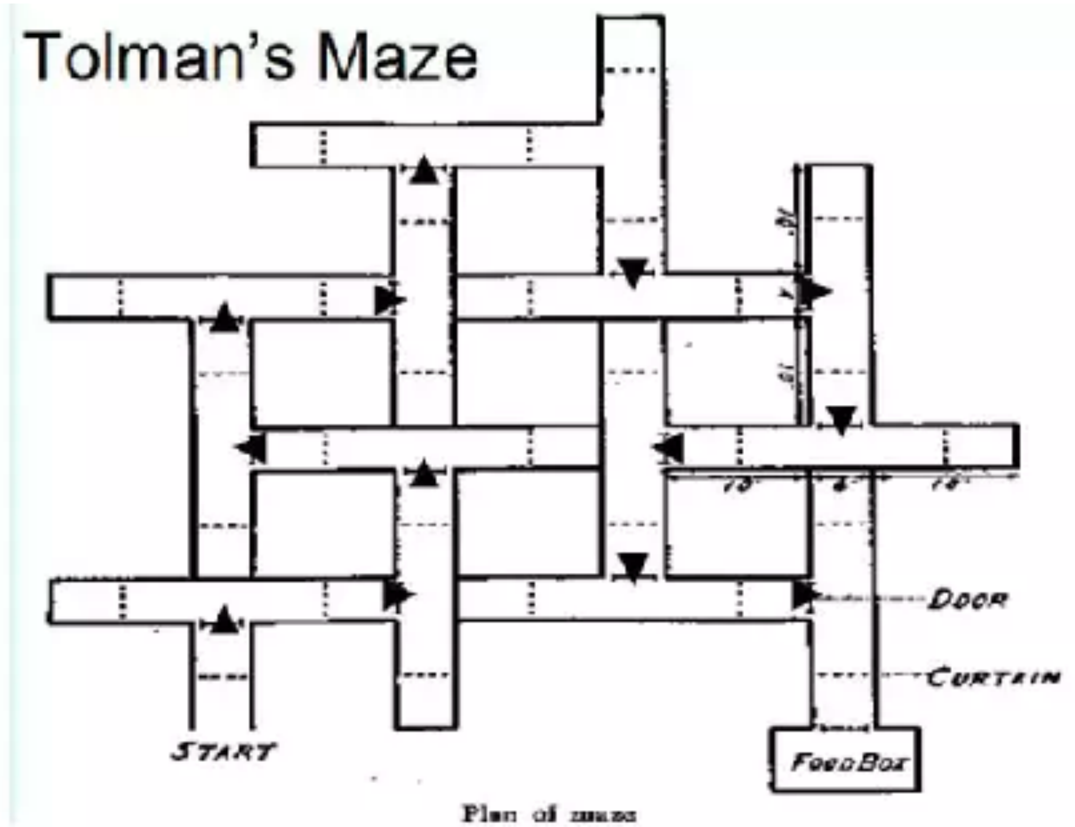
Representation Learning in the Hippocampal-Entorhinal circuit

Kim Stachenfeld
DeepMind

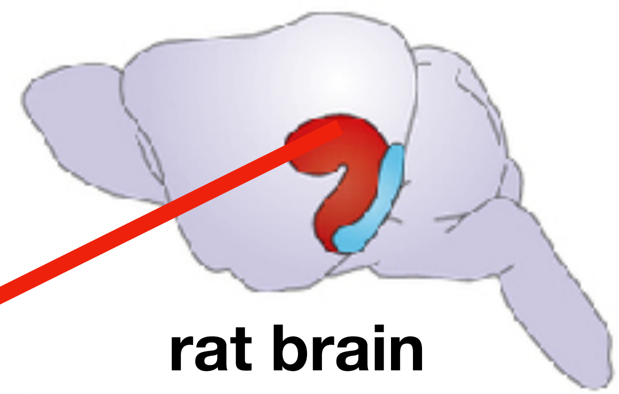
Bridging AI and Cognitive Science
ICLR 2020 Workshop

Tolman and the Cognitive Map



"We assert that the central office itself is far more like a map control room than it is like an old fashioned telephone exchange.
The stimuli, which are allowed in, are not connected by just simple one-to-one switches to the outgoing response. Rather, the incoming impulses are usually worked over and elaborated in the central control room into a tentative, cognitive-like map of the environment. And it is this tentative map, indicating routes and paths and environmental relationships, which finally determines what responses, if any, the animal will finally release" (Tolman 1948, p.192).

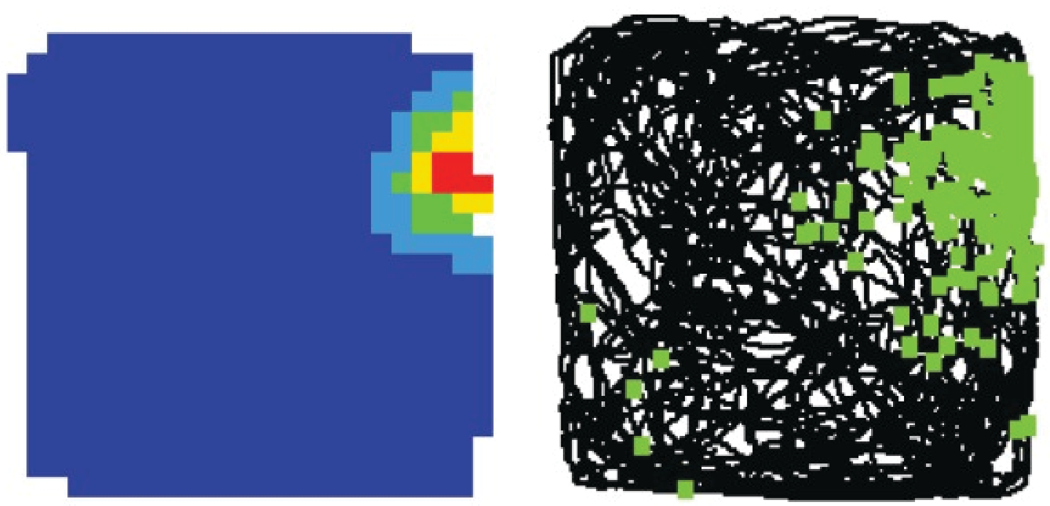
Hippocampus as a "Cognitive Map"



rat brain

Hippocampus
Entorhinal Cortex

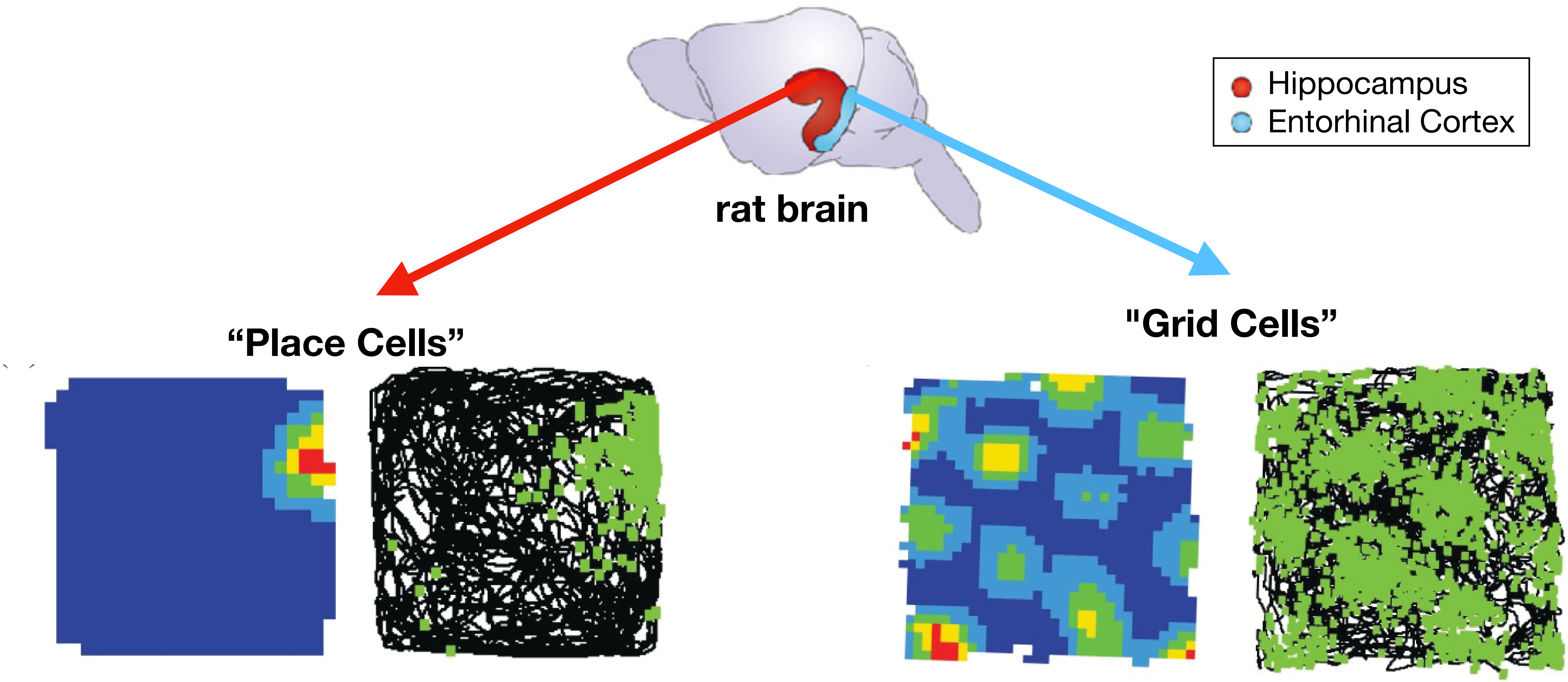
"Place Cells"



O'Keefe & Dostrovsky (1971)

Figures from Hartley et al 2014, Strange et al 2014

Hippocampus as a “Cognitive Map”

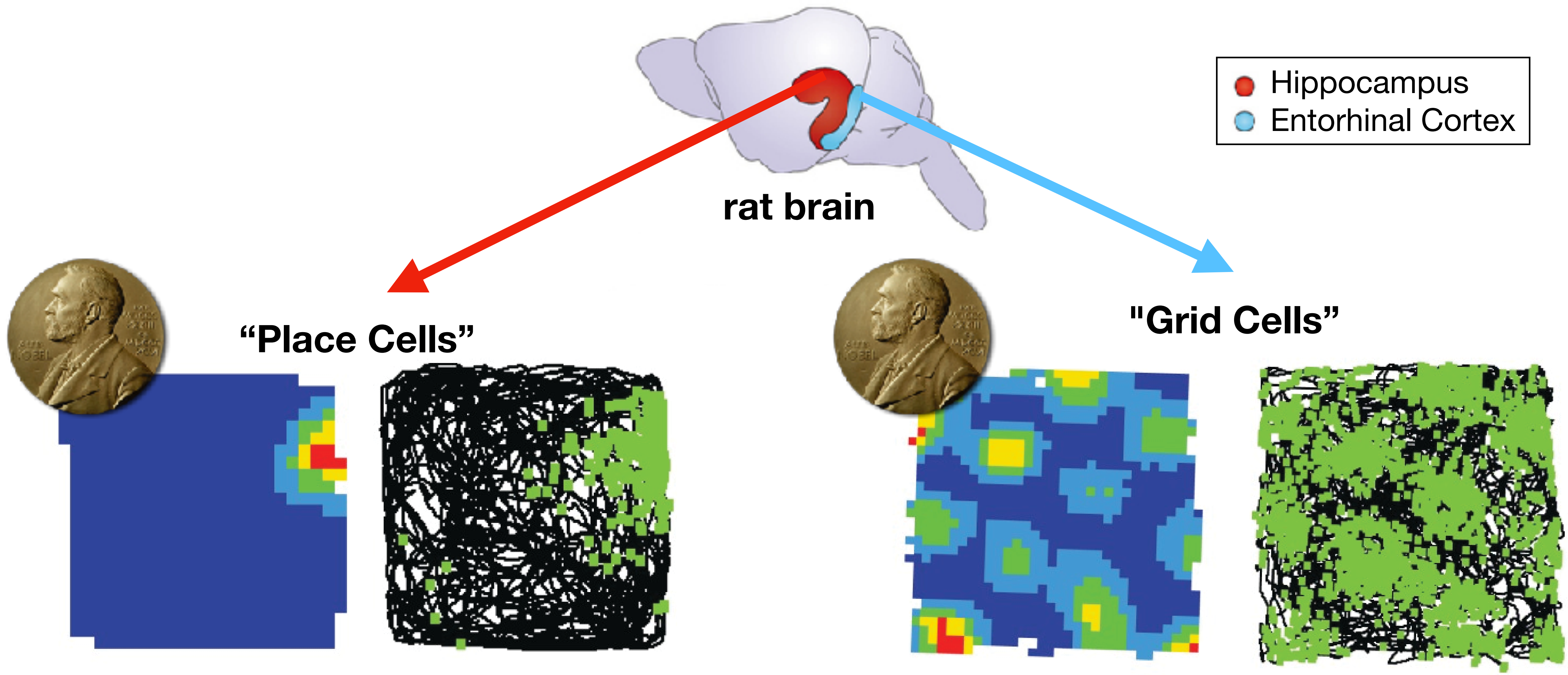


O’Keefe & Dostrovsky (1971)

Hafting, Fyhn, Molden, Moser, Moser (2005)

Figures from Hartley et al 2014, Strange et al 2014

Hippocampus as a "Cognitive Map"

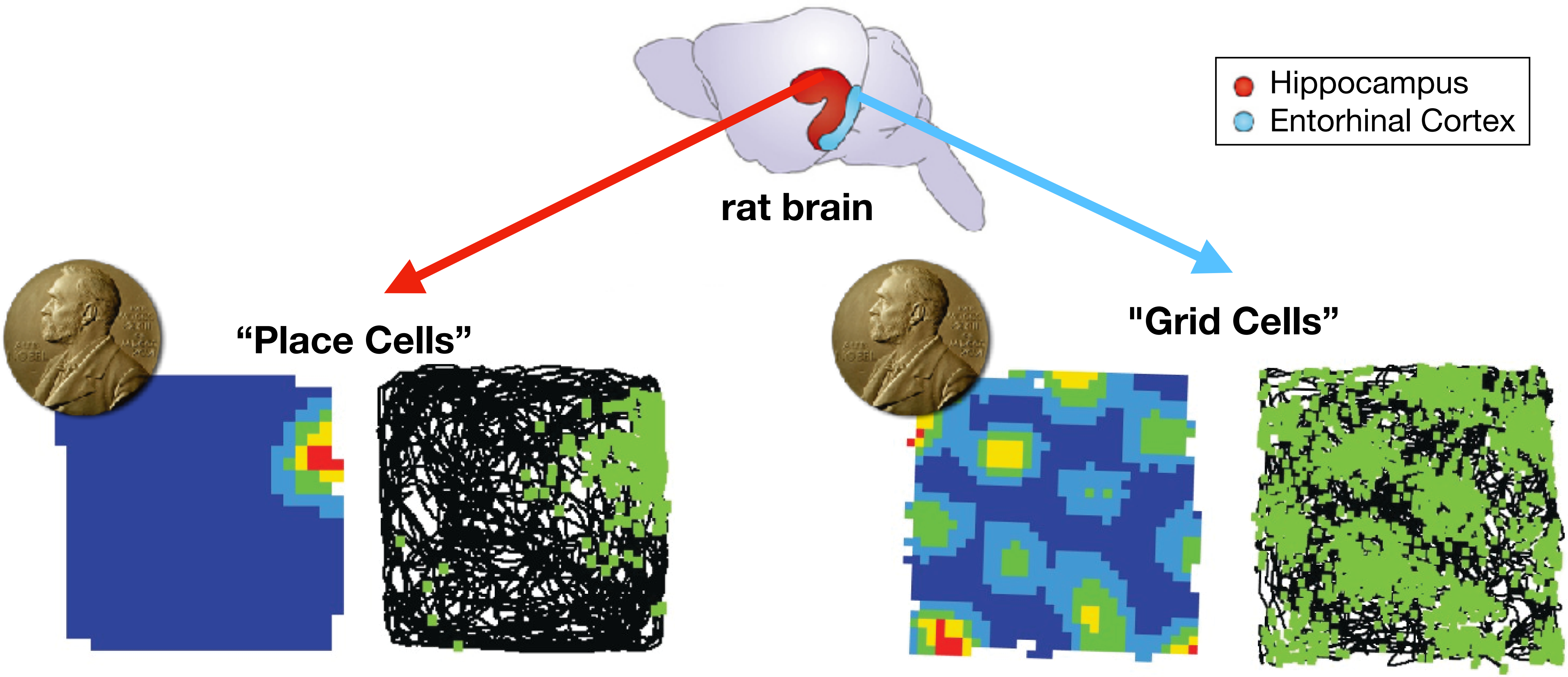


O'Keefe & Dostrovsky (1971)

Hafting, Fyhn, Molden, Moser, Moser (2005)

Figures from Hartley et al 2014, Strange et al 2014

Hippocampus as a "Cognitive Map"



O'Keefe & Dostrovsky (1971)

Hafting, Fyhn, Molden, Moser, Moser (2005)

See also: cells encoding head direction, boundaries, objects….

Figures from Hartley et al 2014, Strange et al 2014

"Place Cells"

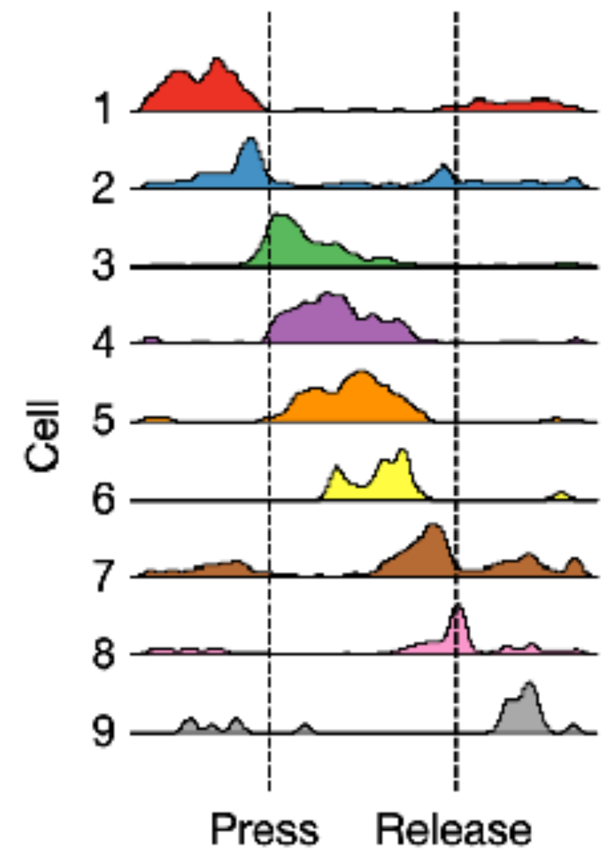
"Place Cells"

- Classically selective for physical location

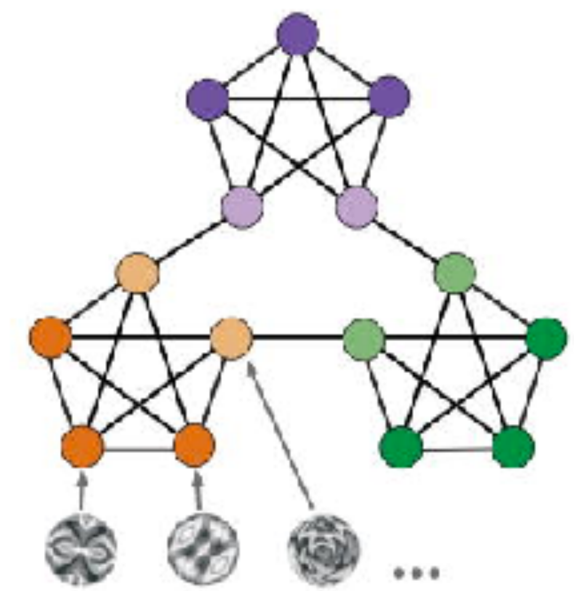
"Place Cells"

- Classically selective for physical location
- Encode non-spatial variables too

e.g. Sound
Aronov et al (2017)



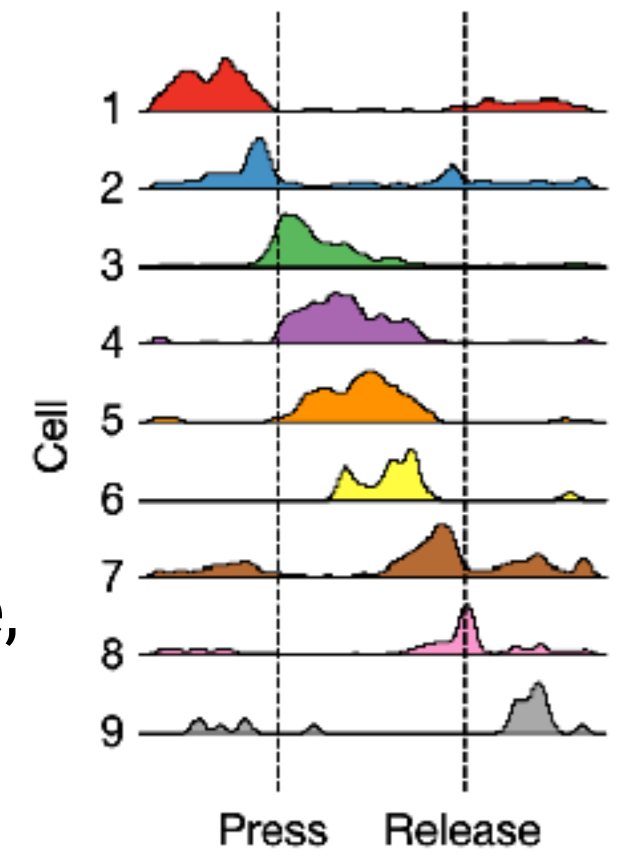
Graph Clusters
Schapiro et al (2016)



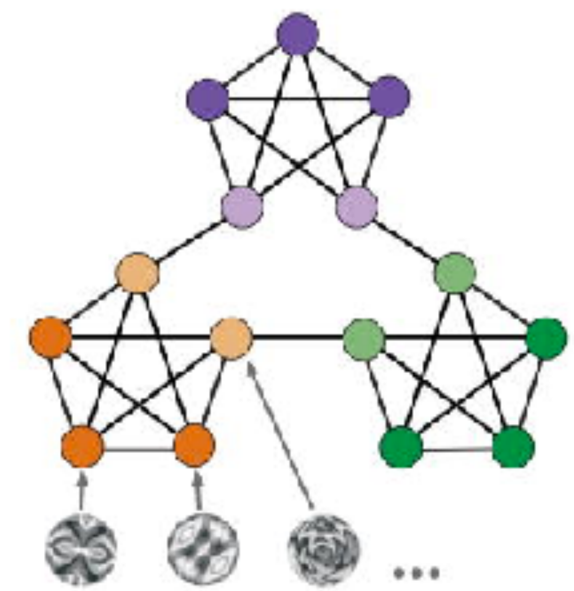
"Place Cells"

- Classically selective for physical location
- Encode non-spatial variables too
- Place fields are altered by experience, in particular CA1 place fields long-lasting "prospective" shifts

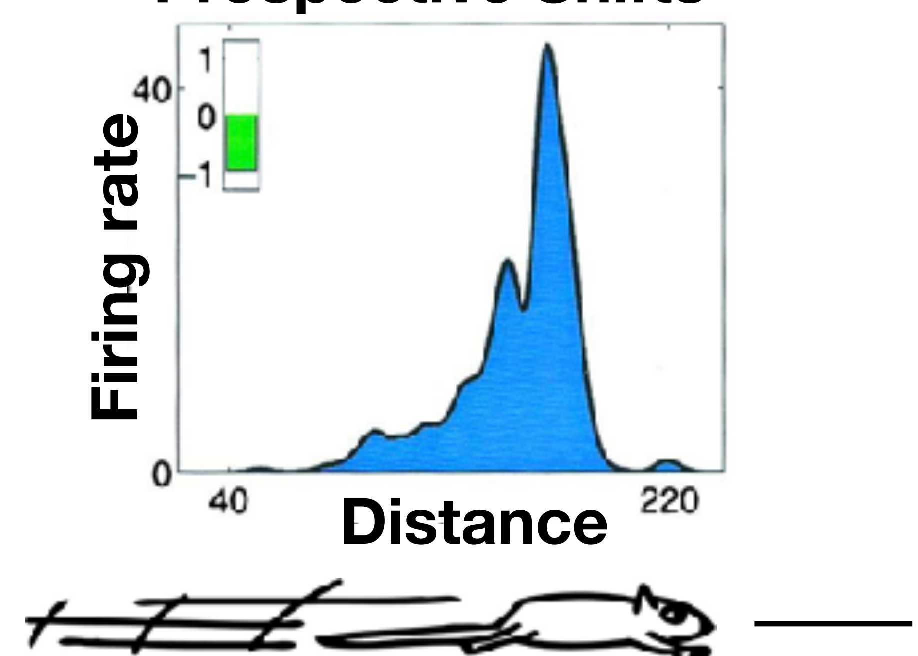
e.g. Sound
Aronov et al (2017)



Graph Clusters
Schapiro et al (2016)



Prospective Shifts

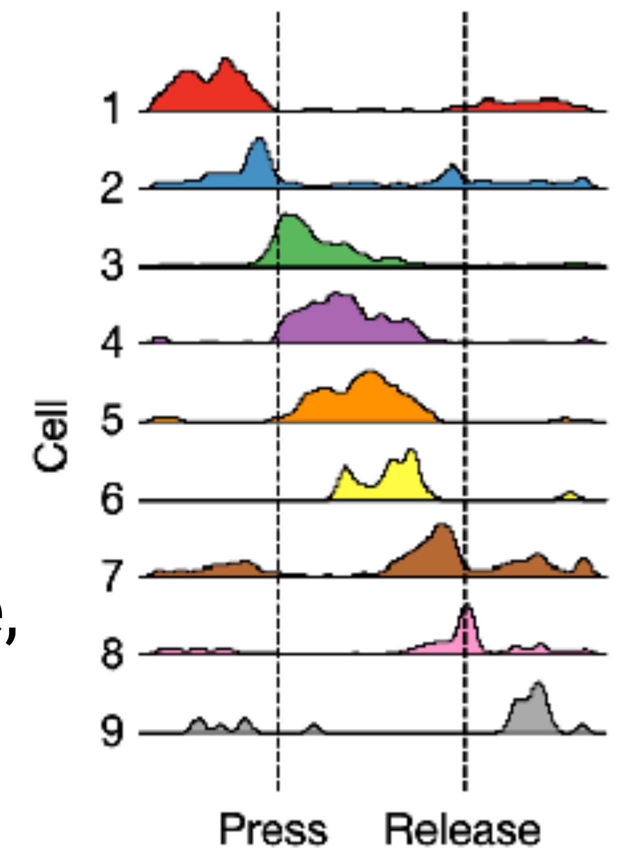


Mehta *et al* 2000

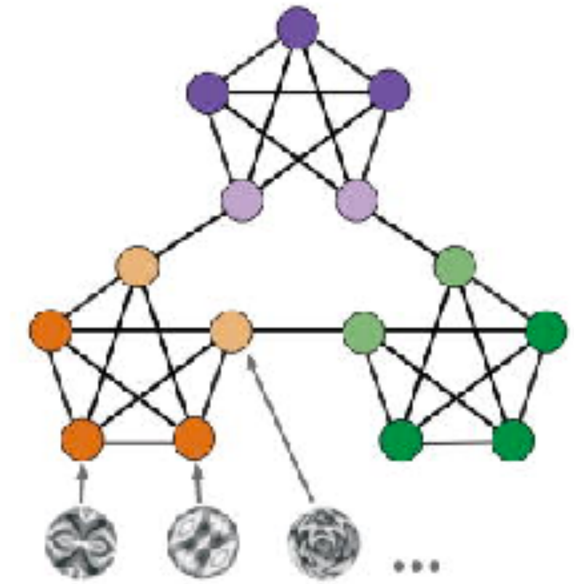
"Place Cells"

- Classically selective for physical location
- Encode non-spatial variables too
- Place fields are altered by experience, in particular CA1 place fields long-lasting “prospective” shifts
- “Global remapping”

