Cognitive Maps in Mice and Machines

Kim Stachenfeld
Research Scientist at DeepMind
MIND Summer School
10 August 2019
Extended version, with additional notes + citations for extra edification

Reinforcement Learning Problem

Reinforcement Learning Problem

- **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 Problem

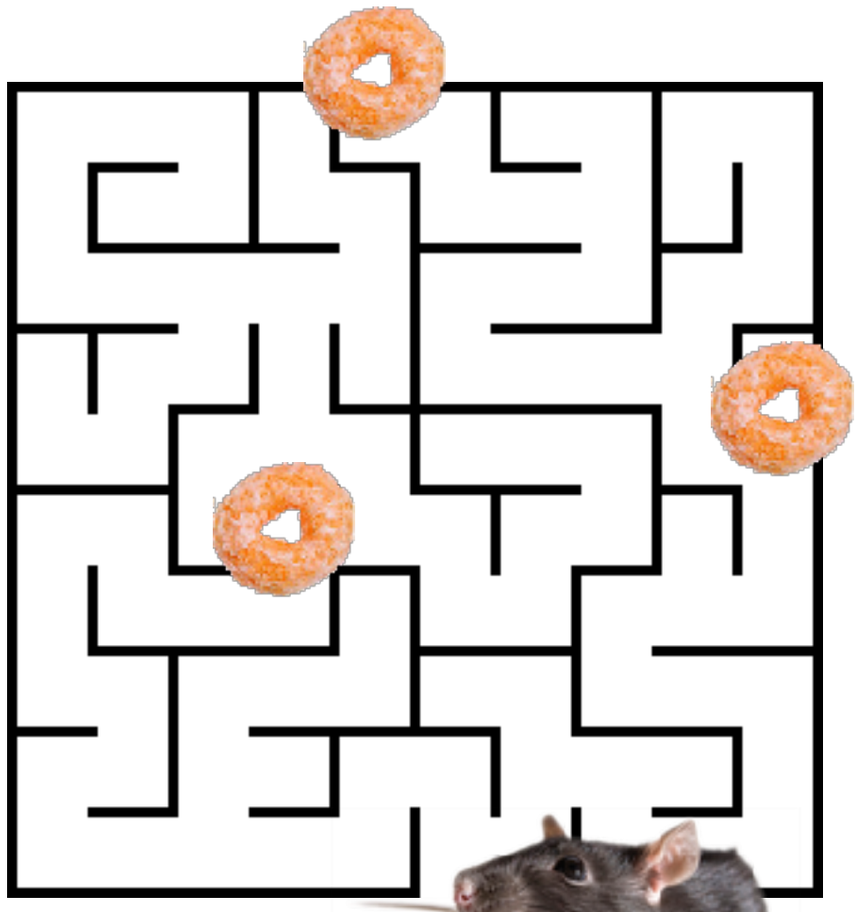
- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative reward, or value, over future states by trial and error

Reinforcement Learning Problem

- **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 Problem

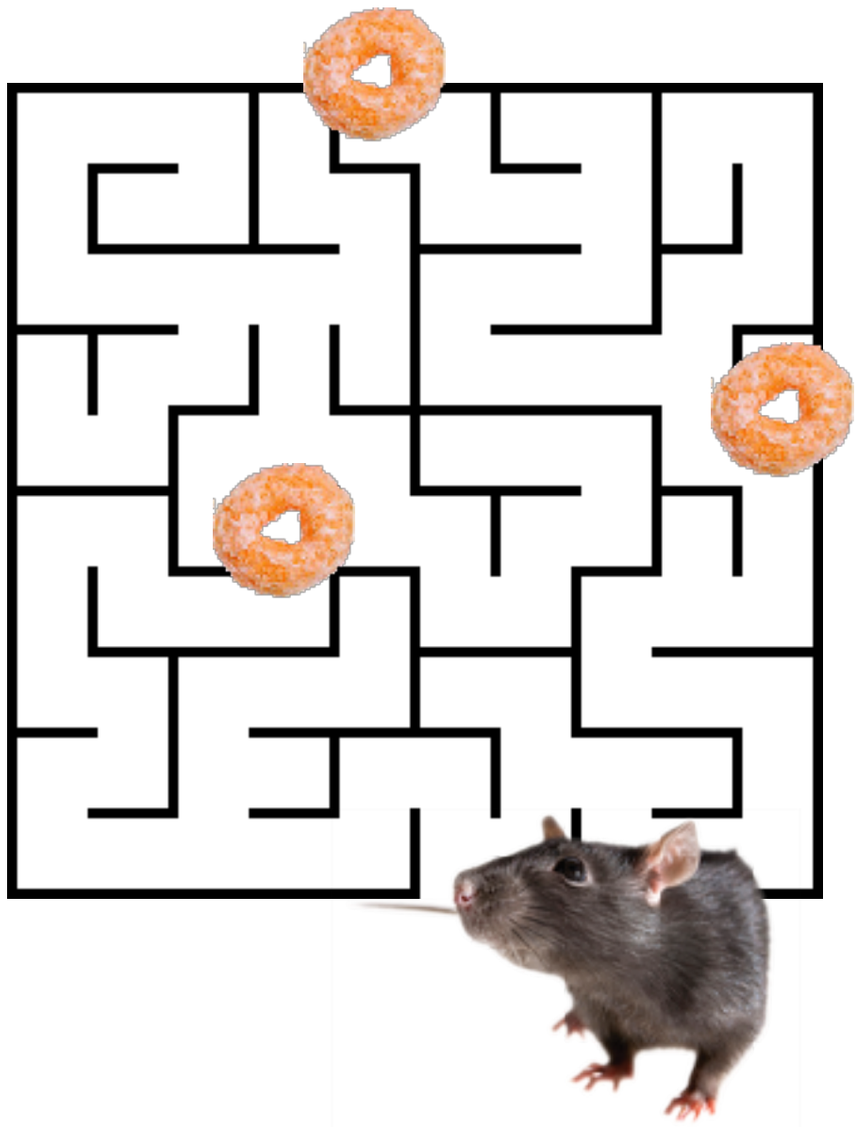
- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative reward, or value, over future states by trial and error



Reinforcement Learning Problem

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

- But this can be very hard in complex problems:

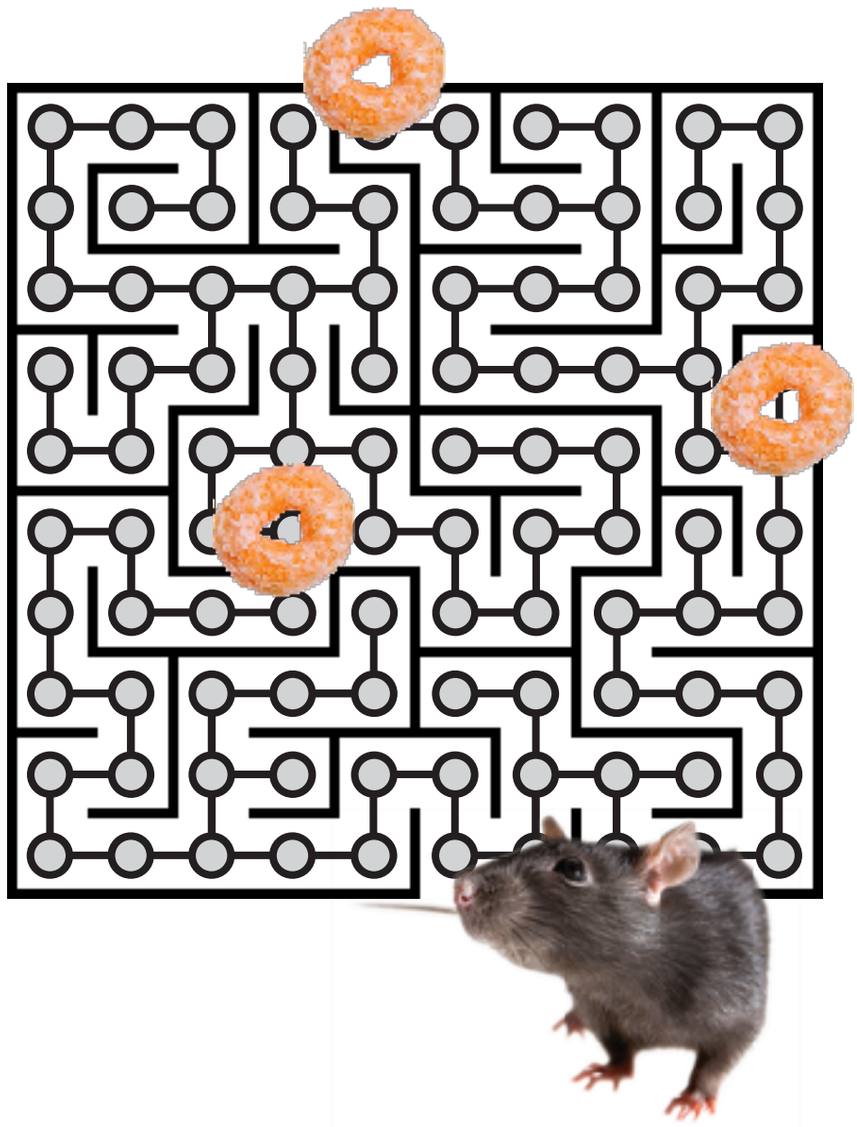


Reinforcement Learning Problem

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

- But this can be very hard in complex problems:

  - Lots of states to learn about



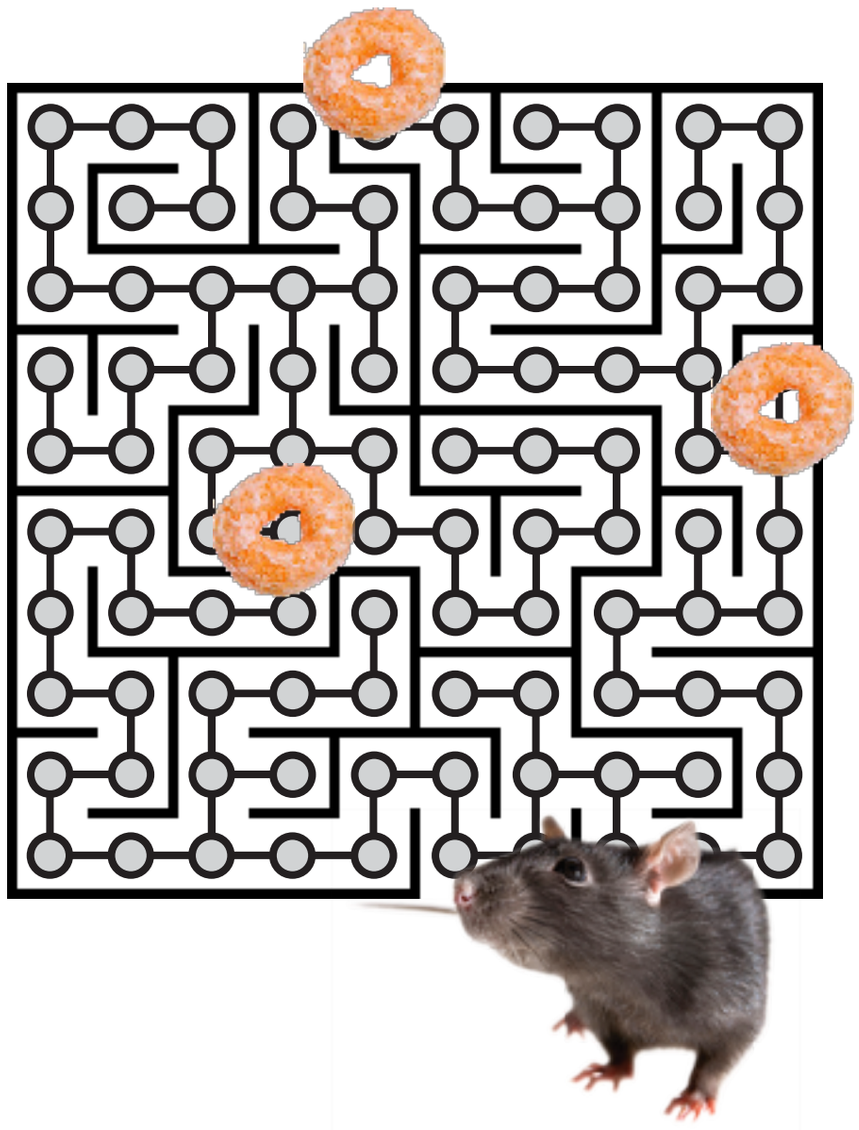
Reinforcement Learning Problem

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

- But this can be very hard in complex problems:

