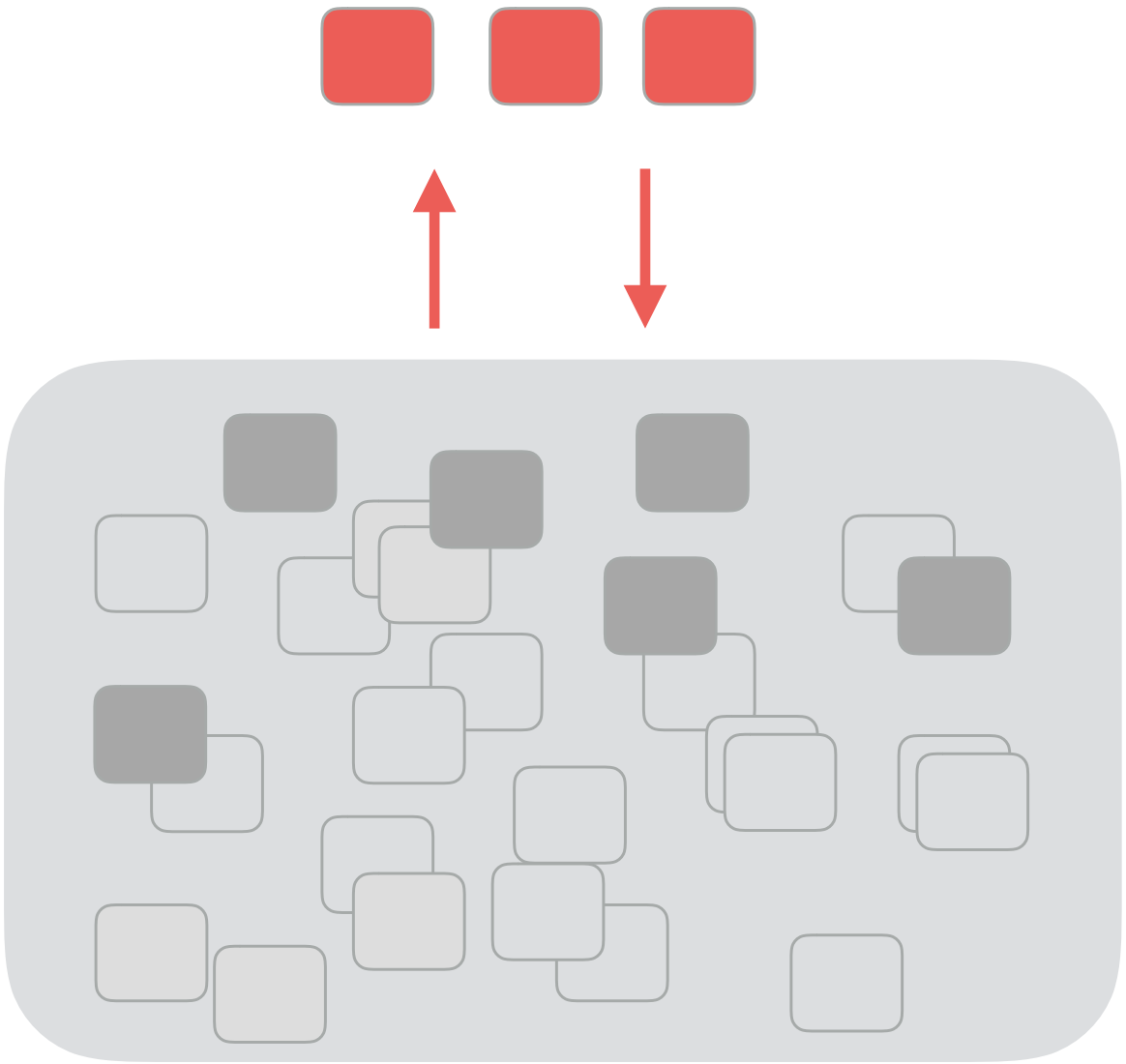
## The Sarsa Algorithm

- **Initialisation**
  - set $Q(s, a)$ to small random values for all $s$ and $a$
- Repeat:
  - initialise $s$
  - choose action $a$ using the current policy
  - repeat:
    - take action $a$ and receive reward $r$
    - sample new state $s'$
    - choose action $a'$ using the current policy
    - update $Q(s, a) \leftarrow Q(s, a) + \mu(r + \gamma Q(s', a') - Q(s, a))$
    - $s \leftarrow s'$, $a \leftarrow a'$
  - for each step of the current episode
- Until there are no more episodes


RL modules as a Cognitive Operating System

# Modules Hypothesis

## Routines can be learned with reinforcement and run in small groups



Forebrain contains a vast number of remembered modules

**A module is an independent perception to action unit**

**A small number of active modules governed by working memory**

**Each module has its own rewards**

**Active modules may be swapped out on a 300ms time scale to meet changing contingencies**

# Setting: Sensori-motor behaviors

Modules have to agree on an action

$$\mathcal{M}_i = \{S_i, A, T_i, R_i\}$$

One way: Average

$$Q(s_t, a_t) = \sum_{i=1}^{M} Q(s_t^{(i)}, a_t^{(i)})$$

Better way: Softmax using

$$P(a_t^{(j)} | Q(s_t^{(1)}, a_t), ..., Q(s_t^{(M)}, a_t)) = \frac{e^{Q(s_t^{(j)}, a_t^{(j)})/\tau}}{\sum_{i=1}^{M} e^{Q(s_t^{(i)}, a_t^{(i)})/\tau}}$$

# Q-Learning variant of Temporal Difference Learning

The goal of RL is to find a policy $\pi$ that maps from the set of states $S$ to actions $A$ so as to maximize the expected total discounted future reward

$$V^{\pi}(s) = E^{\pi}\Big(\sum_{t=0}^{\infty} \gamma^t r_t\Big) \tag{1}$$

Alternatively, the values can be parametrized by state and action pairs, denoted by $Q^{\pi}(s, a)$.

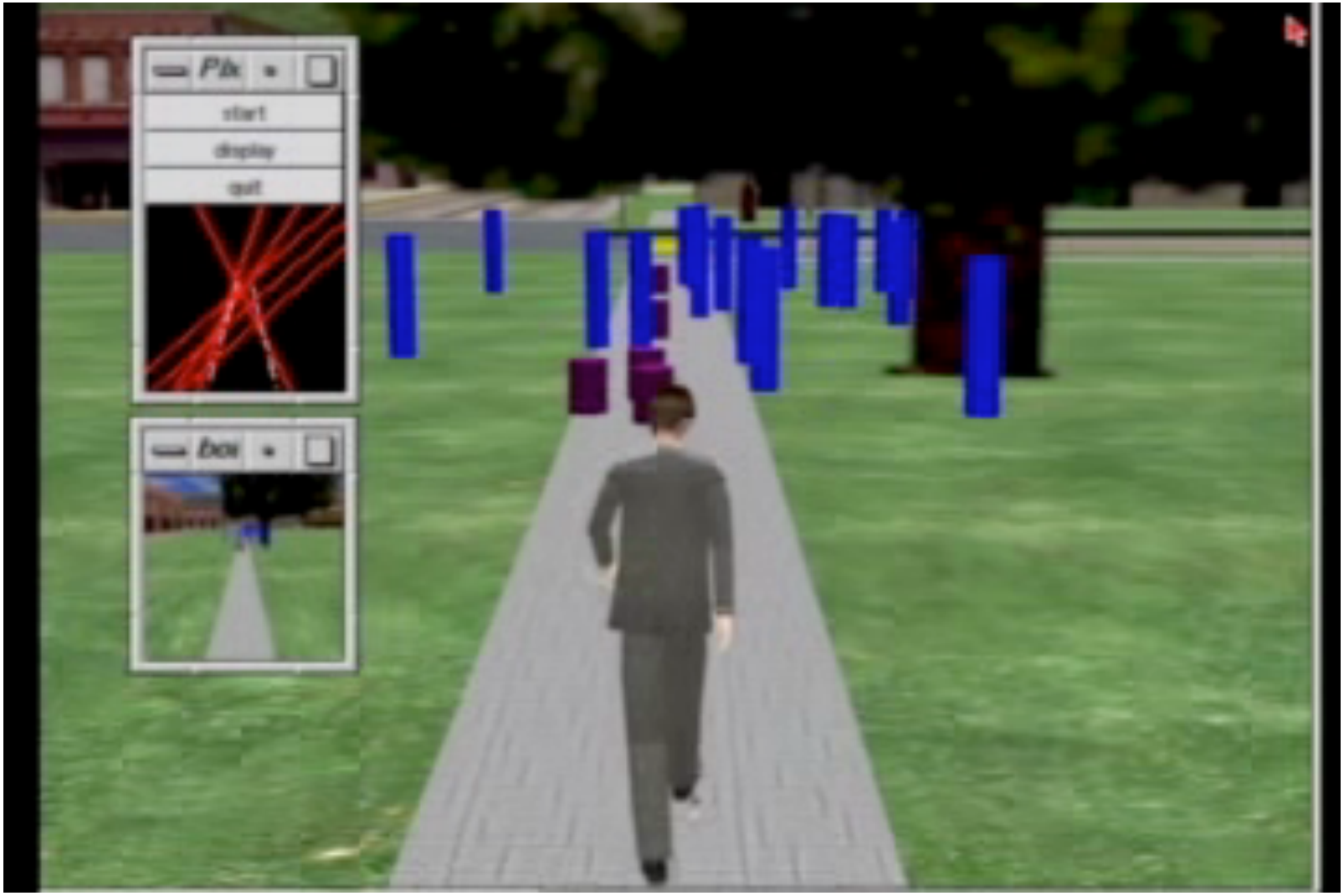
$$Q^{*}(s, a) = \sum_{r} rP(r|s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \max_{a'} Q^{*}(s', a') \tag{2}$$

Temporal difference learning

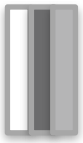
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta_Q \tag{3}$$

$$\delta_Q = r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{4}$$

Agent “Walter” walks down a sidewalk, avoiding blue obstacles and picking up purple litter