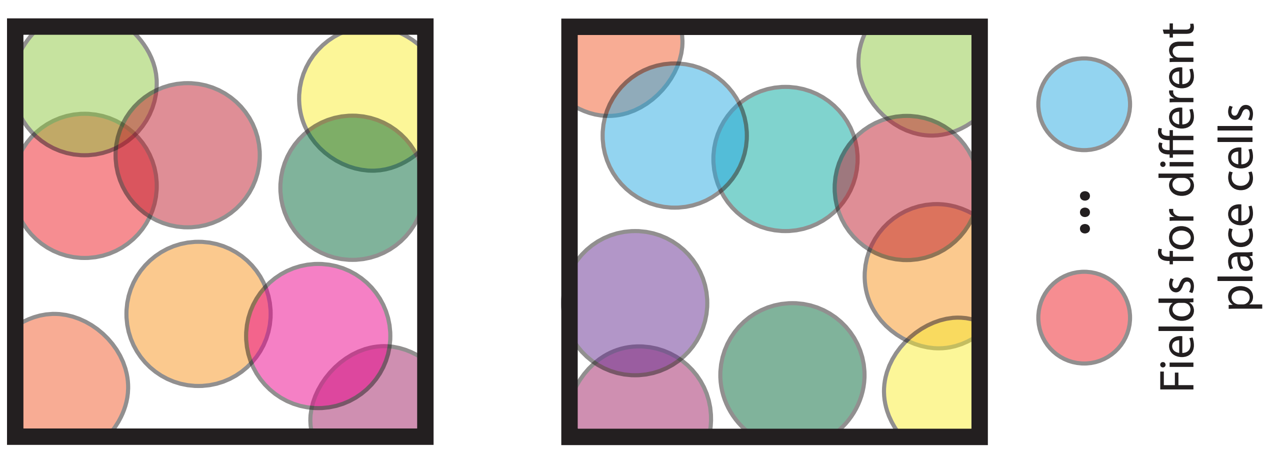
e.g. Sound
Aronov et al (2017)



Graph Clusters
Schapiro et al (2016)



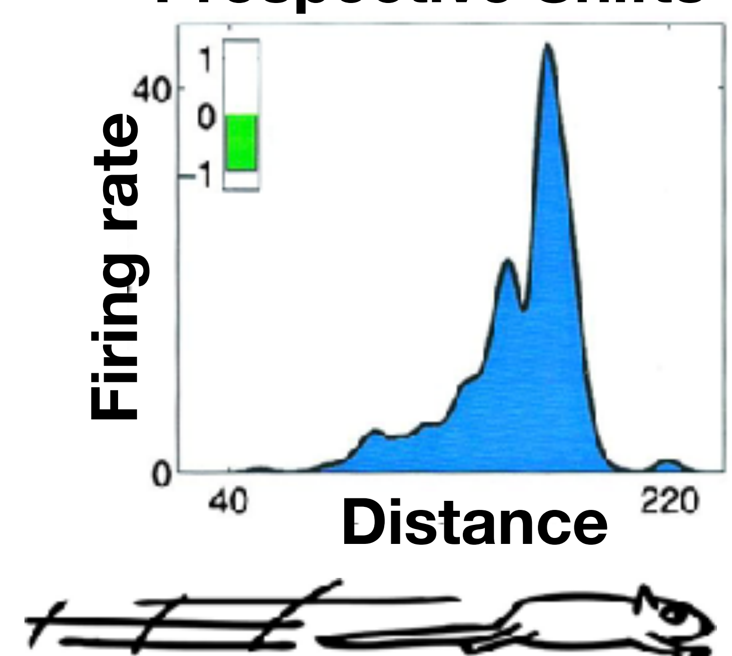
Global remapping
of hippocampal place cells



room 1 **room 2**

Behrens *et al* 2019

Prospective Shifts



Mehta *et al* 2000

Grid cells

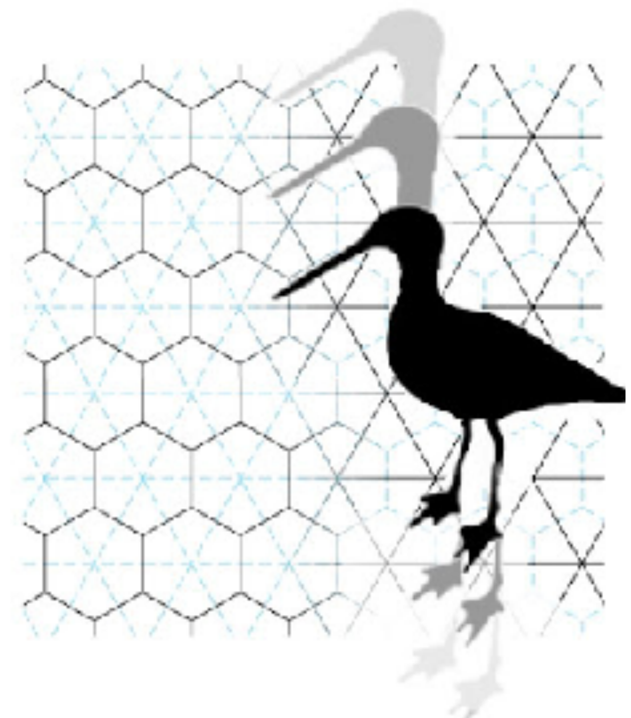
Grid cells

- Firing patterns are spatially periodic

Grid cells

- Firing patterns are spatially periodic
- Also encode non-spatial variables

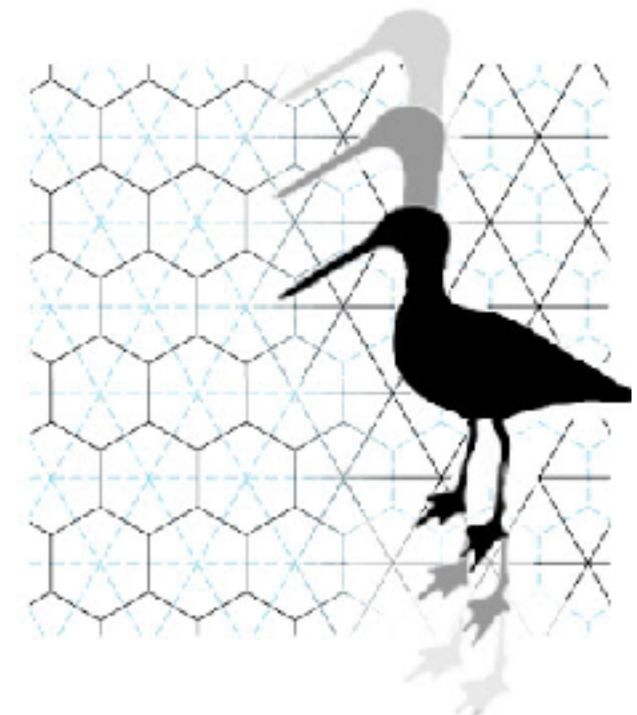
e.g. "Stretchy birds"
Continescu et al (2017)



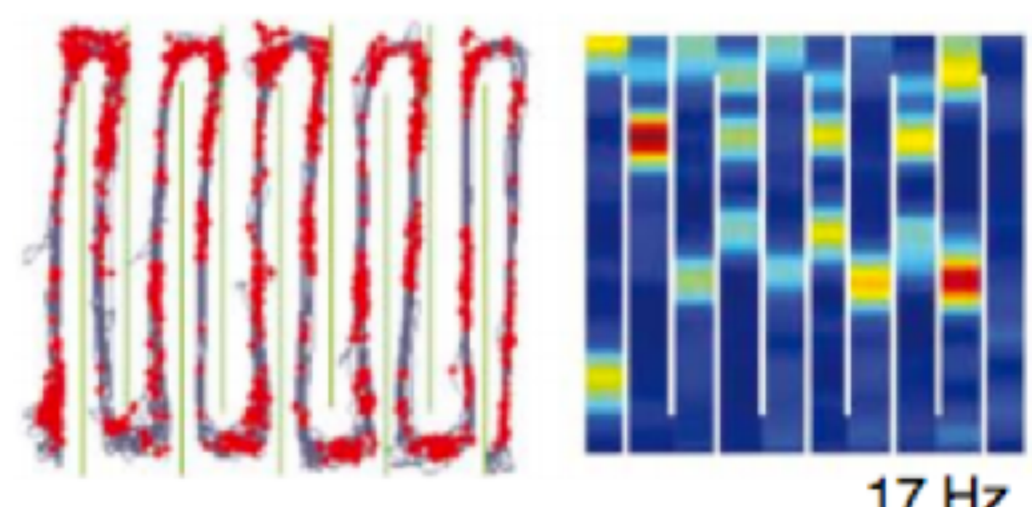
Grid cells

- Firing patterns are spatially periodic
- Also encode non-spatial variables
- Firing properties influenced by boundaries of environment

e.g. “Stretchy birds”
Continescu et al (2017)



Receptive Field in Hairpin Maze

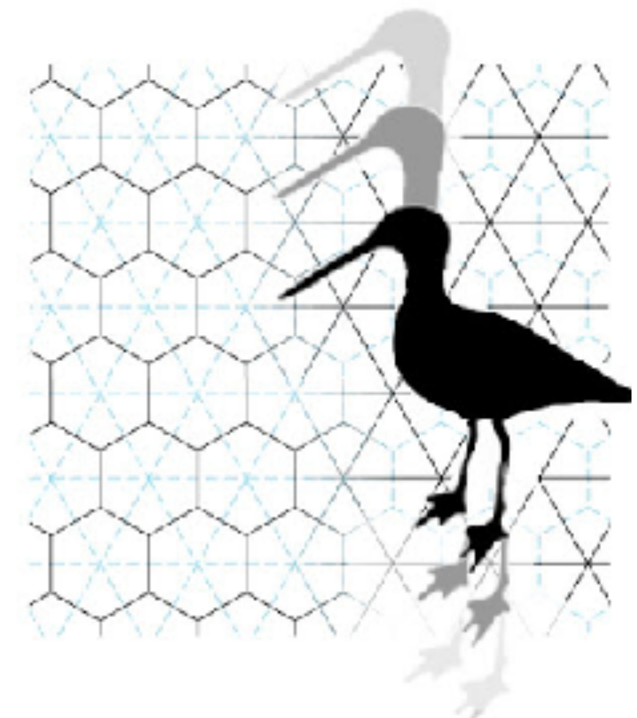


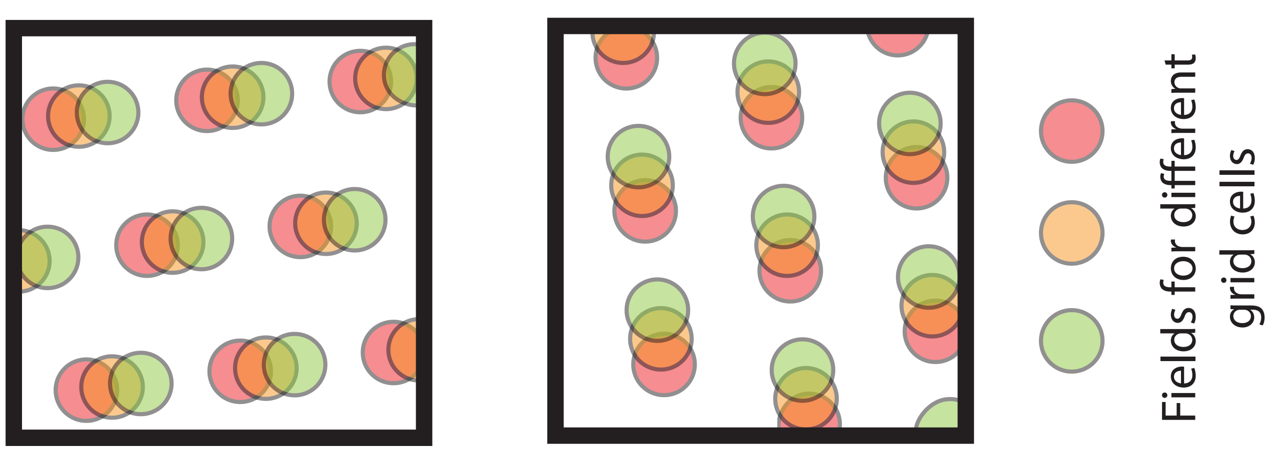
Derdikman *et al* 2009

Grid cells

- Firing patterns are spatially periodic
- Also encode non-spatial variables
- Firing properties influenced by boundaries of environment
- No global remapping

e.g. “Stretchy birds”
Continescu et al (2017)

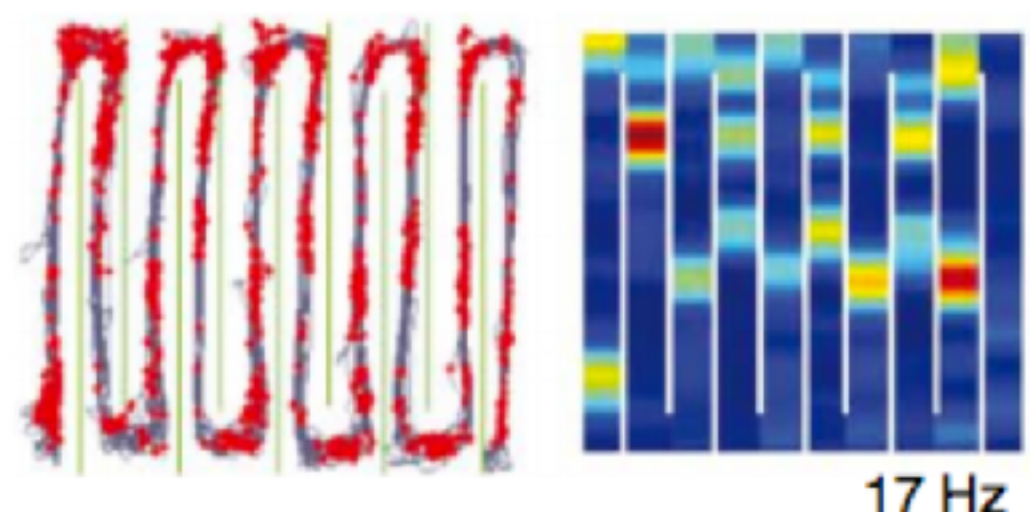




room 1 **room 2**

Behrens *et al* 2019

Receptive Field in Hairpin Maze



Derdikman *et al* 2009

Reinforcement Learning (RL)

Reinforcement Learning (RL)

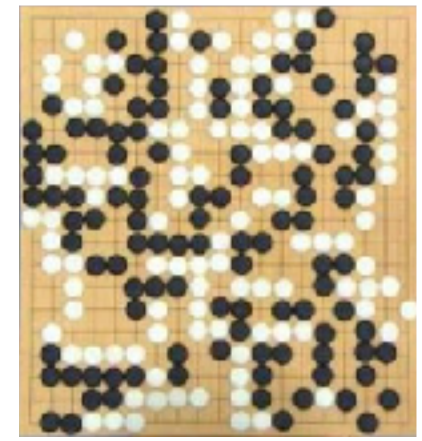
- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

Reinforcement Learning (RL)

- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

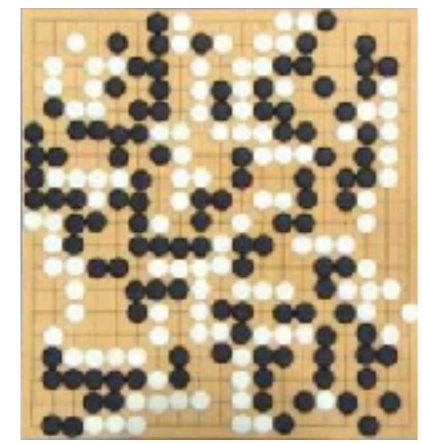
Reinforcement Learning (RL)

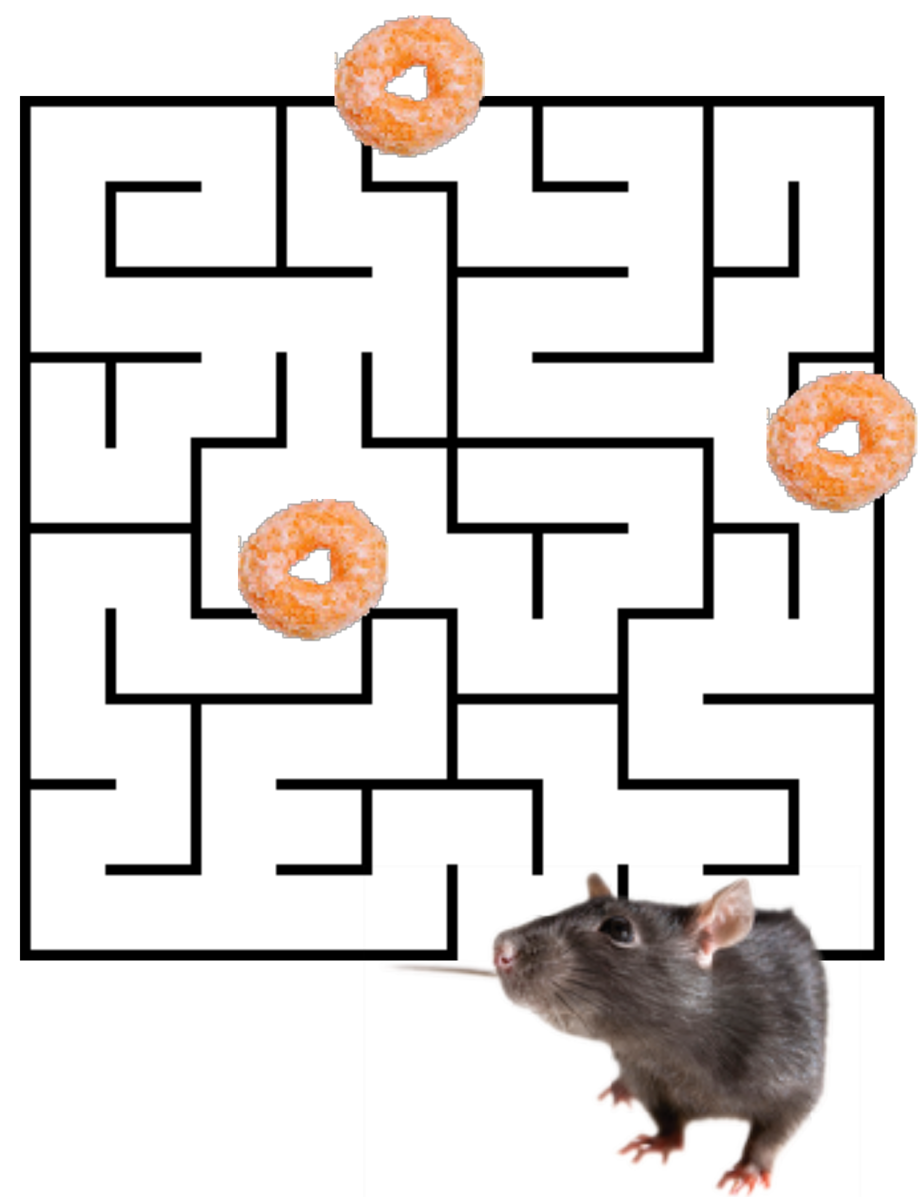
- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error



Reinforcement Learning (RL)

- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

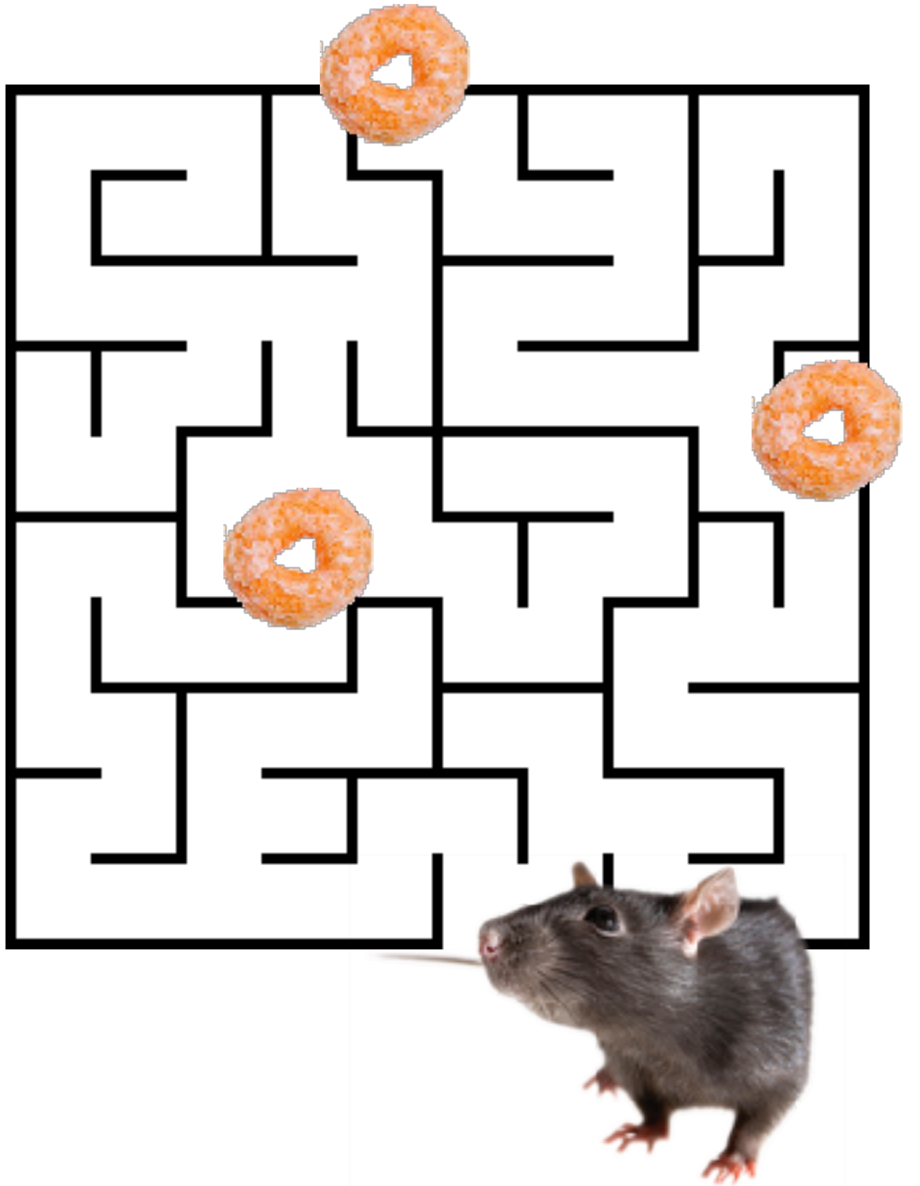




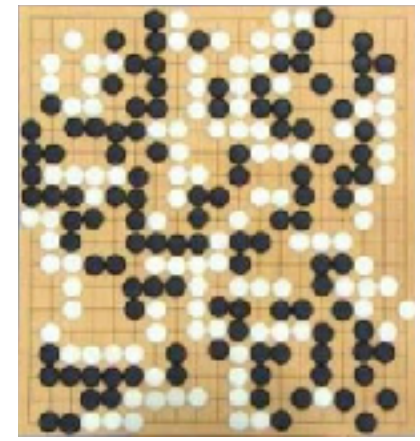
Reinforcement Learning (RL)

- **Reinforcement learning problem:** Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

- RL can be very sample inefficient:

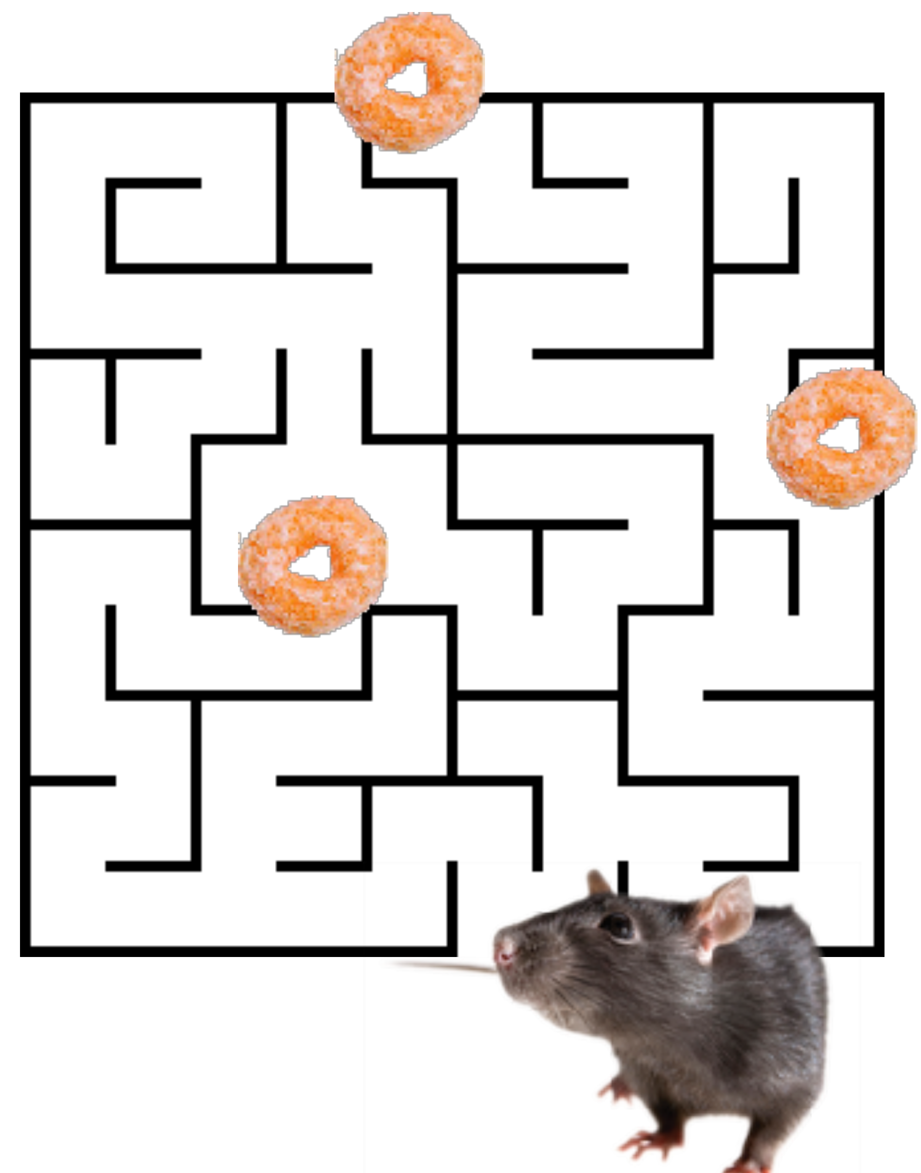


Reinforcement Learning (RL)



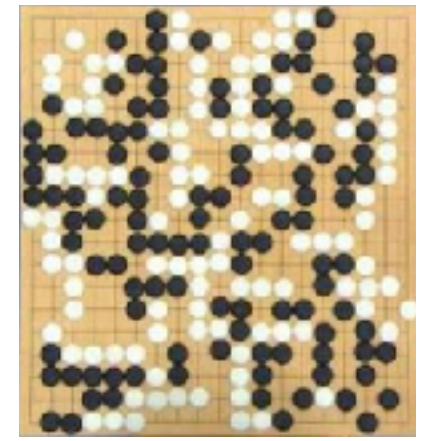
- **Reinforcement learning problem:** Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

- RL can be very sample inefficient:
    - RL agents discard a lot of information not related to reward

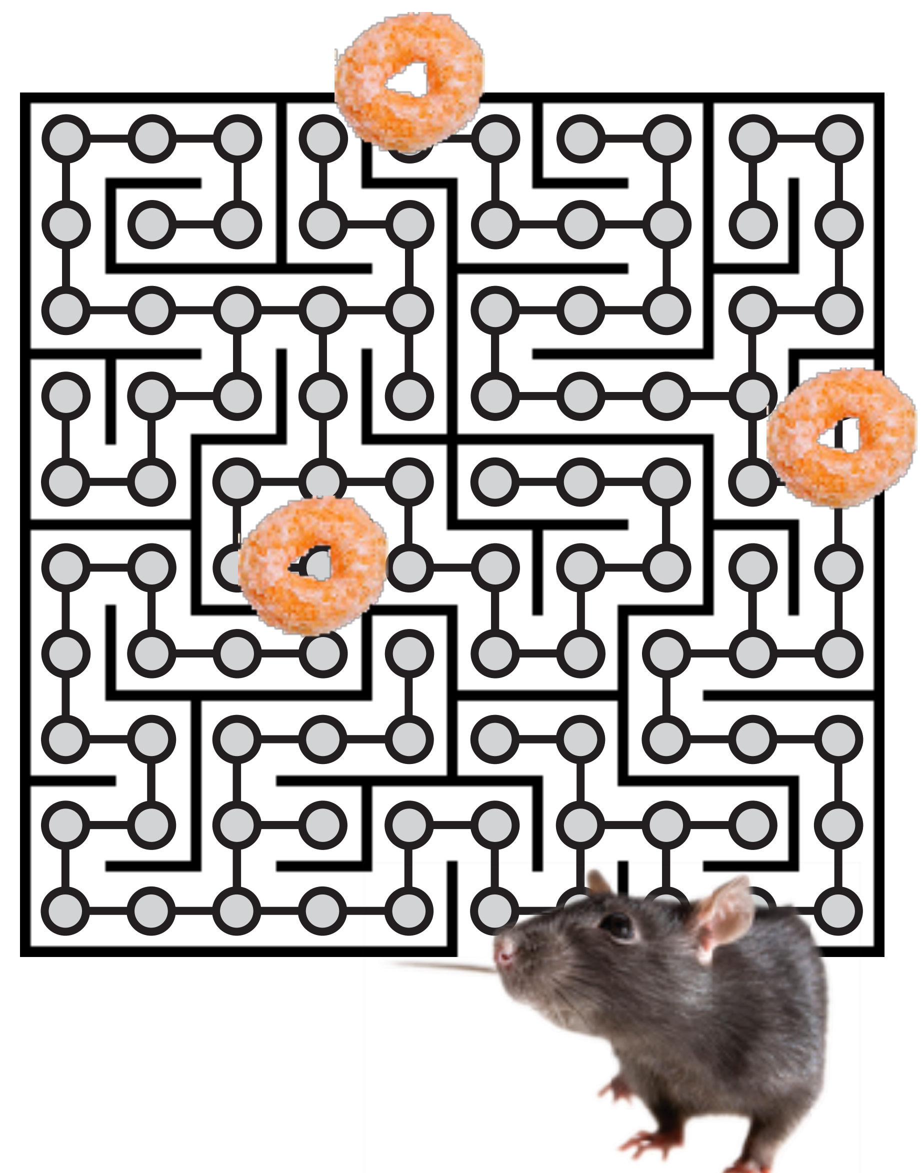


Reinforcement Learning (RL)

- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error



- RL can be very sample inefficient:
  - RL agents discard a lot of information not related to reward
  - High dimensional state spaces and slow coverage

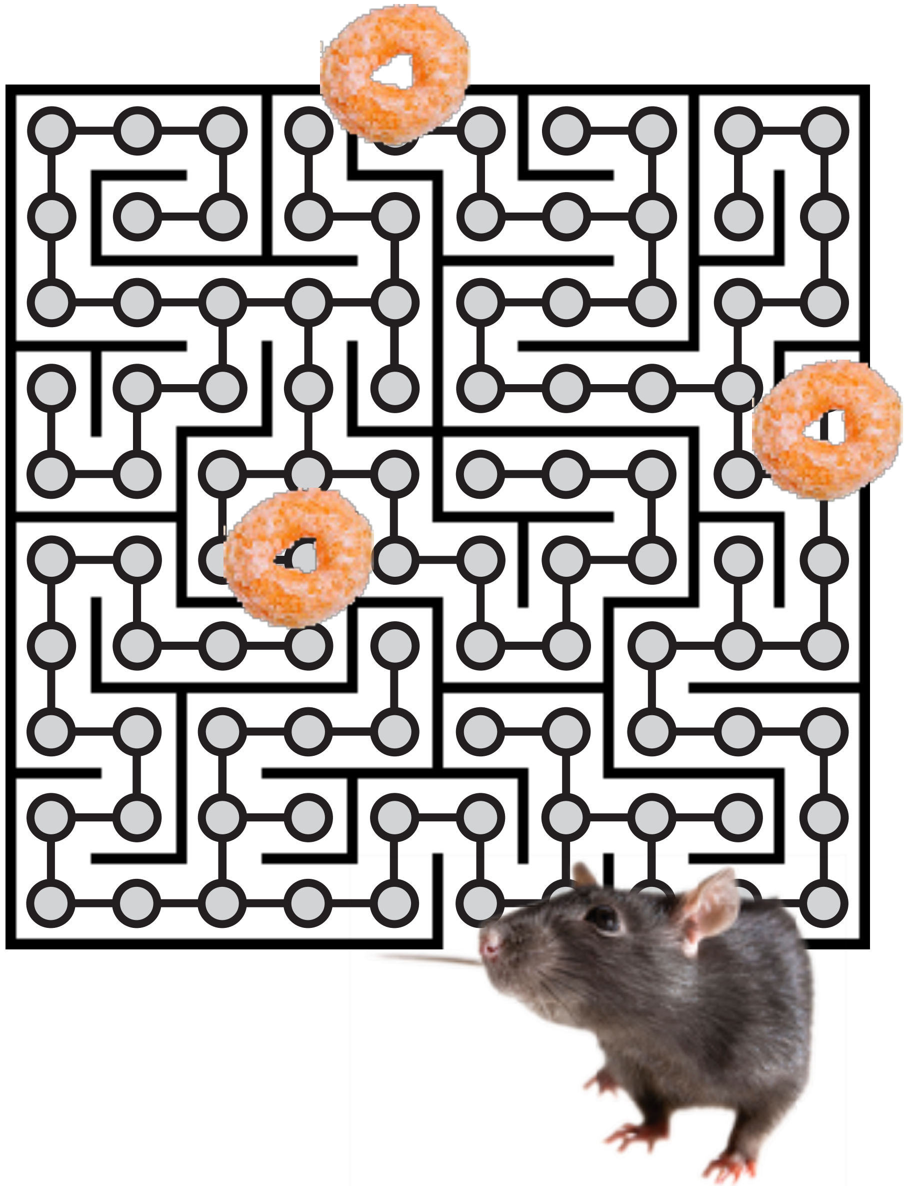


Reinforcement Learning (RL)

- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

- RL can be very sample inefficient:
  - RL agents discard a lot of information not related to reward
  - High dimensional state spaces and slow coverage

- How do neural representations capture structure and use it for efficient + flexible learning and inference?



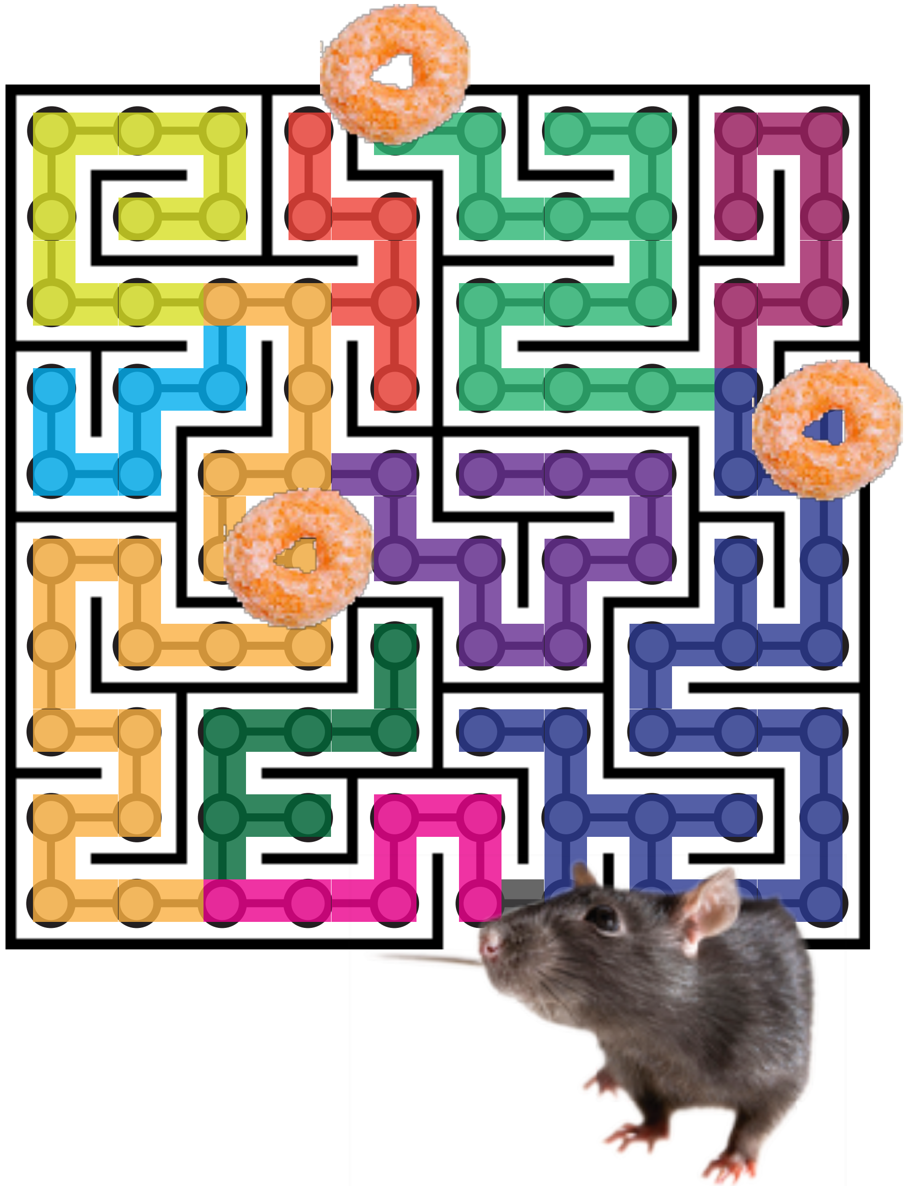
Reinforcement Learning (RL)

- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative **reward**, or value, over future states by trial and error

- RL can be very sample inefficient:
  - RL agents discard a lot of information not related to reward
  - High dimensional state spaces and slow coverage

- How do neural representations capture structure and use it for efficient + flexible learning and inference?

- How do we get machines to do that?



What do we need in a representation?

What do we need in a representation?

What additional structure, besides reward, should we learn?

What do we need in a representation?

What additional structure, besides reward, should we learn?

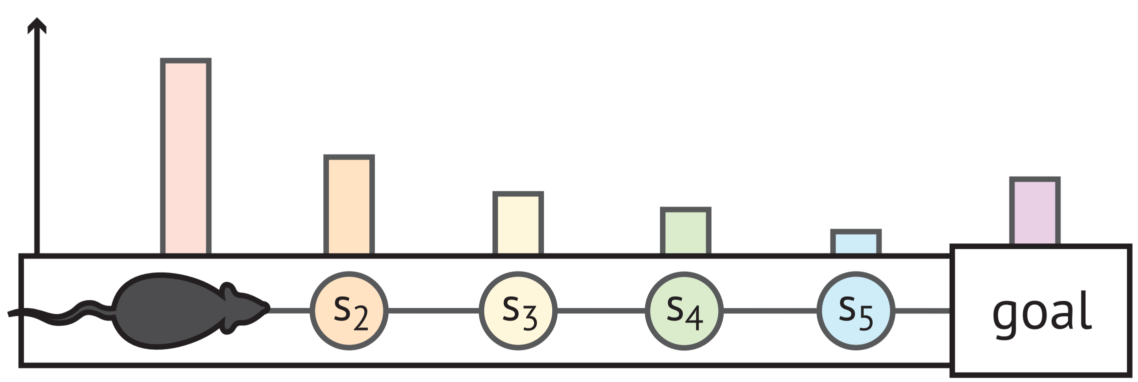
How can we make it so the downstream RL process has fewer things to learn about?

What do we need in a representation?

What additional structure, besides reward, should we learn?

How can we make it so the downstream RL process has fewer things to learn about?

Predictions
Represent states in terms of the predictions they make

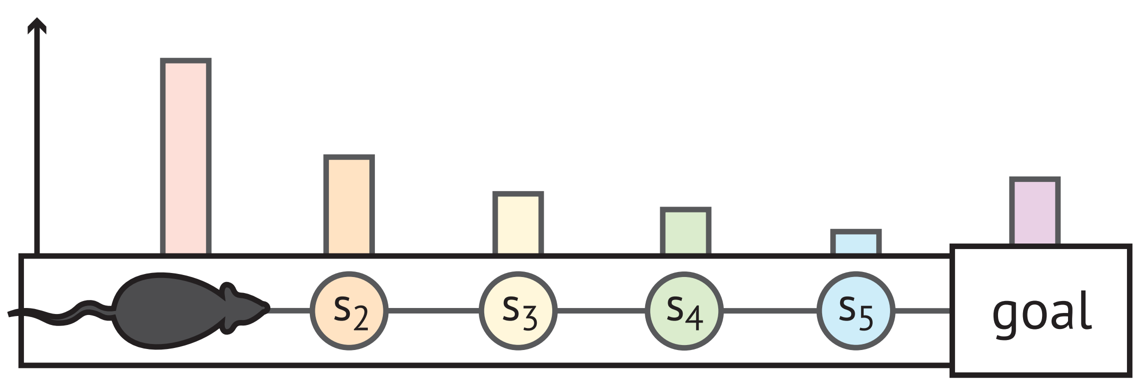


Predictive Representations:
Predictive State Representations (Singh et al 2012)
Successor features (Dayan 1993, Barreto et al 2016, Kulkarni et al 2016)
Universal Value Function Approximators (Schaul et al 2015)
Contrastive predictive coding (Oord et al 2018)
MERLIN (Wayne et al 2018)

What do we need in a representation?

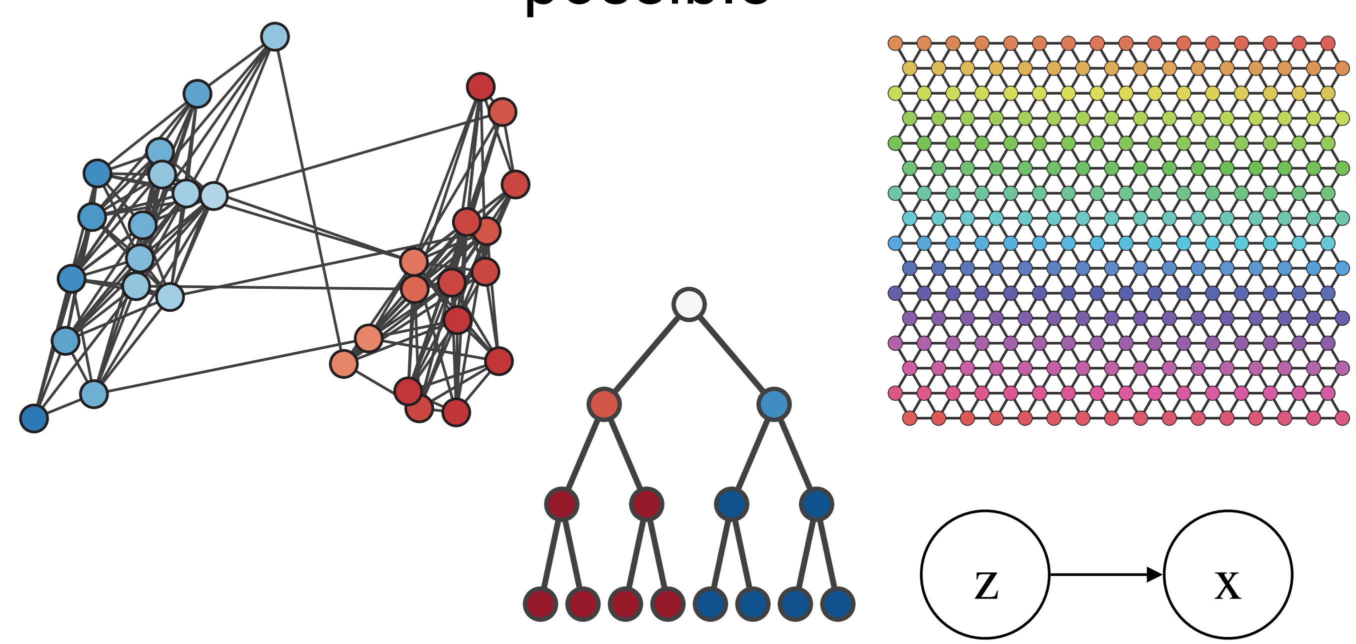
What additional structure, besides reward, should we learn?

Predictions
Represent states in terms of the predictions they make



How can we make it so the downstream RL process has fewer things to learn about?

Compression
Learn as short description as possible



Predictive Representations:
Predictive State Representations (Singh et al 2012)
Successor features (Dayan 1993, Barreto et al 2016, Kulkarni et al 2016)
Universal Value Function Approximators (Schaul et al 2015)
Contrastive predictive coding (Oord et al 2018)
MERLIN (Wayne et al 2018)

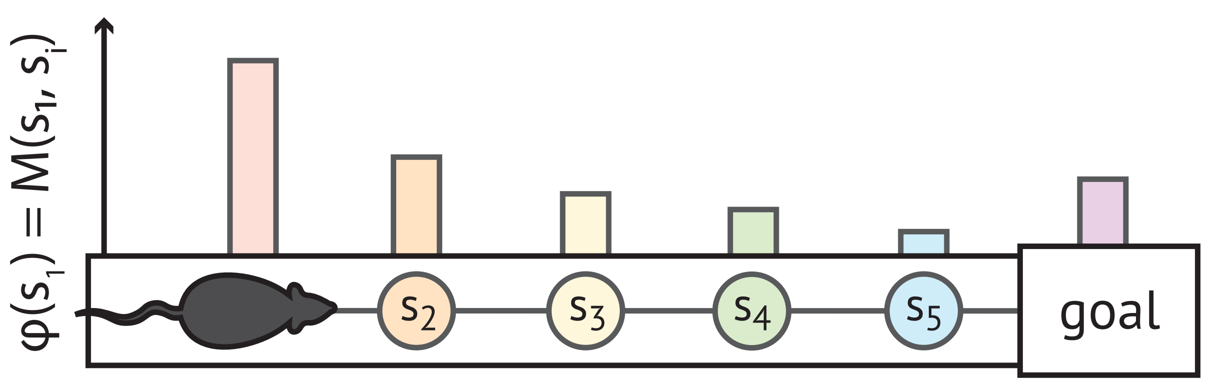
What do we need in a representation?

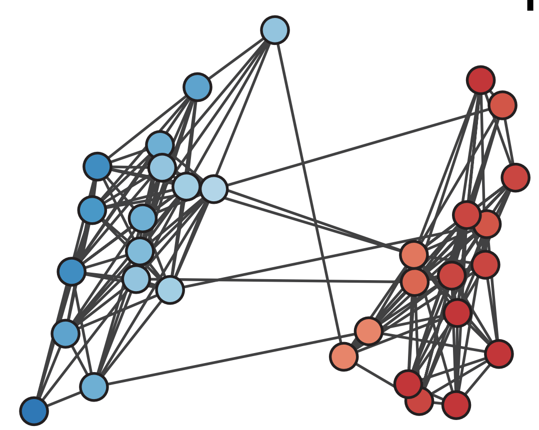
What additional structure, besides reward, should we learn?

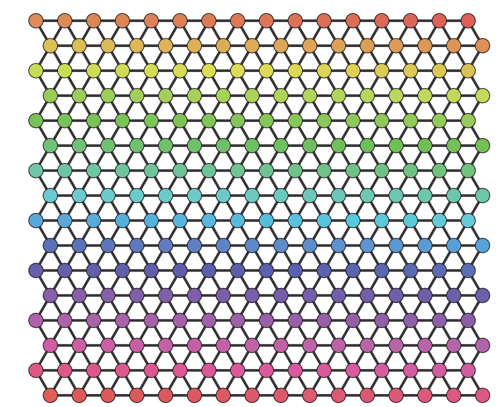
How can we make it so the downstream RL process has fewer things to learn about?

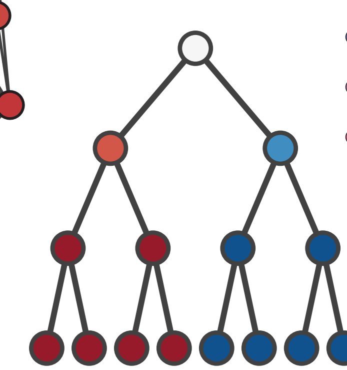
Predictions
Represent states in terms of the predictions they make

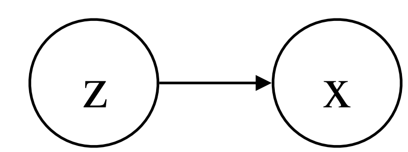
Compression
Learn as short description as possible

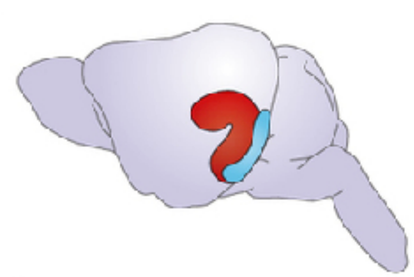












Place Cells

For each state, learn how often you will visit other states?

Grid cells

Capture low-dimensional structure among predictions

Stachenfeld, Botvinick, Gershman (2017)

Predictive representations

Dayan (1993)

Predictive representations

$$V(s) = E[\ R(s_{t0}) + \gamma R(s_{t1}) + \gamma^2 R(s_{t2}) + \ldots]$$

value

add up expected rewards for states over time
discount with $\gamma<1$ to prioritize immediacy

Dayan (1993)

Predictive representations

$$V(s) = E[\, R(s_{t0}) + \gamma R(s_{t1}) + \gamma^2 R(s_{t2}) + \ldots]$$

value

add up expected rewards for states over time
discount with γ<1 to prioritize immediacy

$$V = \left(\sum_{t=0}^{\infty} \gamma^t T^t \right) R$$

expected visits
different states

reward

Dayan (1993)

Predictive representations

$$V(s) = E[\, R(s_{t0}) + \gamma R(s_{t1}) + \gamma^2 R(s_{t2}) + \ldots]$$

value

add up expected rewards for states over time
discount with γ<1 to prioritize immediacy

$$V = \Big(\sum_{t=0}^{\infty} \gamma^t T^t \Big) \; R$$

expected visits different states

reward

$$V = M \, R$$

successor representation matrix

reward

Dayan (1993)

Predictive representations

$$V(s) = E[\, R(s_{t0}) + \gamma R(s_{t1}) + \gamma^2 R(s_{t2}) + \ldots]$$

value

add up expected rewards for states over time
discount with γ<1 to prioritize immediacy

$$V = \left(\sum_{t=0}^{\infty} \gamma^t T^t \right) R$$

expected visits different states

reward

$$V = M\,R$$

successor representation matrix

reward

Instantaneous reward

Dayan (1993)

Predictive representations

$$V(s) = E[\, R(s_{t0}) + \gamma R(s_{t1}) + \gamma^2 R(s_{t2}) + \ldots]$$

value

add up expected rewards for states over time
discount with γ<1 to prioritize immediacy

$$V = \Big(\sum_{t=0}^{\infty} \gamma^t T^t \Big) \; R$$

expected visits different states

reward

$$V = M\,R$$

Caches long-term effect of dynamics

successor representation matrix

reward

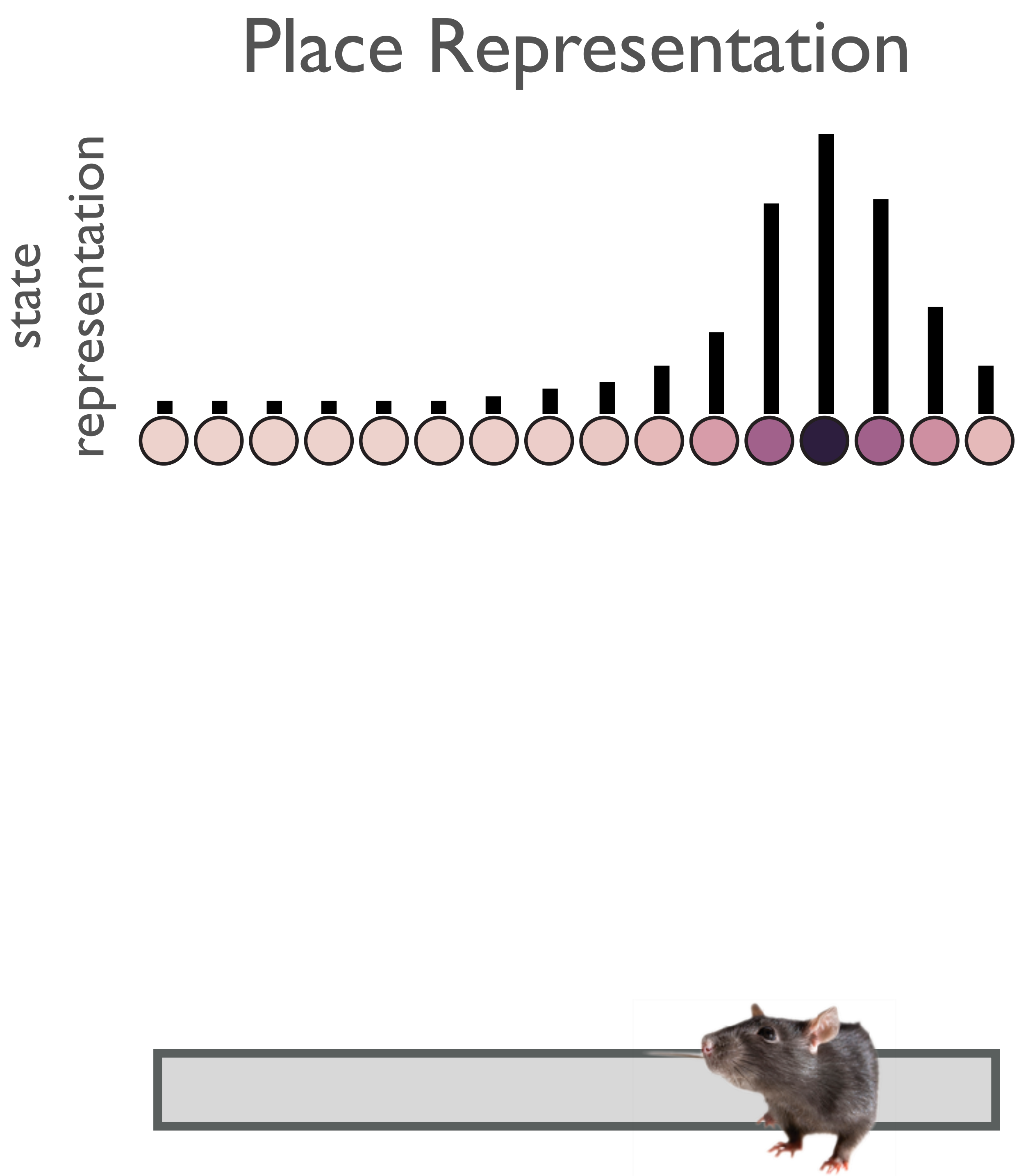
Instantaneous reward

Dayan (1993)

Successor representations in hippocampus

Stachenfeld, Botvinick, Gershman (2017)

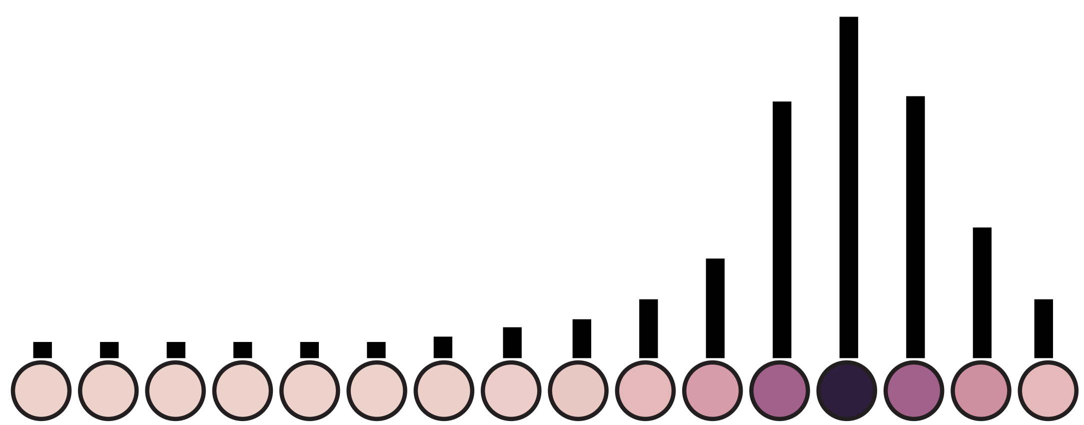
Successor representations in hippocampus



Stachenfeld, Botvinick, Gershman (2017)

Successor representations in hippocampus

Place Representation

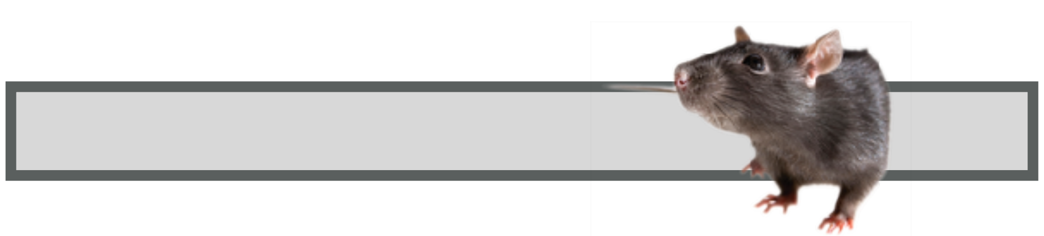


Cell i encoding distance from μ_i

Firing of cell i
At location j

$$Ae^{-\left(\frac{x_j-\mu_i}{2\sigma}\right)^2}$$

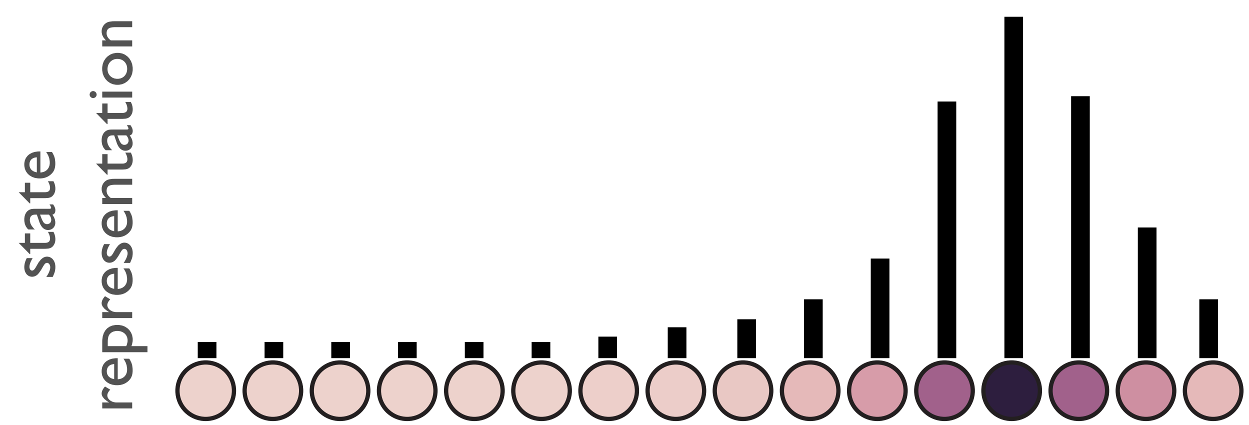
Euclidean Gaussian with center i
centered at μ_i, width σ, height A