  - Lots of states to learn about

  - Not very many states are immediately informative about reward



Reinforcement Learning Problem

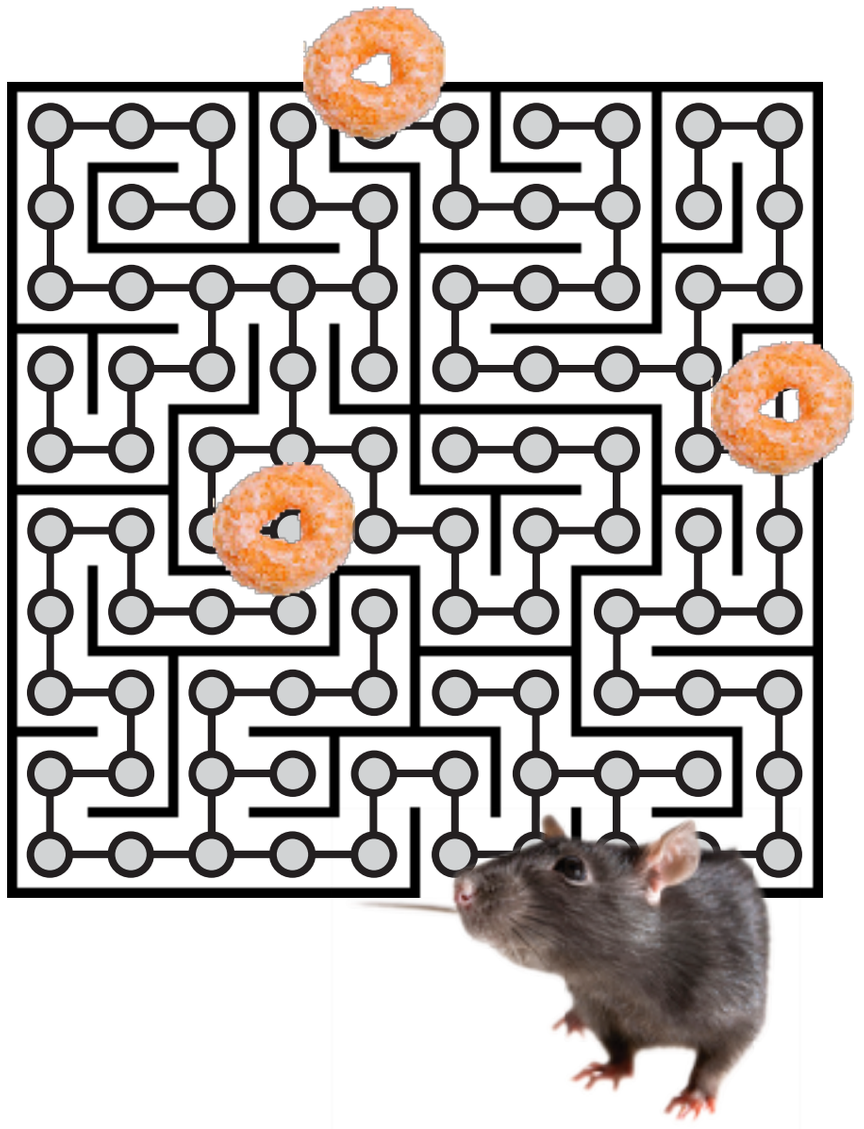
- **Reinforcement learning problem:**
  Given some task, learn a policy to maximize expected cumulative reward, or value, over future states by trial and error

- But this can be very hard in complex problems:

  - Lots of states to learn about

  - Not very many states are immediately informative about reward

  - Exploring the environment takes a long time



Reinforcement Learning Problem

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

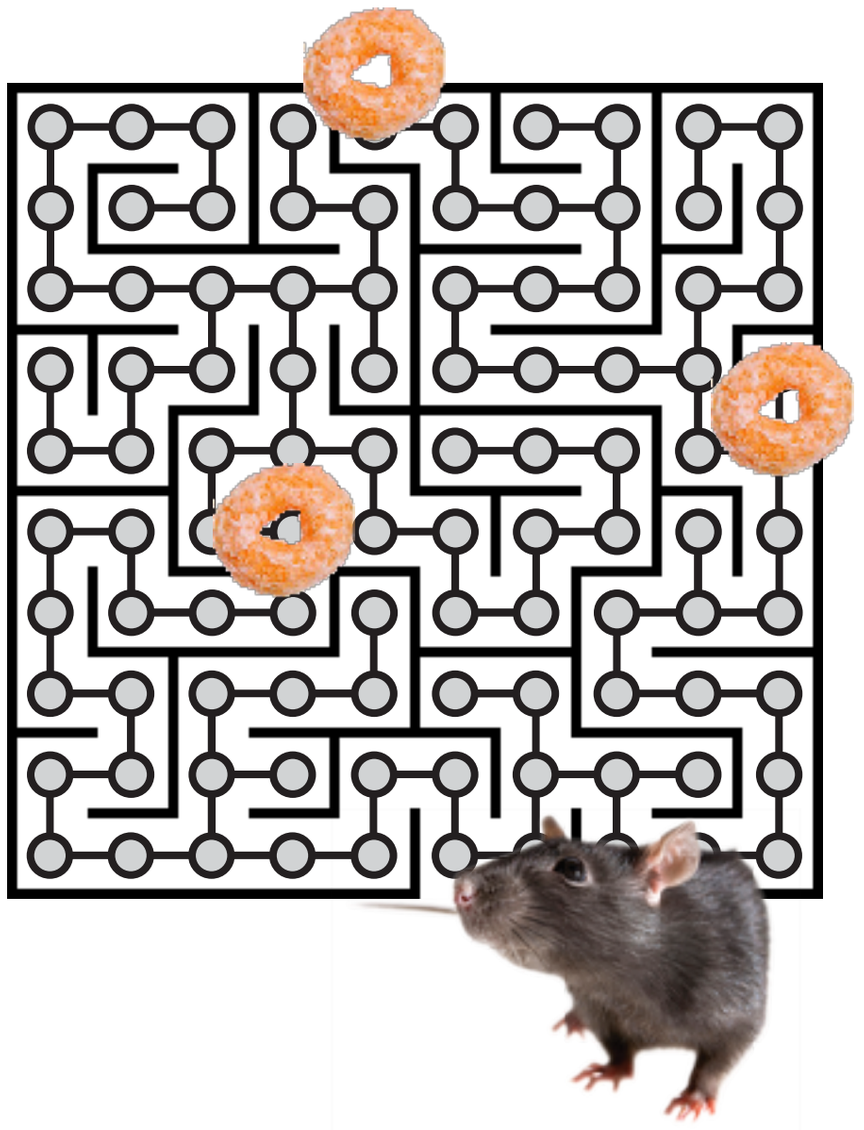
- But this can be very hard in complex problems:

  - Lots of states to learn about

  - Not very many states are immediately informative about reward

  - Exploring the environment takes a long time

  - Out of the box RL methods are not sample efficient



Reinforcement Learning Problem

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

- But this can be very hard in complex problems:

  - Lots of states to learn about

  - Not very many states are immediately informative about reward

  - Exploring the environment takes a long time

  - Out of the box RL methods are not sample efficient

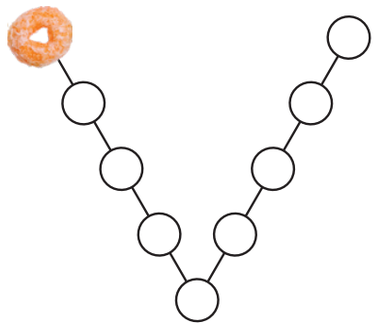
- **Does the hippocampal cognitive map constitute a structured representation of task structure that supports efficient RL?**

Learning benefits from recognising structure

An inefficient thing to do:

Learning benefits from recognising structure

An inefficient thing to do:

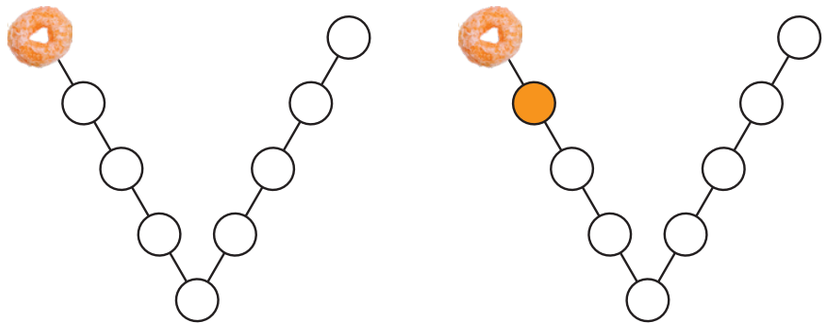


start

1st reward
“I guess that state over there is rewarding!”

Learning benefits from recognising structure

An inefficient thing to do:



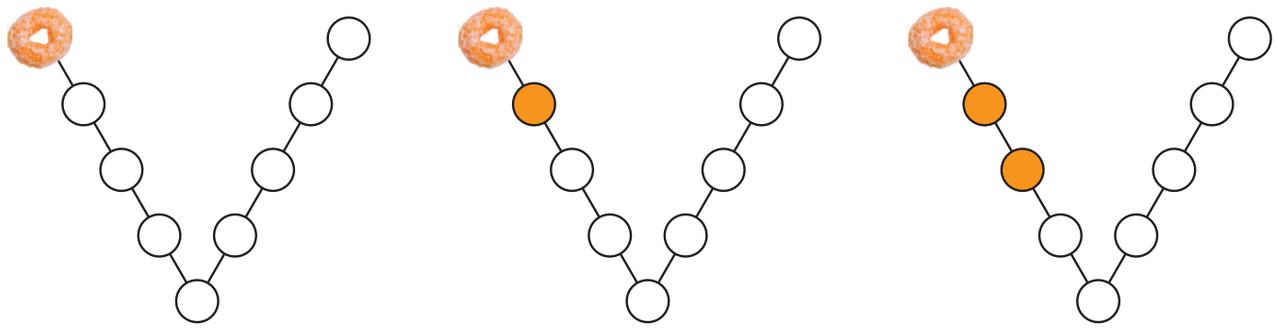
start

1st reward
“I guess that state over there is rewarding!”

2nd reward
“Neat, another valuable state!”

Learning benefits from recognising structure

An inefficient thing to do:



start

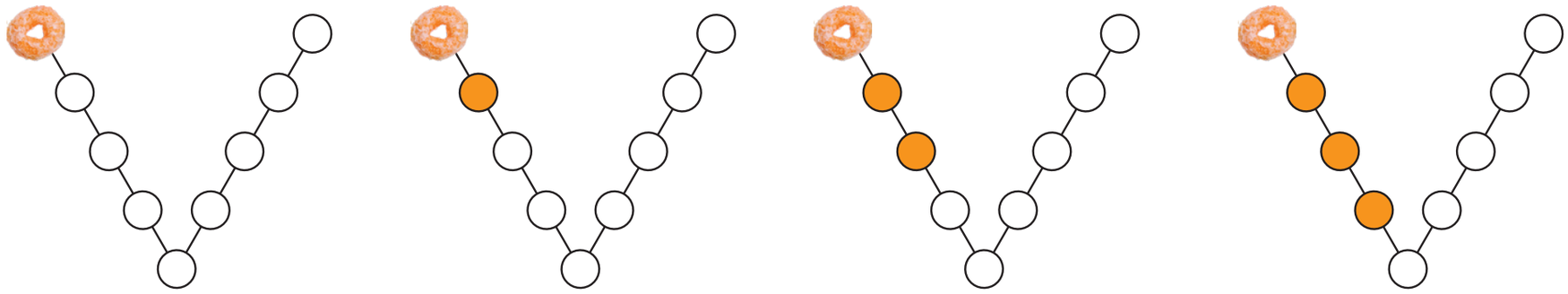
1st reward
“I guess that state over there is rewarding!”

2nd reward
“Neat, another valuable state!”

3rd reward
“Yet another state that predicts value! What luck. If only there was some sort of pattern.”

Learning benefits from recognising structure

An inefficient thing to do:



start

1st reward
"I guess that state over there is rewarding!"

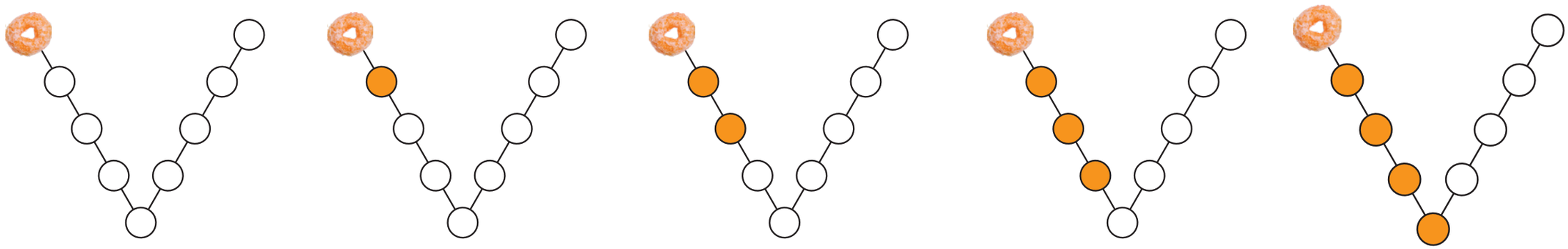
2nd reward
"Neat, another valuable state!"

3rd reward
"Yet another state that predicts value! What luck. If only there was some sort of pattern."

4th reward
…

Learning benefits from recognising structure

An inefficient thing to do:



start

1st reward
“I guess that state over there is rewarding!”

2nd reward
“Neat, another valuable state!”

3rd reward
“Yet another state that predicts value! What luck. If only there was some sort of pattern.”

4th reward
…

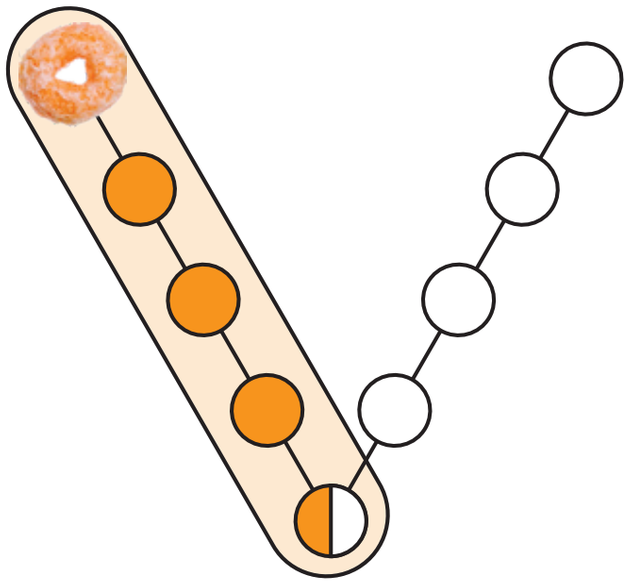
5th reward
“I should go left.”