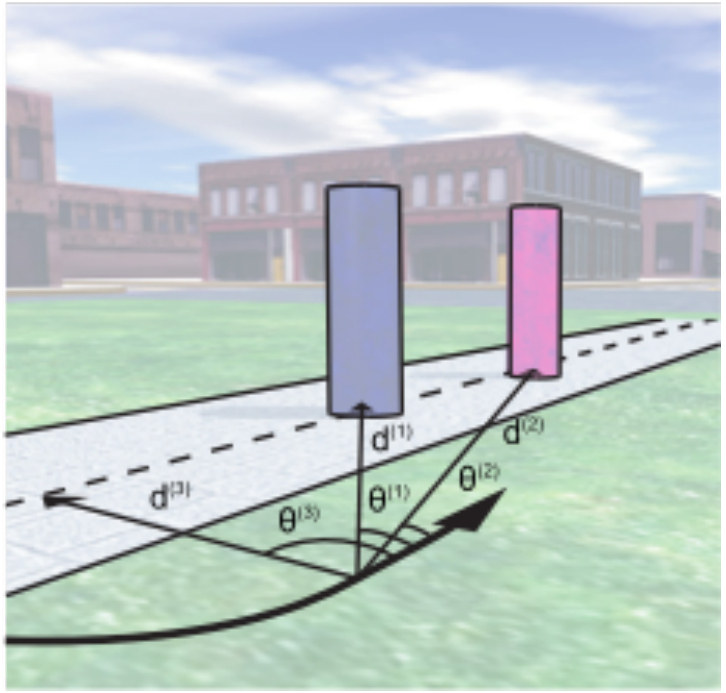
Sidewalk



Obstacle

Litter

# State Spaces definitions



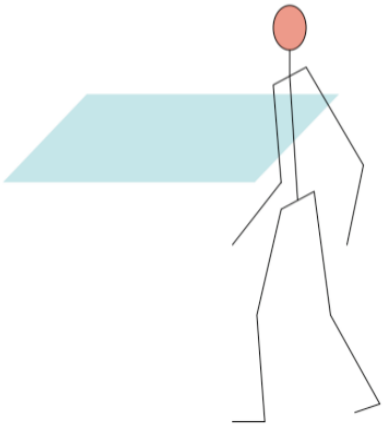
# Module for Litter Cleanup

1. Visual Routine

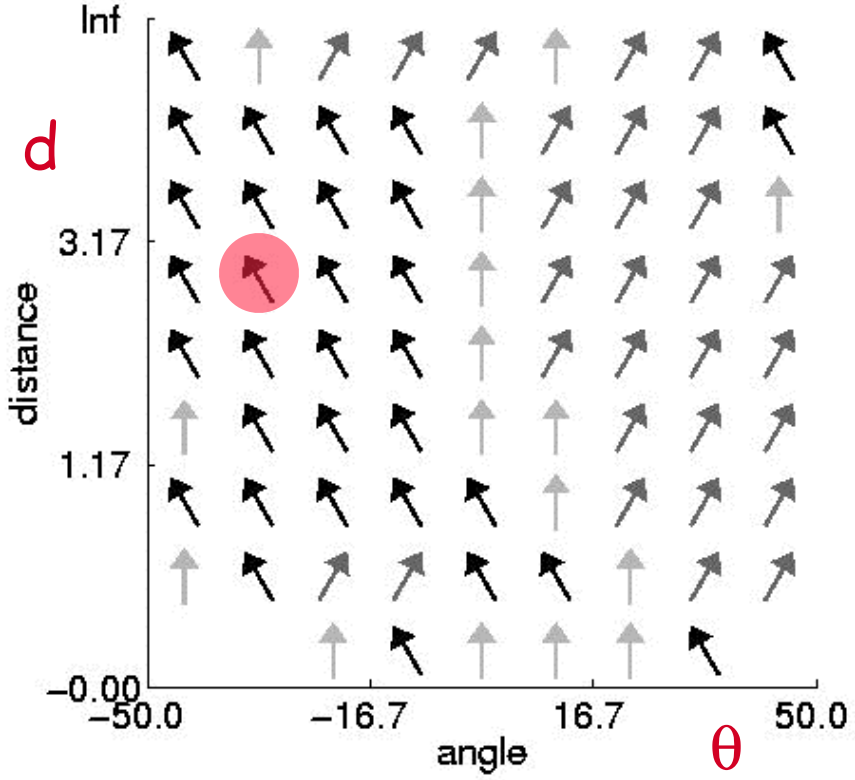
→ θ,d

↓

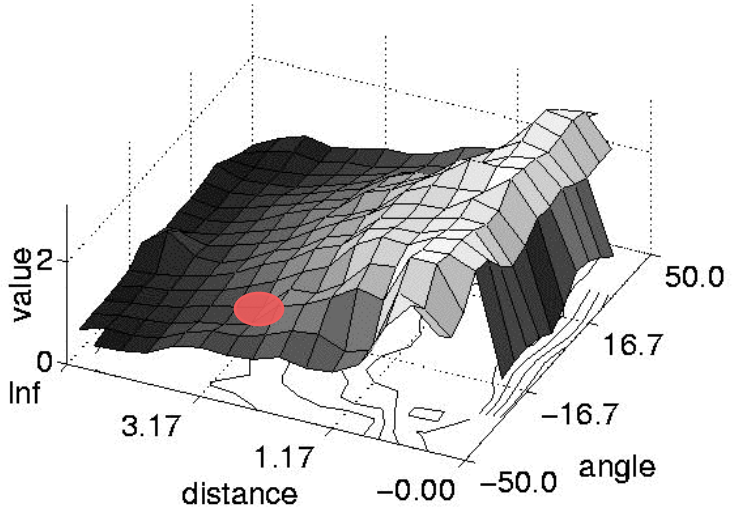
2a. Policy



2b. V is value of Policy



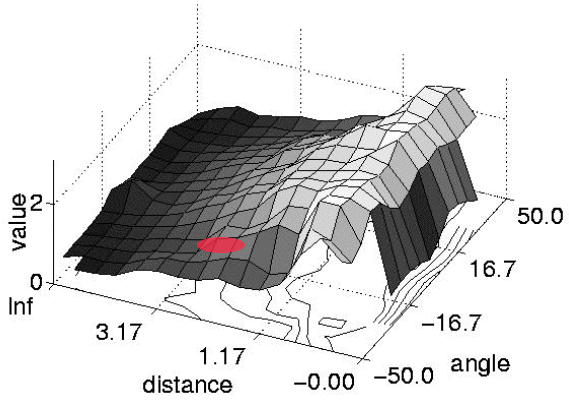

Heading from Walter's perspective

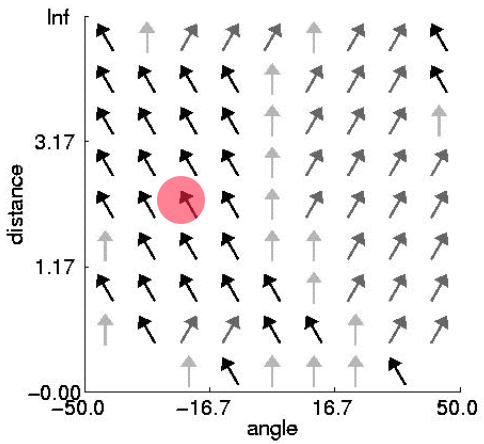



$V(s) = \max_a Q(s,a)$

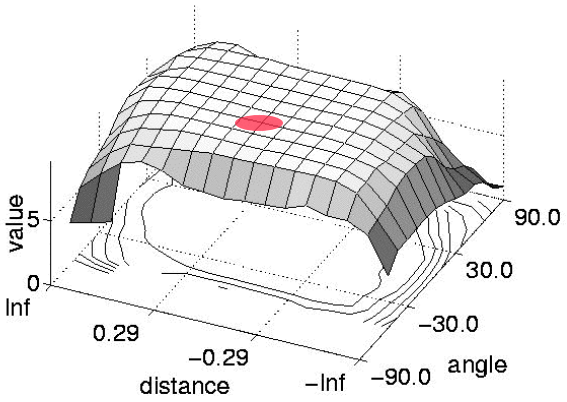
# Learned Module Behaviors

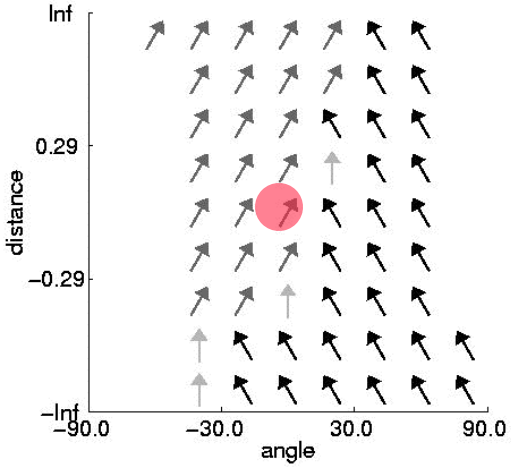
## Litter



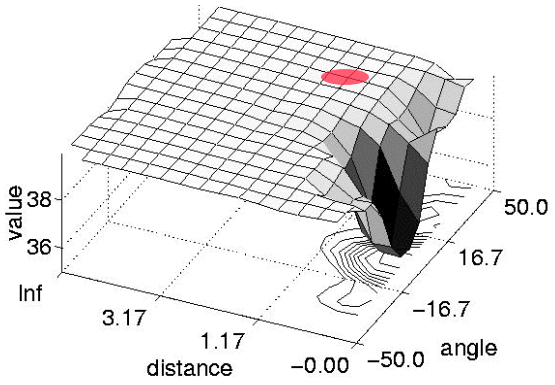



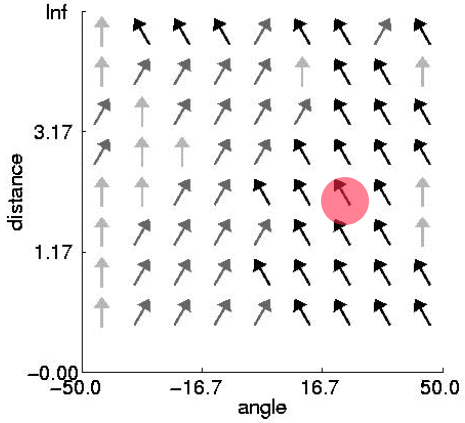

## Sidewalk



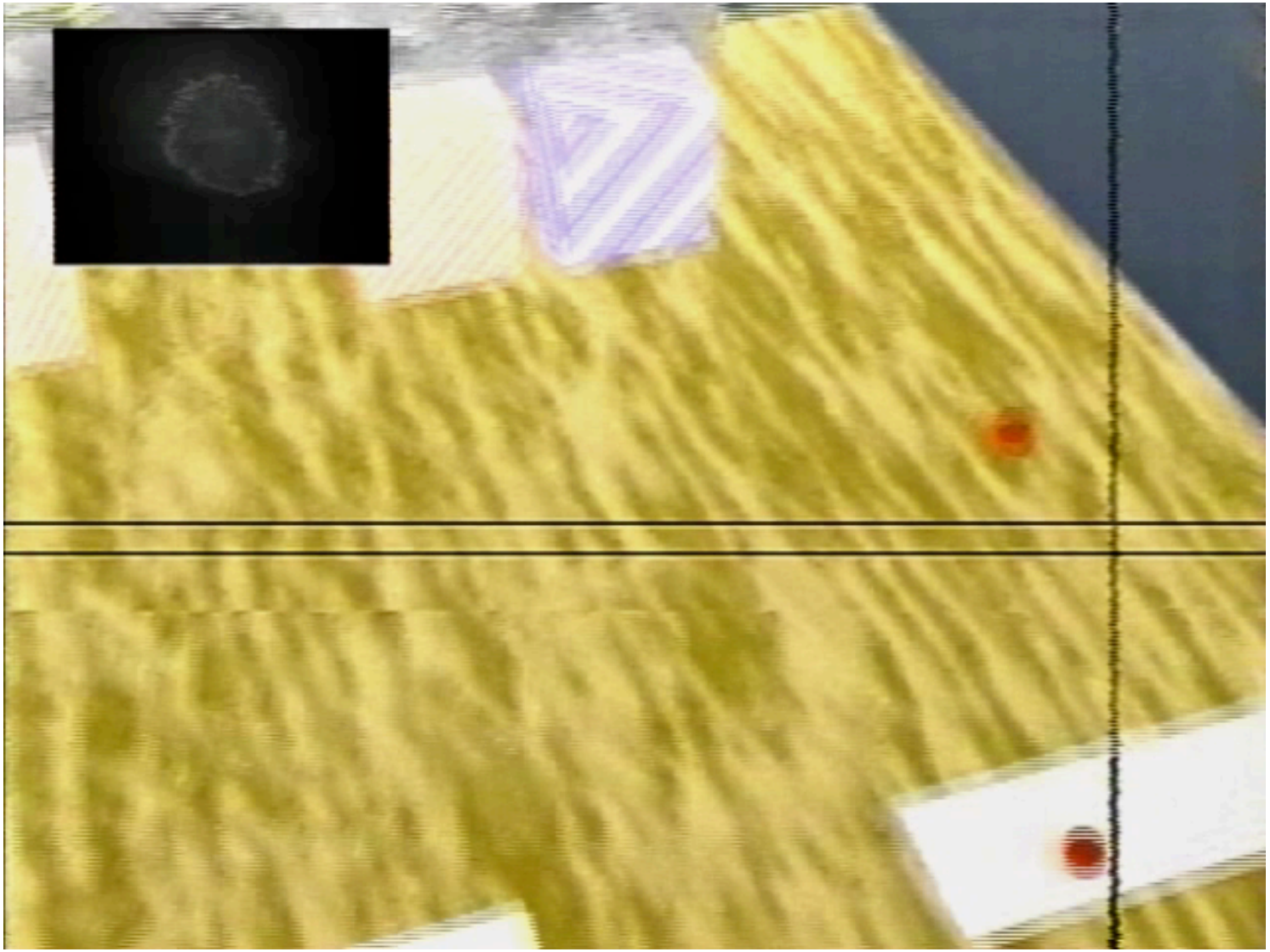




## Obstacles



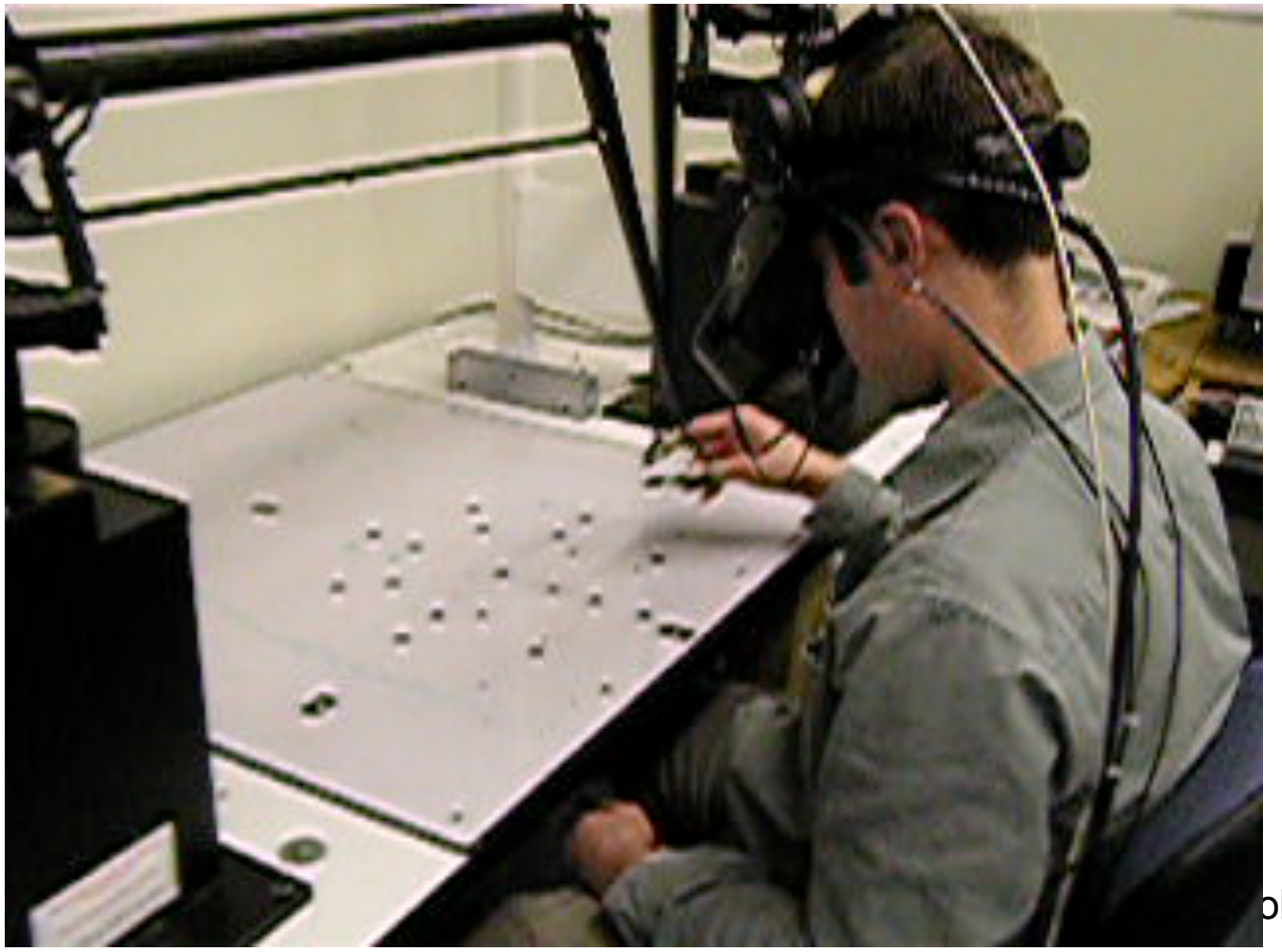



# Every fixation has a specific purpose



Hayhoe Lab data Droll et al JEP 2003