$\mathbf{x_j}$

Successor representations in hippocampus

Place Representation



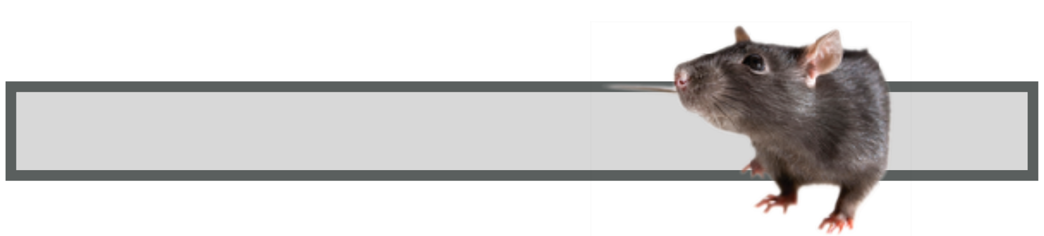
Cell i encoding distance from μ_i

Firing of cell i
At location j

$$Ae^{-\left(\frac{x_j-\mu_i}{2\sigma}\right)^2}$$

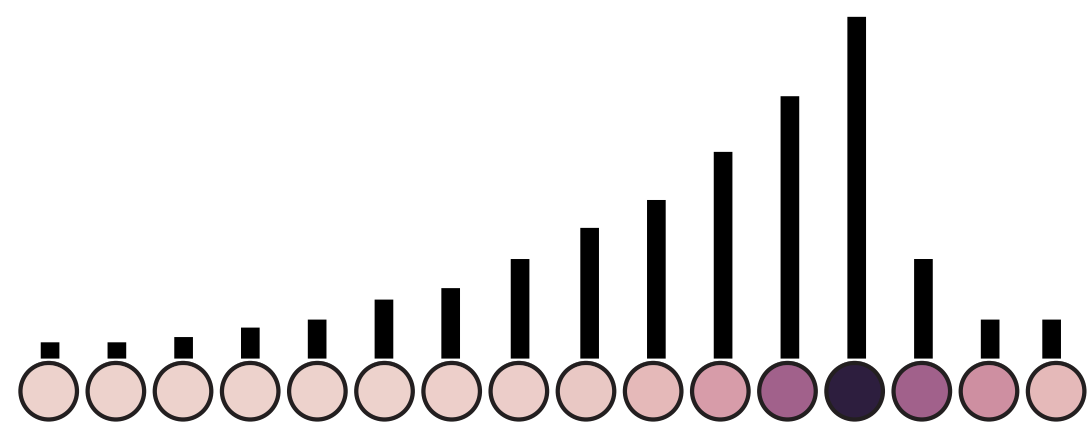
Euclidean Gaussian with center i

centered at μ_i, width σ, height A



$\mathbf{x_j}$

Successor Representation

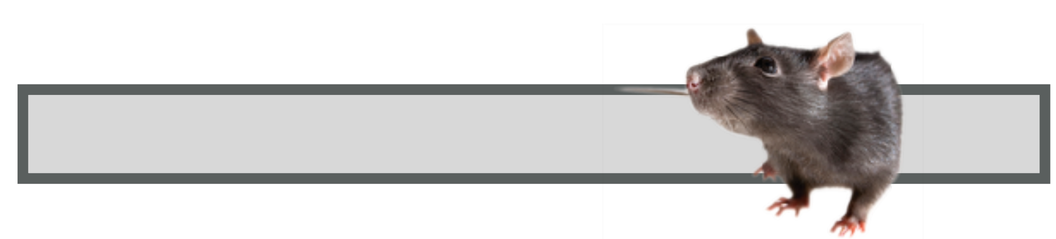


Cell i encoding predictions about state i

row j of matrix $M = \sum_{t=0}^{\infty} \gamma^t T^t$

Expected #visits to state i

Discount γ

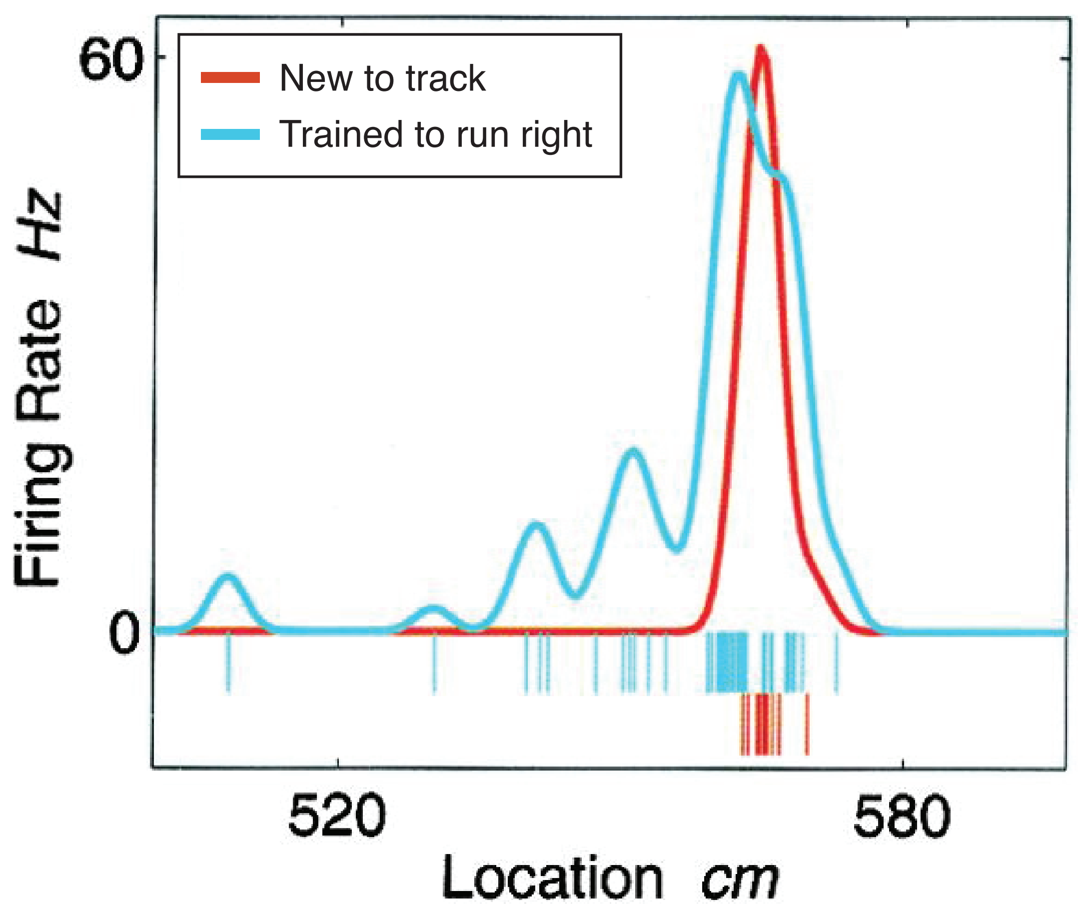


$\mathbf{s_j}$

Place fields along a track

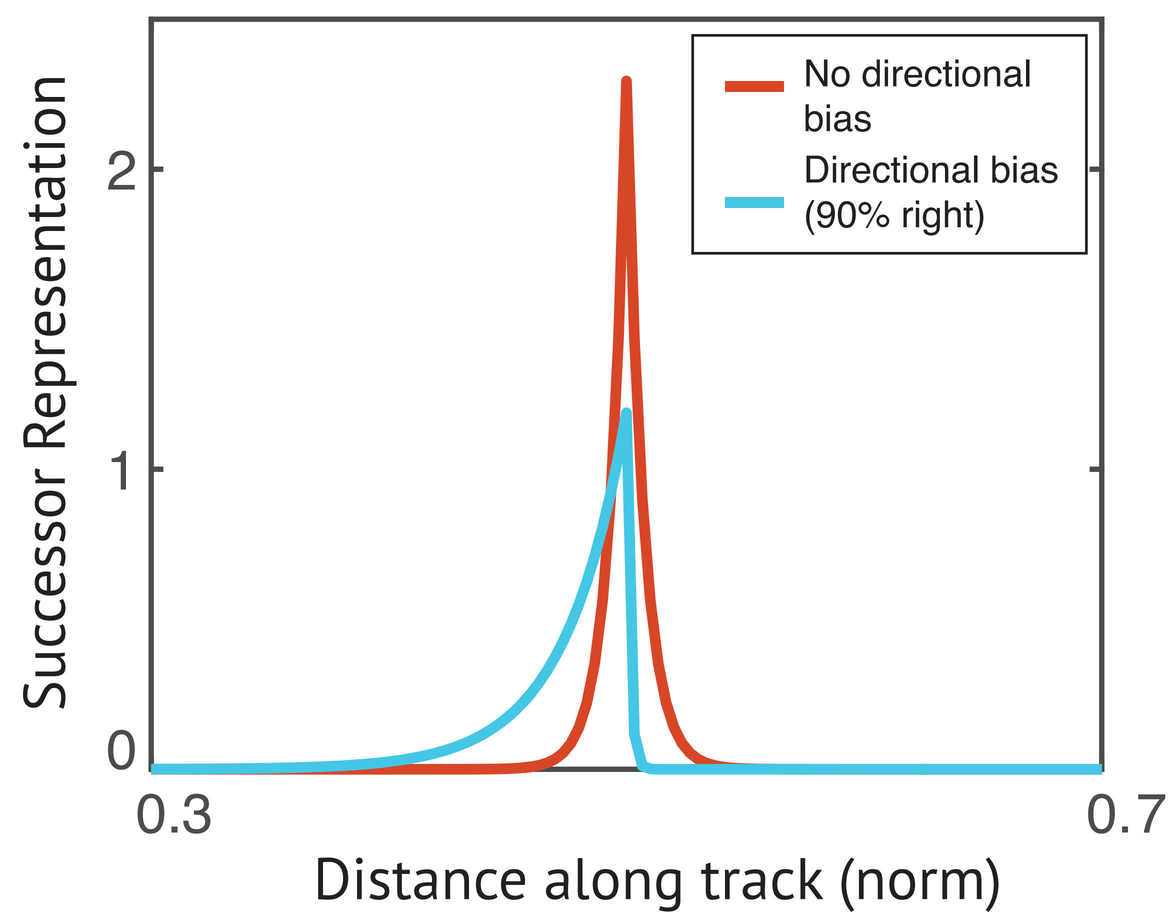
Experiment

A **Mehta *et al.* (2000)**



Model

B **Simulated SR Place cells**

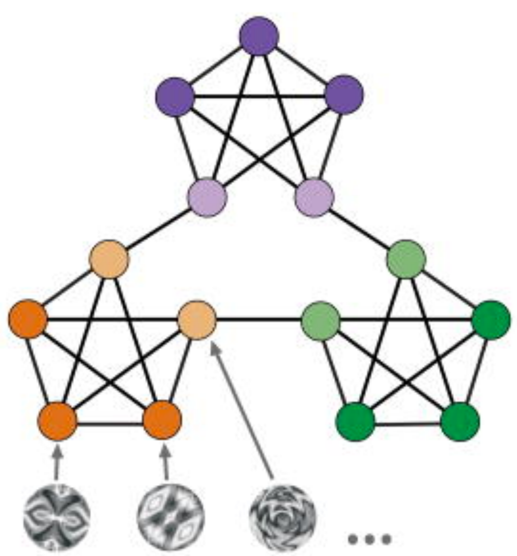


direction of motion

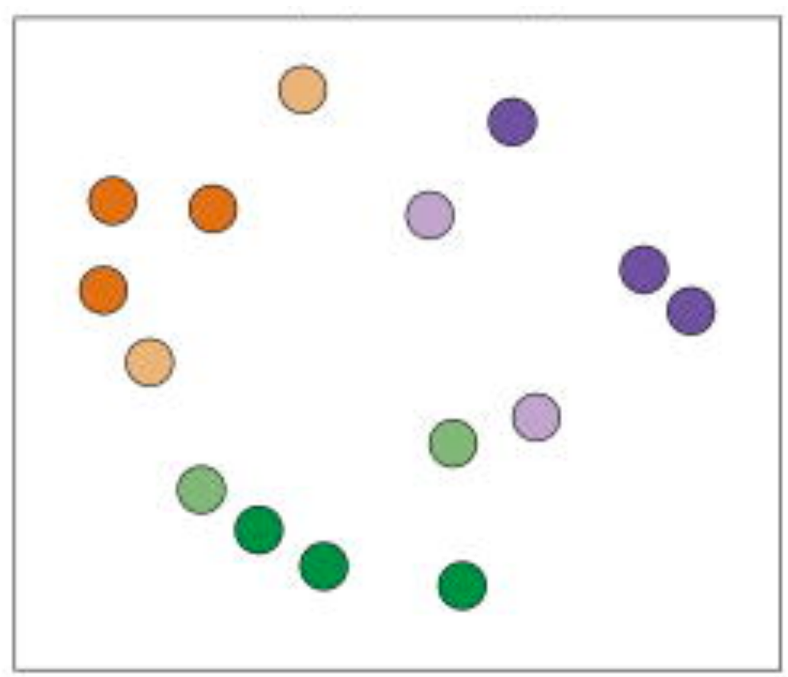
Stachenfeld, Botvinick, Gershman 2017

Clustering by temporal structure

Experiment



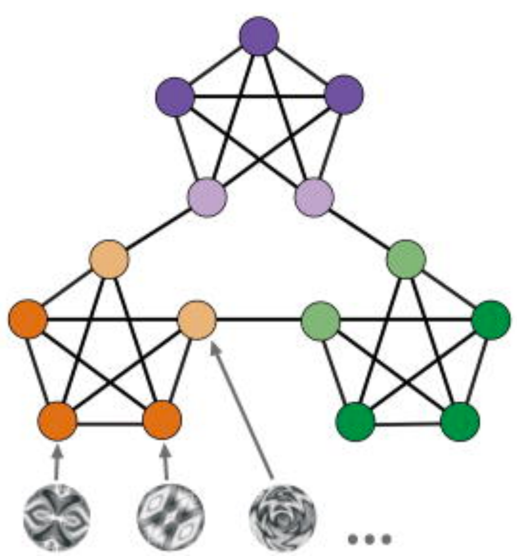
Bilateral hippocampus
MDS



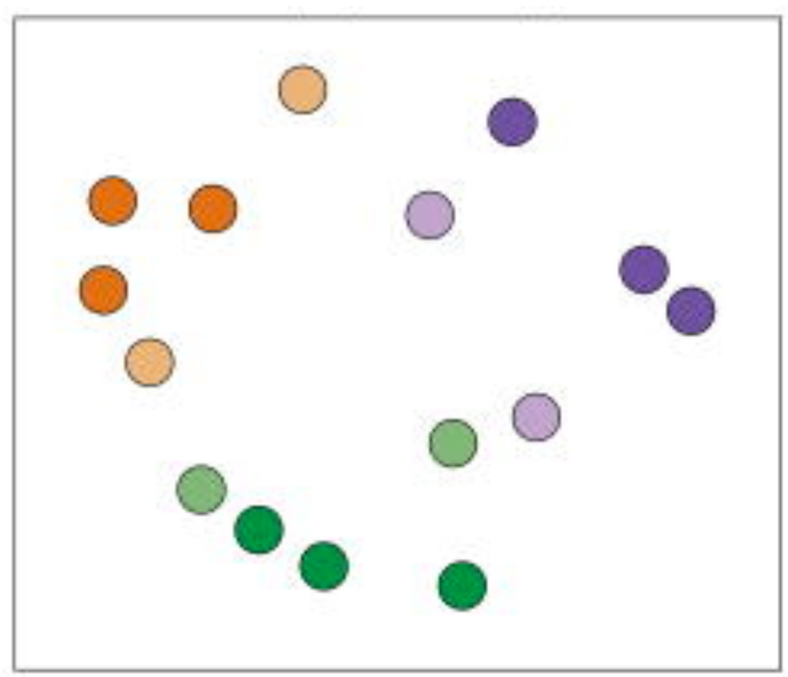
Schapiro et al. (2013)

Clustering by temporal structure

Experiment

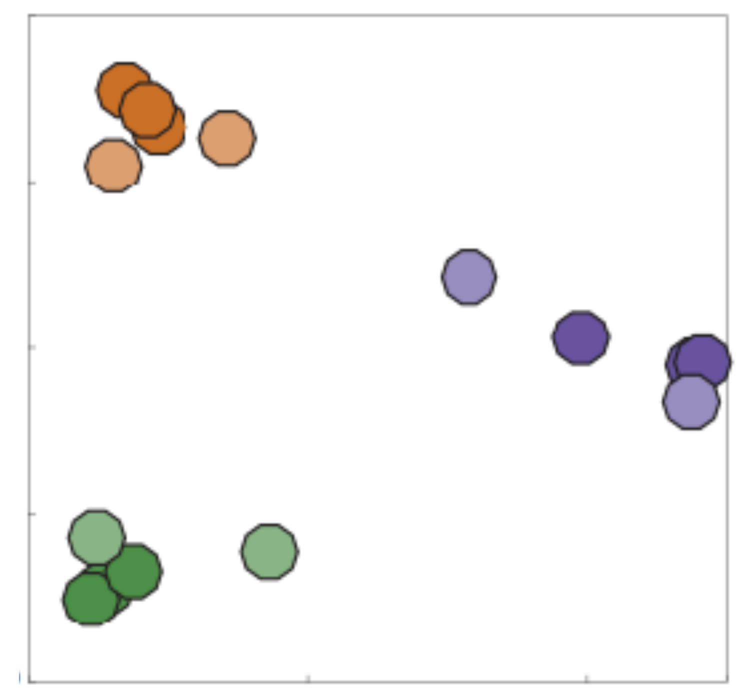


Bilateral hippocampus
MDS



Simulation

Successor representation
MDS

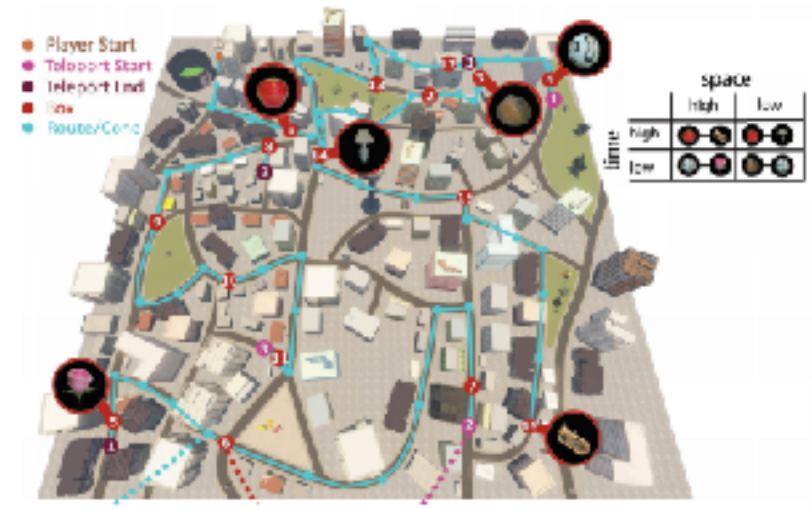


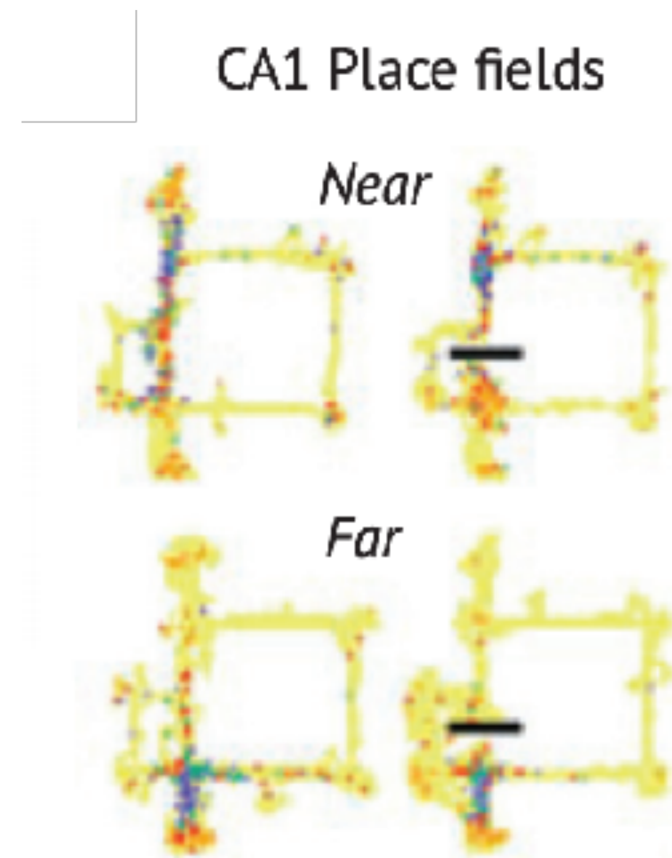
Schapiro et al. (2013)

Stachenfeld, Botvinick, Gershman (2017)

And more!

- Mixed spatiotemporal distance effects (Decker et al 2016)
- Obstacle-induced effects (Alvernhe et al 2011)
- Some aspects of goal-related firing (Hollup et al 2001)
- For more successor representation experiments, see also: Russek et al (2017), Momennejad et al (2017), Garvert et al (2017)





Stachenfeld, Botvinick, Gershman (2017)

Why is this useful?

Why is this useful?

- Model defined with respect to general variables like state + temporal adjacency

Why is this useful?

- Model defined with respect to general variables like state + temporal adjacency
- Learnable (Dayan 1993) + Scalable (Kulkarni et al 2018)

Why is this useful?

- Model defined with respect to general variables like state + temporal adjacency
- Learnable (Dayan 1993) + Scalable (Kulkarni et al 2018)
- Flexible computation of value when reward changes* because transition information is preserved in the successor representation

Why is this useful?

- Model defined with respect to general variables like state + temporal adjacency
- Learnable (Dayan 1993) + Scalable (Kulkarni et al 2018)
- Flexible computation of value when reward changes* because transition information is preserved in the successor representation
- * **Caveat**: still not as flexible as model-based planning because predictions are averaged over different possible futures

$$V = M\,R$$

successor representation matrix

reward

Instantaneous reward

Caches long-term effect of dynamics

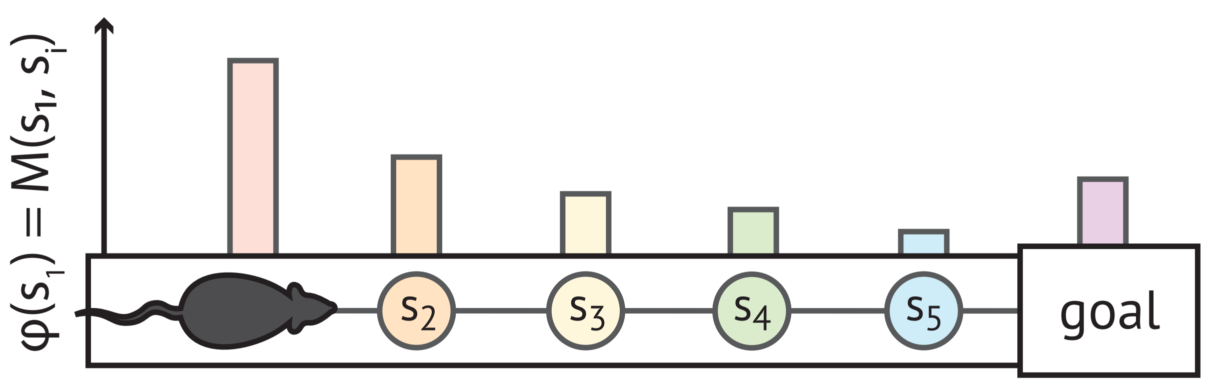
What do we need in a representation?

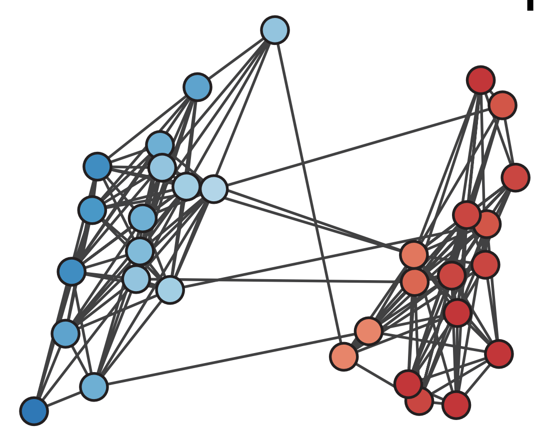
What additional structure, besides reward, should we learn?

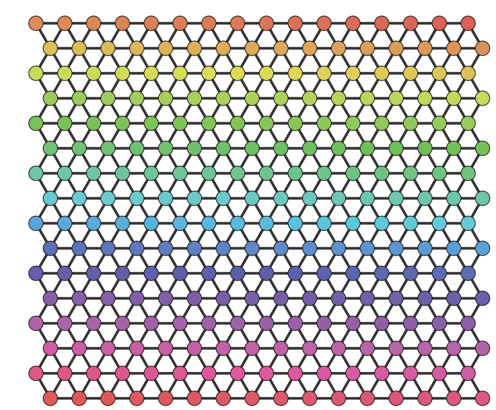
How can we make it so the downstream RL process has fewer things to learn about?

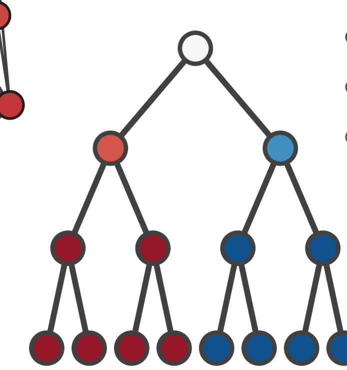
Predictions
Represent states in terms of the predictions they make

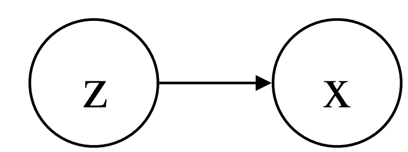
Compression
Learn as short description as possible

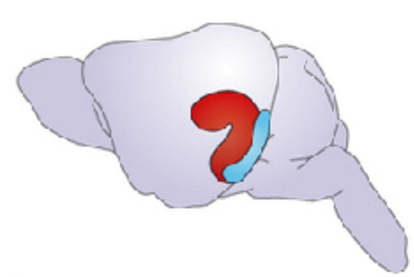












Place Cells

For each state, learn how often you will visit other states?

Grid Cells

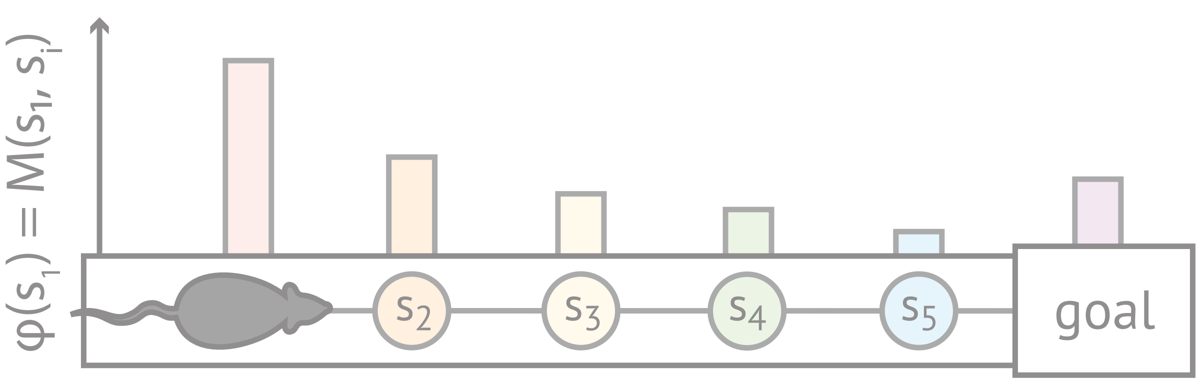
Capture low-dimensional structure among predictions

Stachenfeld, Botvinick, Gershman (2017)

What do we need in a representation?