Learning benefits from recognising structure

A more reasonable thing to do:

Learning benefits from recognising structure

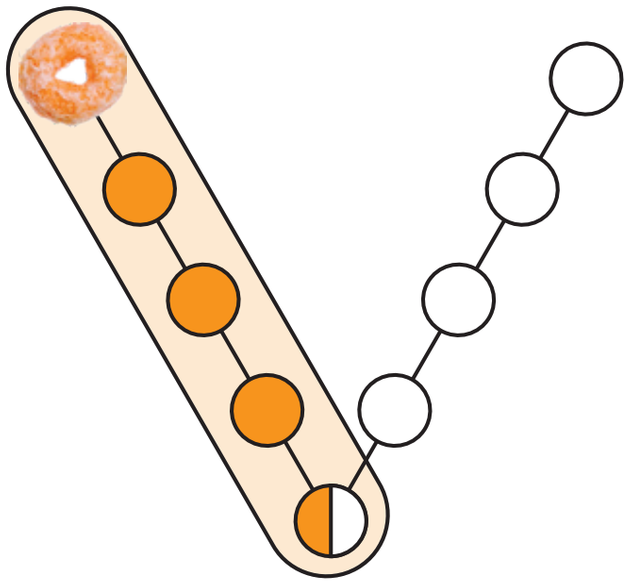
A more reasonable thing to do:



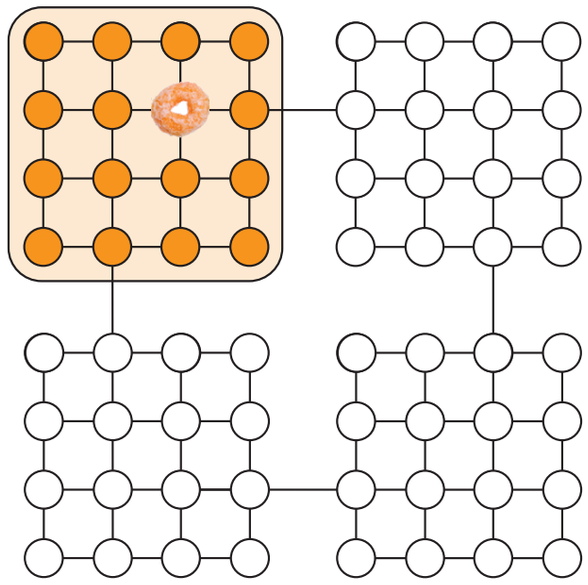
1st left
“I should go left.”

Learning benefits from recognising structure

A more reasonable thing to do:



1st left
“I should go left.”



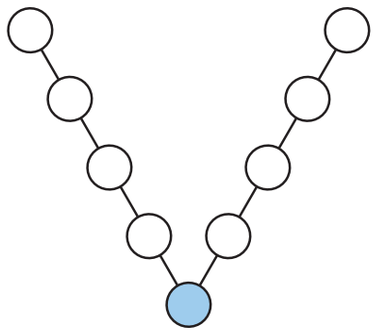
1st visit to top left room
“I should go to the northwest room”

Exploration benefits from structure

Another inefficient thing to do:

Exploration benefits from structure

Another inefficient thing to do:

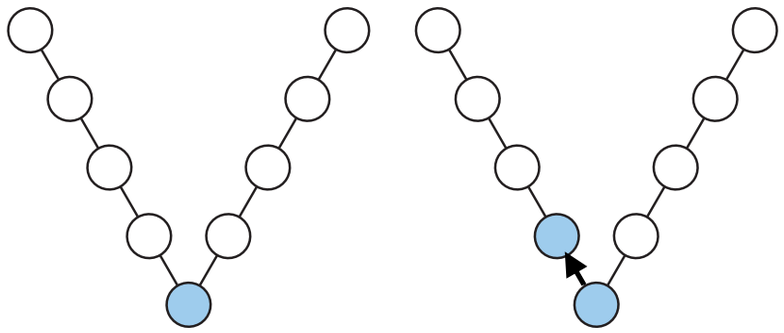


start

blue = visited state

Exploration benefits from structure

Another inefficient thing to do:

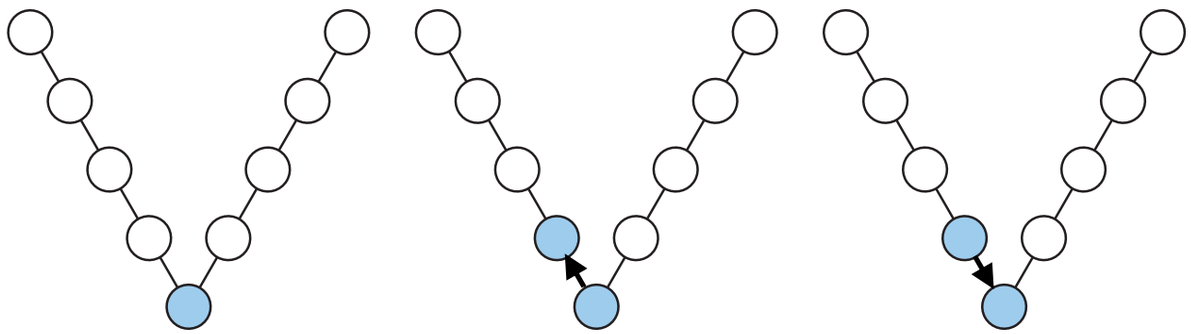


start **1st step**

blue = visited state

Exploration benefits from structure

Another inefficient thing to do:

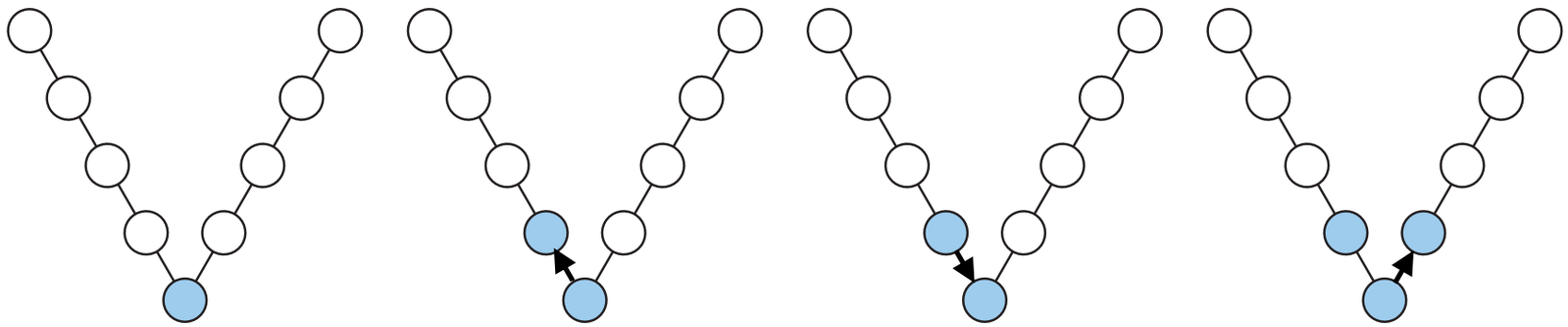


start **1st step** **2nd step**

blue = visited state

Exploration benefits from structure

Another inefficient thing to do:

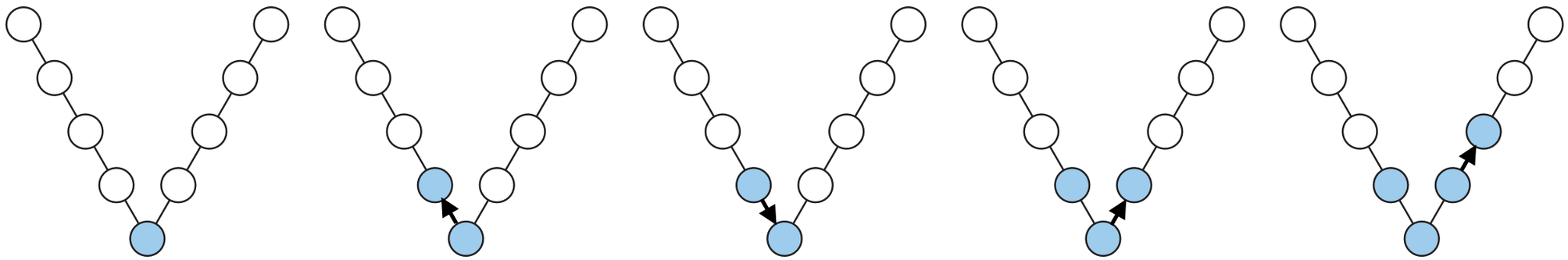


start **1st step** **2nd step** **3rd step**

blue = visited state

Exploration benefits from structure

Another inefficient thing to do:

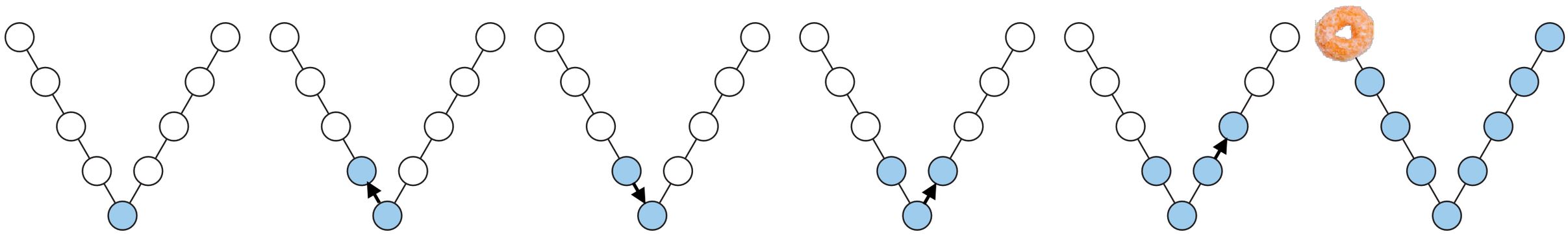


start **1st step** **2nd step** **3rd step** **4th step**

blue = visited state

Exploration benefits from structure

Another inefficient thing to do:



start **1st step** **2nd step** **3rd step** **4th step** **…eventually**

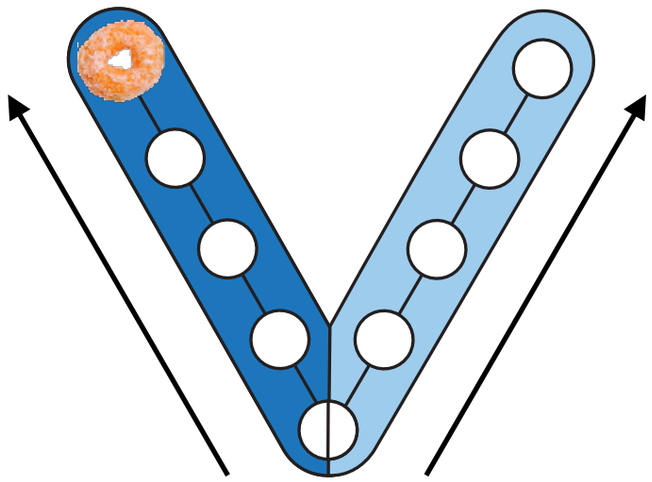
blue = visited state

Exploration benefits from structure

A more reasonable thing to do:

Exploration benefits from structure

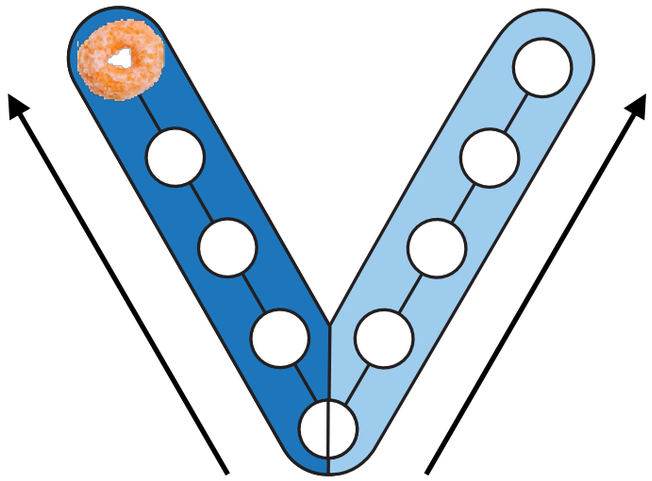
A more reasonable thing to do:



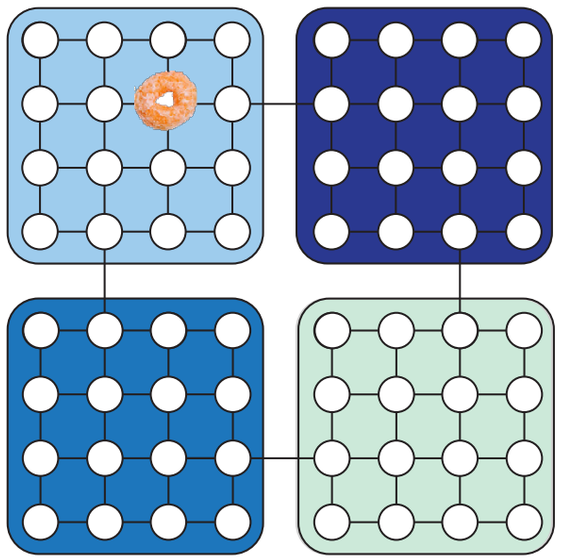
“I should check arm 1 and then arm 2”

Exploration benefits from structure

A more reasonable thing to do:



“I should check arm 1 and then arm 2”



“I should check each room”

What makes a good representation?