# Every fixation has a specific purpose



oll et al JEP 2003

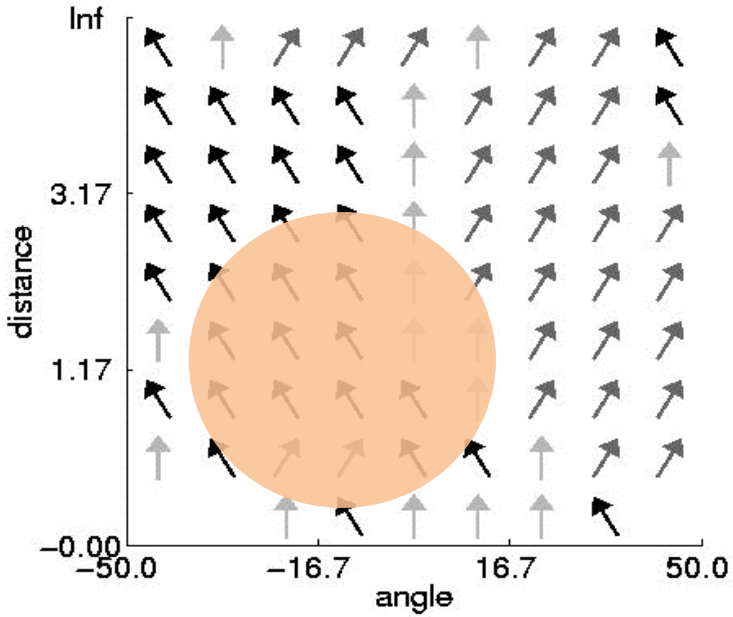
# The eye's **Fixations** can't be handled with direct reinforcement

The difference in cost between two different fixation sequences is too small to be usefully significant

Have to find an alternative constraint

# Which module should get the gaze vector?

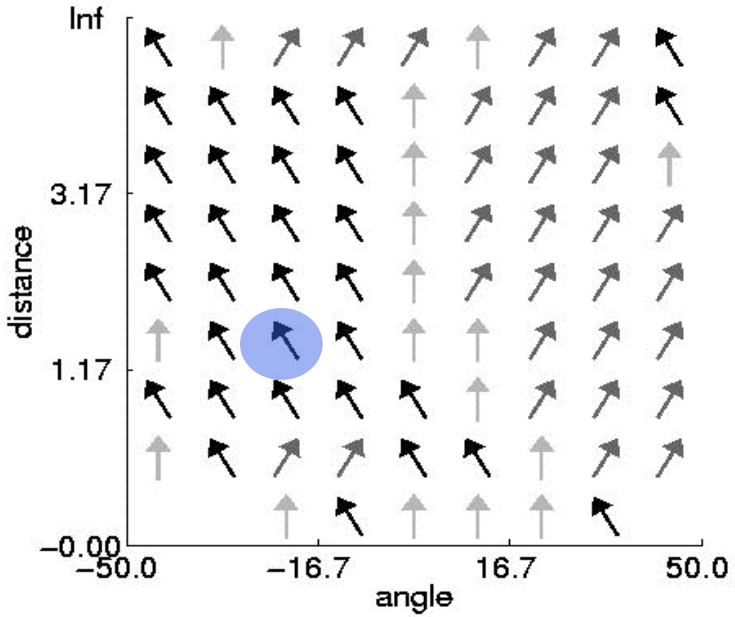
## Before Observation




Without gaze, the location in state space becomes increasingly uncertain.

# Which module should get the gaze vector?

## After Observation



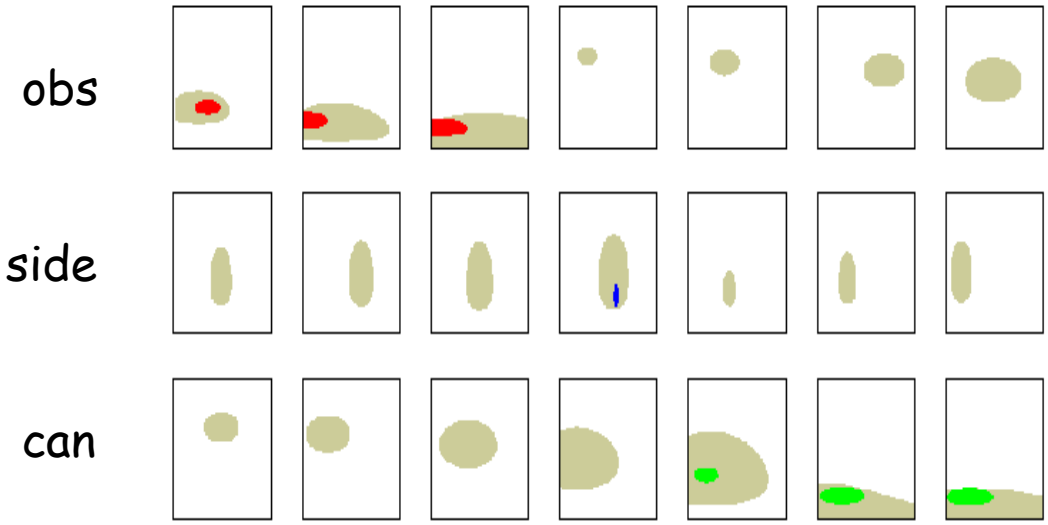

Making a measurement with a fixation reduces this uncertainty

# Sprague's hypothesis

$$gain_b = E\left[\max_a \left(Q_b(s_b, a) + \sum_{i \in B, i \neq b} Q_i^E(s_i, a)\right)\right] - \sum_i Q_i^E(s_i, a_E)$$