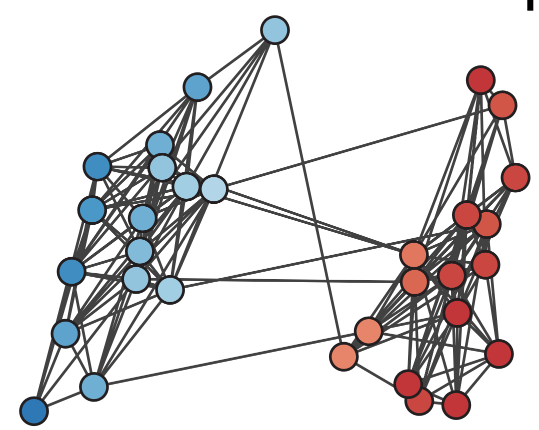
What additional structure, besides reward, should we learn?

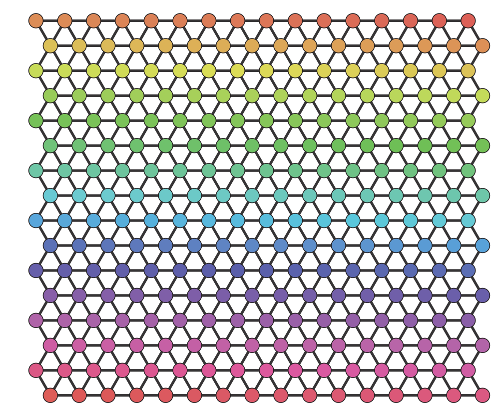
Predictions
Represent states in terms of the predictions they make

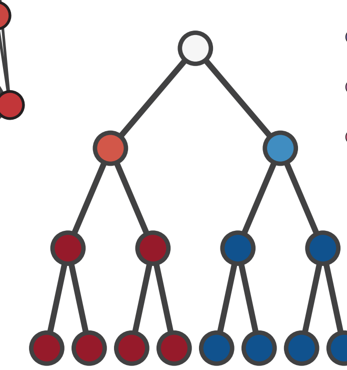


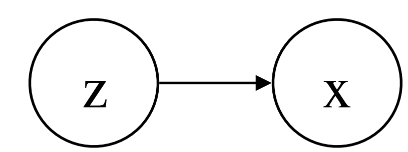
How can we make it so the downstream RL process has fewer things to learn about?

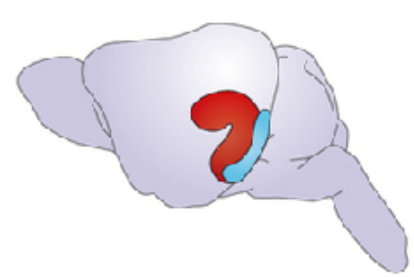
Compression
Learn as short description as possible











Place Cells

For each state, learn how often you will visit other states?

Grid Cells

Capture low-dimensional structure among predictions

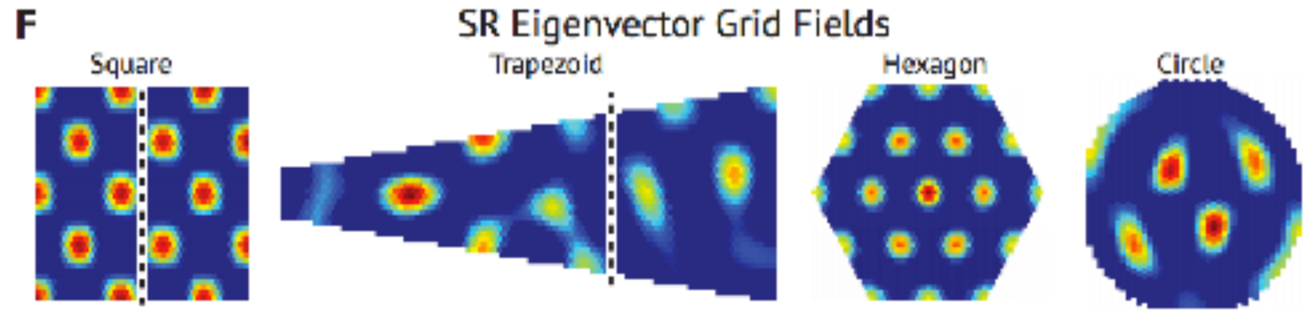
Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

Stachenfeld, Botvinick, Gershman (2017)

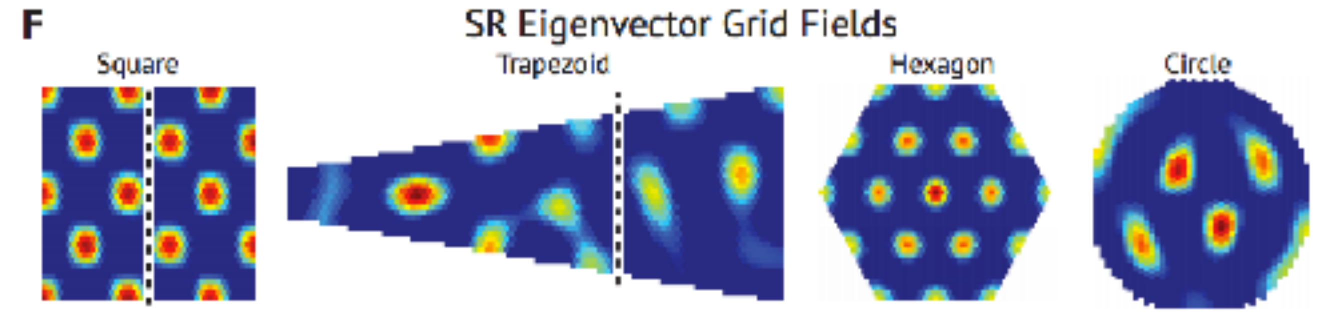
Simulating Grid Cells

Eigenvectors of successor representation



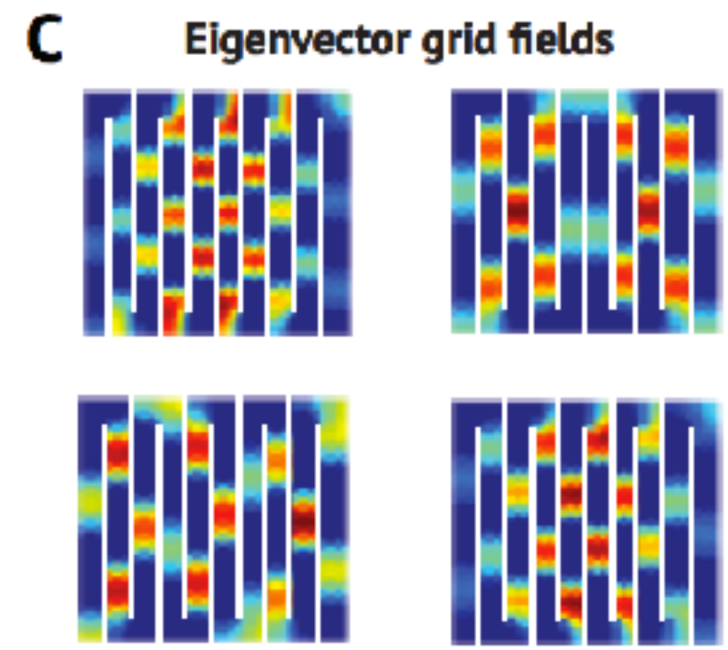
Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells



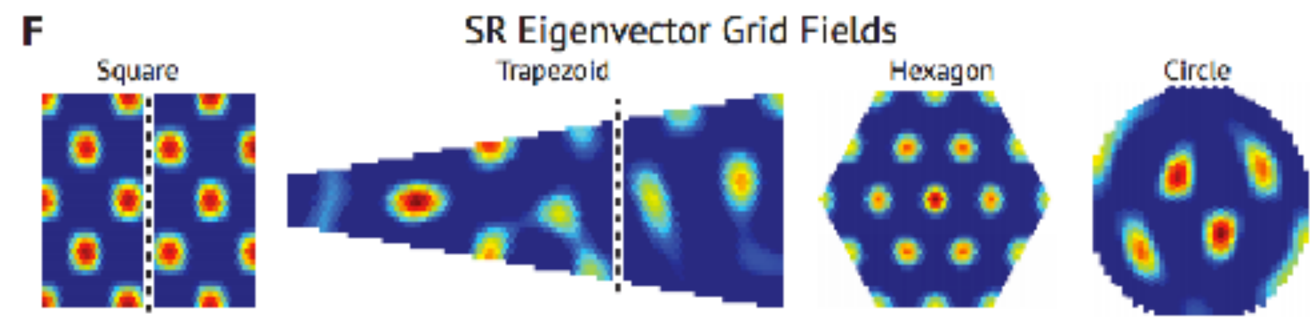
Eigenvectors of successor representation

- Grid fields in geometric environments
  Krupic et al. (2013)



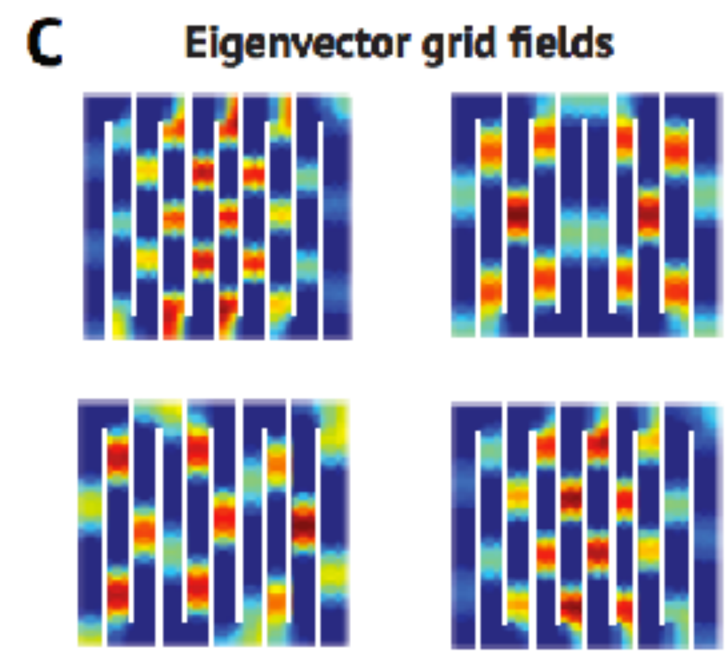
Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

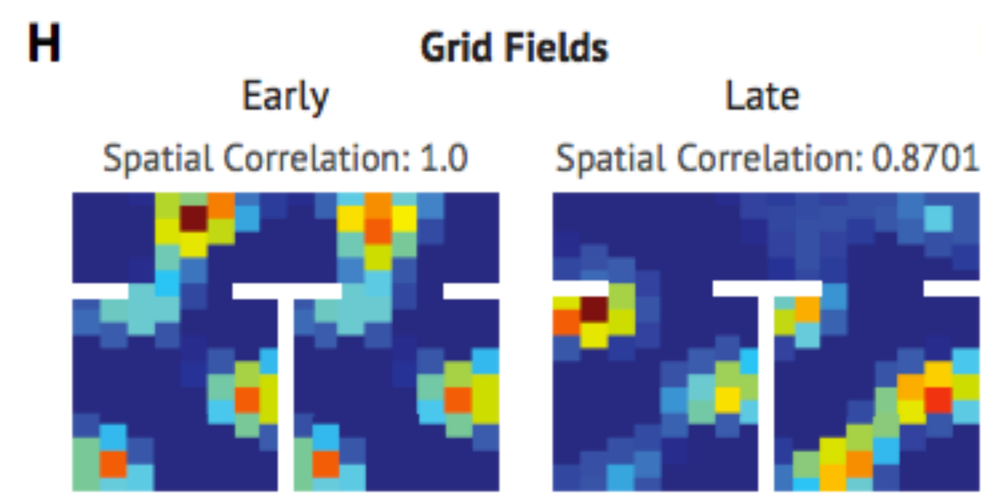


Eigenvectors of successor representation

- Grid fields in geometric environments
  Krupic et al. (2013)



- Fragmentation in Hairpin Maze
  Derdikman et al. (2009)



Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

Eigenvectors of successor representation

- Grid fields in geometric environments
  Krupic et al. (2013)

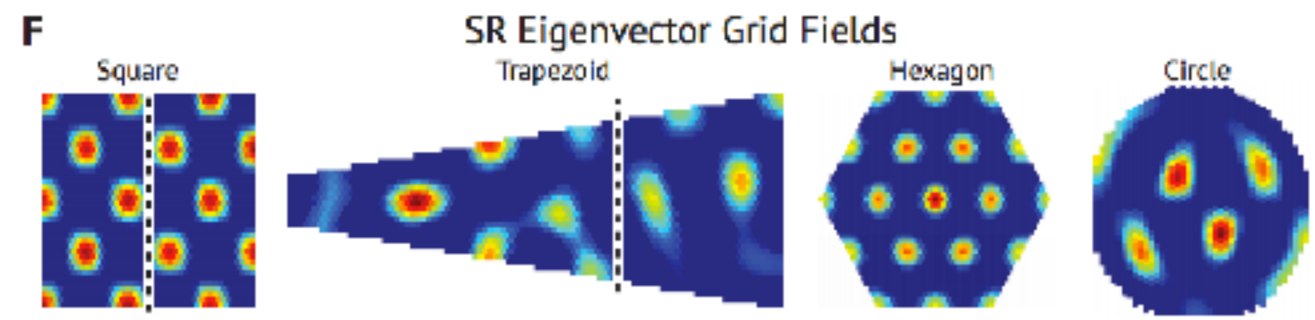
- Fragmentation in Hairpin Maze
  Derdikman et al. (2009)

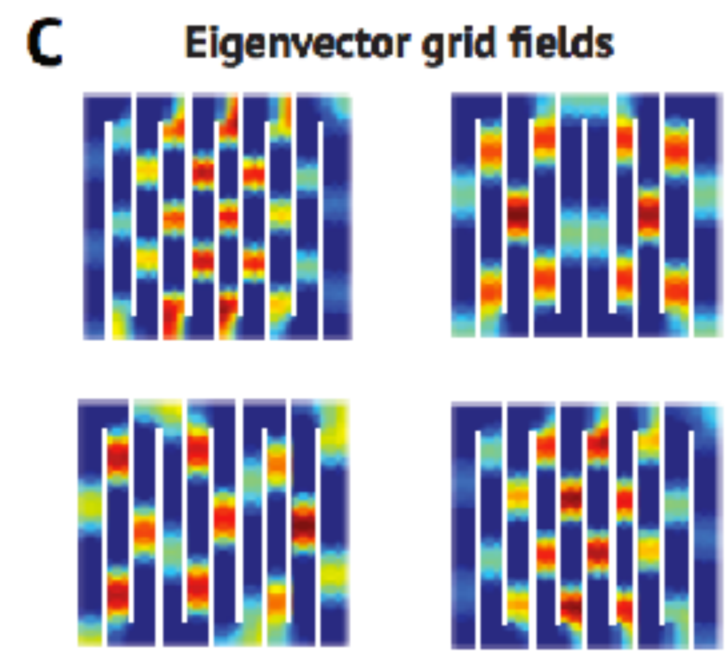
- Grid field learning in a multicompartment environment
  Carpenter et al. (2017)

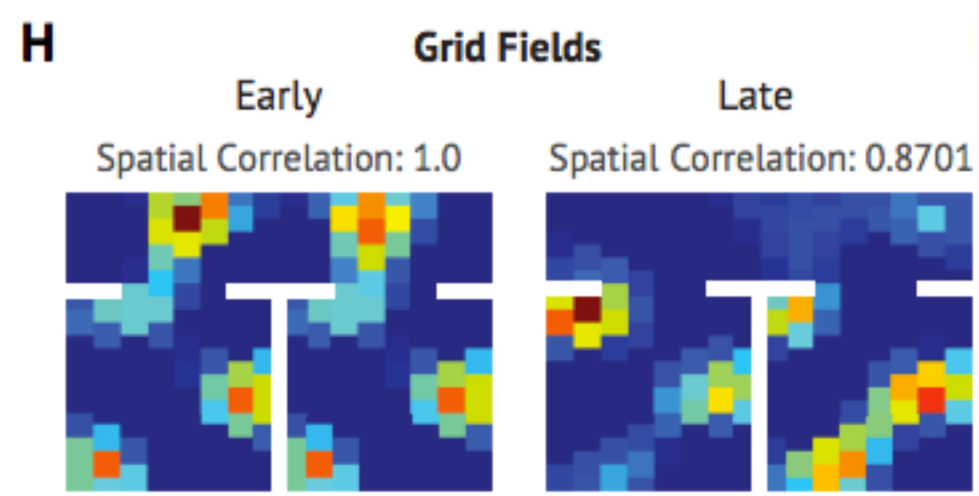
Takeaway: we capture many aspects of how environmental geometry and topology are known to affect grid cells.

Related approach to other spectral methods, slow feature analysis

See also: Dordek et al 2016, Banino et al 2018, Sorscher et al 2019, Whittington et al 2019, Mok & Love 2019

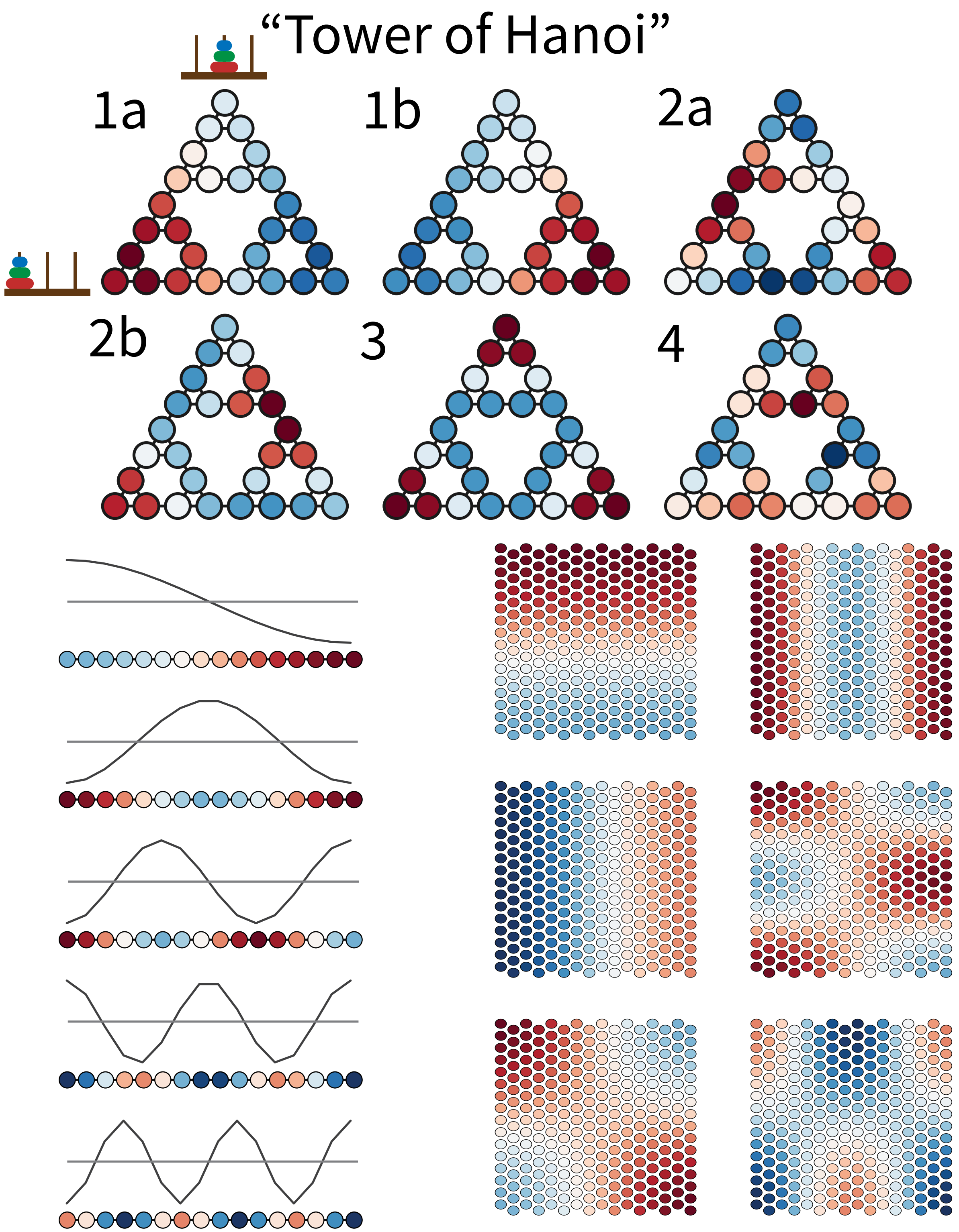






Stachenfeld, Botvinick, Gershman (2017)

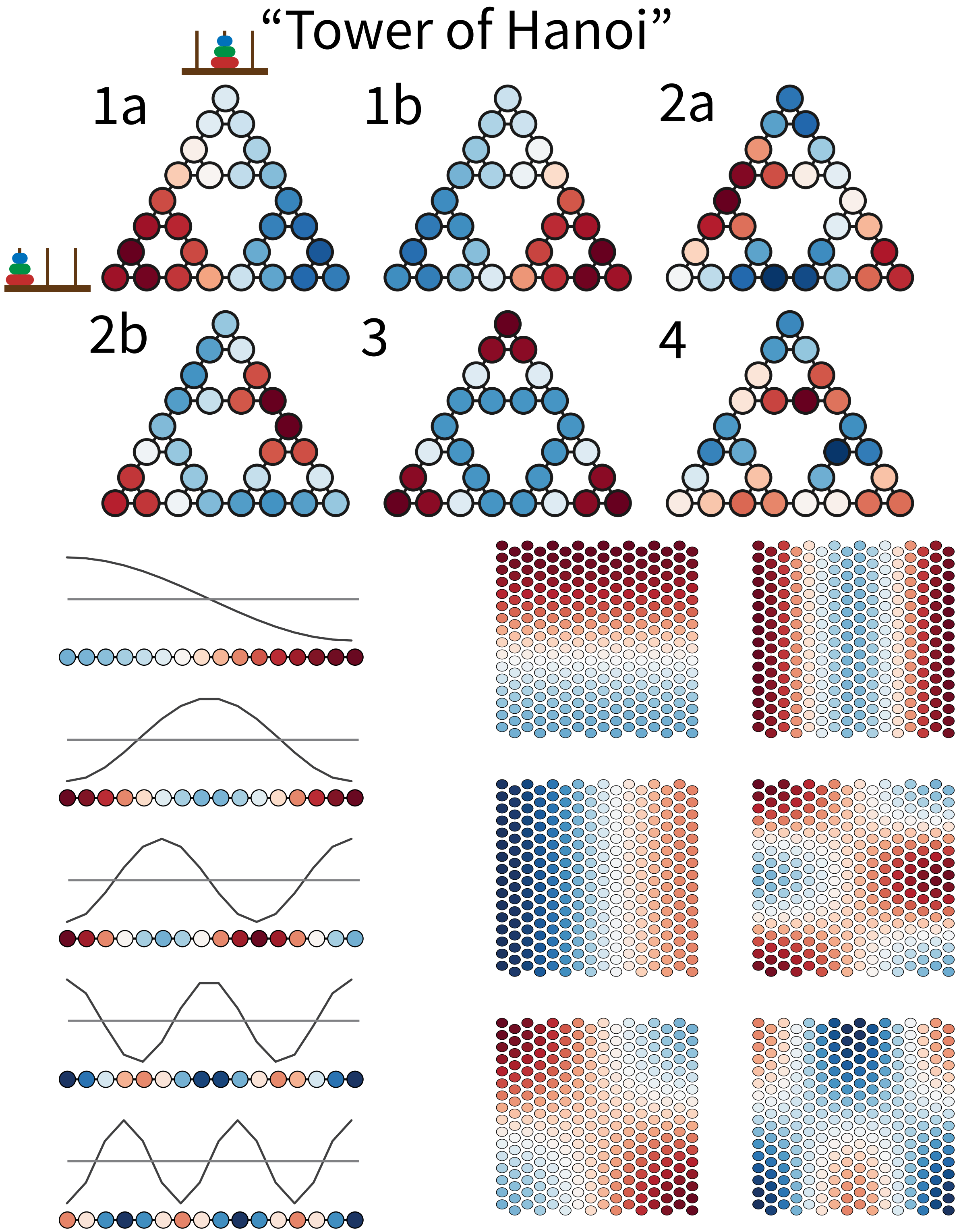
Eigenvectors have a lot of nice applications



Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

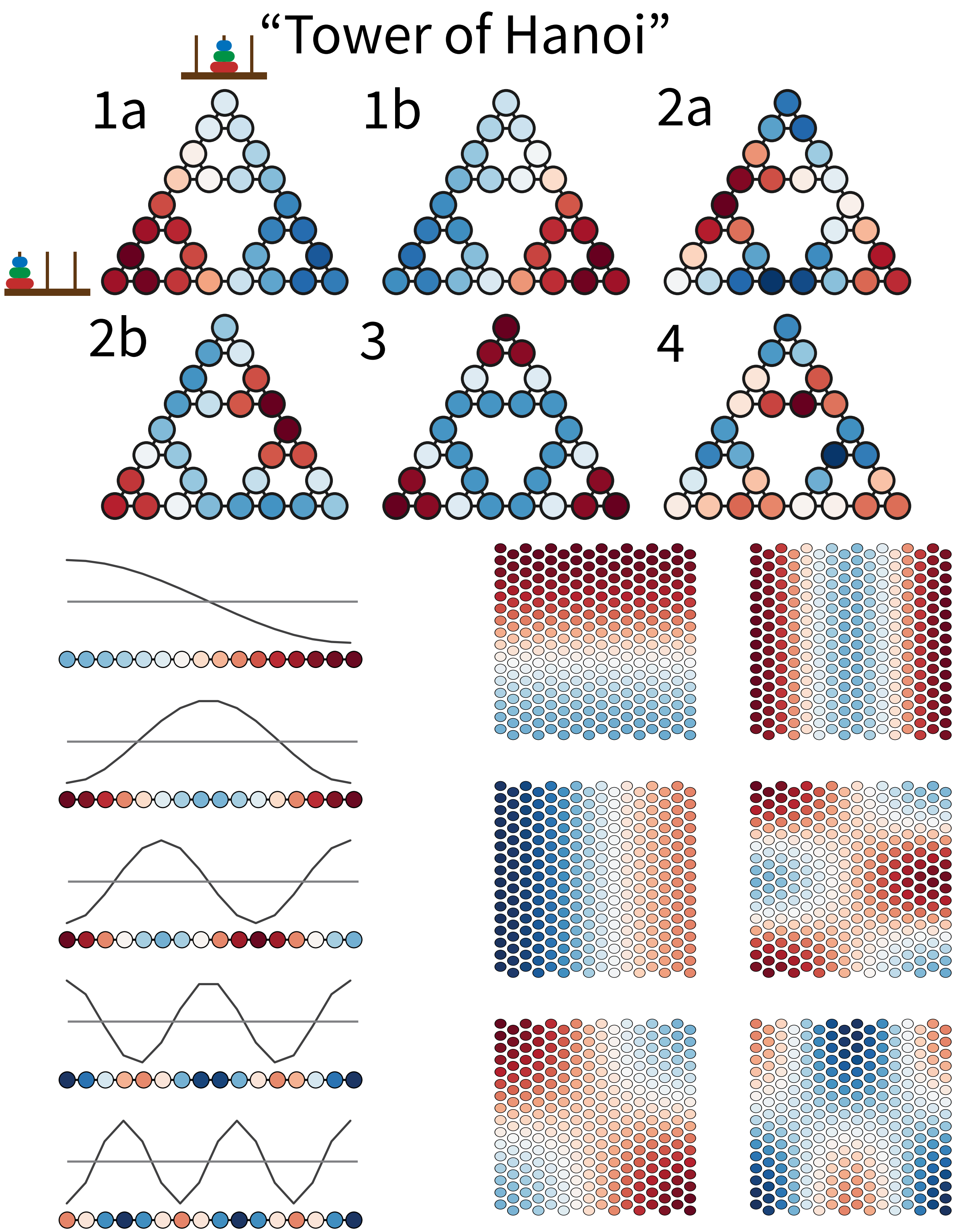
- **For compression — sort of like PCA over transition structure**



Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

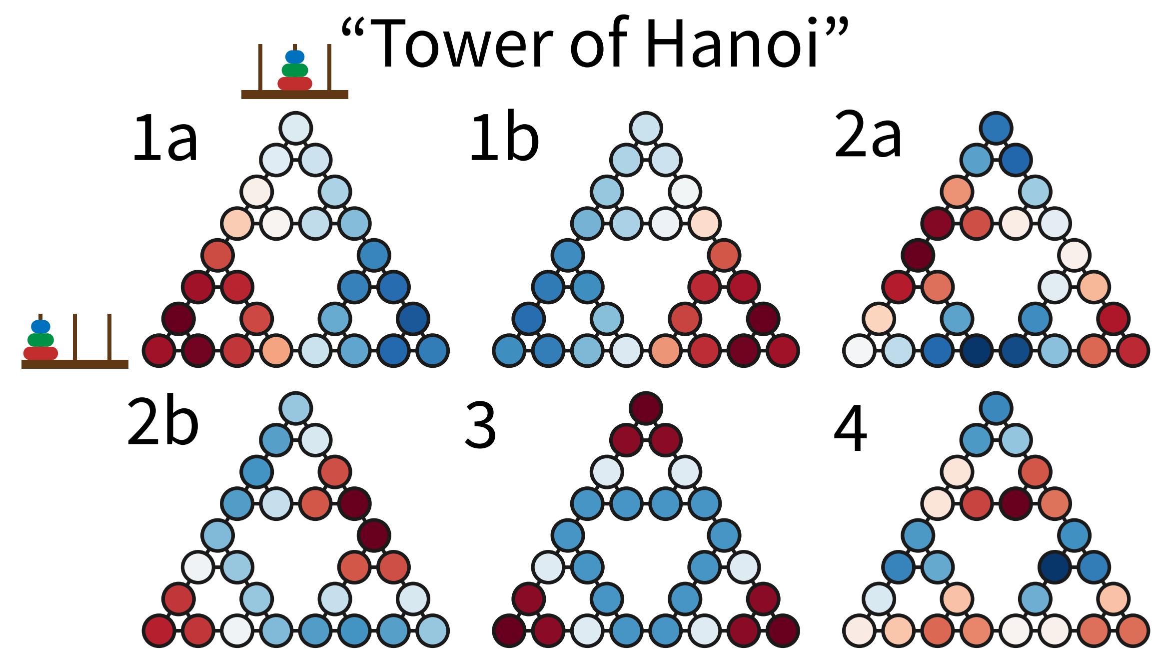
- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**

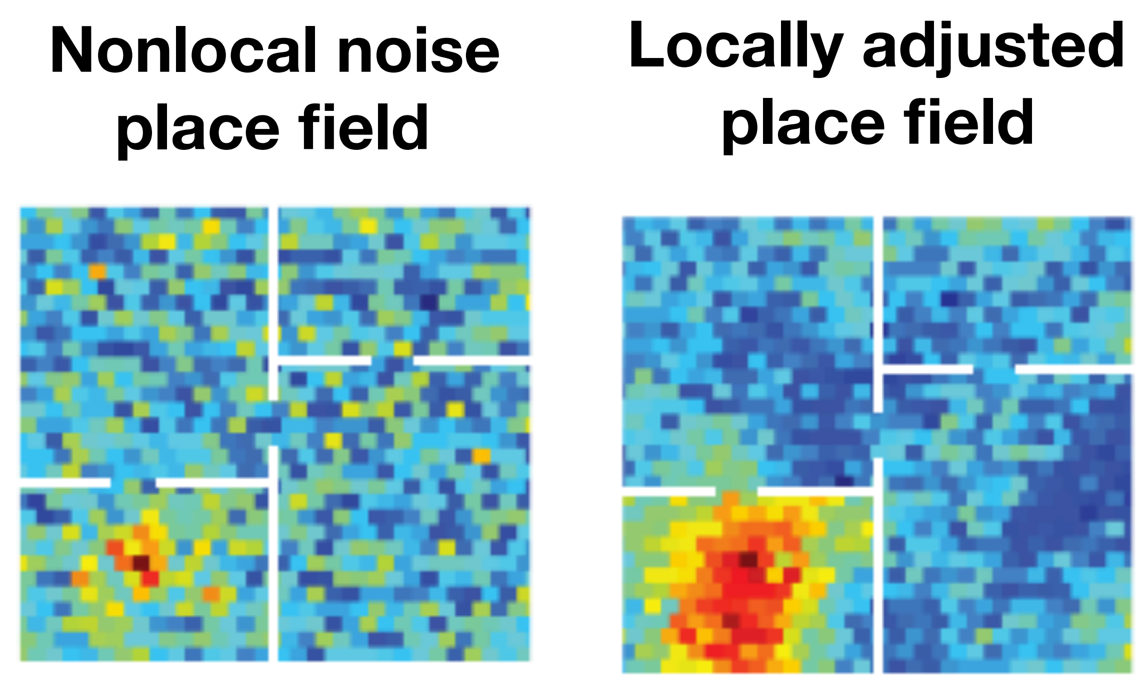


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**

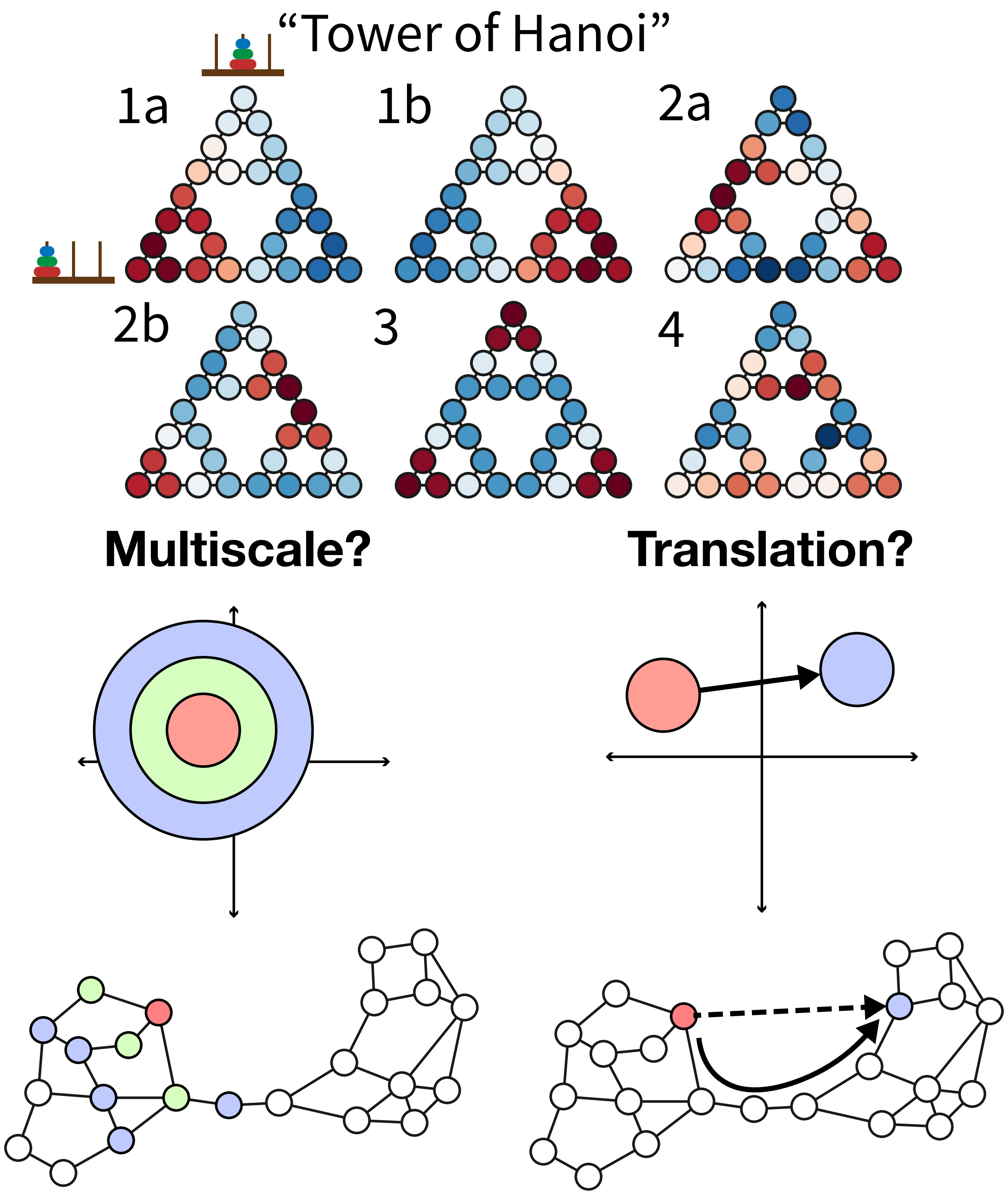




Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

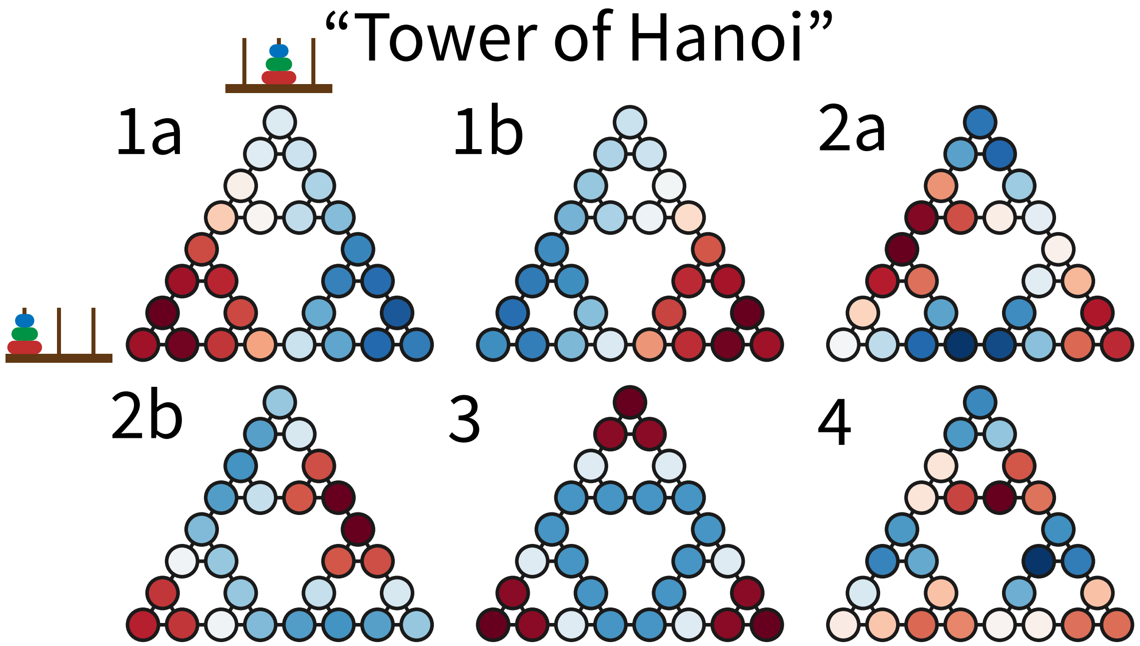
- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**
- **For "geometry on graphs"** Generalize convolution and spatial features to arbitrary geometries

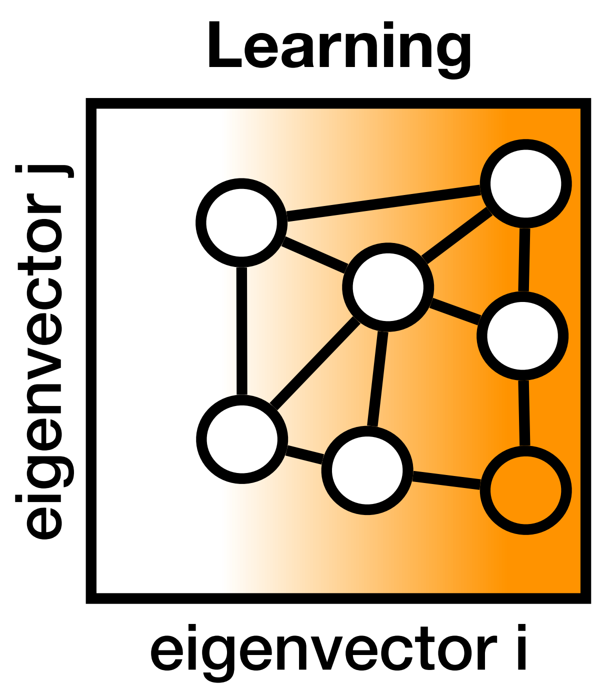


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**
- **For "geometry on graphs"** Generalize convolution and spatial features to arbitrary geometries
- **For supporting learning:** "Protovalue functions" (Mahadevan (2007))

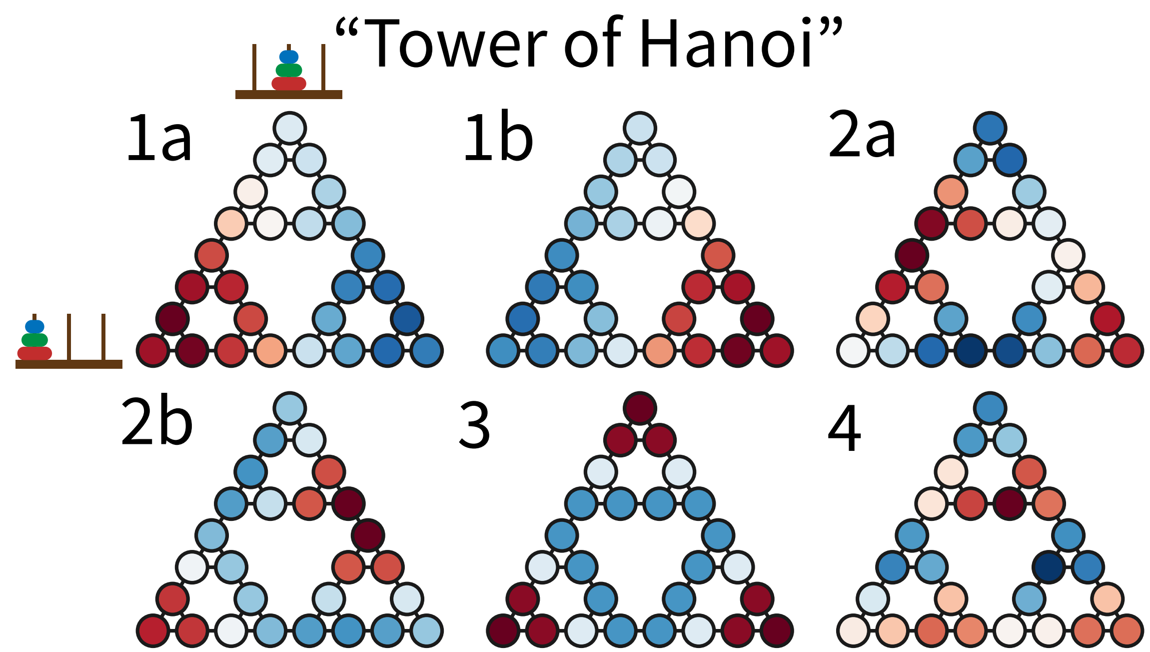


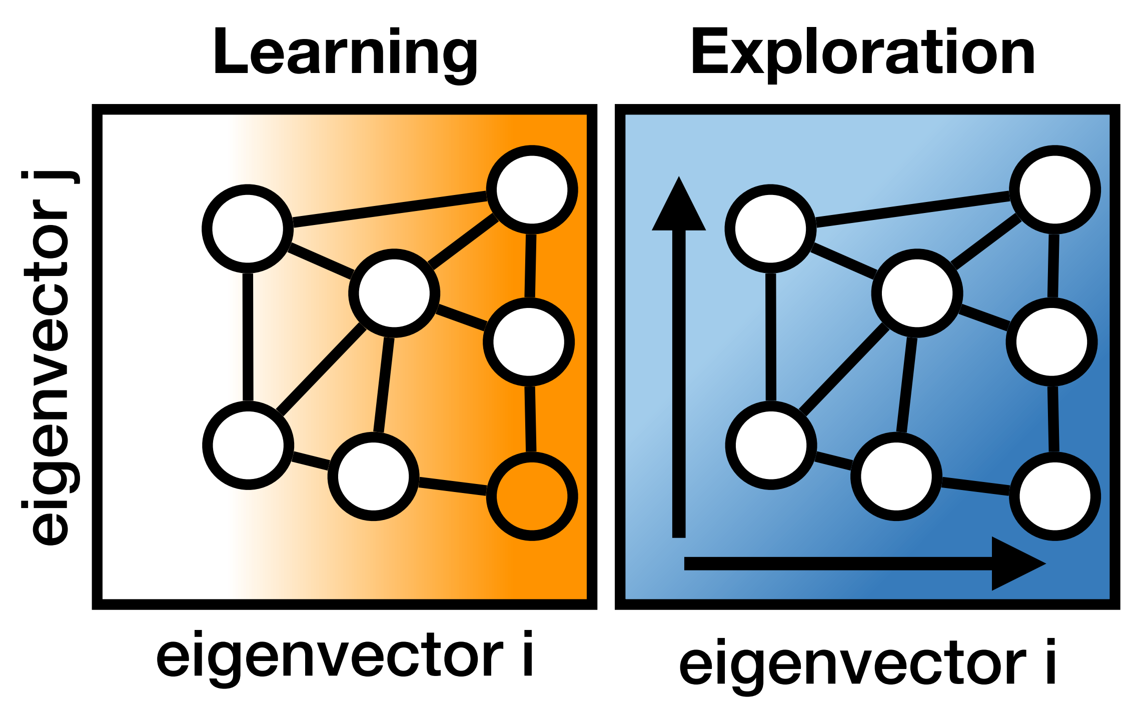


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**
- **For "geometry on graphs"** Generalize convolution and spatial features to arbitrary geometries
- **For supporting learning:** "Protovalue functions" (Mahadevan (2007))
- **For supporting exploration:** "Eigenoptions" Machado et al (2017a,b—2018)

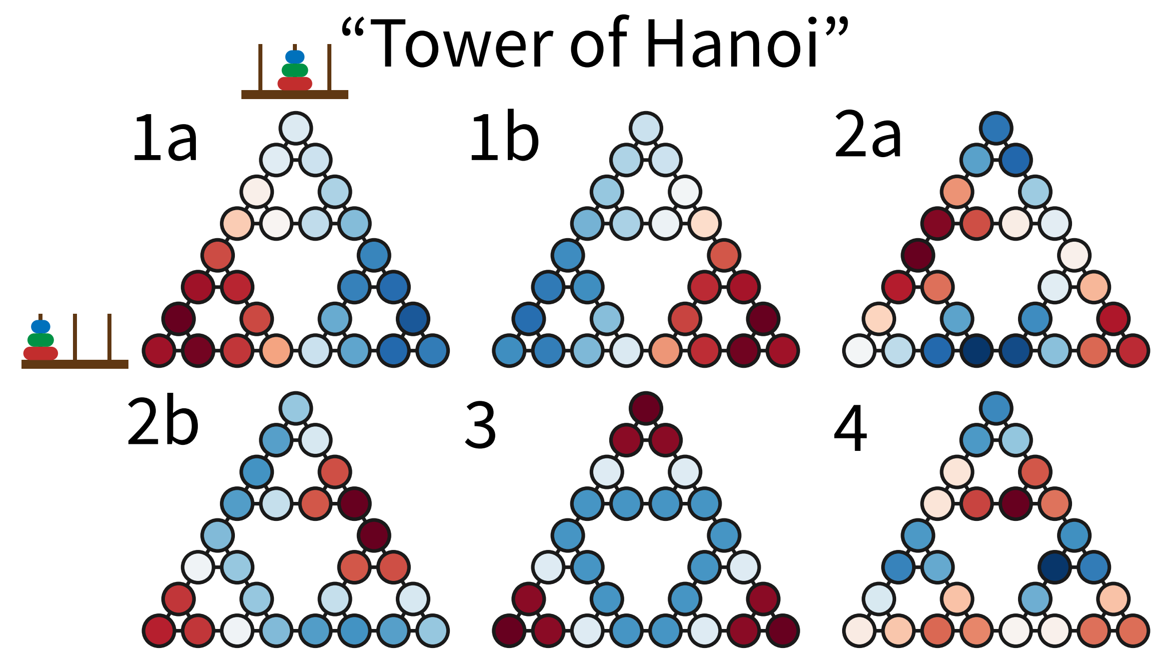


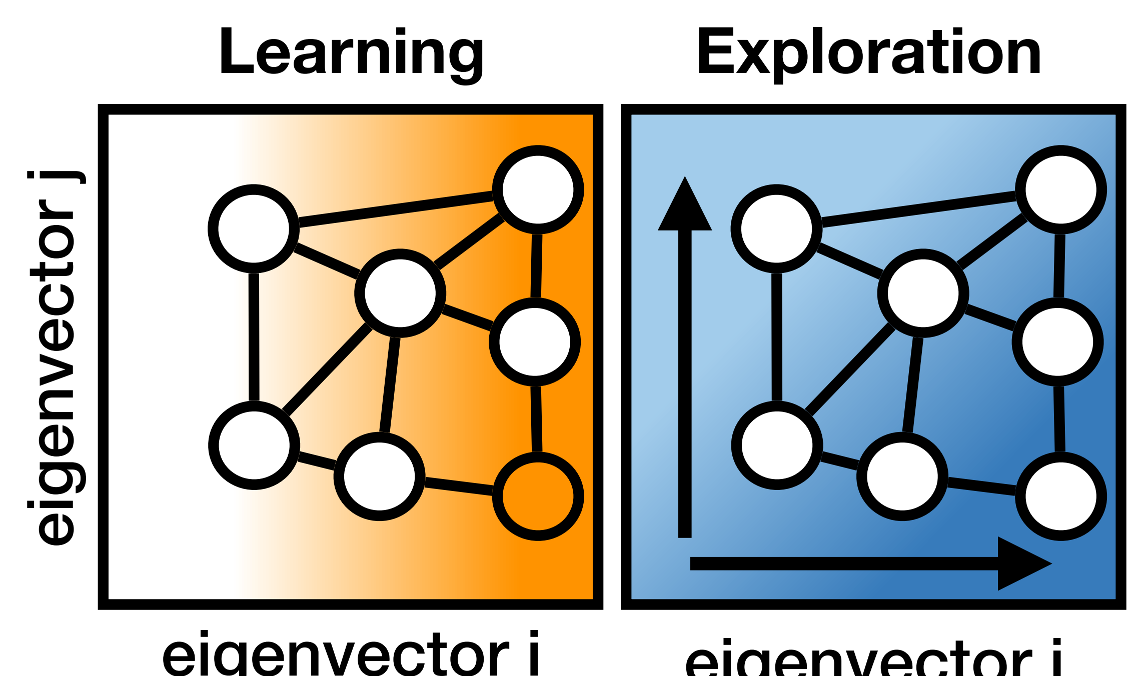


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**
- **For "geometry on graphs"** Generalize convolution and spatial features to arbitrary geometries
- **For supporting learning:** "Protovalue functions" (Mahadevan (2007))
- **For supporting exploration:** "Eigenoptions" Machado et al (2017a,b—2018)
- **For multiscale/hierarchical learning + exploration**

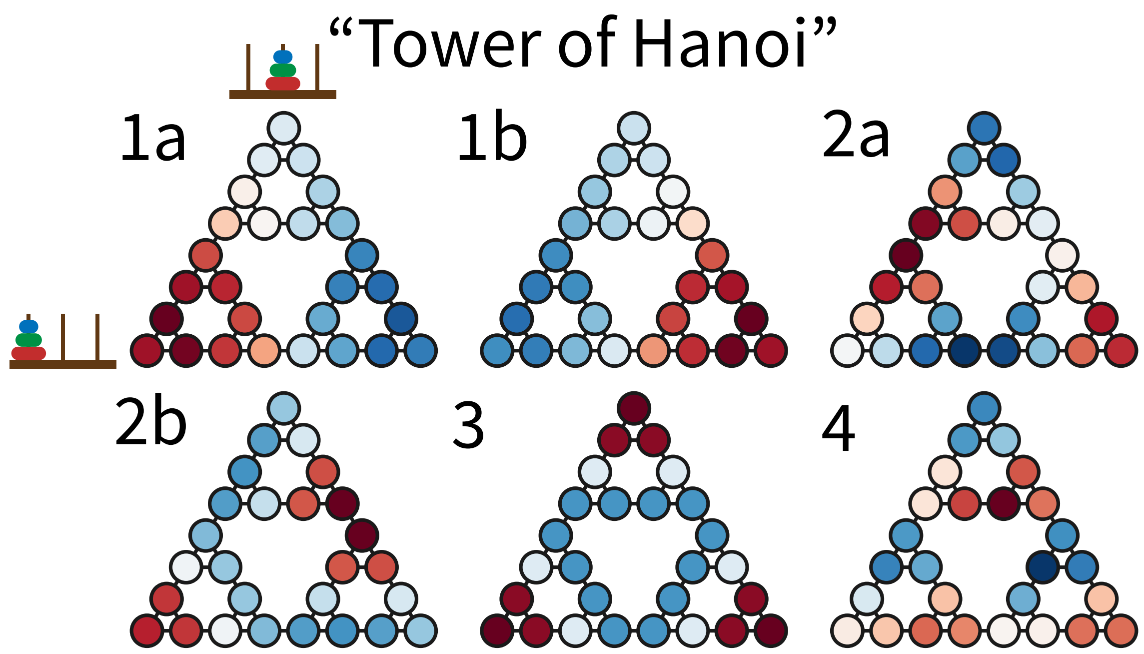


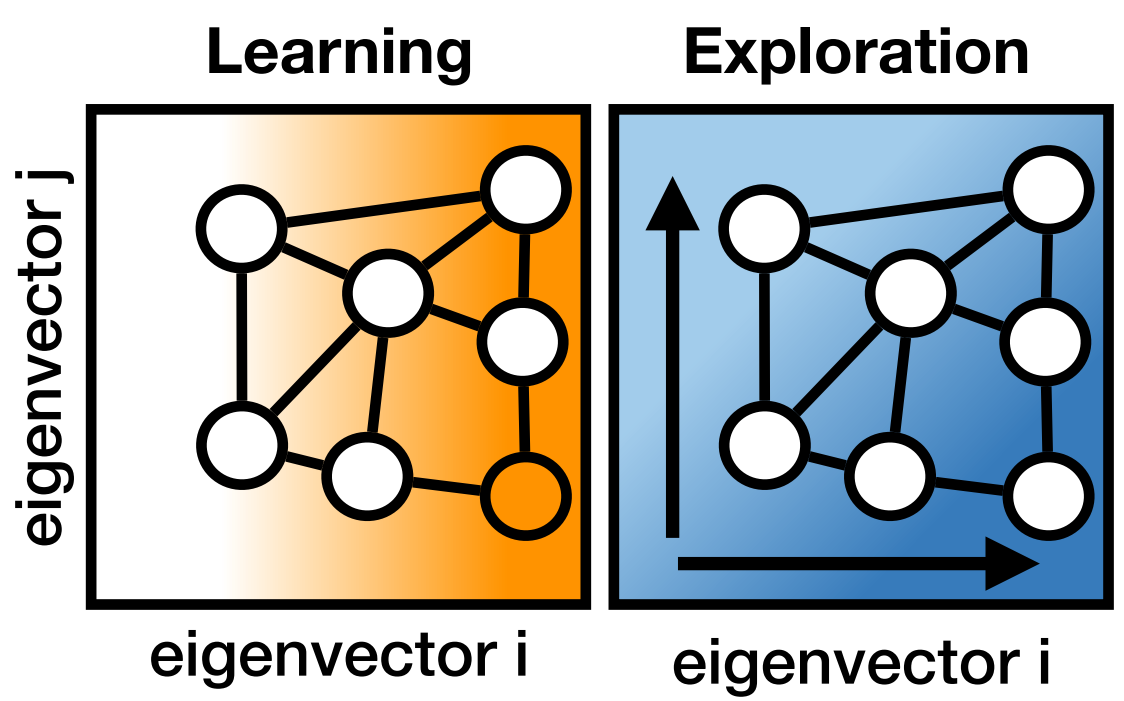


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Eigenvectors have a lot of nice applications

- **For compression — sort of like PCA over transition structure**
- **Generalization of Fourier to graphs**
- **Noise correction + filling in missing data**
- **For "geometry on graphs"** Generalize convolution and spatial features to arbitrary geometries
- **For supporting learning:** "Protovalue functions" (Mahadevan (2007))
- **For supporting exploration:** "Eigenoptions" Machado et al (2017a,b—2018)
- **For multiscale/hierarchical learning + exploration**
- **Graph Convolutional Neural Nets** Bruna et al 2013