- **Other examples of work using a "representation learning" perspective:** Banino et al (2018), Yamins et al (2014), Gardner et al (2018), Momennejad et al (2017), Gustafson & Daw (2011), Niv et al (2015), Wang et al (2015), Radulescu et al (2019), Gershman & Niv (2010), Doll et al (2012), Behrens et al (2018), Stachenfeld et al (2017)

- ^Not exhaustive

Abstract learning and exploration

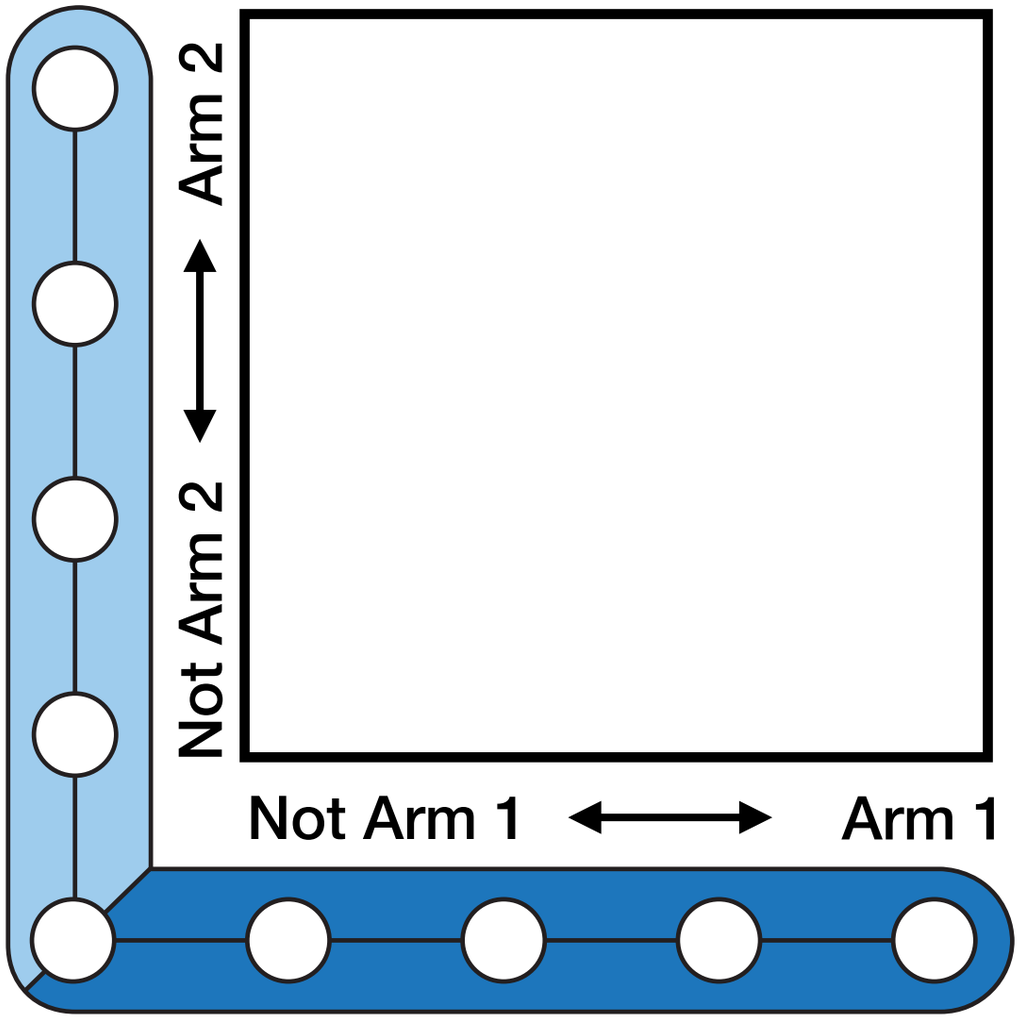
Abstract learning and exploration

Learn in representation space

Value function approximation in representation space

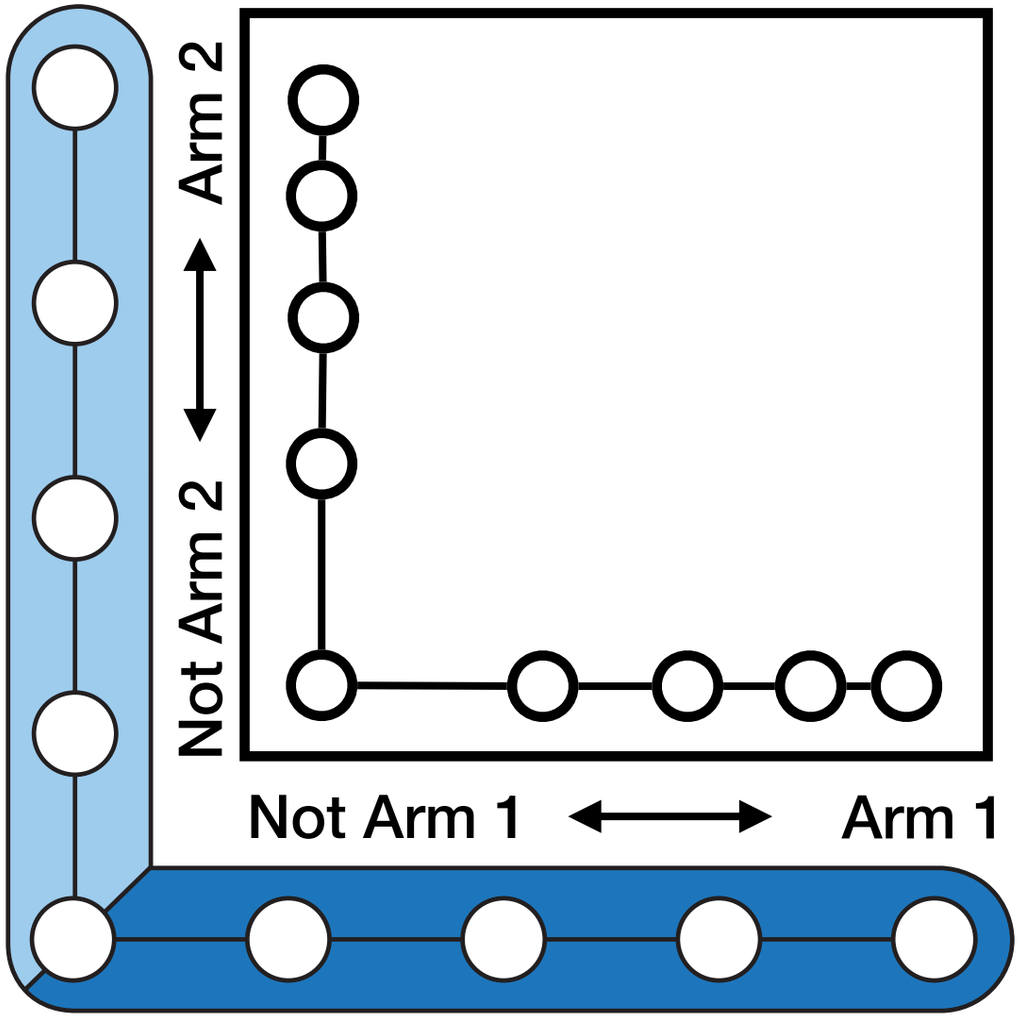
Abstract learning and exploration

Learn in representation space
Value function approximation in representation space



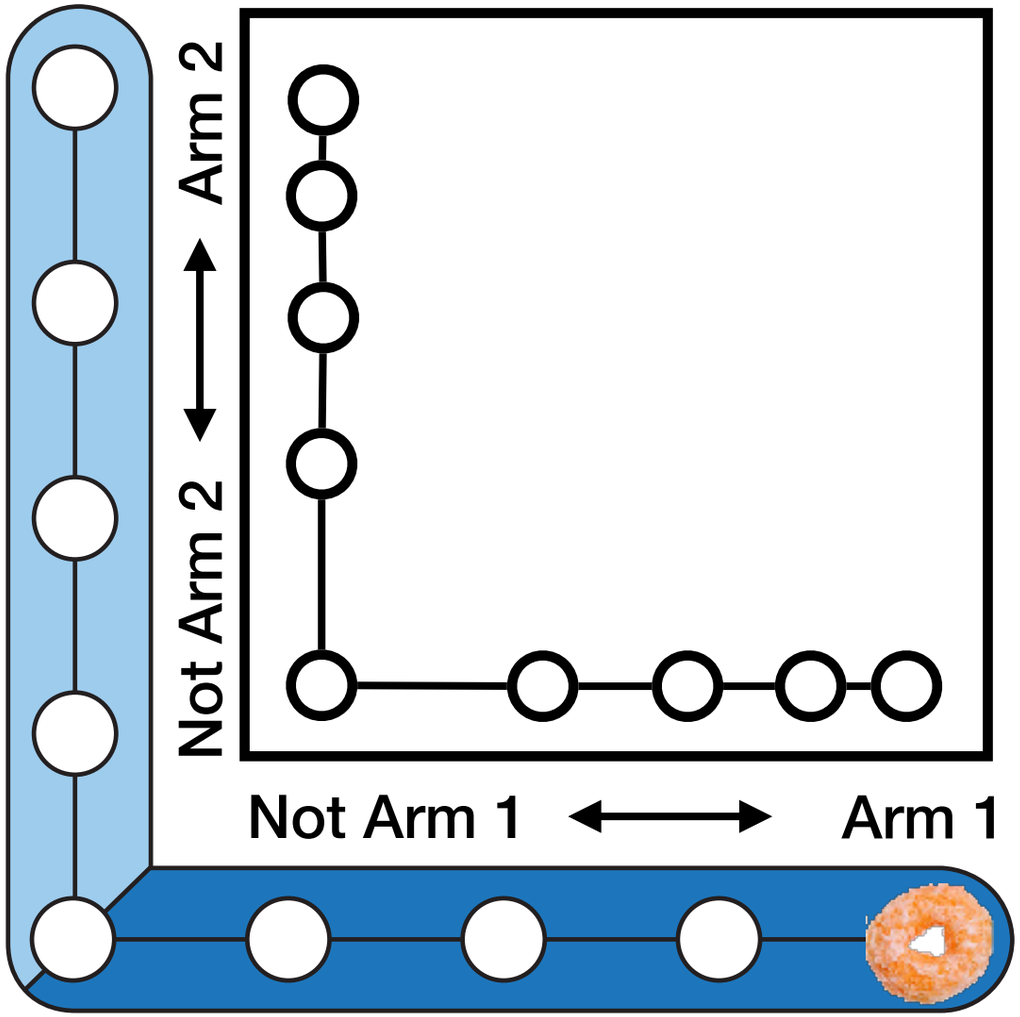
Abstract learning and exploration

Learn in representation space
Value function approximation in representation space



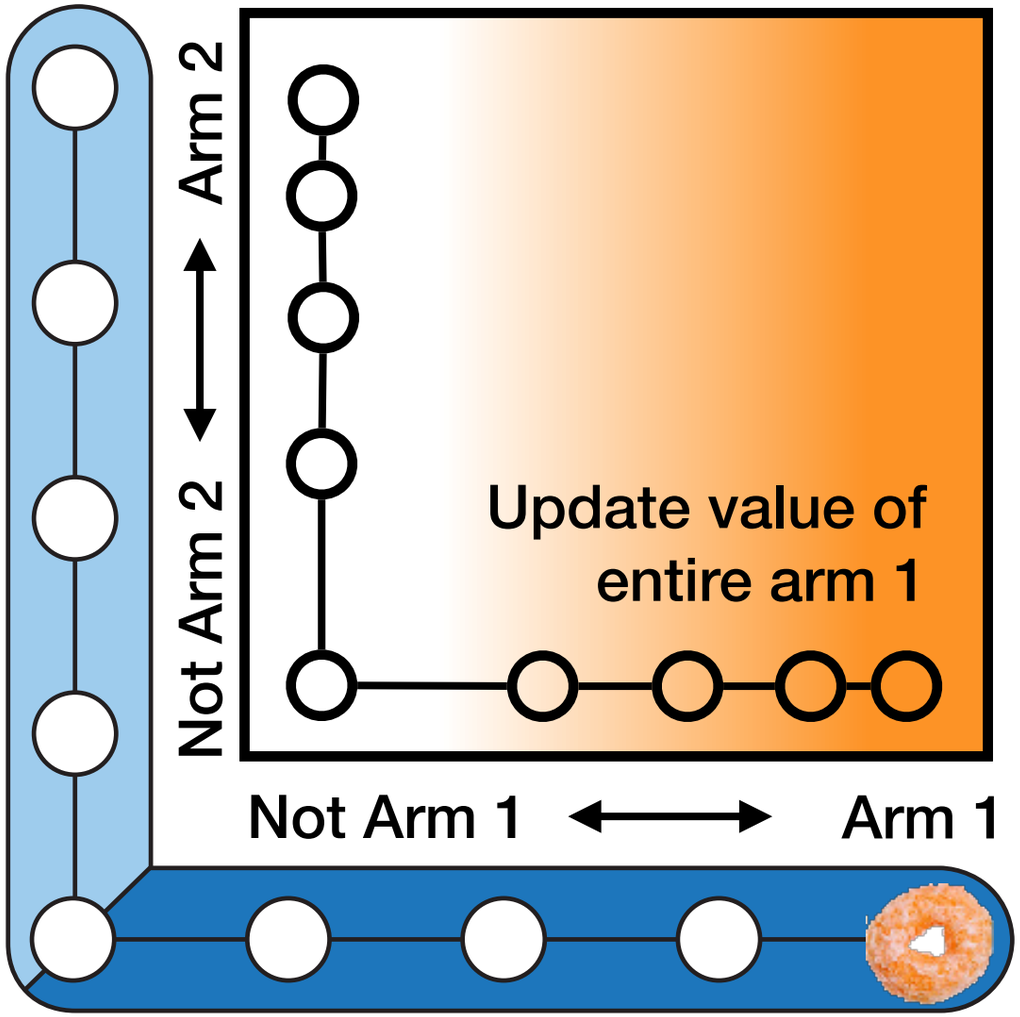
Abstract learning and exploration

Learn in representation space
Value function approximation in representation space