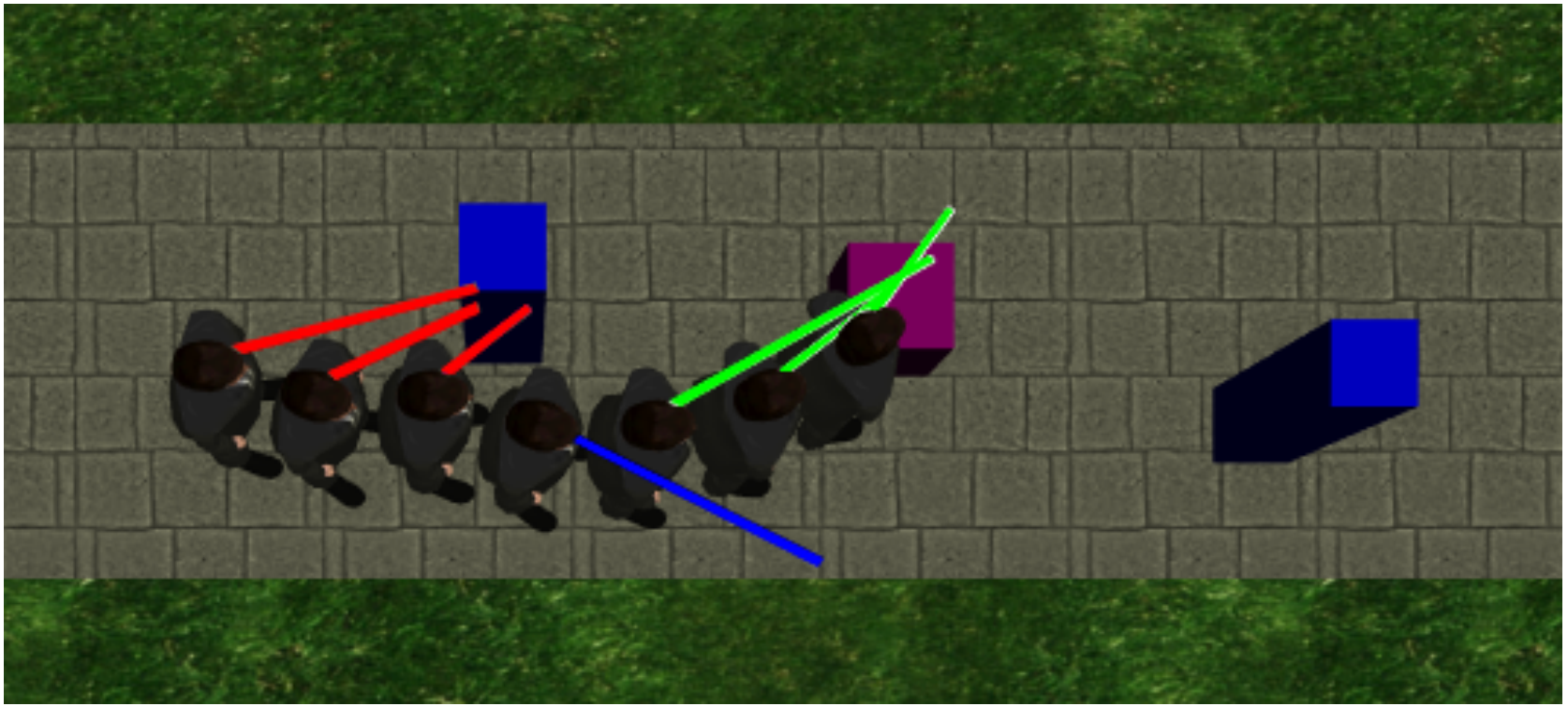
Reward where *b* updates using gaze

Average reward no updates

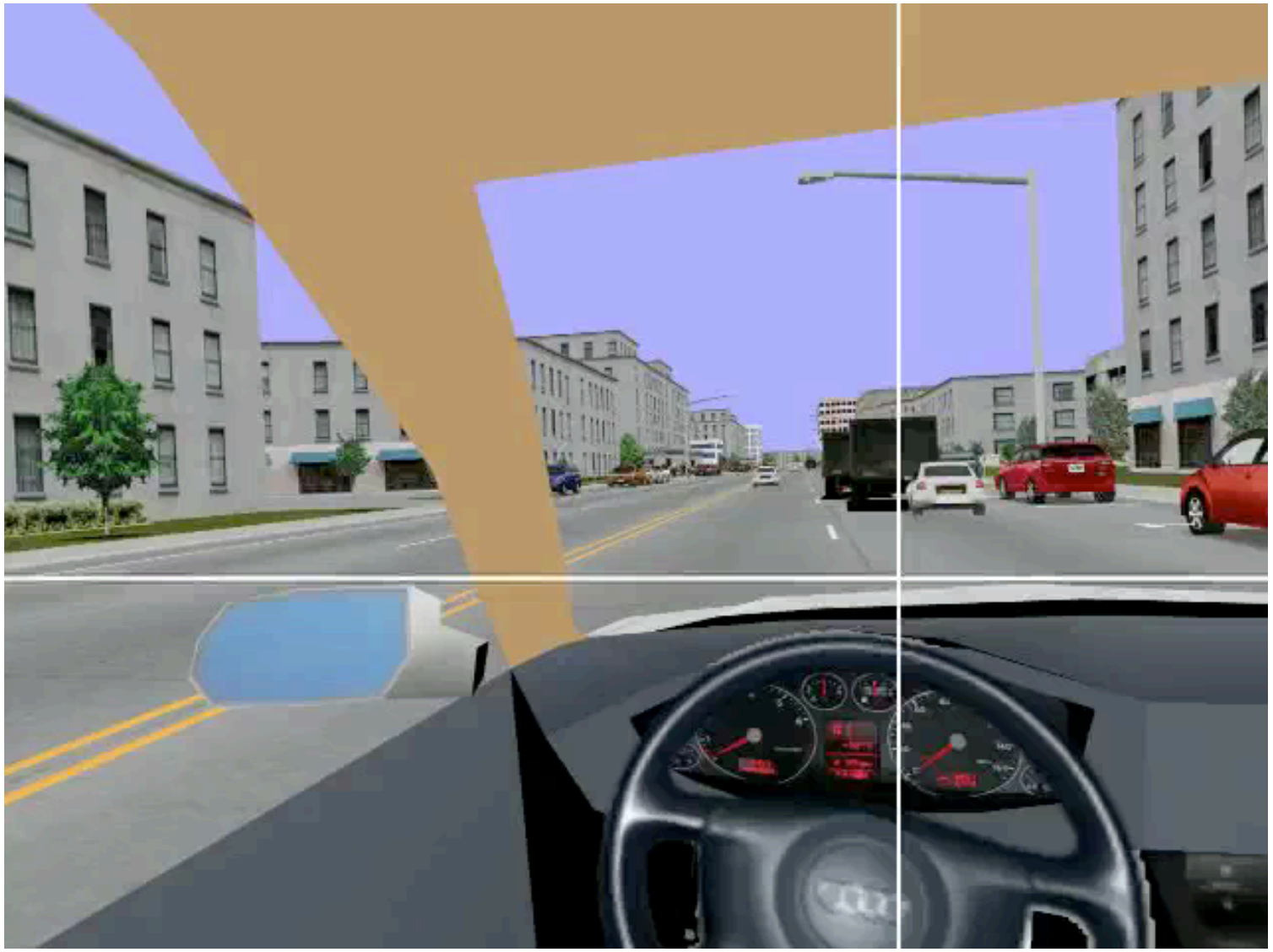



A priori uncertainty

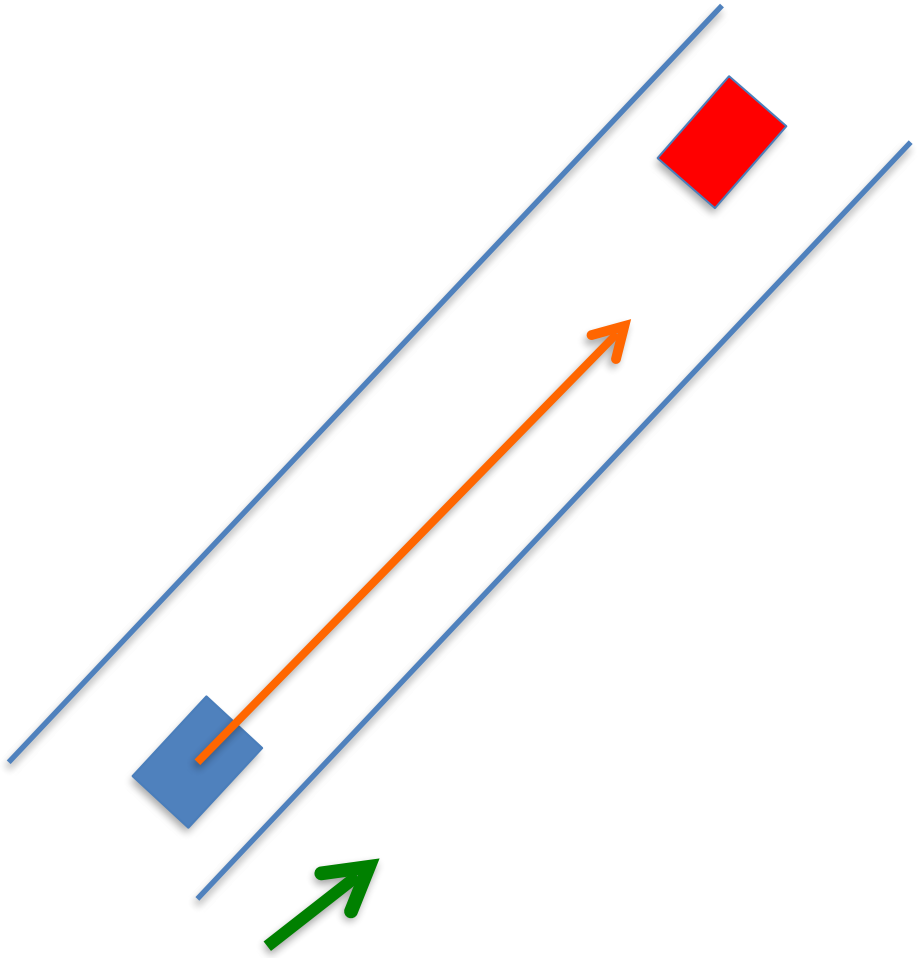
A posteriori uncertainty



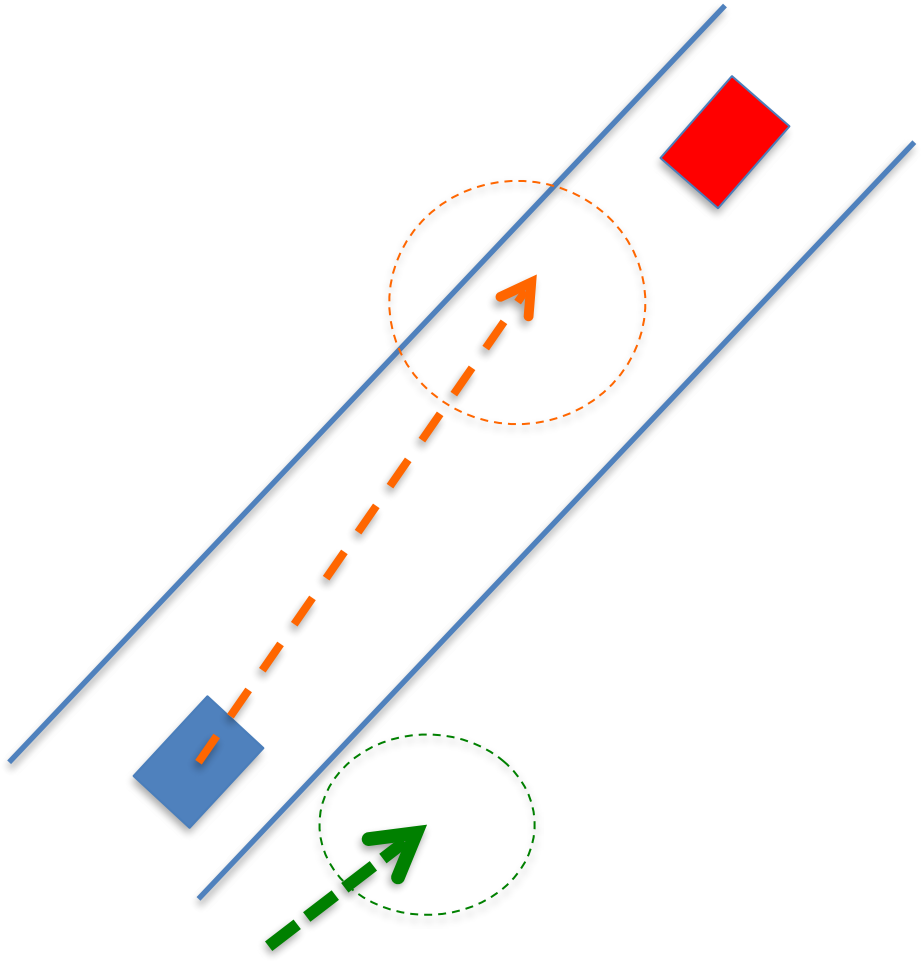
# Predicting Human Visuomotor Behavior in a Driving Task



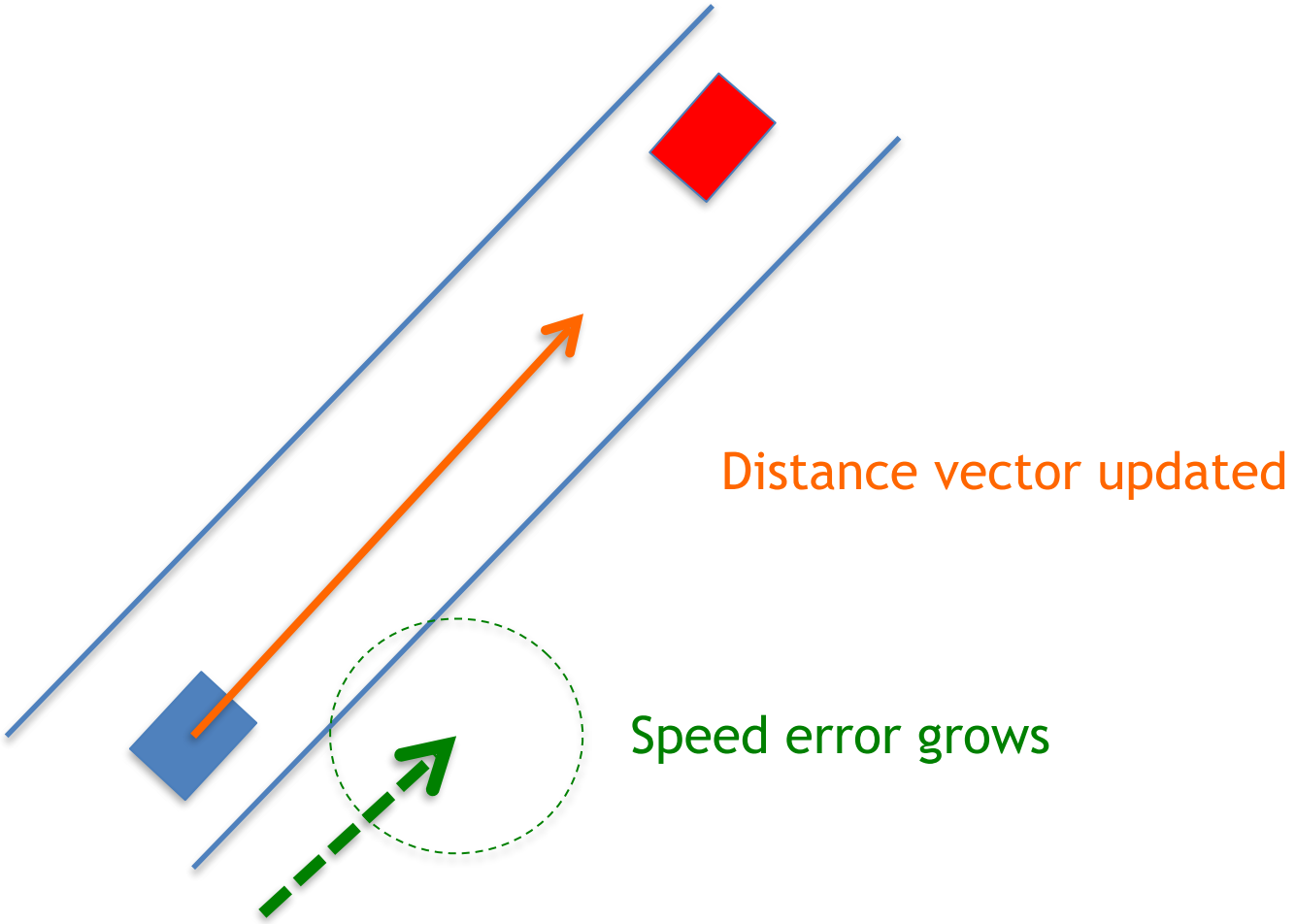
Tasks: Follow a lead car and maintain a speed



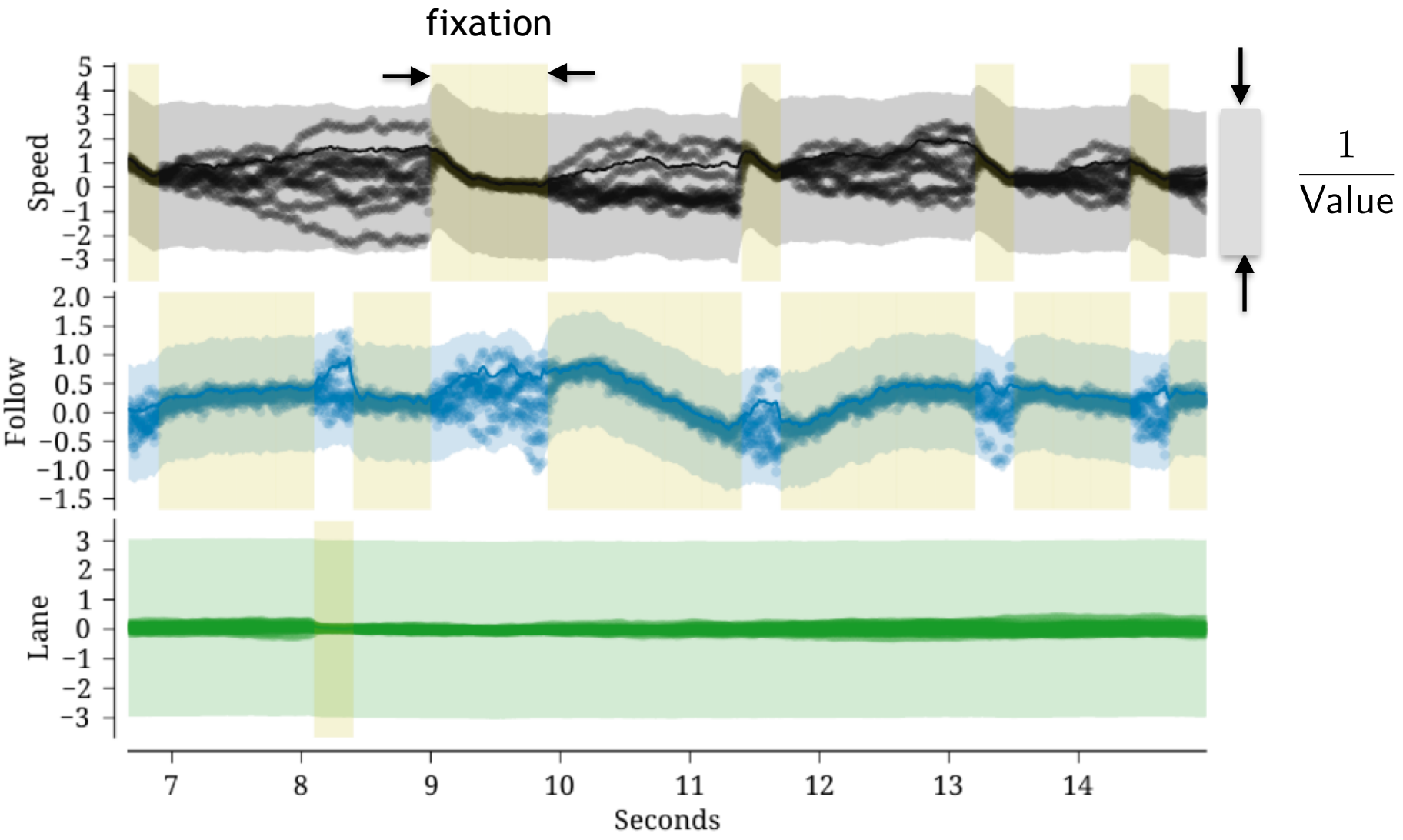
# PD Control: In absence of gaze, set point estimates drift