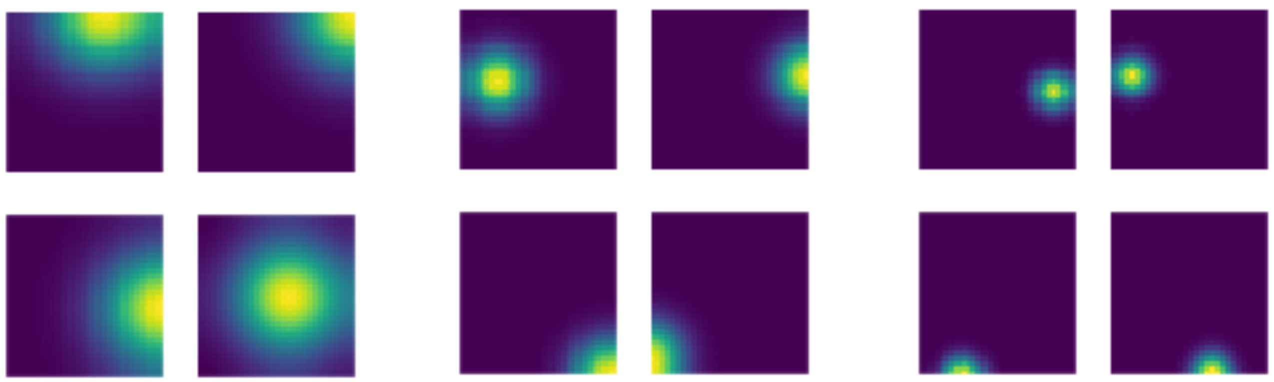


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

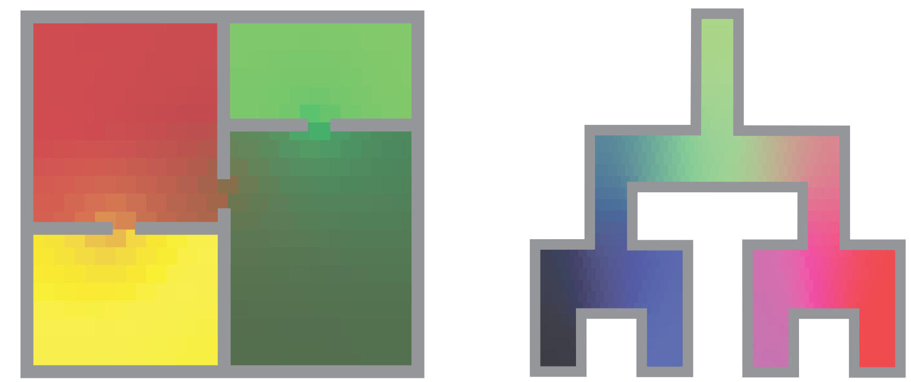
Hierarchical Learning + Exploration

Dan McNamee

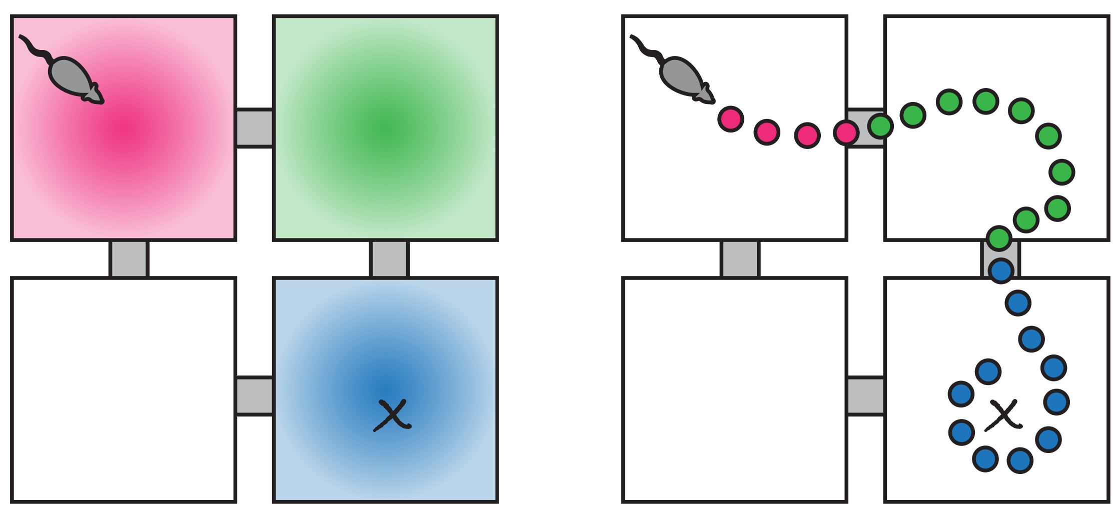
Multi-scale



Detect structural boundaries



Supports hierarchical learning + planning

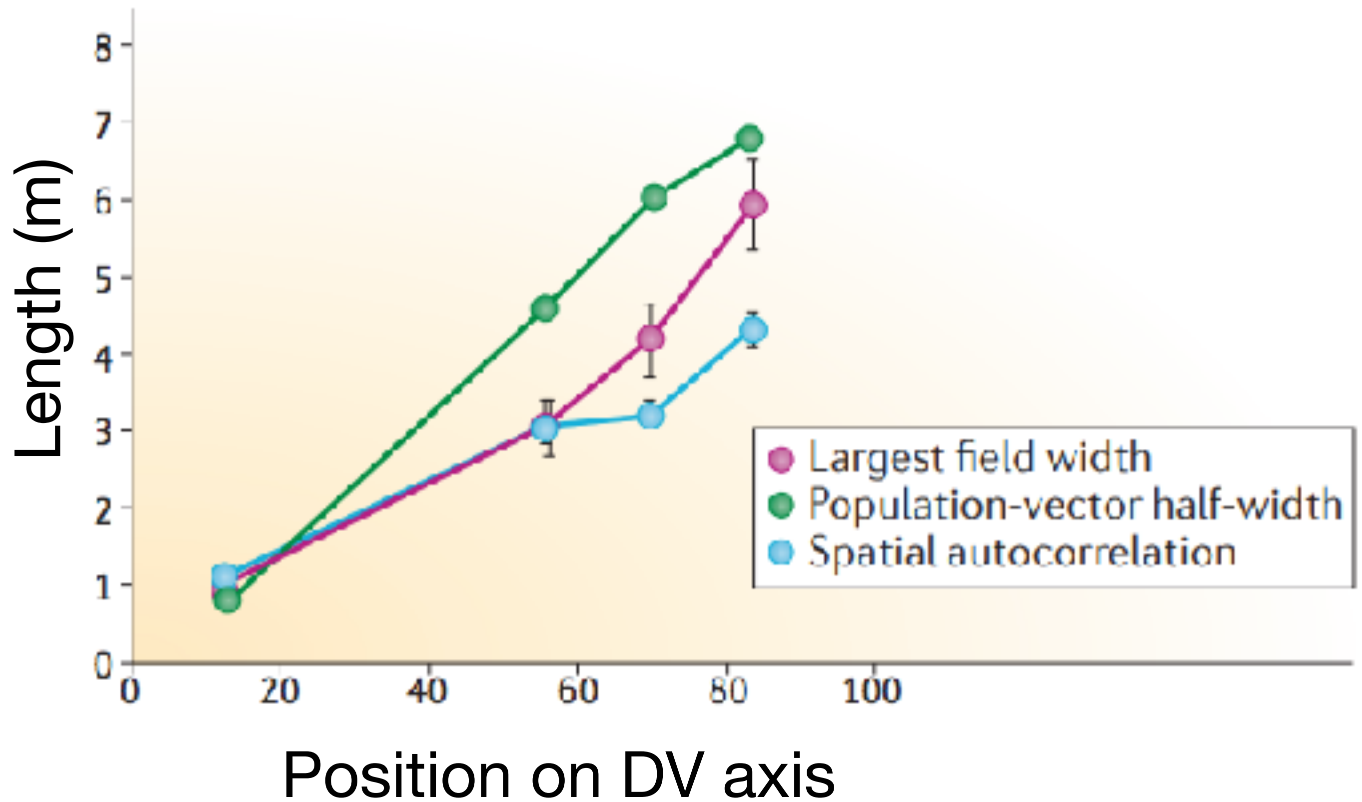


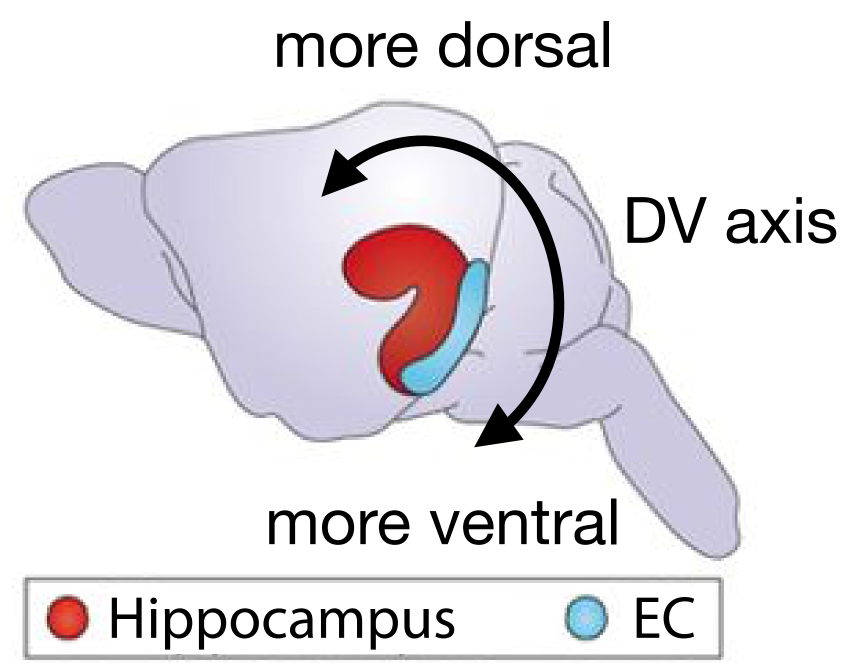
High Level

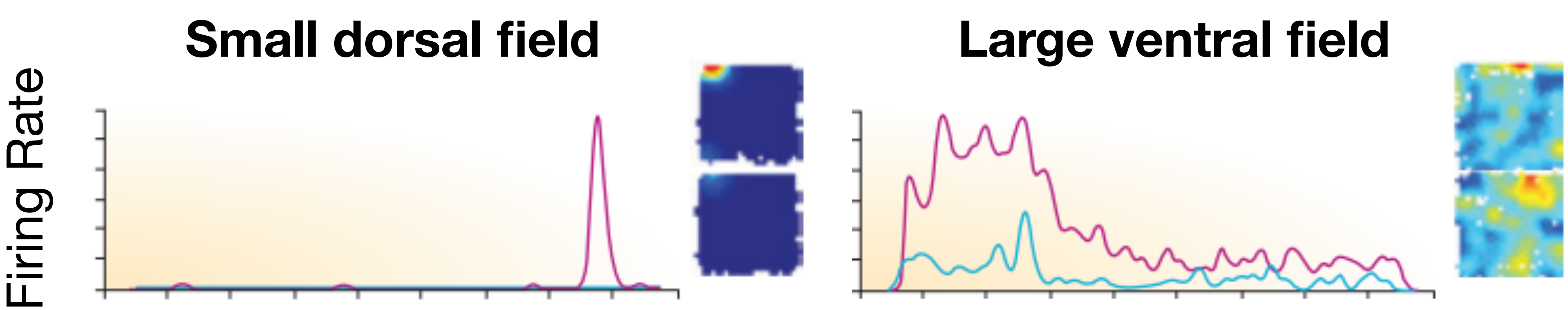
Low Level

Multi-scale representations

Range of spatial scales along longitudinal axis of hippocampus + MEC



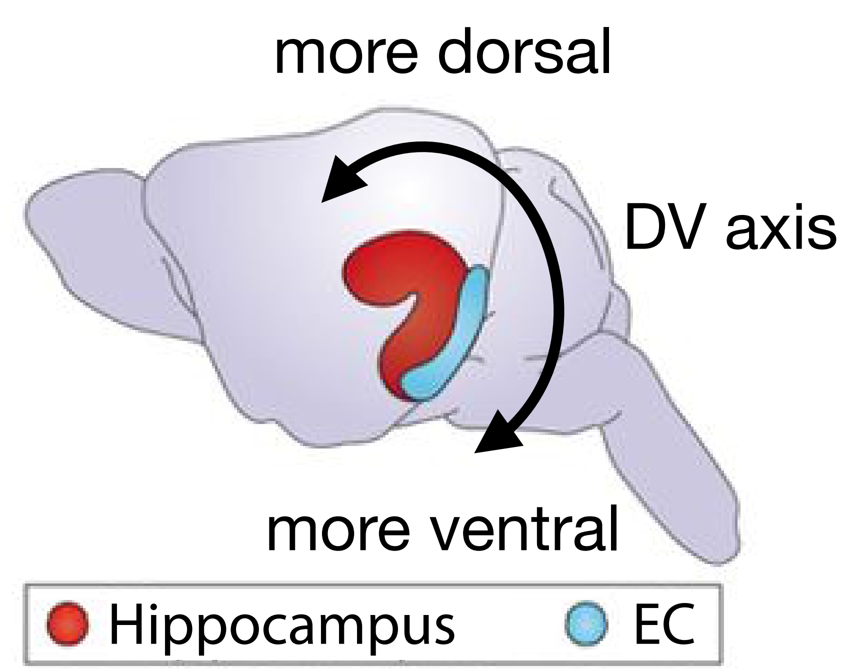




Review by Strange et al (2014)

Multi-scale representations

Same eigenvectors support different scales

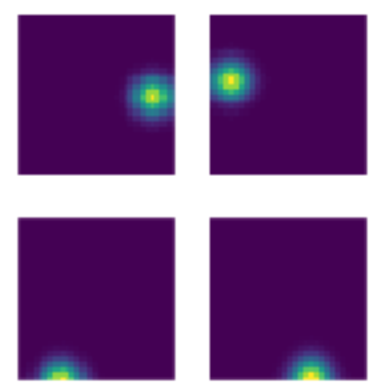


Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

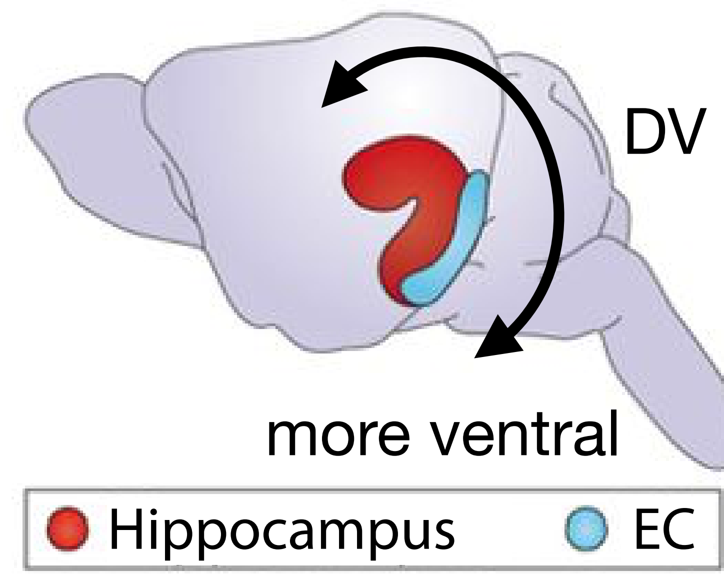
Multi-scale representations

Same eigenvectors support different scales

Simulated place fields with different scales



more dorsal
smaller discount factor

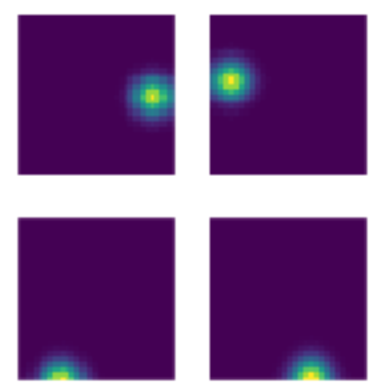


Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

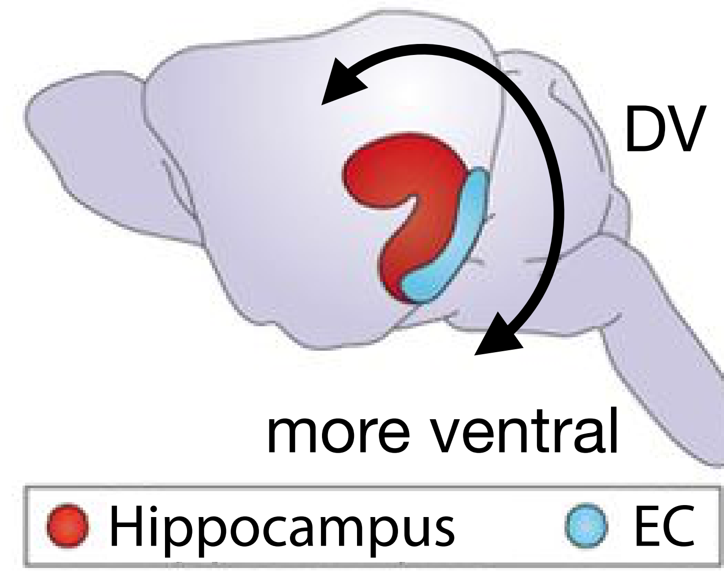
Multi-scale representations

Same eigenvectors support different scales

Simulated place fields with different scales

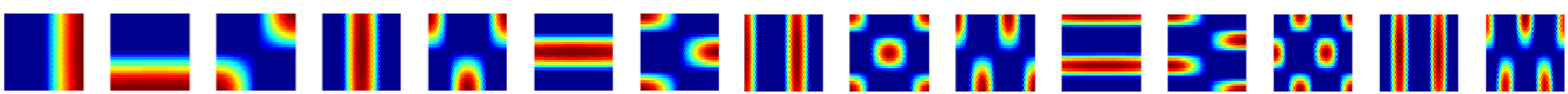


more dorsal
smaller discount factor



Simulated eigenvector grid fields

more ventral



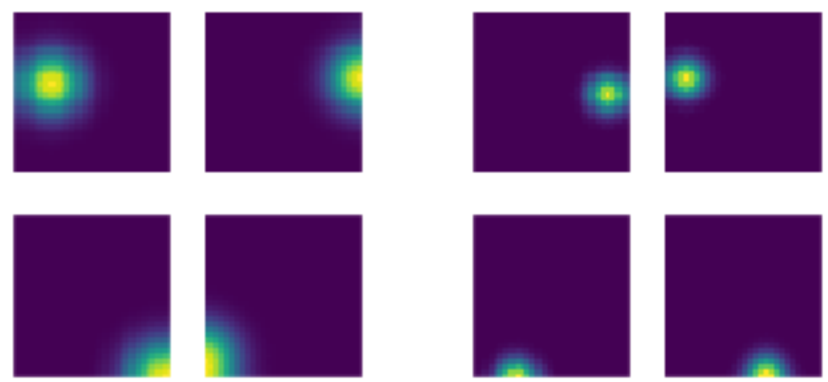
more dorsal

Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

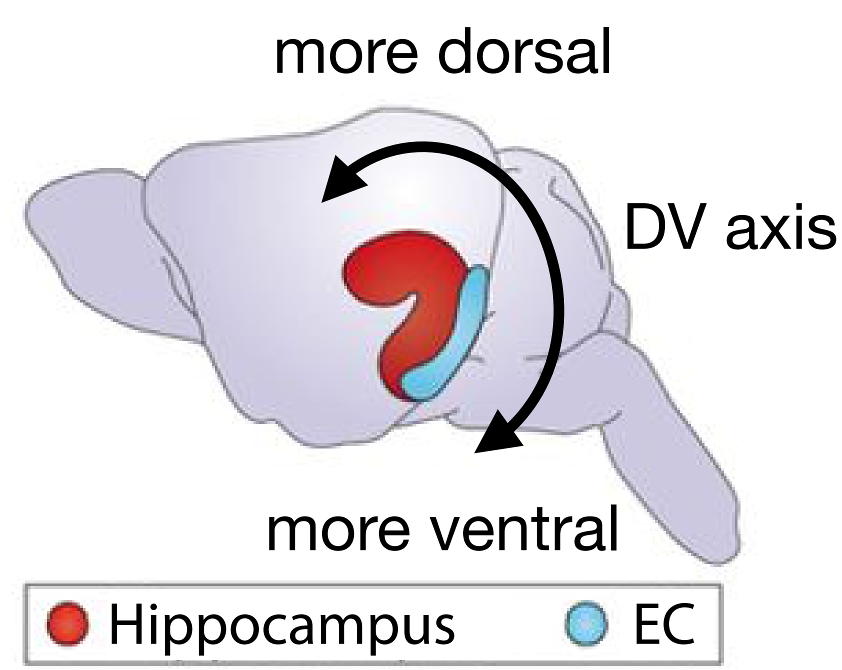
Multi-scale representations

Same eigenvectors support different scales

Simulated place fields with different scales

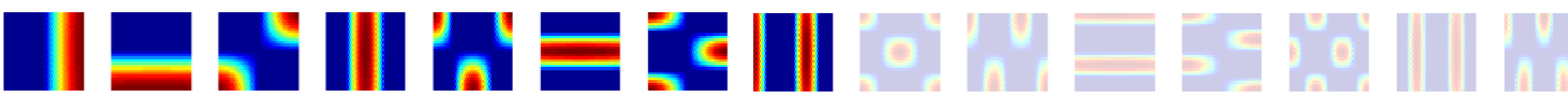


more dorsal
smaller discount factor



Simulated eigenvector grid fields

more ventral

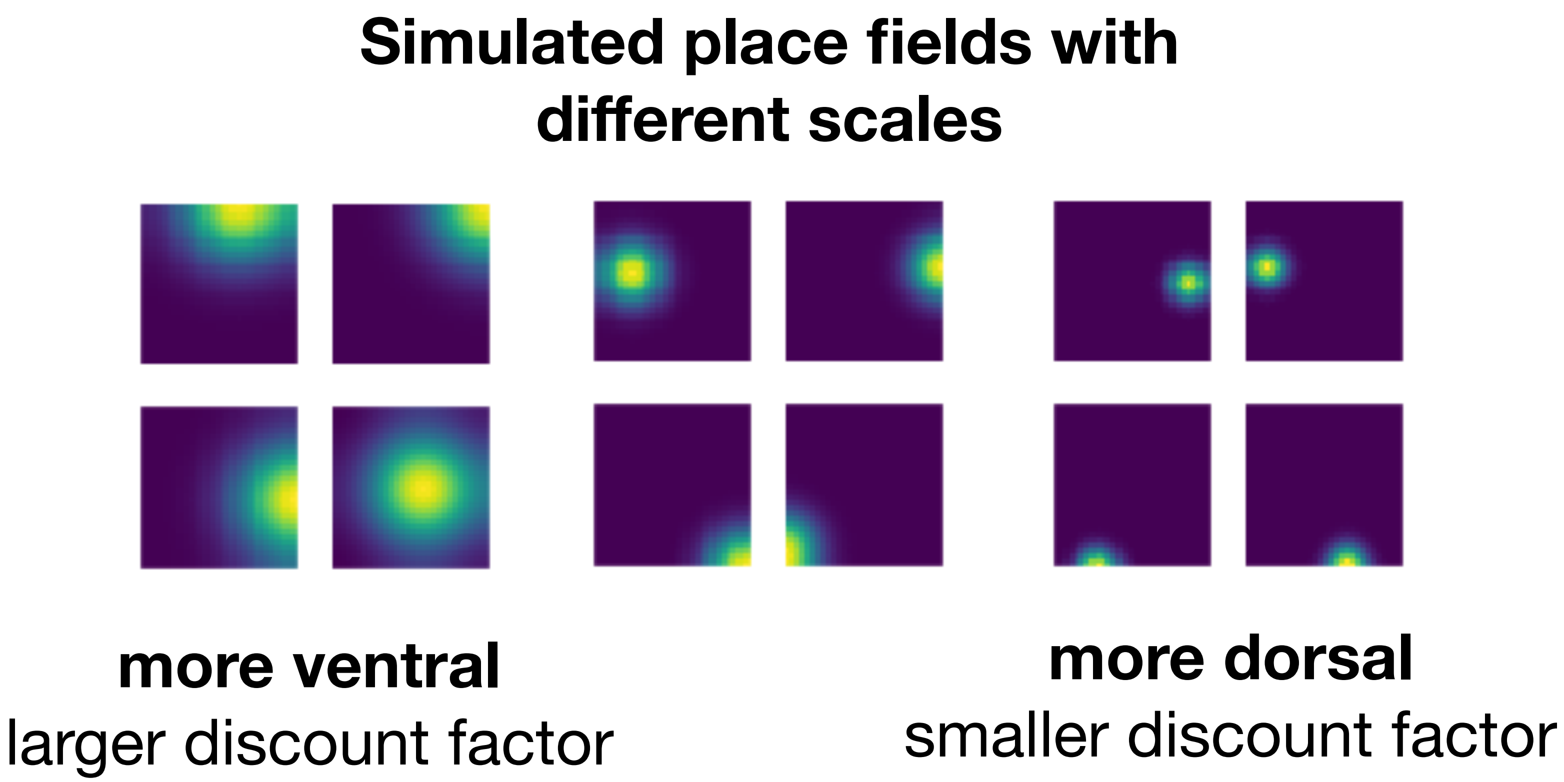


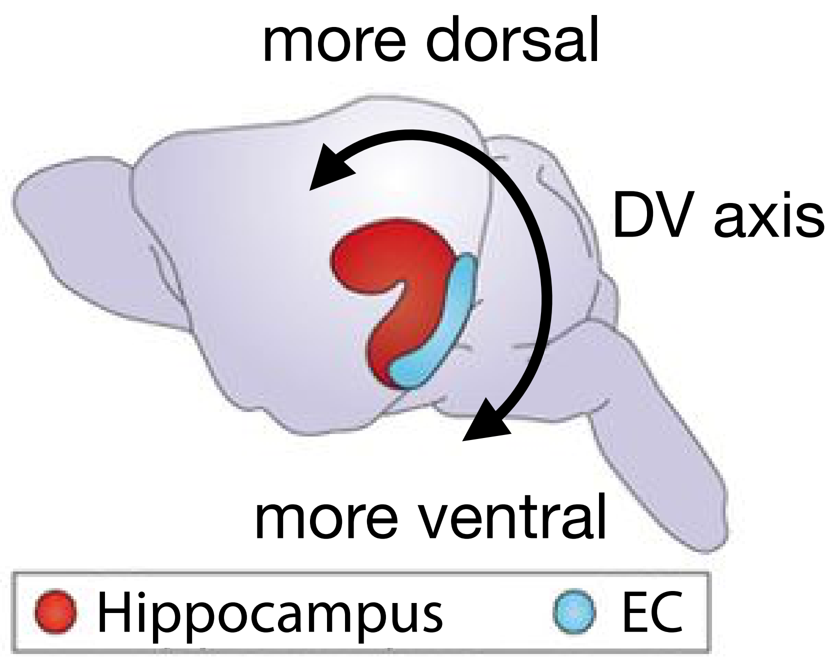
more dorsal

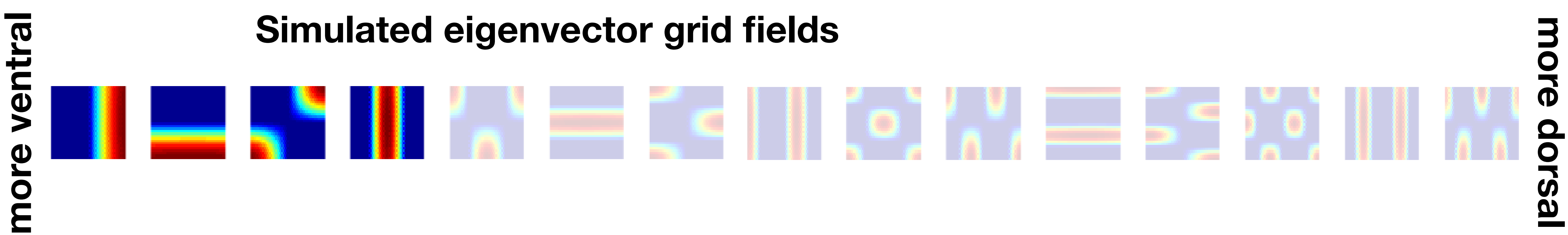
Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

Multi-scale representations

Same eigenvectors support different scales





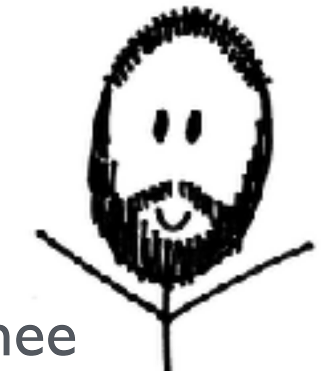


Stachenfeld, Botvinick, Gershman (2017)

also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

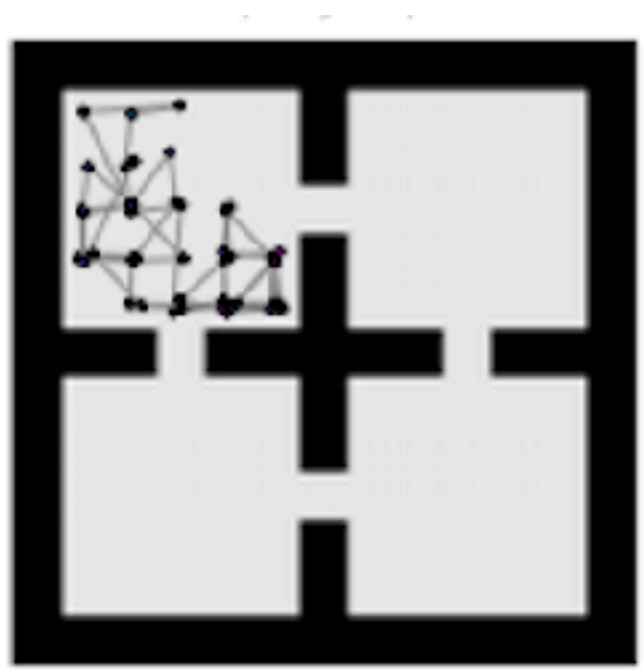
Sample with different statistics

Dan
McNamee



Random Walk

Agent gets stuck in room



McNamee, Stachenfeld, Botvinick, Gershman (in prep)