Abstract learning and exploration

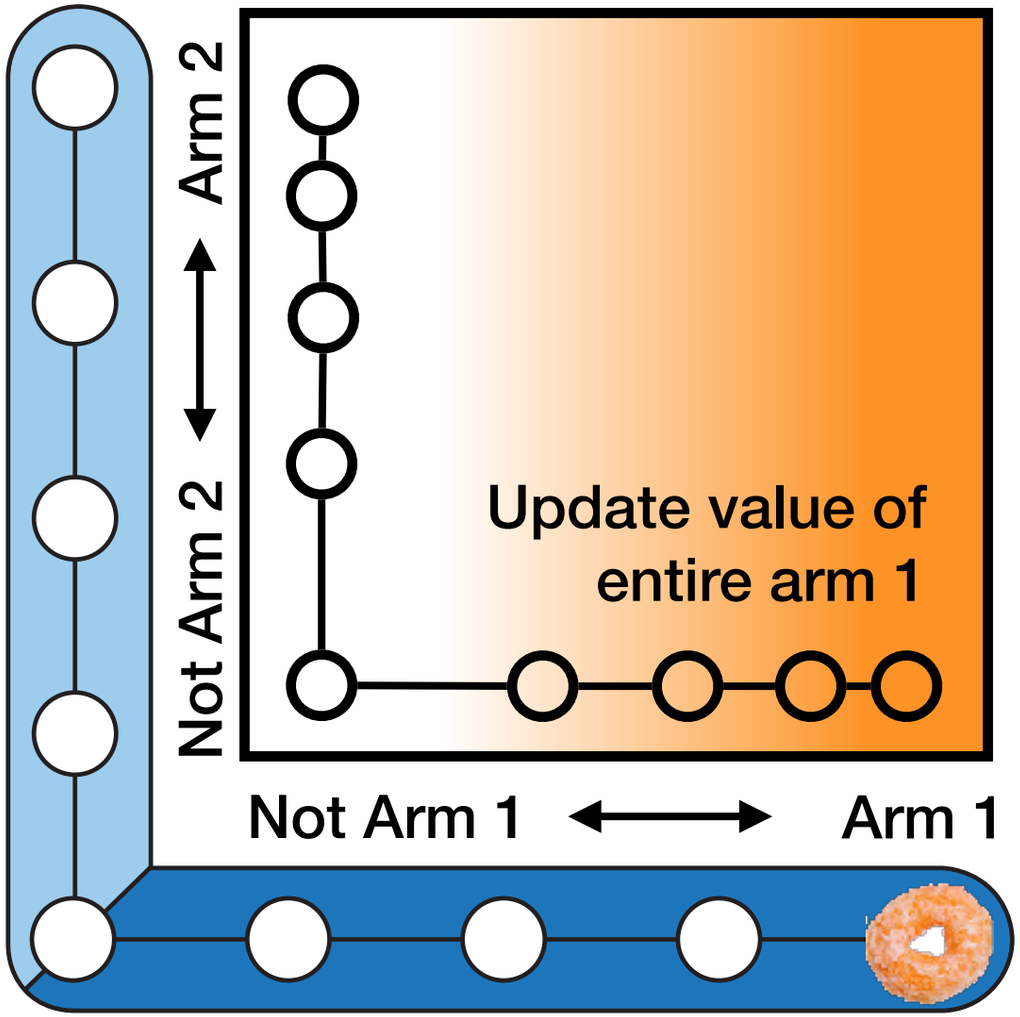
Learn in representation space
Value function approximation in representation space



Abstract learning and exploration

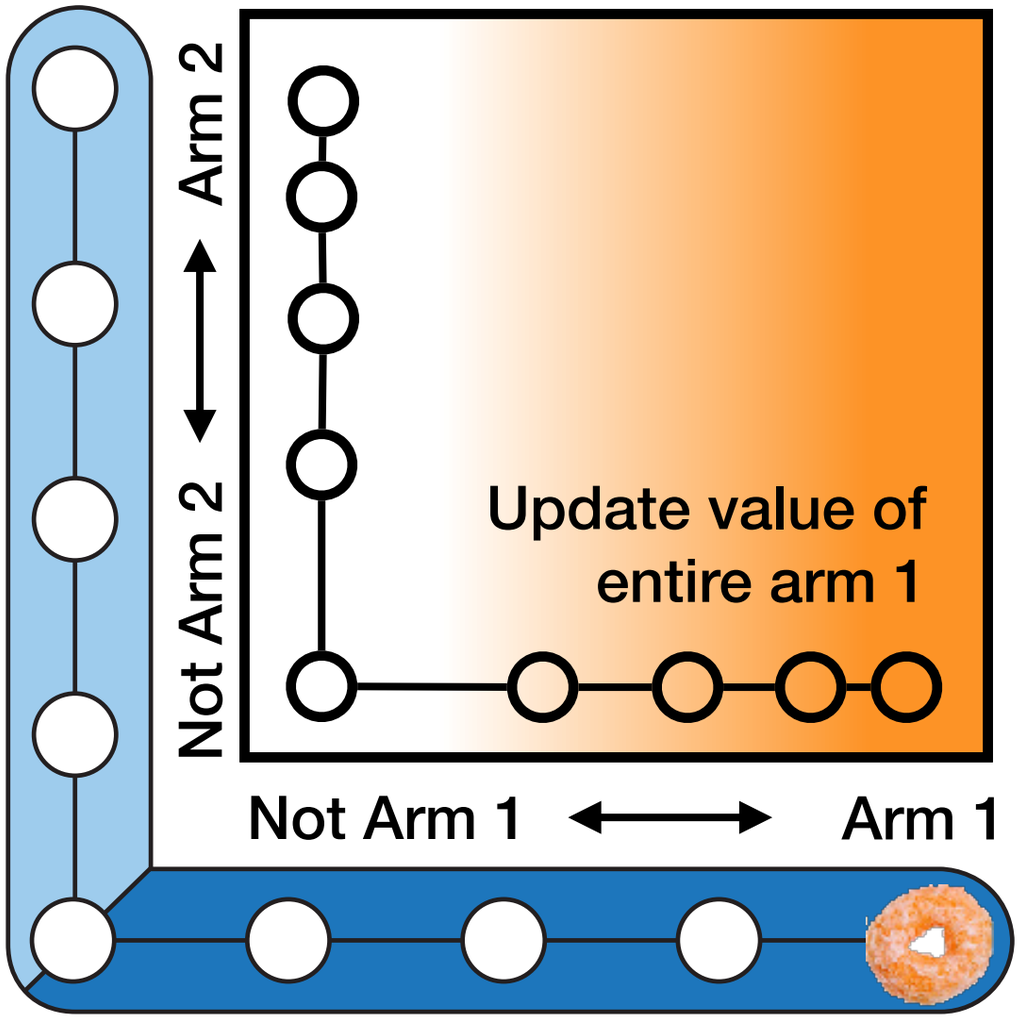
Learn in representation space
Value function approximation in representation space

Explore representation space
Extract sequence of actions that explores an axis in representation space

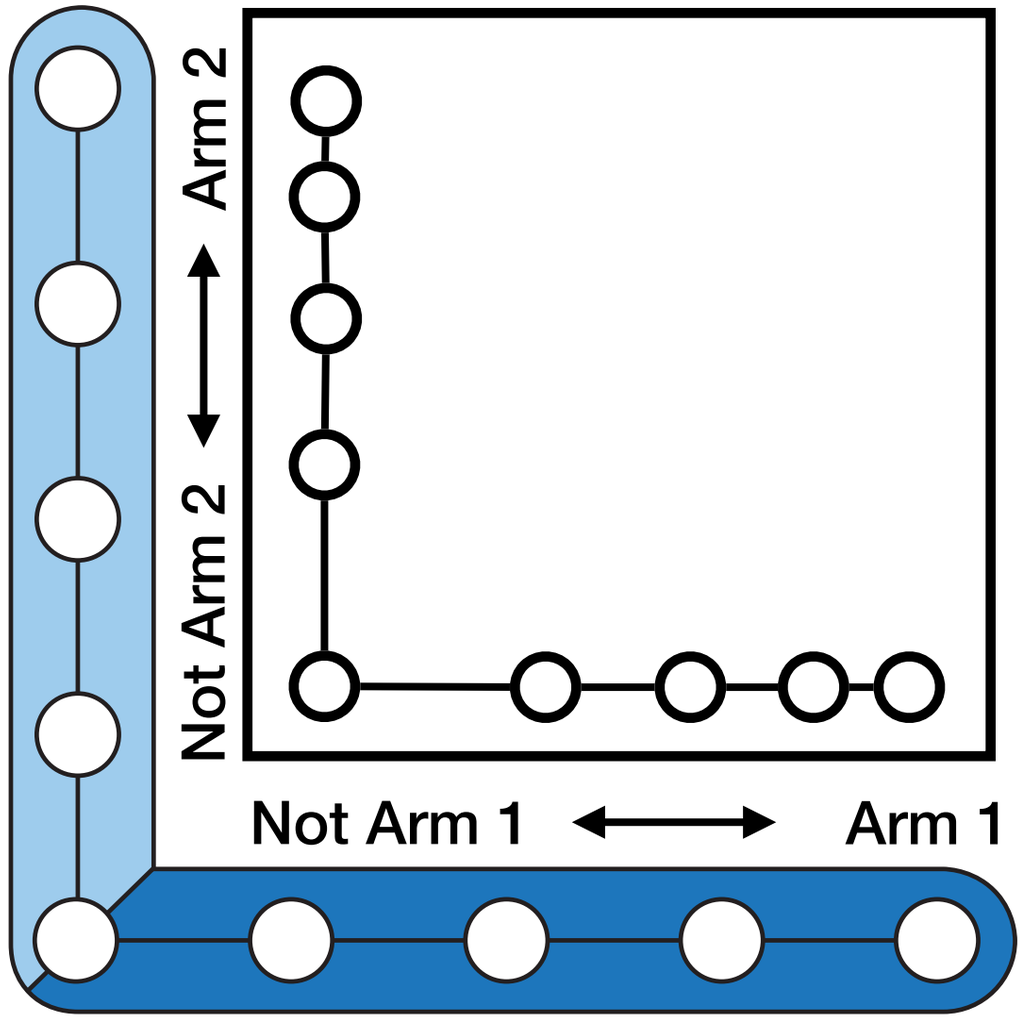


Abstract learning and exploration

Learn in representation space
Value function approximation in representation space

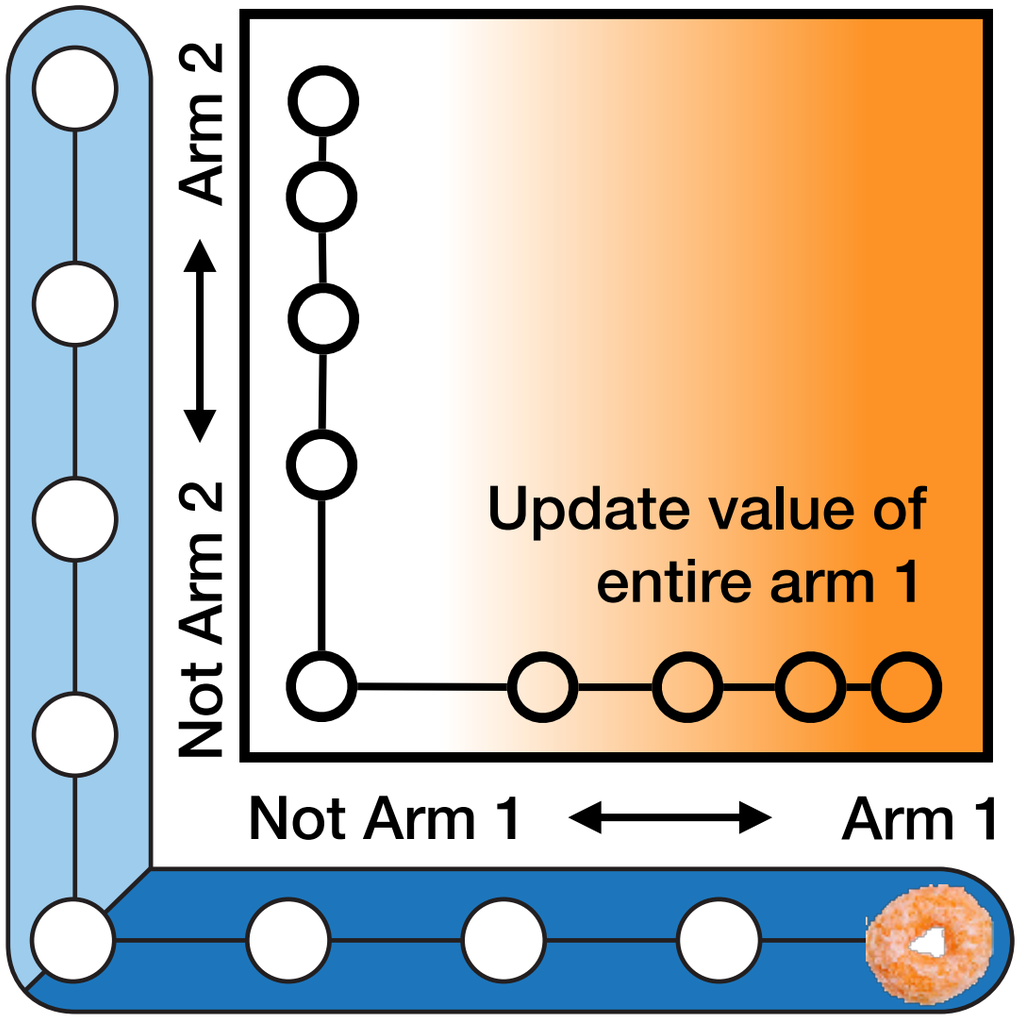


Explore representation space
Extract sequence of actions that explores an axis in representation space