# Barrier models control which estimate is updated

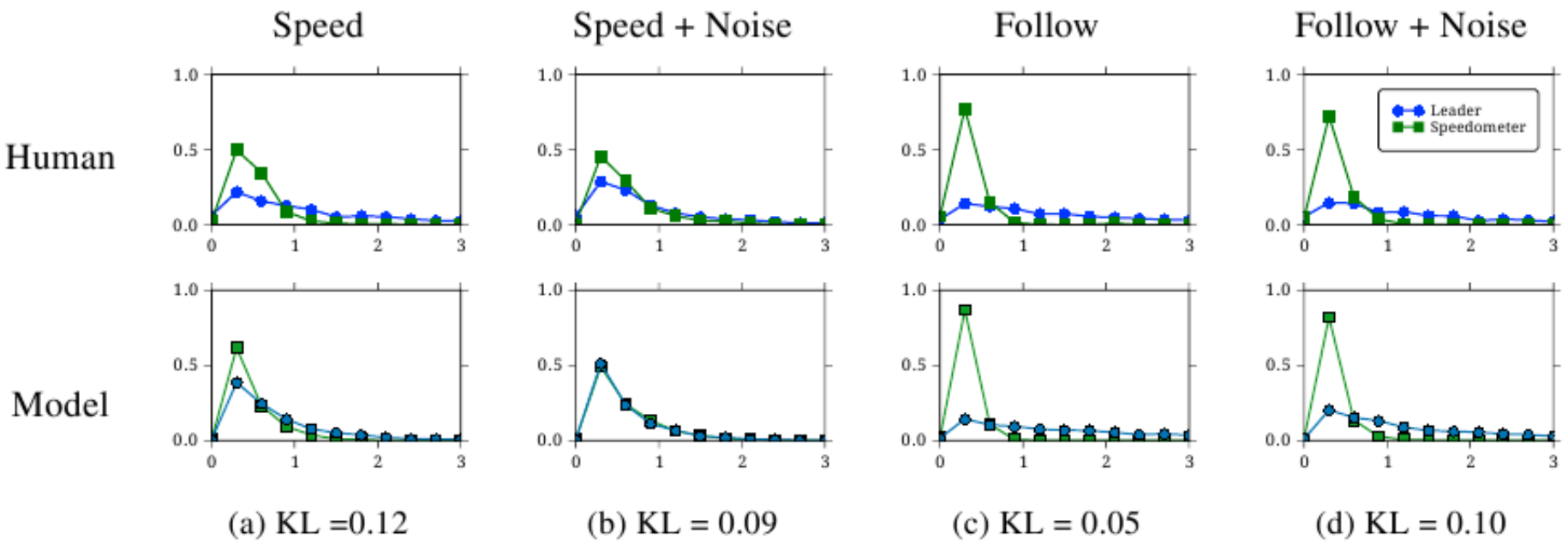


# Multi-particle boundary decision model




Johnson et al *Proc. Roy Soc B* 2014

# KL divergence shows a very close match of fixation interval distributions between human data and boundary drift model




(a) KL =0.12 (b) KL = 0.09 (c) KL = 0.05 (d) KL = 0.10

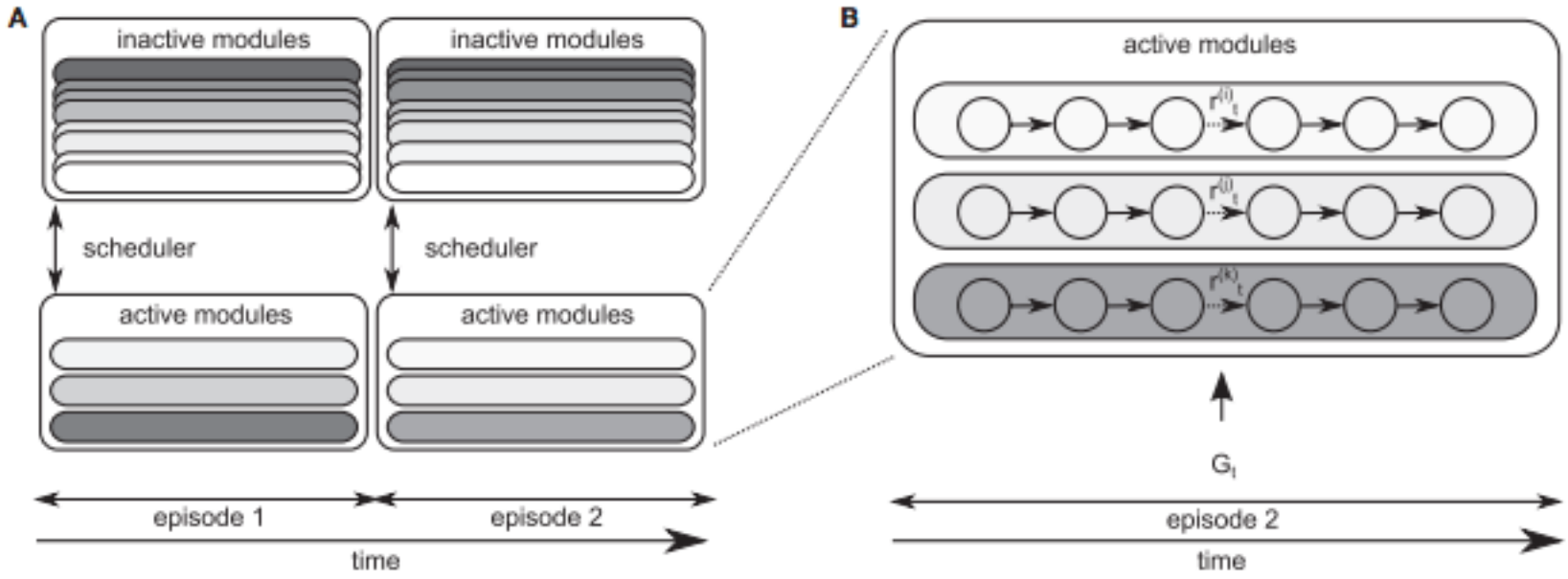
# Credit Assignment: how do modules learn their share of reward?

Situations where you need to calibrate reward

Solving a problem with a new set of modules

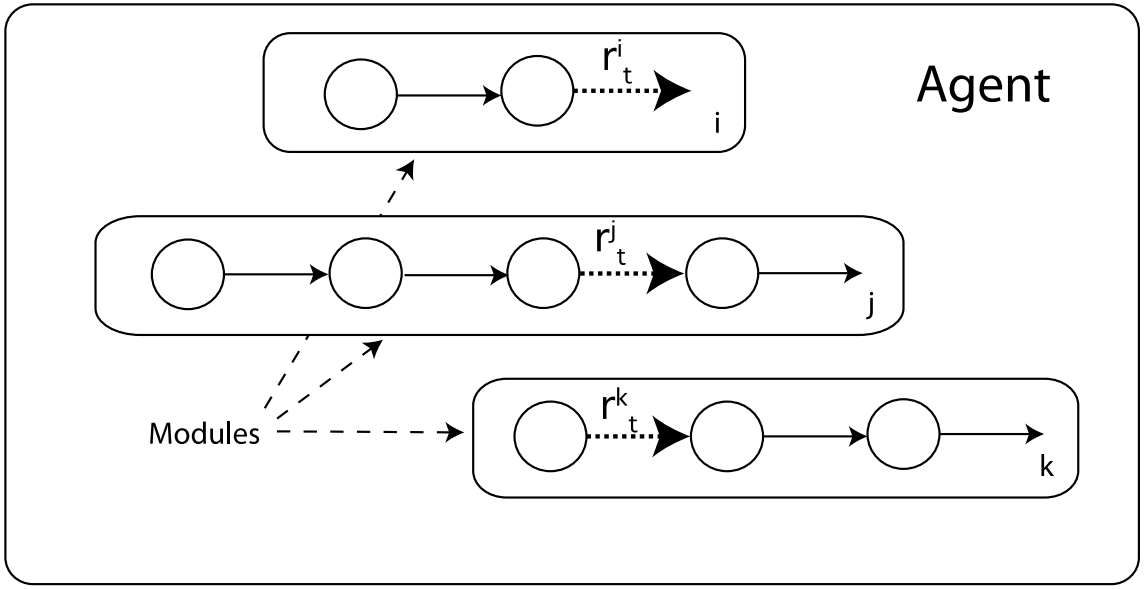
Only the global reward is known
and the problem is to assign credit appropriately

# Modules architecture: Episodes



# Module rewards can be found by bootstrapping

Key assumption: a module knows the estimated rewards of its co-activated module set




$G_t$

Module's estimate

Estimate of the other modules

$$\hat{r}^{(i)} \leftarrow (1-\beta)\hat{r}^{(i)} + \beta\left(G - \sum_{j \in \mathcal{M}, j \neq i} \hat{r}^{(j)}\right)$$

# Variance weighted estimates are superior

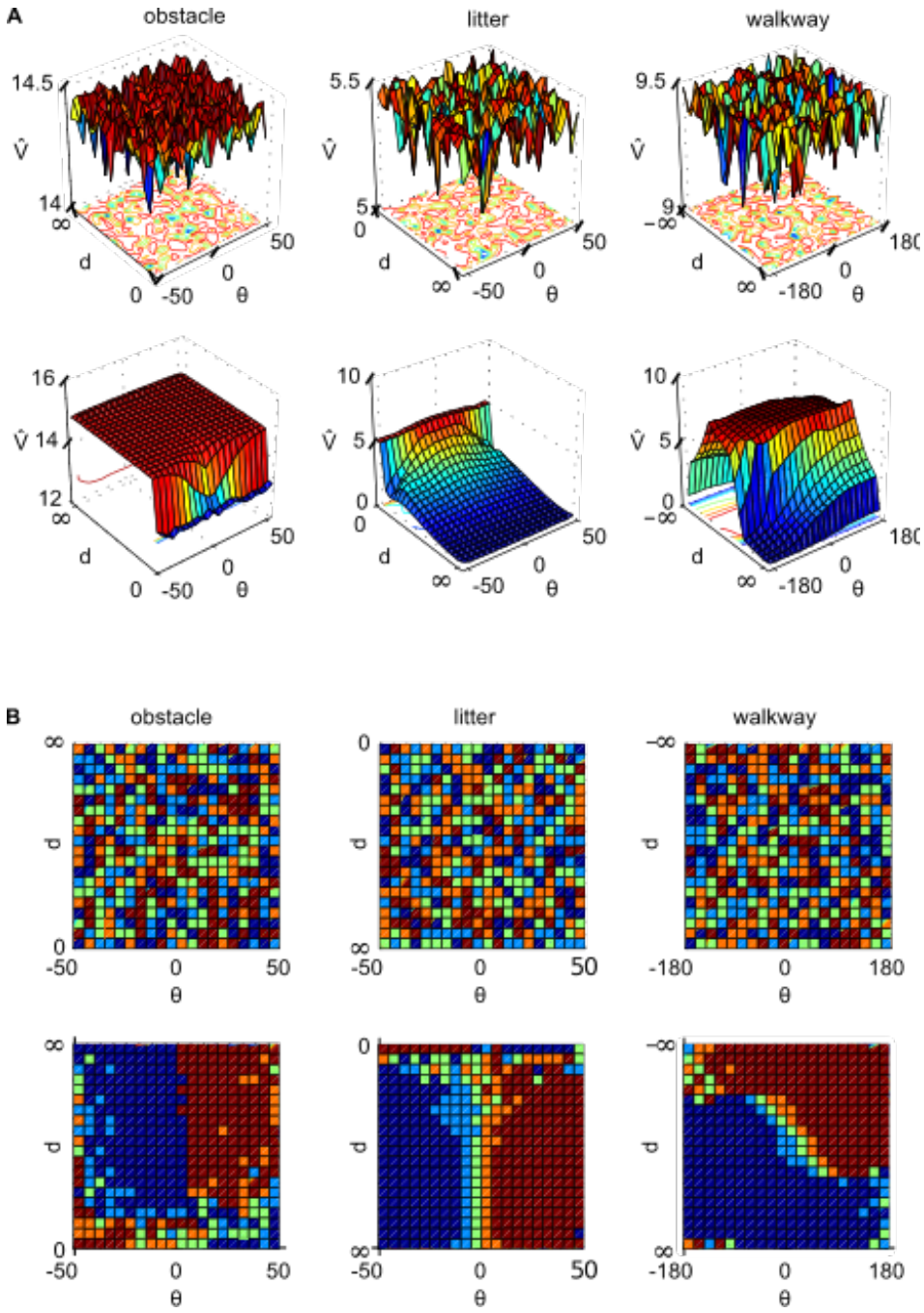
Problem: A module that has a high variance estimate can corrupt others' estimates