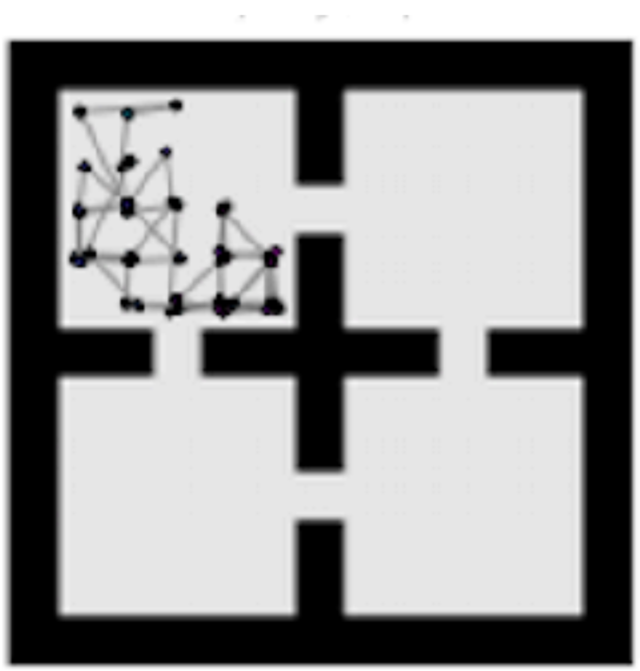
Sample with different statistics

Dan
McNamee

Random Walk

Levy Walk

Agent gets stuck in room

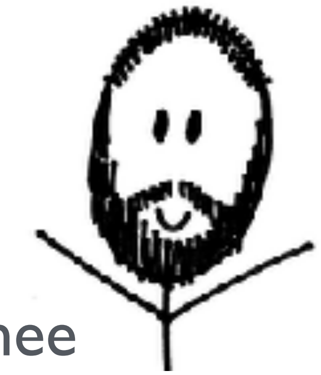


Agent frees itself from room, allowing it to find reward faster

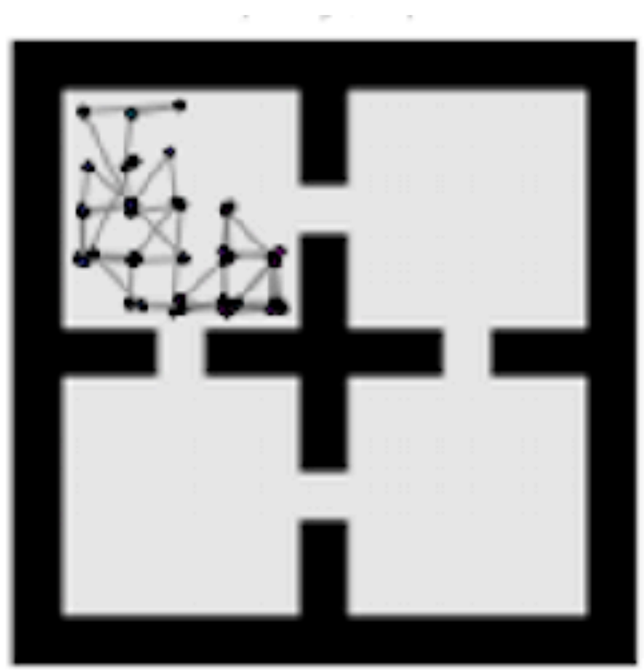
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Sample with different statistics

Dan
McNamee



Random Walk



Agent gets stuck in room

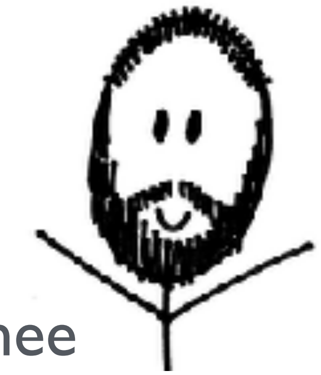
Levy Walk

Agent frees itself from room, allowing it to find reward faster

Levy walks experienced during foraging behavior

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

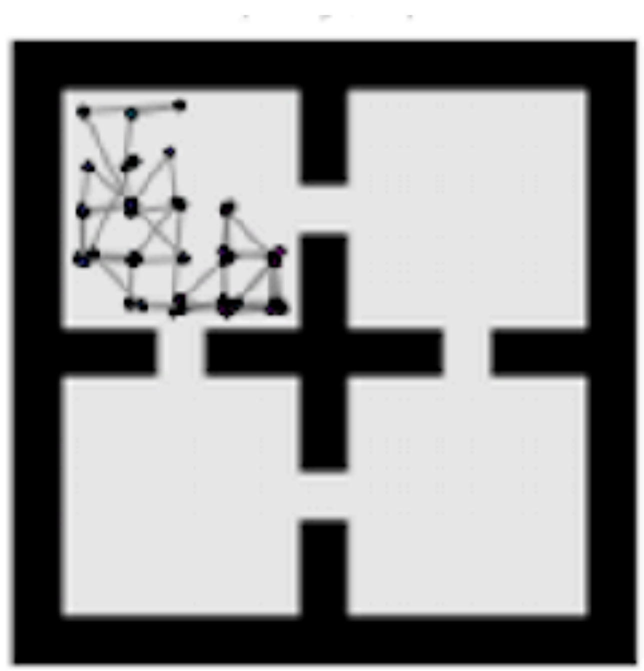
Sample with different statistics



Random Walk

Levy Walk

Agent gets stuck in room



Agent frees itself from room, allowing it to find reward faster

Random walk observed during sleep

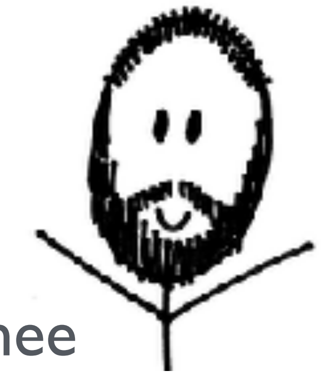
Levy walks experienced during foraging behavior

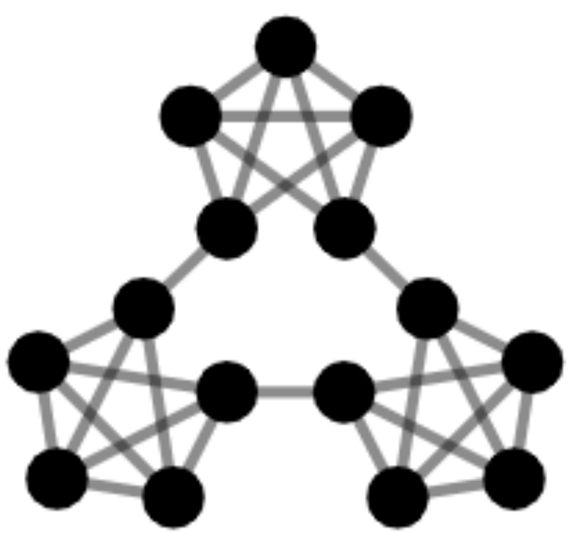
Stella et al (2019)

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Different advantages for different sampling modes

Dan
McNamee

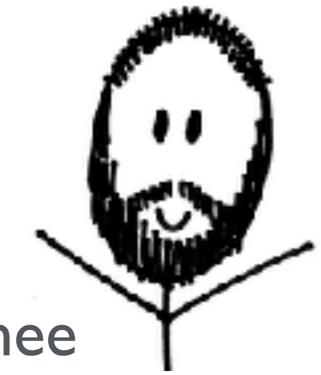


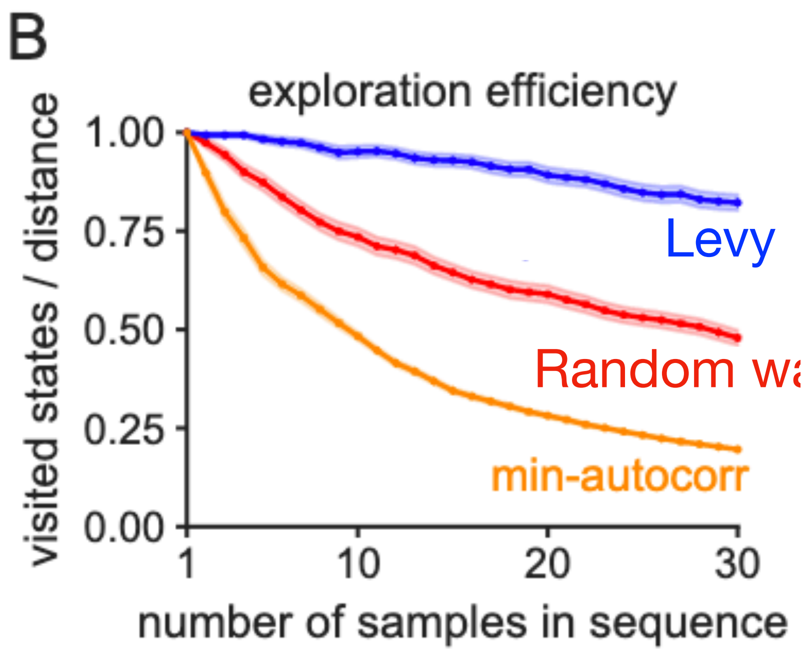


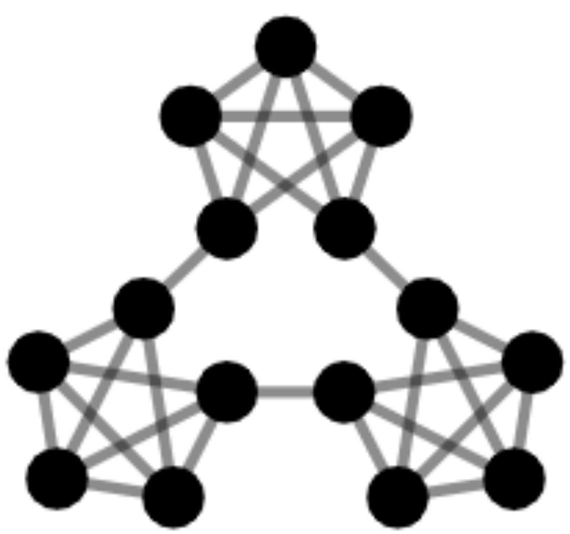
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Different advantages for different sampling modes

Dan
McNamee



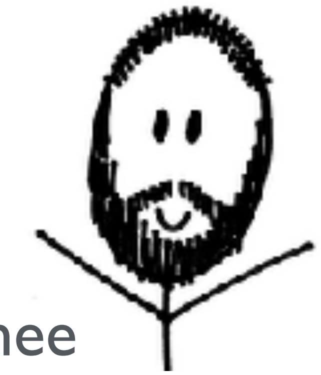


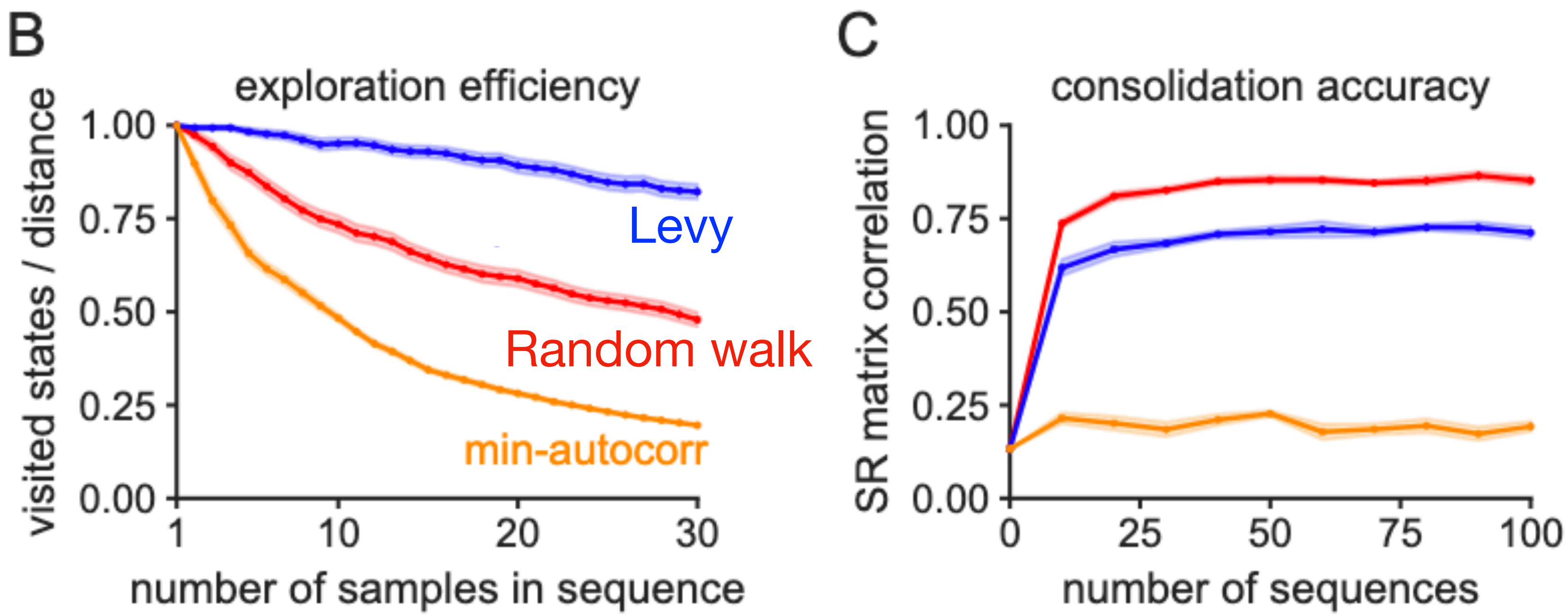


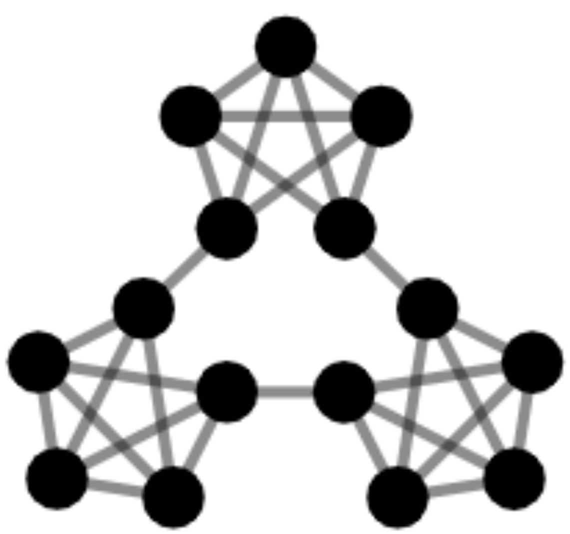
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Different advantages for different sampling modes

Dan McNamee



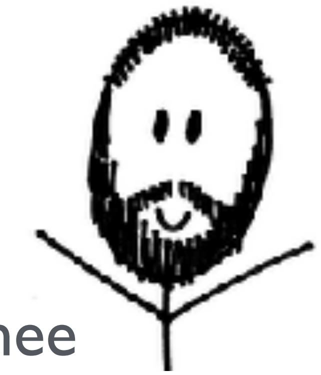


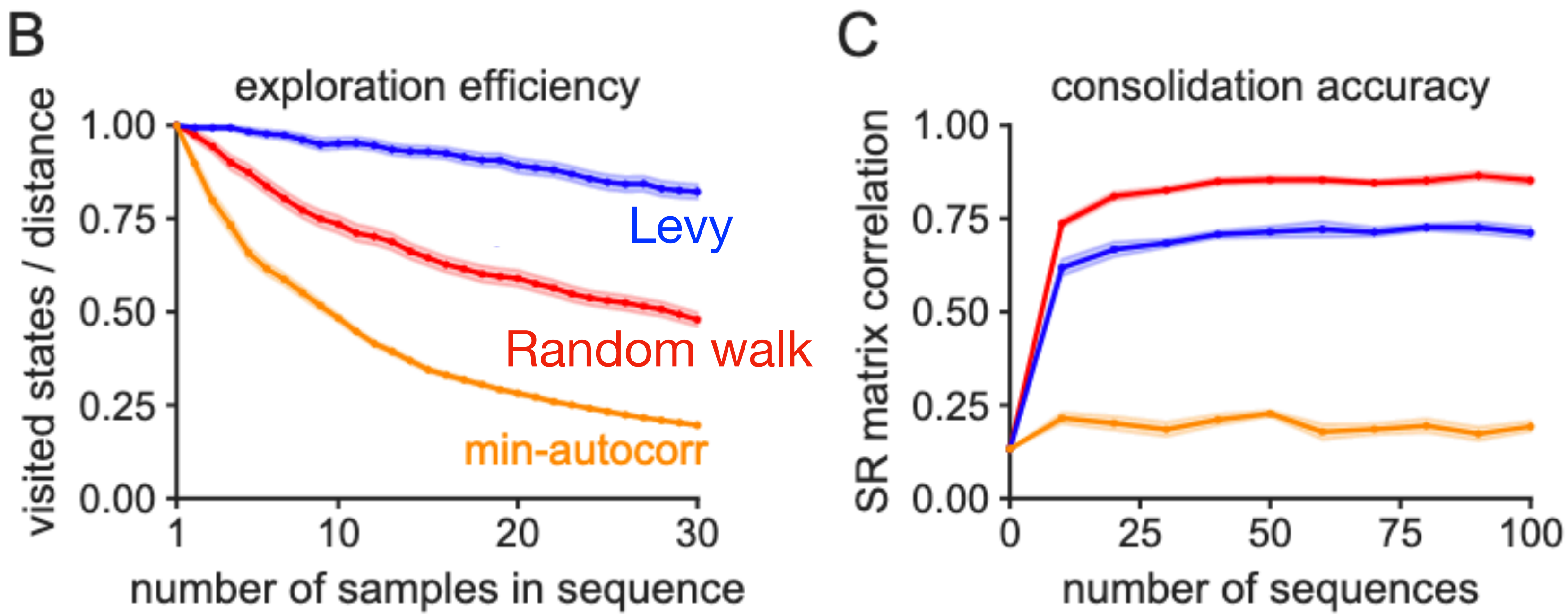


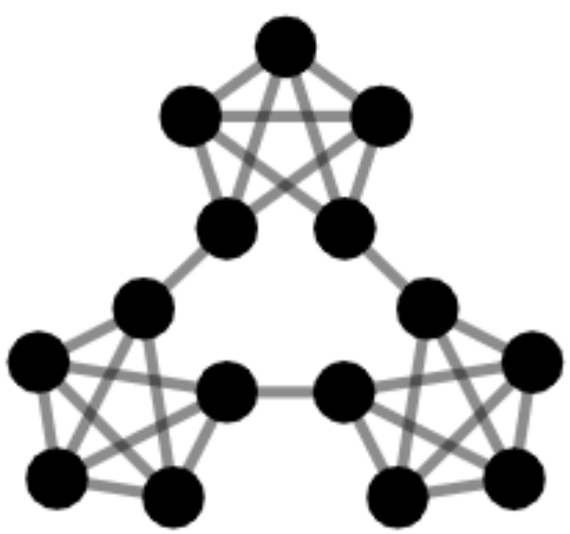
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

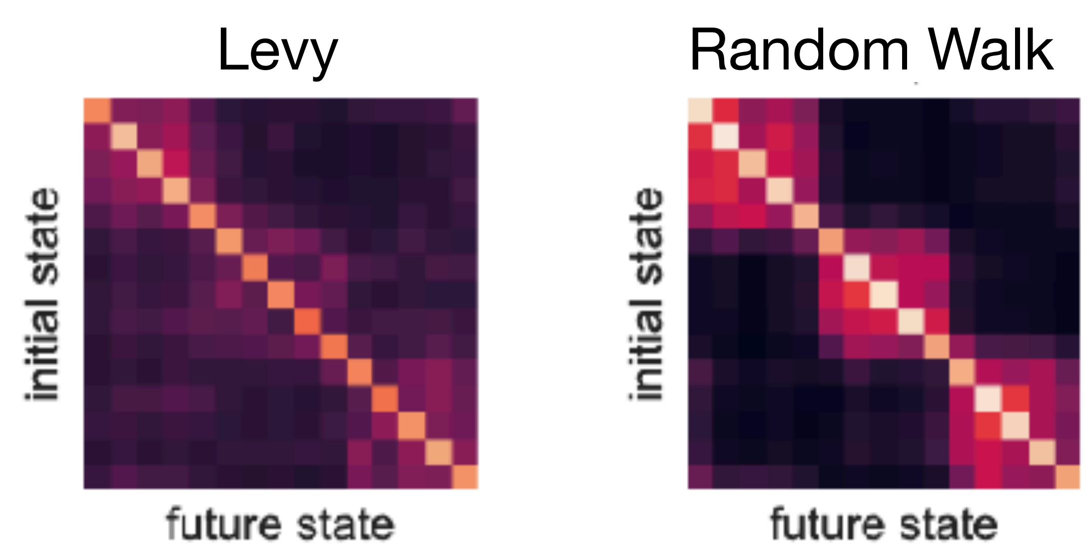
Different advantages for different sampling modes

Dan McNamee





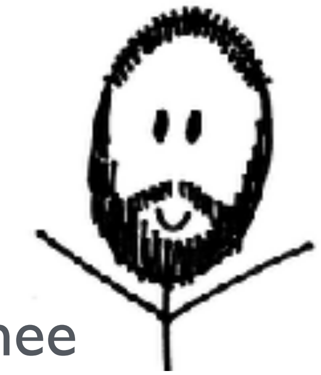


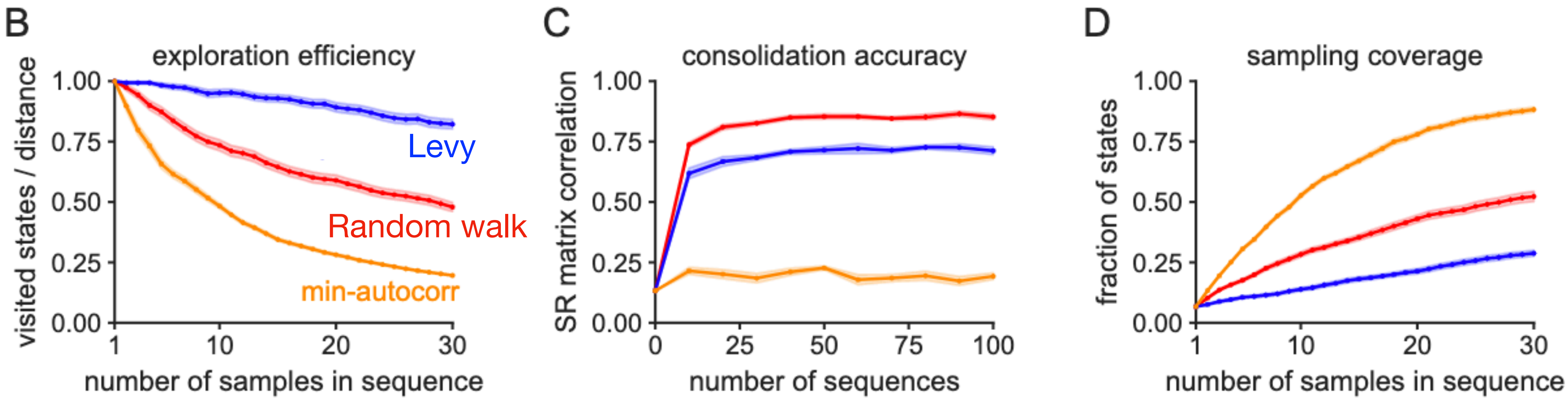


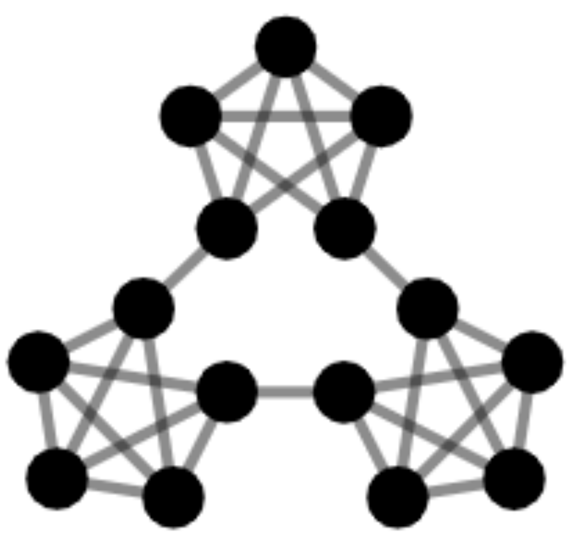
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

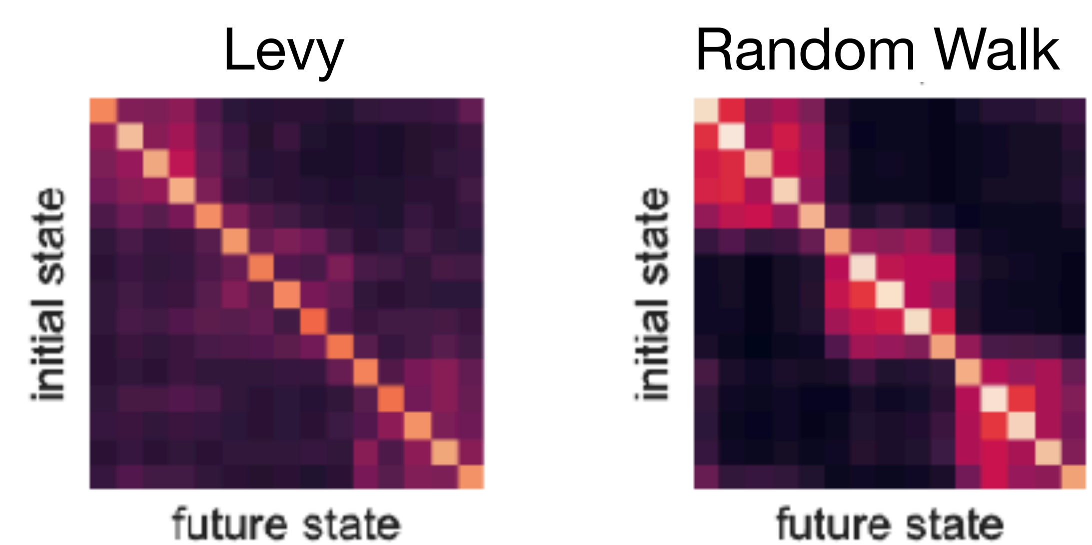
Different advantages for different sampling modes

Dan McNamee









McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Take-aways

Take-aways

- Prediction and compression as important ingredients of representations that support flexible behavior

Take-aways

- Prediction and compression as important ingredients of representations that support flexible behavior
- These are useful terms for understanding the neural mechanisms of cognitive flexibility

Take-aways

- Prediction and compression as important ingredients of representations that support flexible behavior
- These are useful terms for understanding the neural mechanisms of cognitive flexibility
- Representation learning is useful for neuroscience and neuroscience-inspired AI

Take-aways

- Prediction and compression as important ingredients of representations that support flexible behavior
- These are useful terms for understanding the neural mechanisms of cognitive flexibility
- Representation learning is useful for neuroscience and neuroscience-inspired AI
- Hippocampus is central to key cognitive functions we don’t know how to implement in machines

Hippocampus and Neuro-Inspired AI

Navigation
O’Keefe & Nadel 1978

Relational Memory
Eichenbaum et al 1999

Rapid, Episodic memory
Scoville & Milner 1957

Imagination
Hassabis et al 2007

Structure Learning
Schapiro et al 2016

Model-based Planning
Miller et al 2017, Vikhbladh et al 2019

Replay + Sampling
Skaggs & McNaughton 1996

Binding problem
Opitz 2010

Lots of potential for bridging Neuro + AI

Acknowledgements

Dan McNamee

Jesse Geerts

Matt Botvinick

Sam Gershman

Tim Behrens

Neil Burgess

Peter Battaglia

PRINCETON Neuroscience INSTITUTE

DeepMind

Contact Info

stachenfeld@google.com

www.neurokim.com