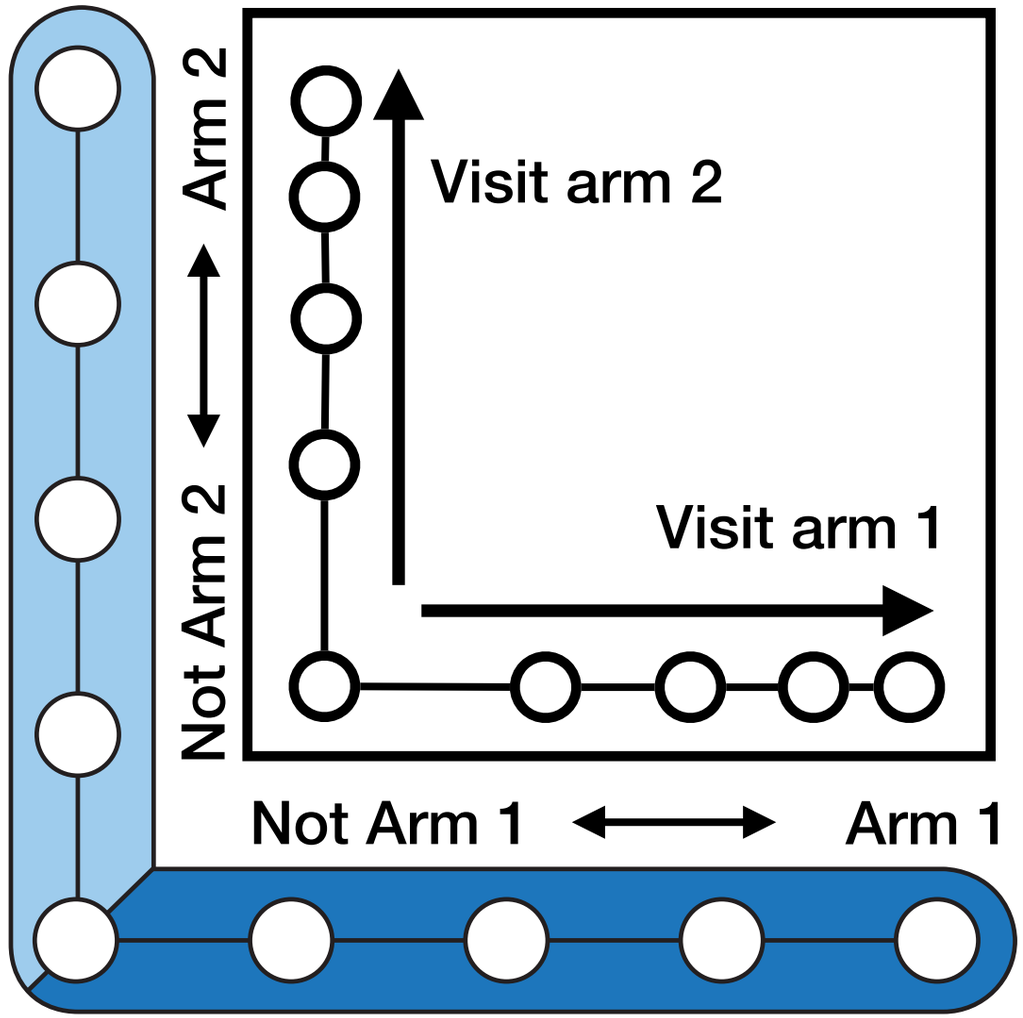


Abstract learning and exploration

Learn in representation space
Value function approximation in representation space

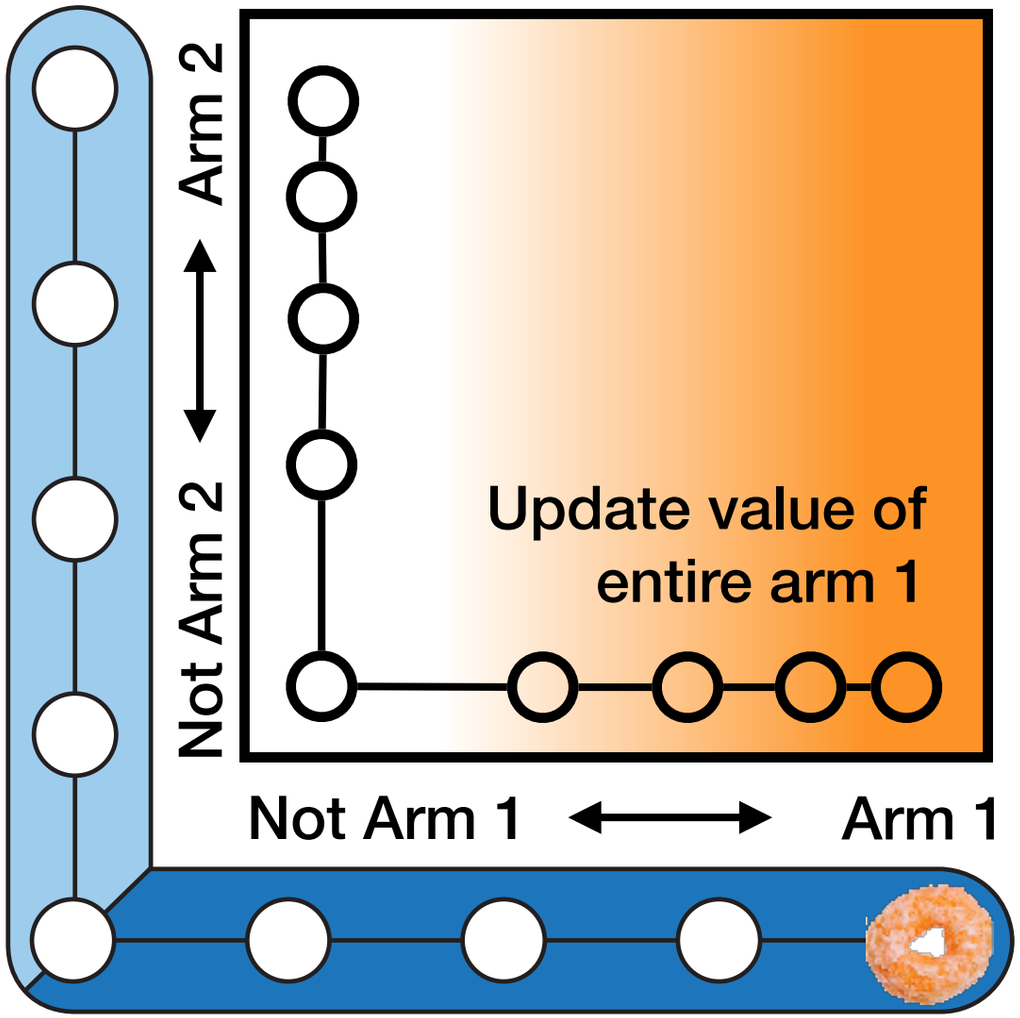


Explore representation space
Extract sequence of actions that explores an axis in representation space

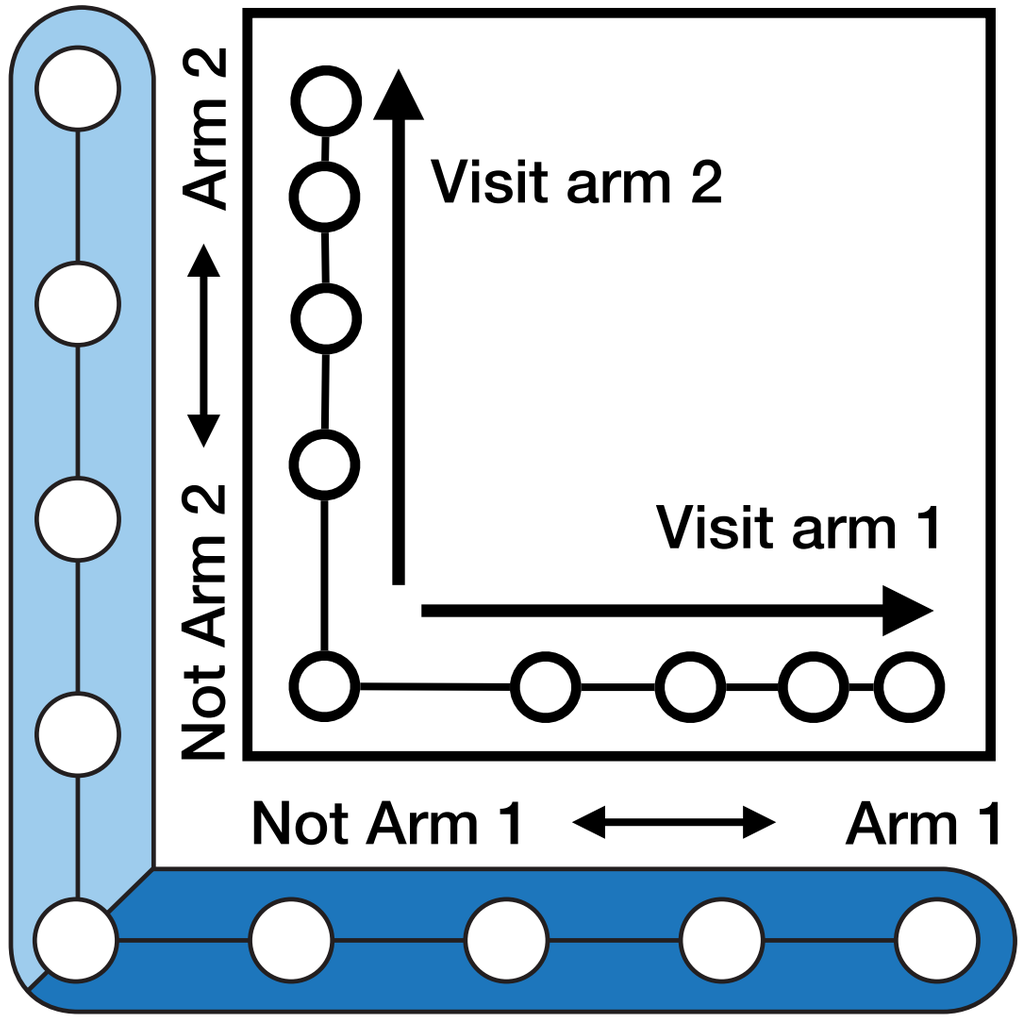


Abstract learning and exploration

Learn in representation space
Value function approximation in representation space



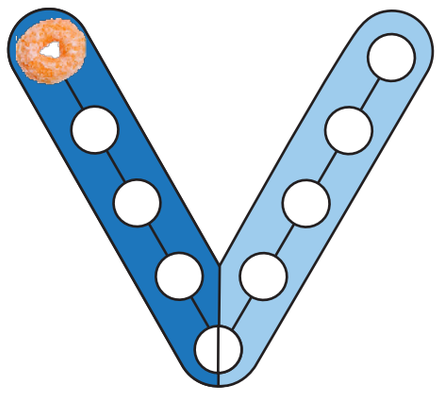
Explore representation space
Extract sequence of actions that explores an axis in representation space



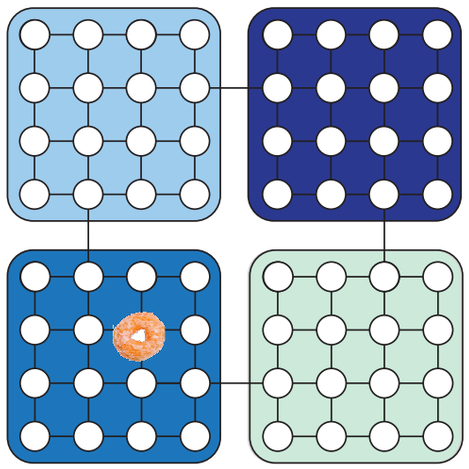
Representation learning goal for both:
cluster states that **generalize**
separate states that **do not**

What makes a good representation?

Maze arms

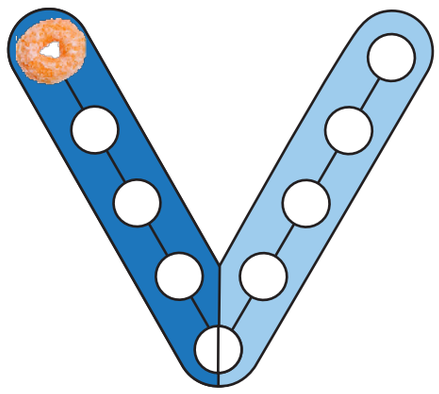


Rooms

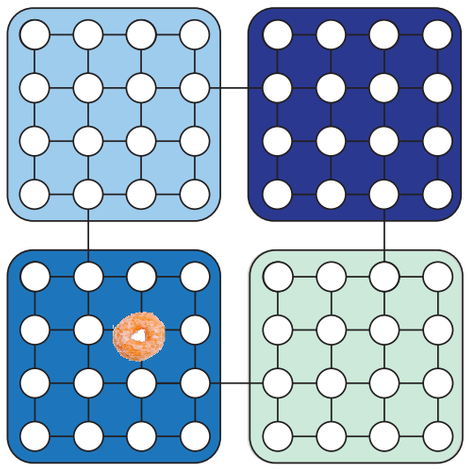


What makes a good representation?