Solution: weight estimates by the variances in reward

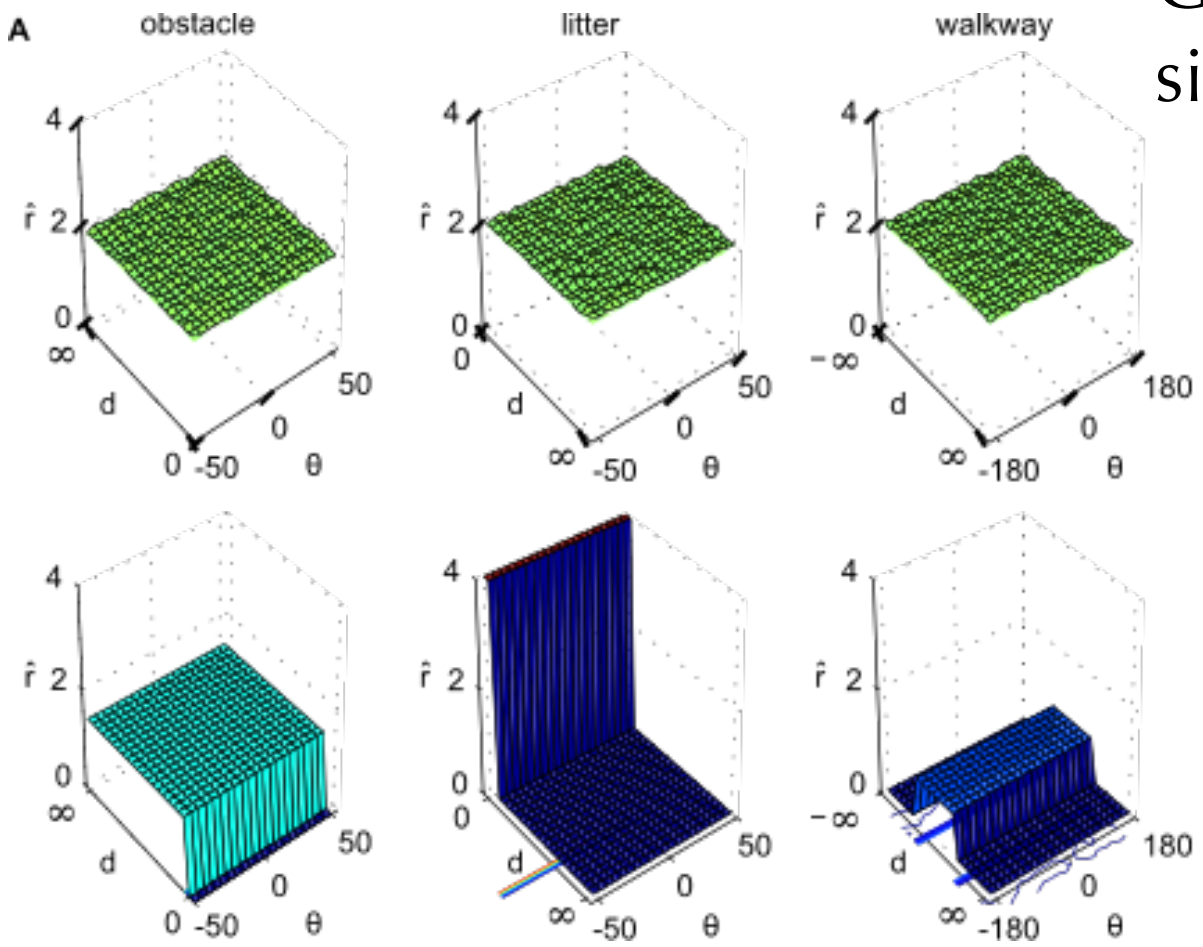
$$\beta_i = \frac{(\sigma^{(i)})^2}{\sum_{j=1}^{N} (\sigma^{(j)})^2}$$

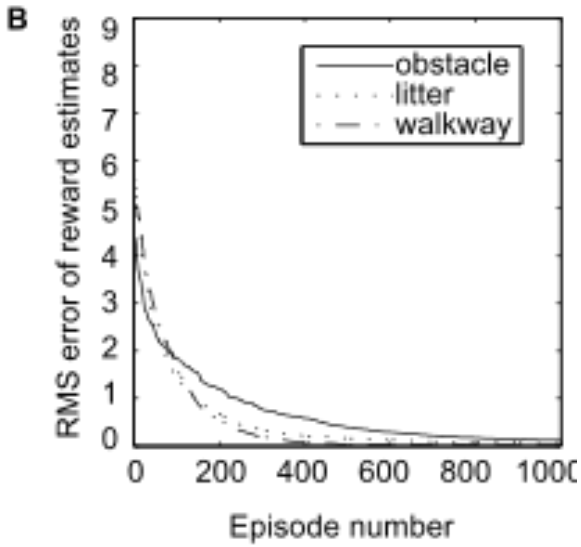
$$= \frac{(\sigma^{(i)})^2}{\sum_{j \neq i}^{N} (\sigma^{(j)})^2 + (\sigma^{(i)})^2}$$

Learning  Q tables
And policies given only
Global reward in sidewalk venue



Learning rewards given only
Global reward for sidewalk venue



# Inverse RL: given a set of {state, action} data how do recover the module weights?

This is important for imitation learning

An observer has the needed set but doesn't know how to weight them.

# The main ideas

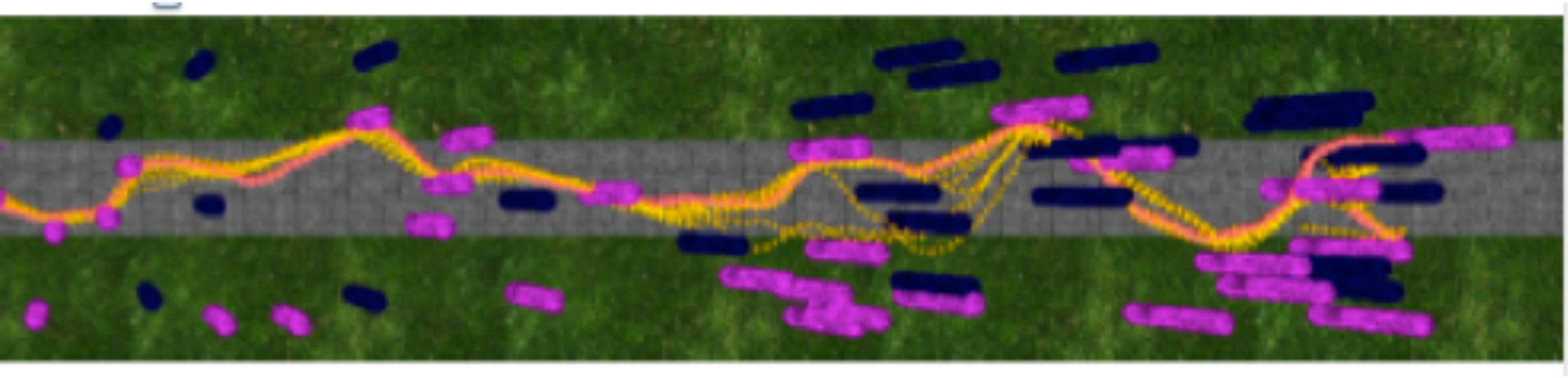
Modular rewards are scaled

Search the space of scale factors to make observed data most probable

$$P(s_j, a_j | Q^*) = \frac{1}{Z} e^{\sum_i c_i Q(s_j^{*(i)}, a_j)}$$

# Inverse Reinforcement learning in the sidewalk navigation example

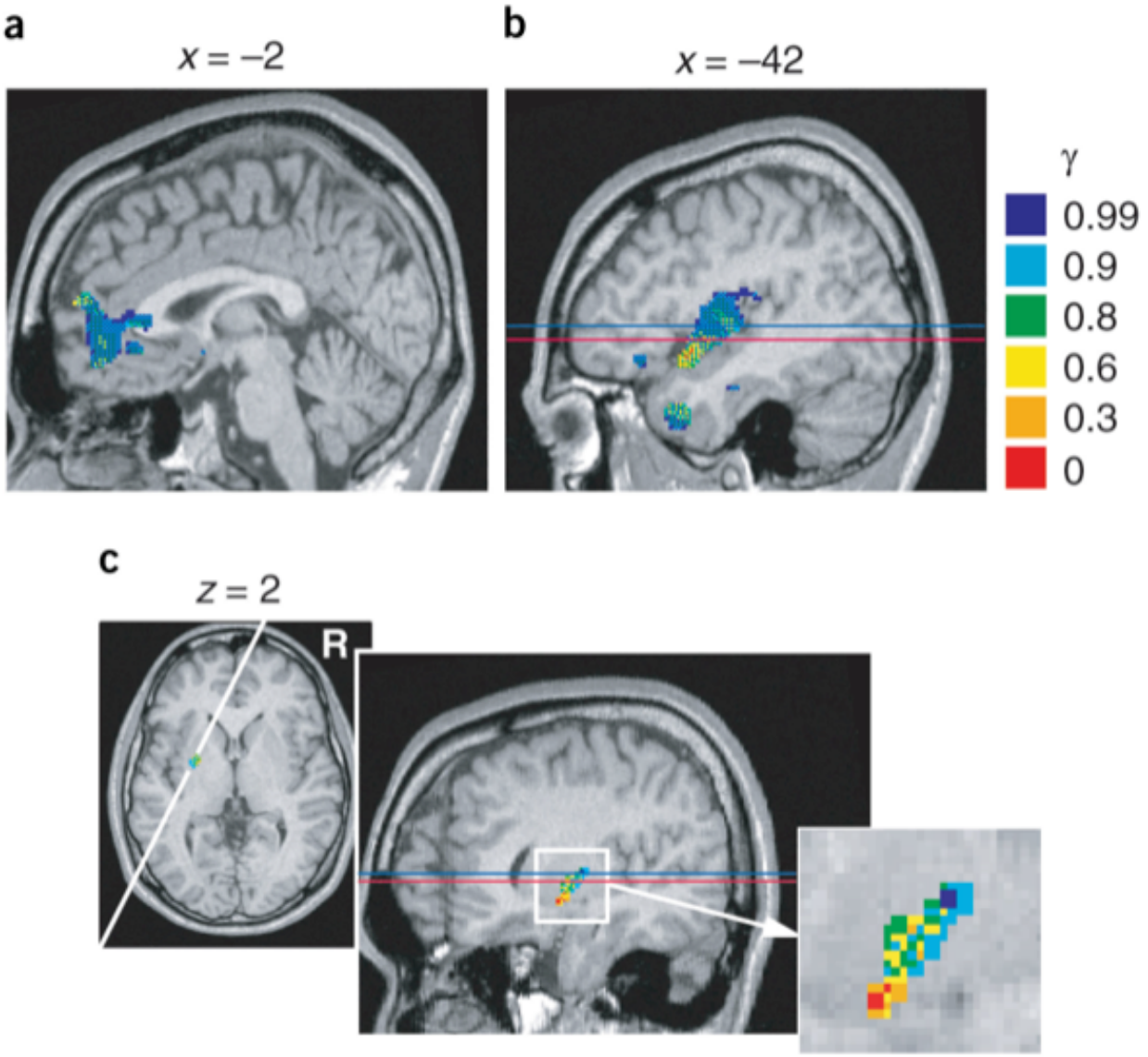
Choose the module weighting that makes the observed data the most probable

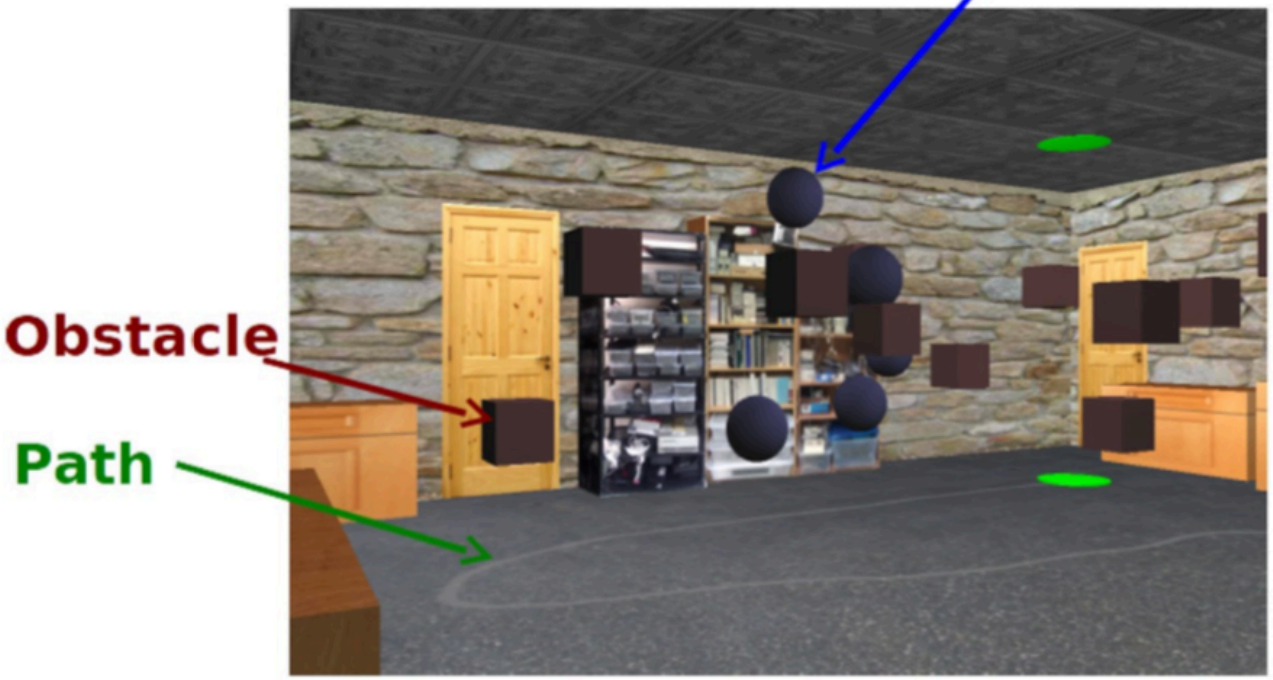


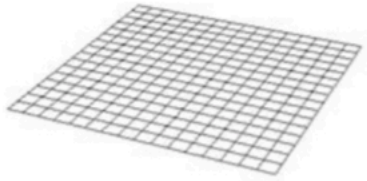
- Original trajectory (data)
- Generated trajectories using recovered weight estimates

**If the optimization problem is slightly reformatted, the model's discount factors can also be estimated**

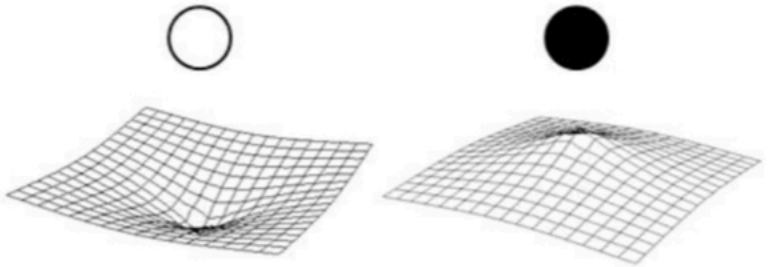
Kenji Doya's fMRI experiments show regions of the cortex that seem to respond differentially depending on the discount factor



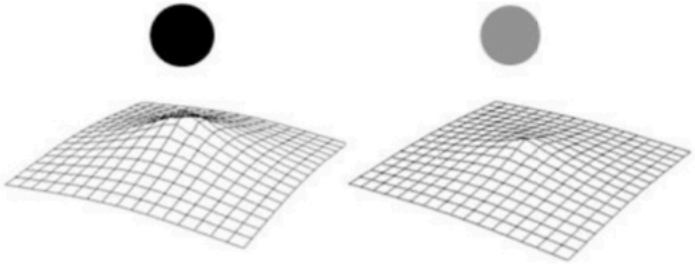

Target
Obstacle
Path



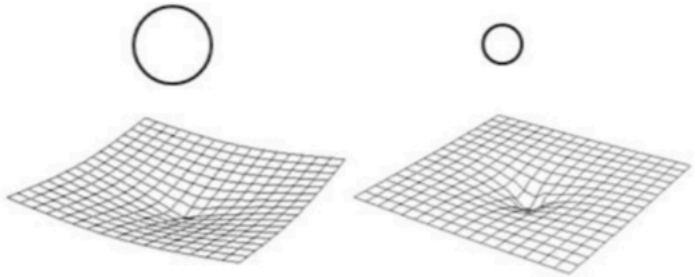
**(A)** Initial value surface



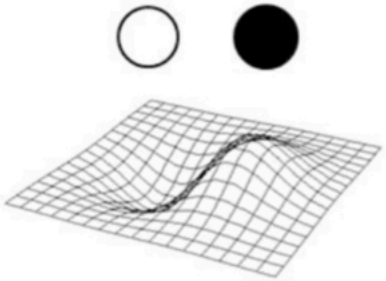
**(B)** Postive reward vs. negative reward



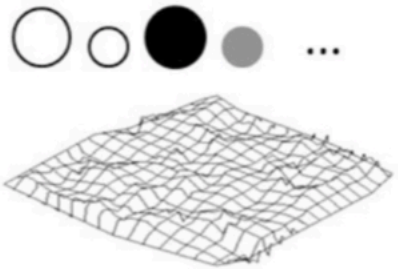
**(C)** Large reward vs. small reward



**(D)** Large discount factor $\gamma$ vs. small $\gamma$



**(E)** Composed surface with two objects



**(F)** Composed surface with many objects

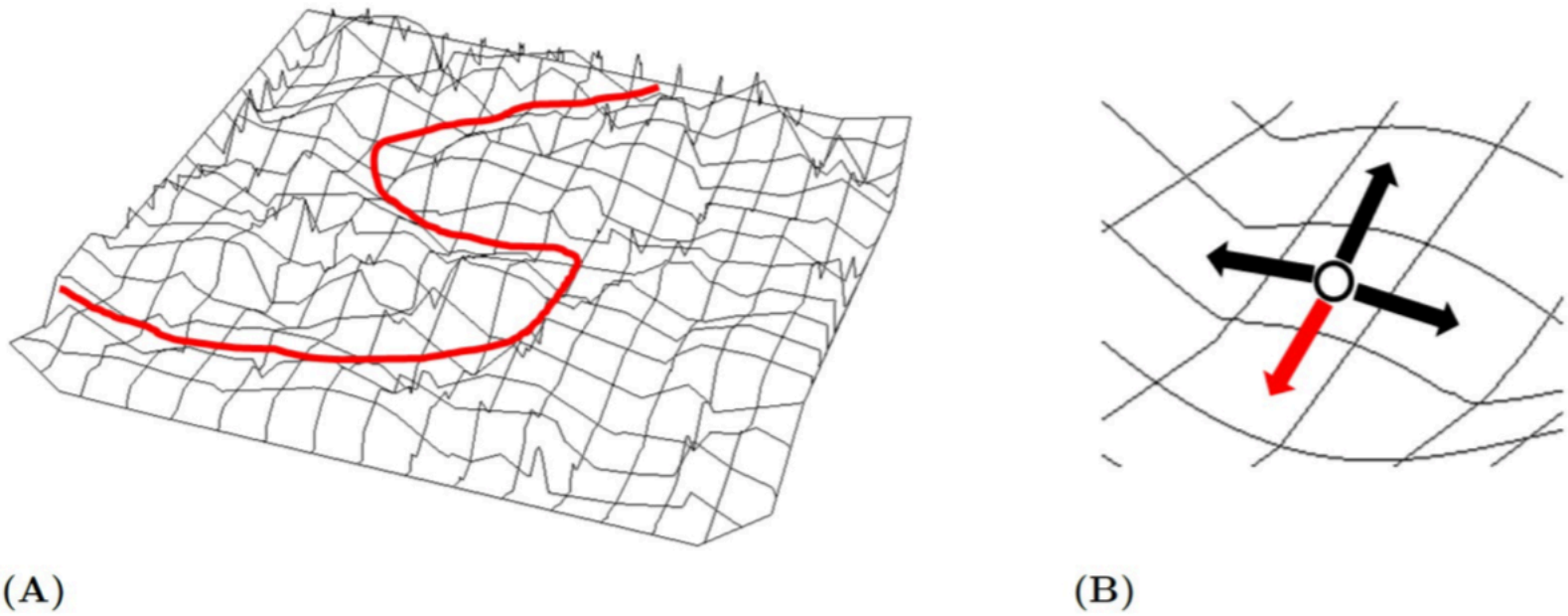
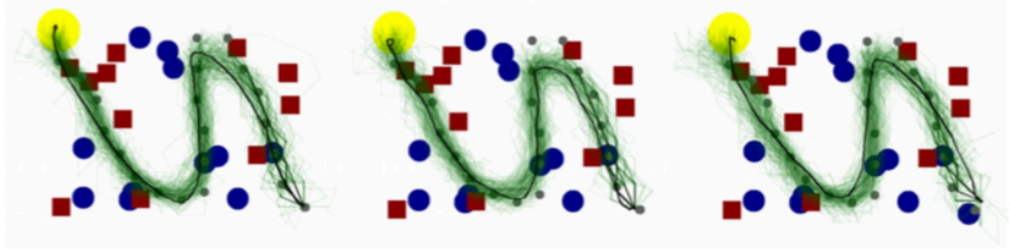
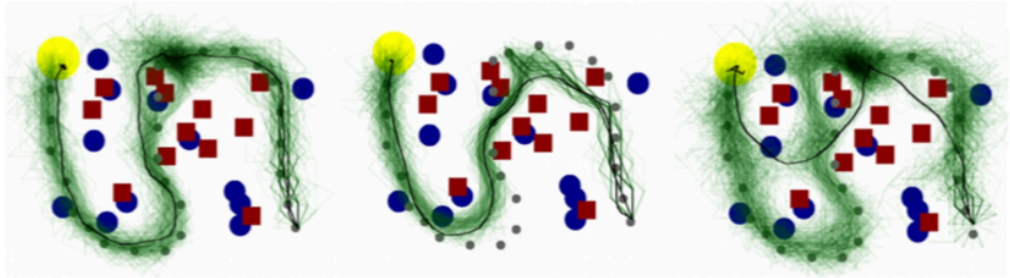



**Fig 3. Maximum likelihood modular inverse reinforcement learning.** (A) From an observed trajectory (a sequence of state-action pairs), the goal of modular IRL is to recover the underlying value surface. (B) Maximum likelihood IRL assumes that the probability of observing a particular action (red) in a state is proportional to its Q-value among all possible actions as in Eq (5).



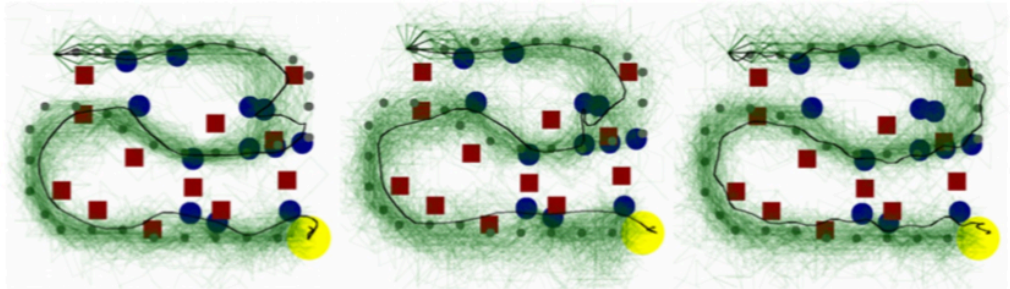
(A) $r : (0.11, 0.38, \mathbf{0.52})$ $\gamma : (0.85, 0.99, 0.88)$ (B) $r : (0.11, 0.25, \mathbf{0.63})$ $\gamma : (0.88, 0.99, 0.94)$ (C) $r : (0.07, 0.39, \mathbf{0.54})$ $\gamma : (0.82, 0.99, 0.94)$

Path



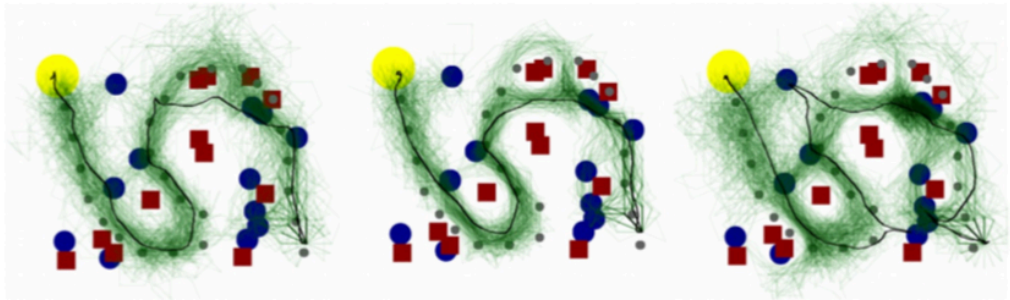
(D) $r : (0.00, \mathbf{0.57}, \mathbf{0.43})$ $\gamma : (0.00, 0.69, 0.91)$ (E) $r : (0.02, \mathbf{0.57}, \mathbf{0.41})$ $\gamma : (0.99, 0.59, 0.97)$ (F) $r : (0.06, \mathbf{0.75}, \mathbf{0.19})$ $\gamma : (0.95, 0.60, 0.88)$

Path & Avoid



(G) $r : (\mathbf{0.30}, 0.22, \mathbf{0.48})$ $\gamma : (0.77, 0.72, 0.89)$ (H) $r : (\mathbf{0.27}, 0.29, \mathbf{0.45})$ $\gamma : (0.69, 0.73, 0.96)$ (I) $r : (\mathbf{0.30}, 0.17, \mathbf{0.52})$ $\gamma : (0.76, 0.99, 0.89)$

Path & Targets



All

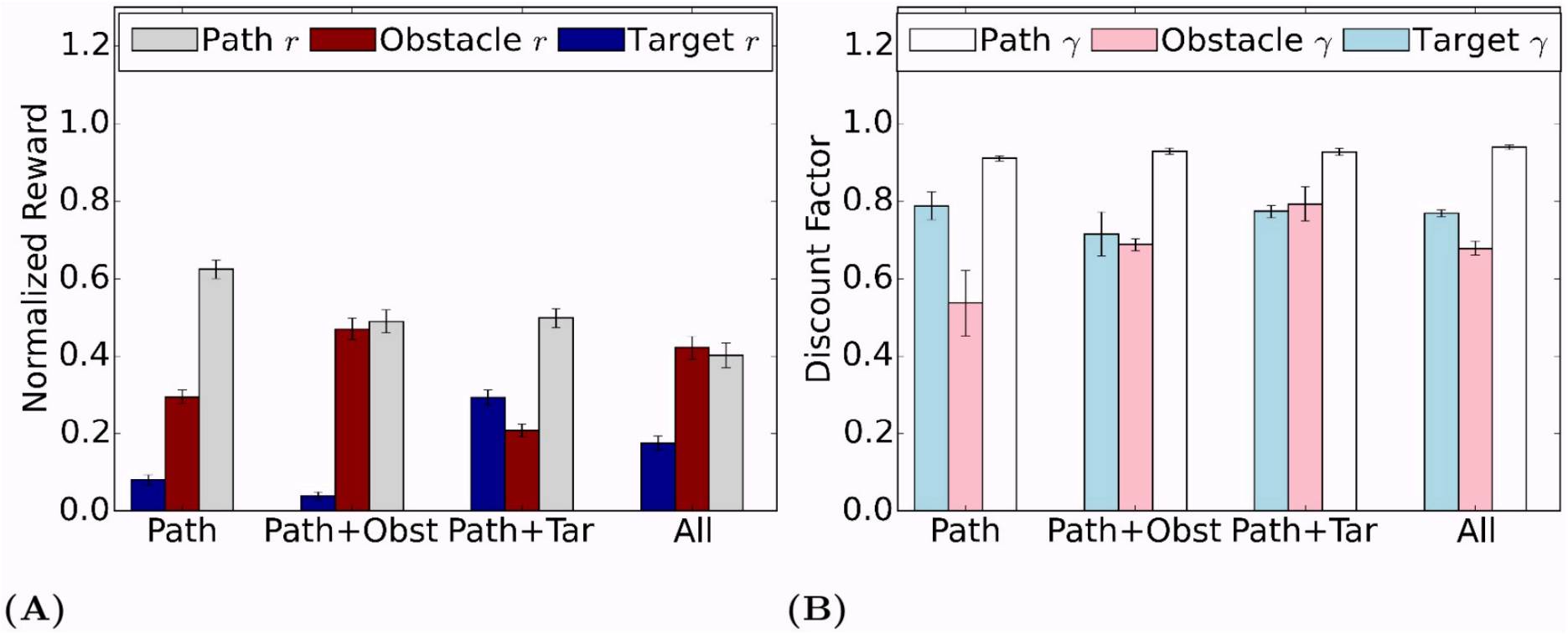



**Fig 5.** (A) Normalized average rewards across different task instructions. The error bar represents the standard error of the mean between subjects ($N = 25$). The obstacle module has negative reward, but to compare with the other two modules its absolute value is taken. The estimated reward agree with task instructions. (B) Average discount factors across different task instructions. The error bar represents the standard error of the mean between subjects ($N = 25$).

# Acknowledgements


Nathan Sprague
James Madison University

Constantin Rothkopf
Technische Universität Darmstadt

Luxin Zhang
Peking University

Leif Johnson
Google Corporation

Rohan Zhang
University of Texas at Austin

FUNDING
National Institutes of Health
National Science Foundation
Google

Zhang, R., Zhang, S., Tong, M.H., Ballard, D.H. & Hayhoe, M.H. (2018) **Modeling sensory-motor decisions in natural behavior**. *PLOS Computational Biology* (in press).

Leif Johnson, Brian Sullivan, Mary Hayhoe and Dana Ballard,(2014)Predicting human visuomotor behaviour in a driving task,*Phil. Trans. R. Soc.*,369,20130044.

Ballard, D. H., Kit, D., Rothkopf, C. A. and Sullivan, B. (2013)A hierarchical modular architecture for embodied cognition, *Multisensory Research*,26(1-2),177-204

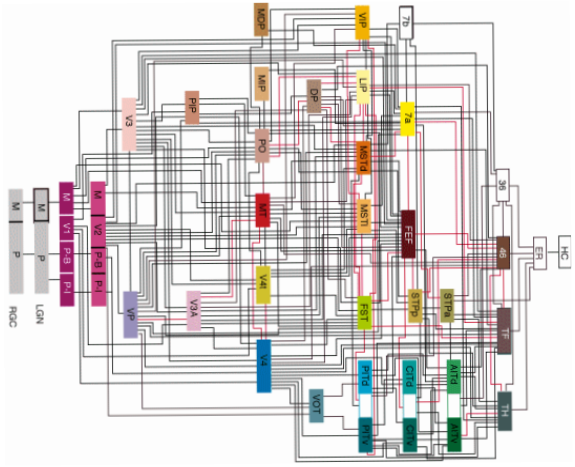
C. A. Rothkopf and Ballard, D. H.(2013)Modular inverse reinforcement learning for visuomotor behavior, *Biologica Cybernetics*, 107(4),477-490

Rothkopf, C. A., and Ballard, D. H. (2010) Credit assignment in multiple goal embodied visuomotor behavior, *Frontiers in Psychology*

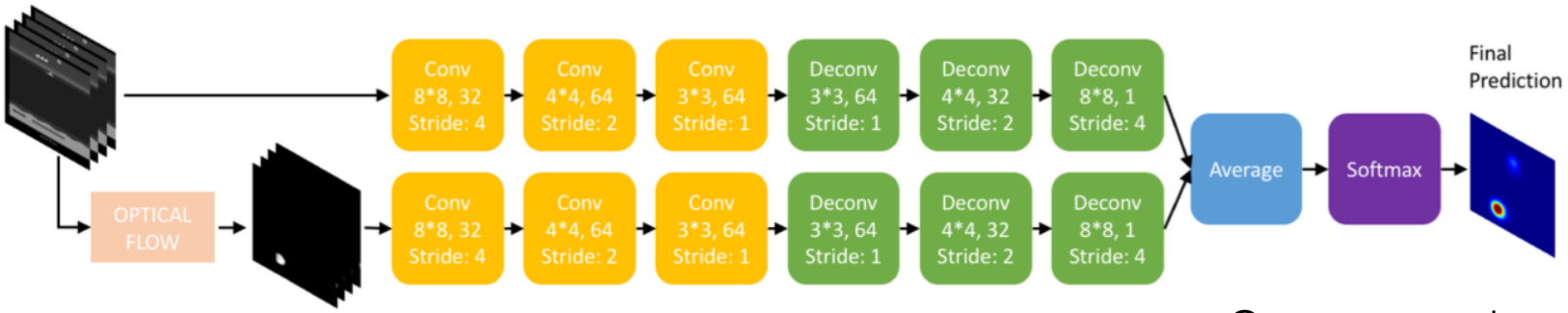
- Rothkopf, C. A., and Ballard, D. H.(2009) Image statistics at the point of gaze during human navigation,*Visual Neuroscience*,26, 81-92

Rothkopf, C. A., Ballard D. H. and Hayhoe, M. M. (2007) Task and context determine where you look, *Journal of Vision,* 7(14) 1-20

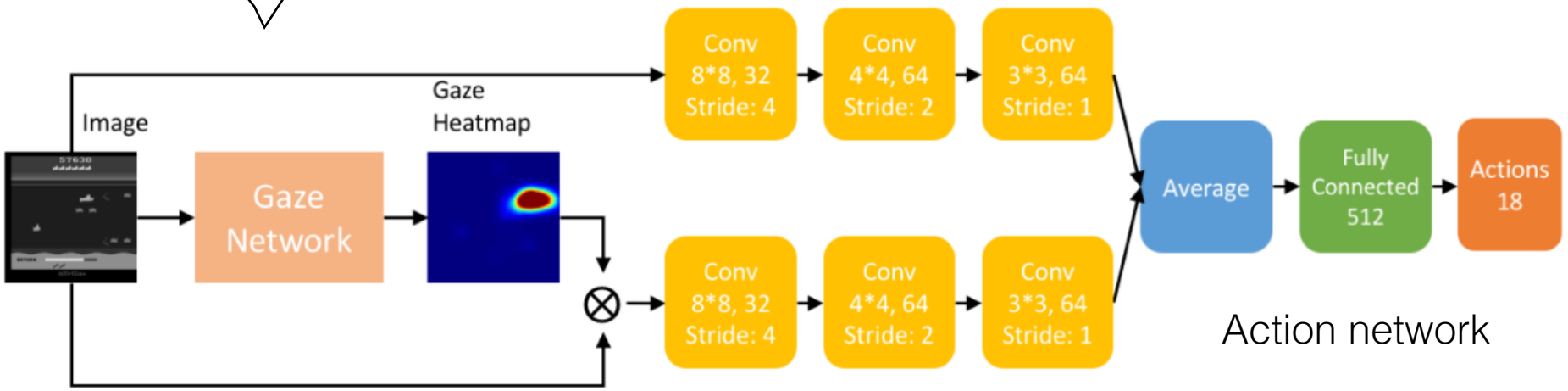
Sprague, N. and Ballard, D. H. (2005) Modeling Embodied Visual Behaviors *ACM Transactions on Applied Perception*



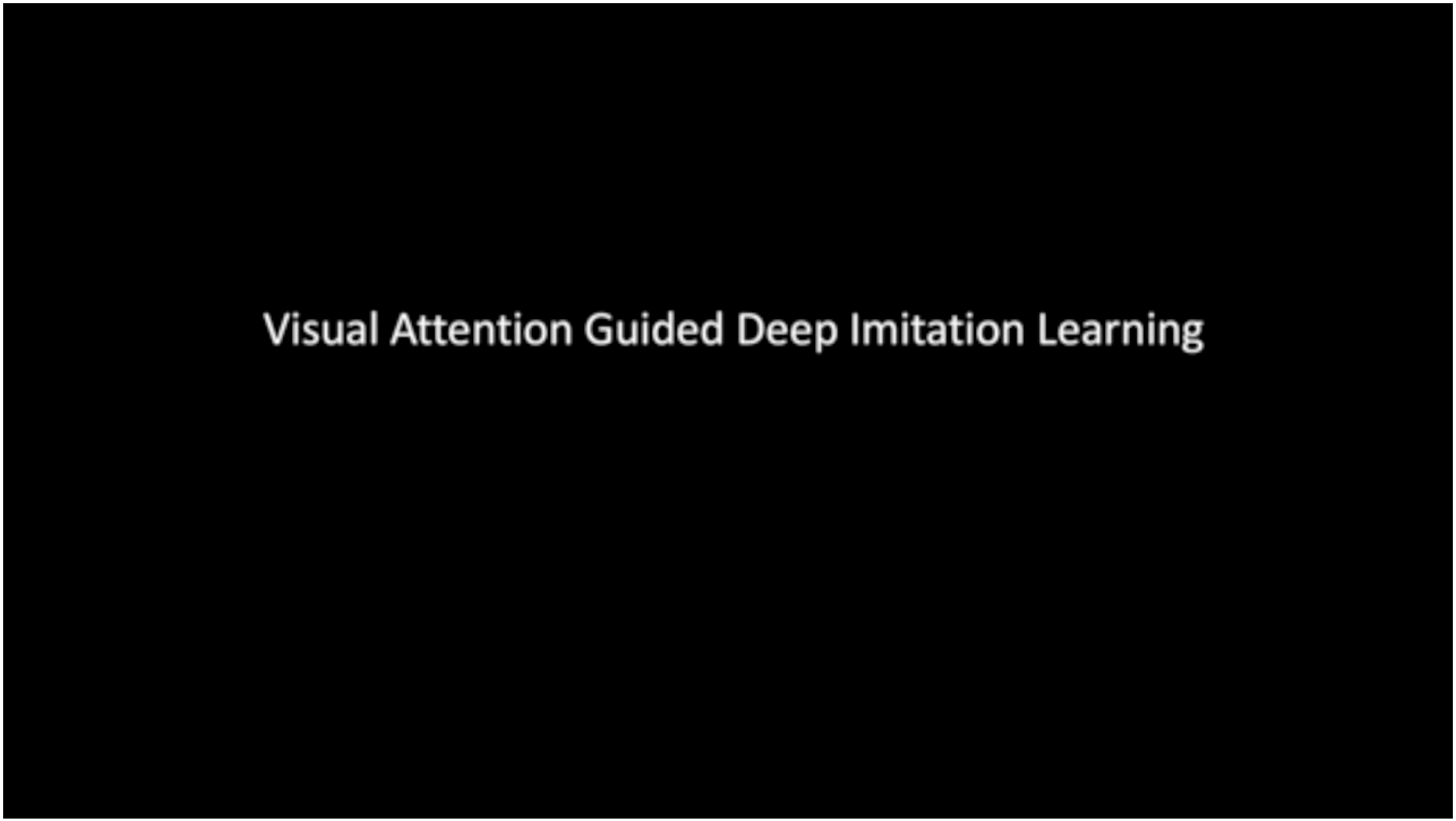
Monkey network




Gaze network




Action network


Visual Attention Guided Deep Imitation Learning

# Hybrid Reward Architecture for Reinforcement Learning


**Harm van Seijen**[1]
harm.vanseijen@microsoft.com

**Mehdi Fatemi**[1]
mehdi.fatemi@microsoft.com

**Joshua Romoff**[12]
joshua.romoff@mail.mcgill.ca

**Romain Laroche**[1]
romain.laroche@microsoft.com

**Tavian Barnes**[1]
tavian.barnes@microsoft.com

**Jeffrey Tsang**[1]
tsang.jeffrey@microsoft.com

[1]Microsoft Maluuba, Montreal, Canada
[2]McGill University, Montreal, Canada


~1,800 modules