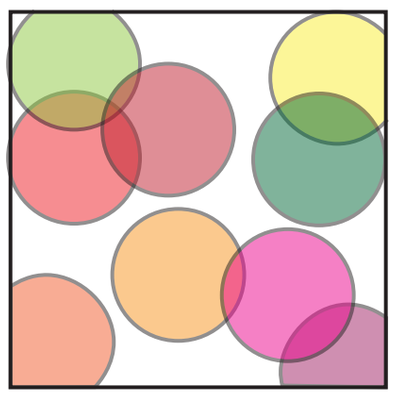
Maze arms



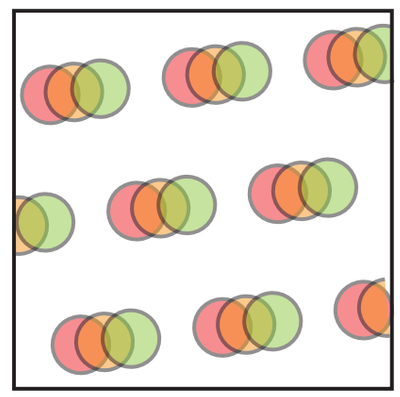
Rooms



Place cells

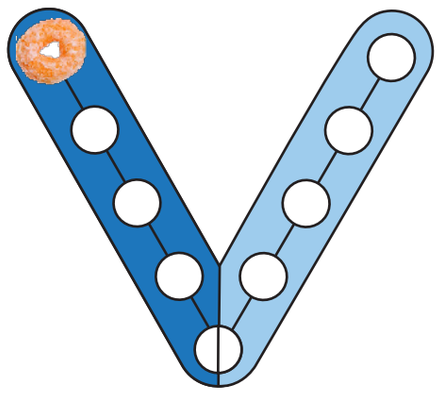


Grid cells

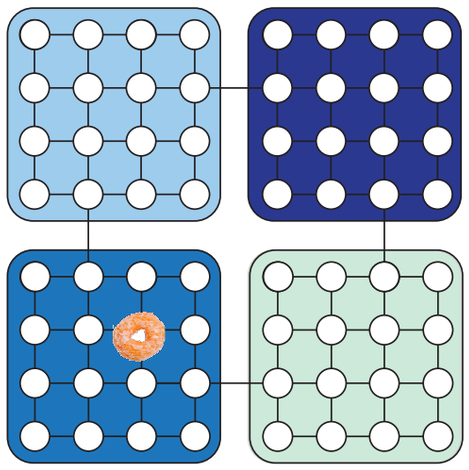


What makes a good representation?

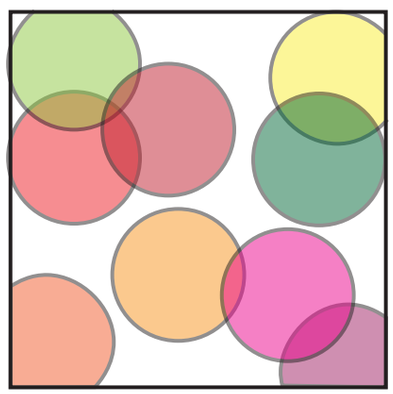
Maze arms



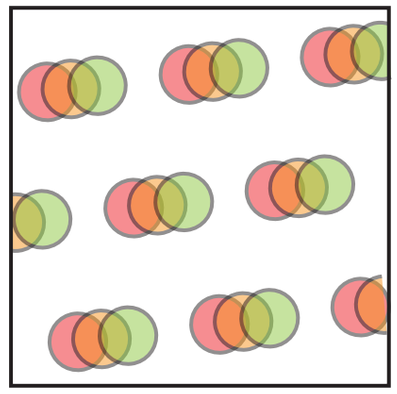
Rooms



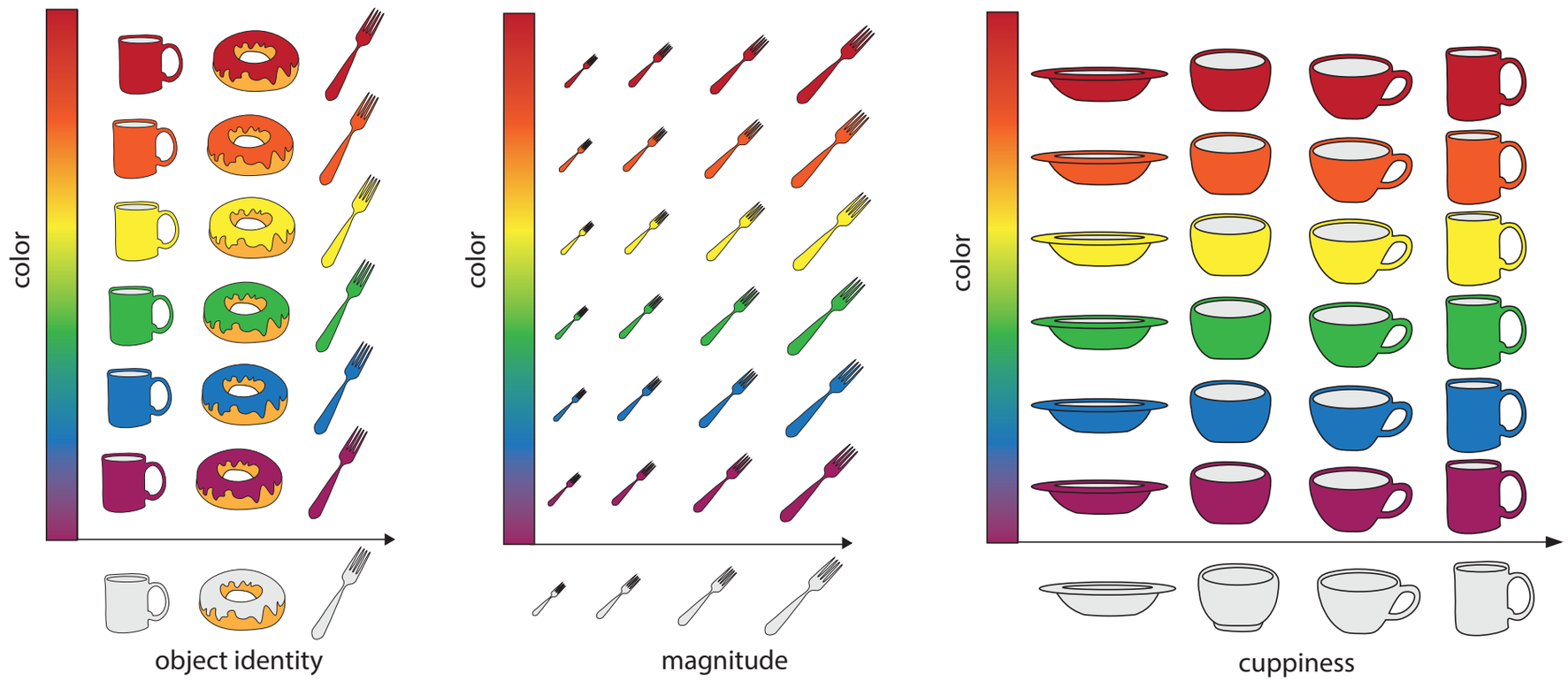
Place cells



Grid cells



Independent/disentangled/factorized components

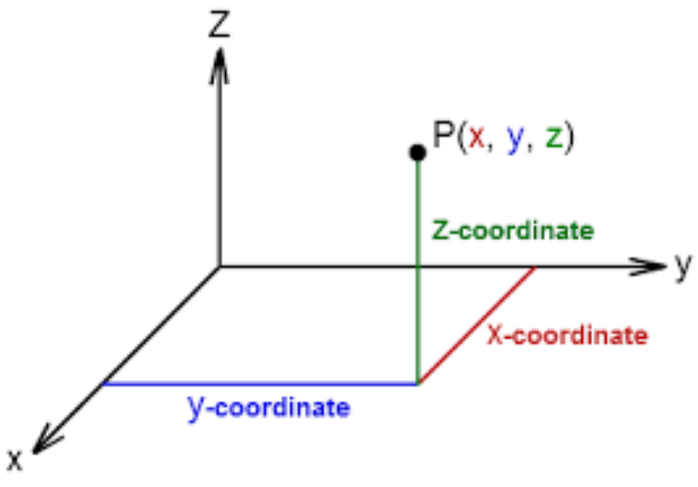


Behrens, Muller, Whittington, Mark, Baram, Stachenfeld, Kurth-Nelson (2018)

Spaces that aren’t Euclidean

Spaces that aren’t Euclidean

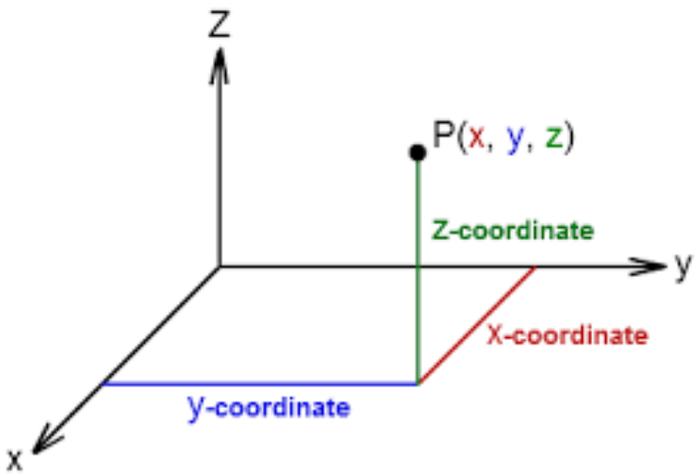
nD Euclidean
As in “of or pertaining to Euclid,”
that guy who did a lot of geometry
before fancy math was invented



hi

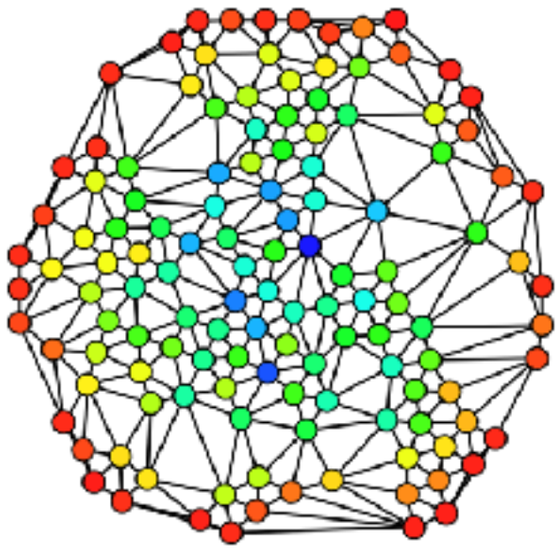
Spaces that aren't Euclidean

nD Euclidean
As in "of or pertaining to Euclid," that guy who did a lot of geometry before fancy math was invented

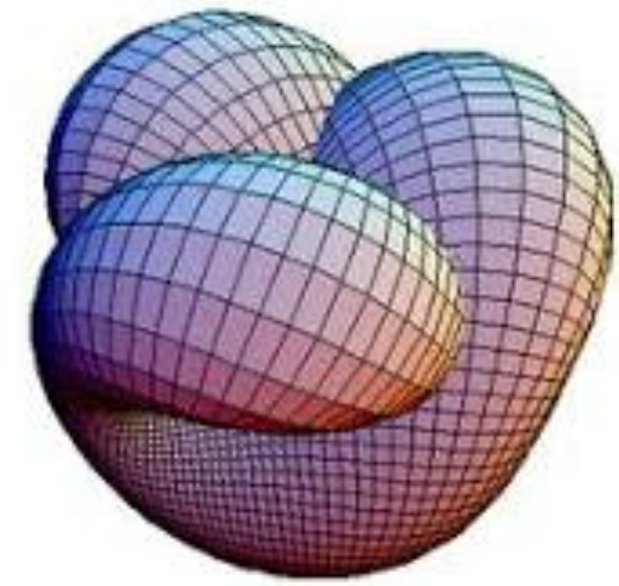


hi

Non-Euclidean
Fancy math spaces made out of more arbitrary relations



Graphs
Nodes connected by edges



Manifolds
"Locally Euclidean"

What makes a good representation?

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”
- **Categorical:** Assign stimuli to discrete object categories

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”
- **Categorical:** Assign stimuli to discrete object categories
- **Causal:** infer latent variables that cause observations, can be perturbed

What makes a good representation?

- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”
- **Categorical:** Assign stimuli to discrete object categories
- **Causal:** infer latent variables that cause observations, can be perturbed

Successor representations

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

- Decompose into future term and reward term…

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

- Decompose into future term and reward term…

value = sum expected future state visits · reward at each state visit

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

- Decompose into future term and reward term…

value = sum expected future state visits · reward at each state visit

depends on dynamics

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

- Decompose into future term and reward term…

value = sum expected future state visits · reward at each state visit

depends on dynamics

depends on reward

Image from www.jessicayung.com

Dayan (1993)

Successor representations

- RL says maximize value

value = sum expected future reward

- Decompose into future term and reward term…

value = sum expected future state visits · reward at each state visit

depends on dynamics

depends on reward

Successor Representation

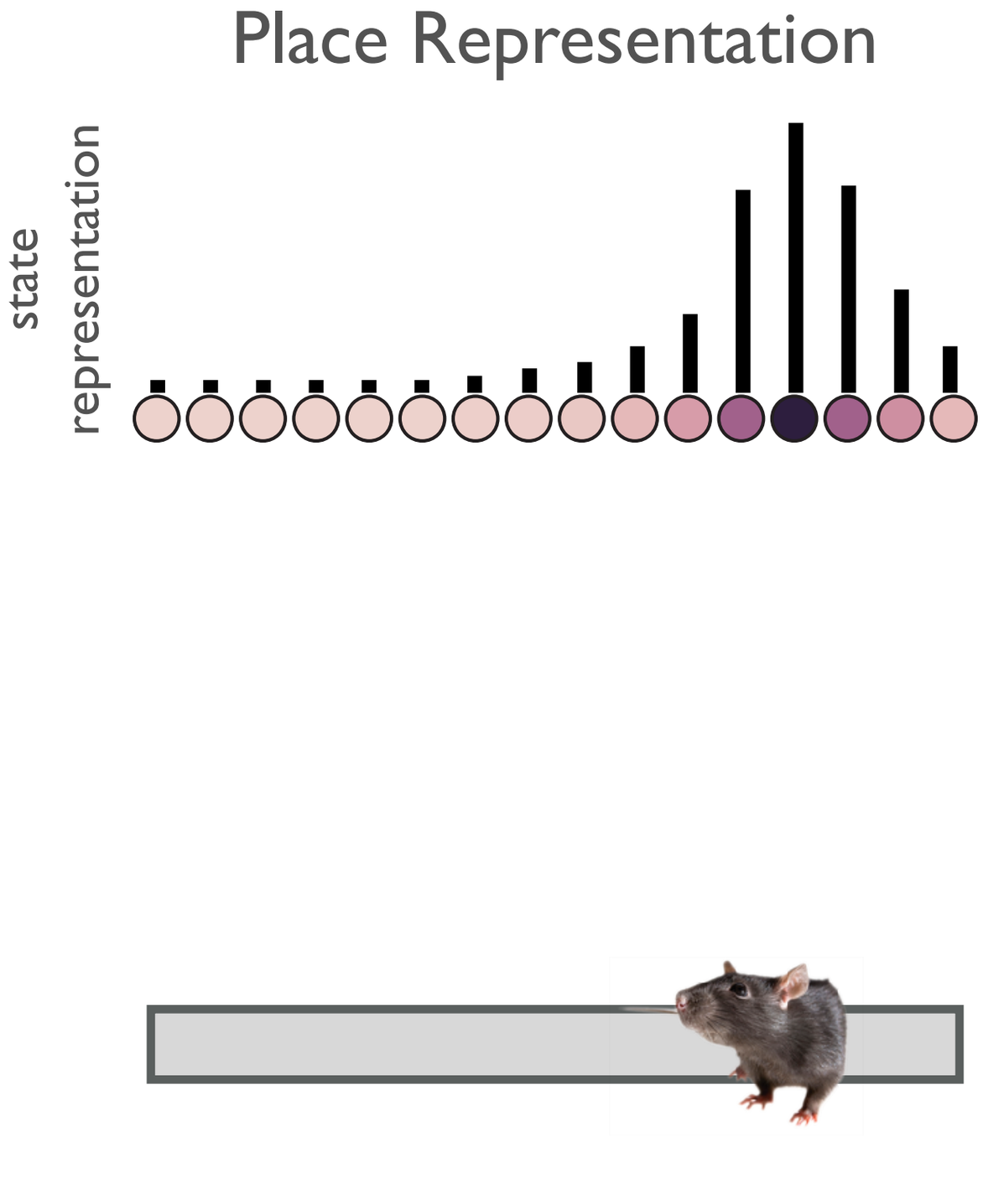
Image from www.jessicayung.com

Dayan (1993)

Successor representations in hippocampus

Stachenfeld, Botvinick, Gershman (2017)

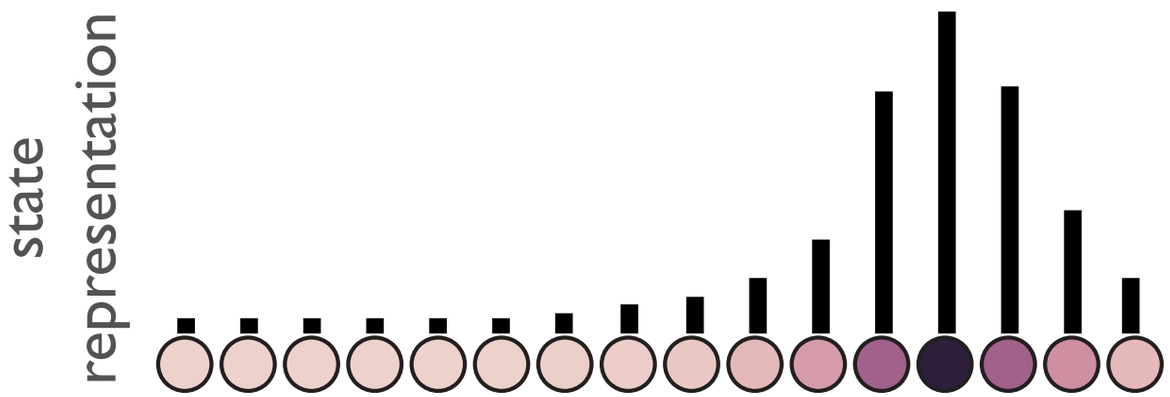
Successor representations in hippocampus



Stachenfeld, Botvinick, Gershman (2017)

Successor representations in hippocampus

Place Representation

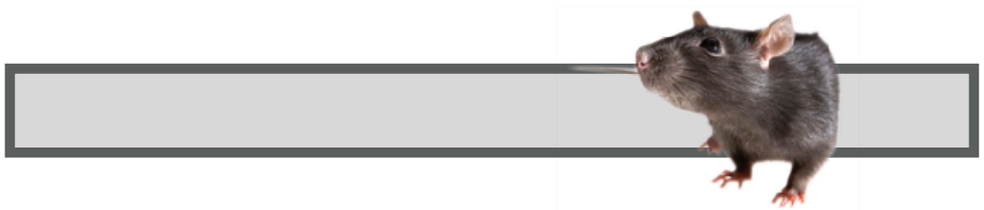


Firing of cell i
At location j

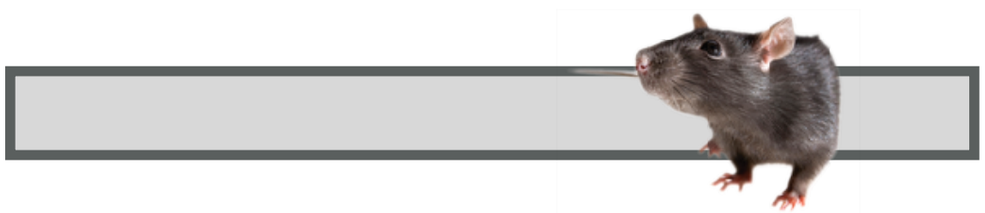
Cell i encoding distance from μ_i

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

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



x_j

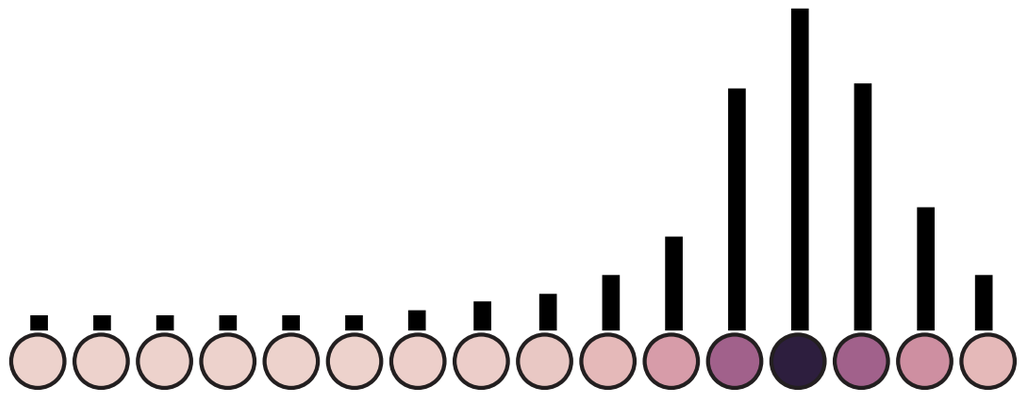


S_j

Successor representations in hippocampus

Place Representation

state representation

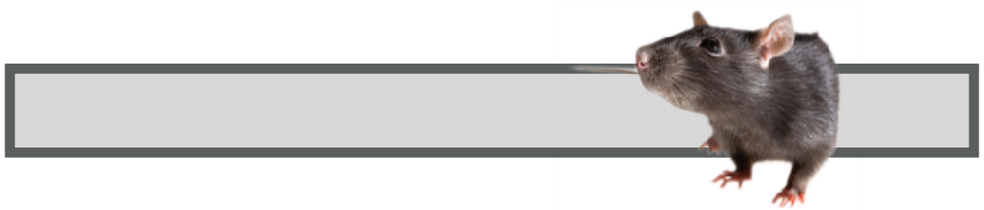


Firing of cell i
At location j

Cell i encoding distance from μ_i

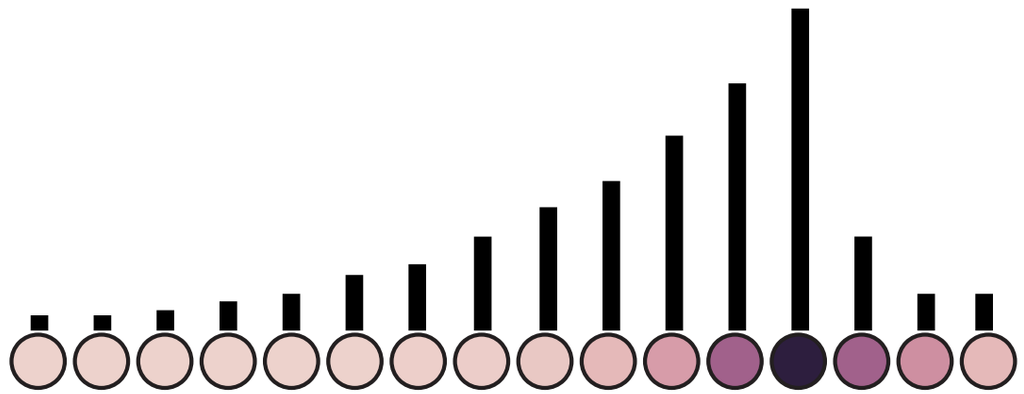
$$Ae^{-\left(\frac{x_j-\mu_i}{\sigma}\right)^2}$$

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



x_j

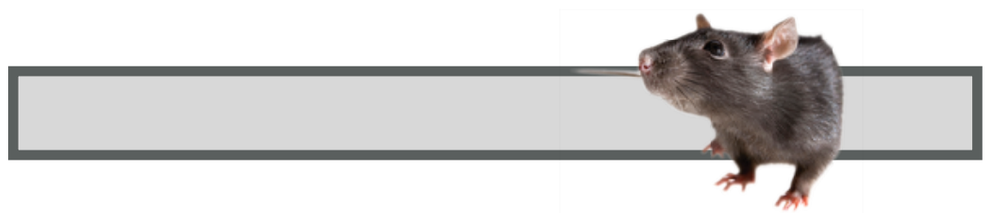
Successor Representation



Cell i encoding predictions about state i

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\mathbb{I}(s_t=i)\mid s_0=s_j\right]$$

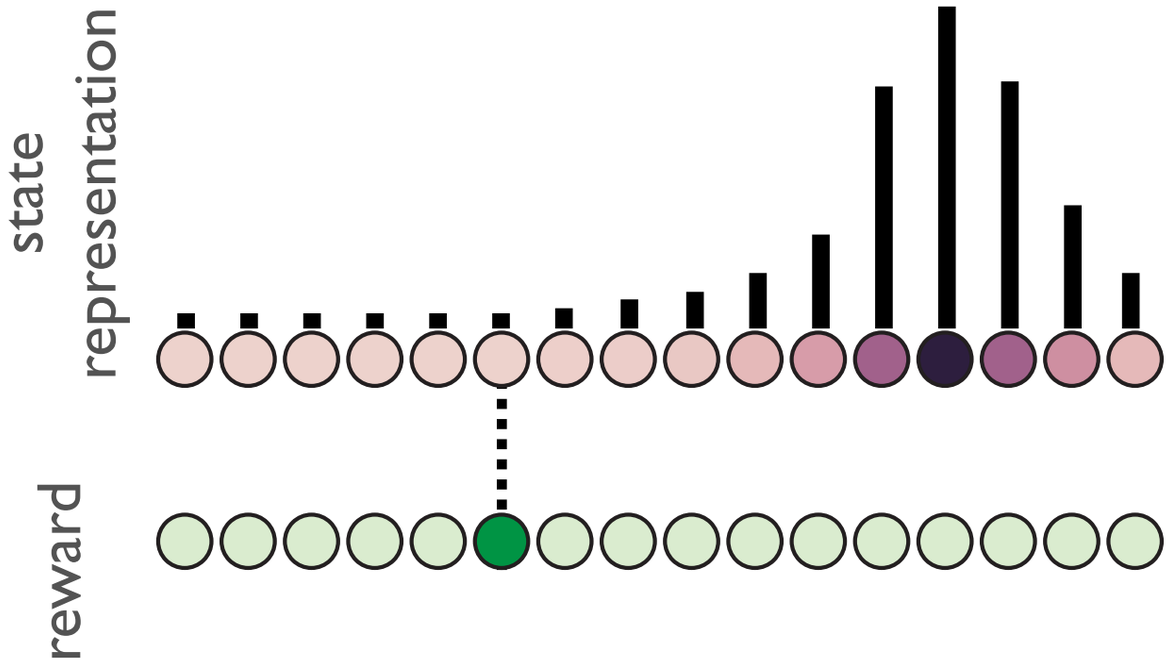
Expected #visits to state i
Discount γ



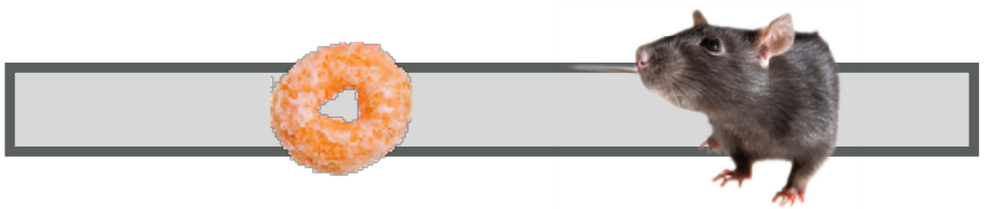
s_j

Successor representations in hippocampus

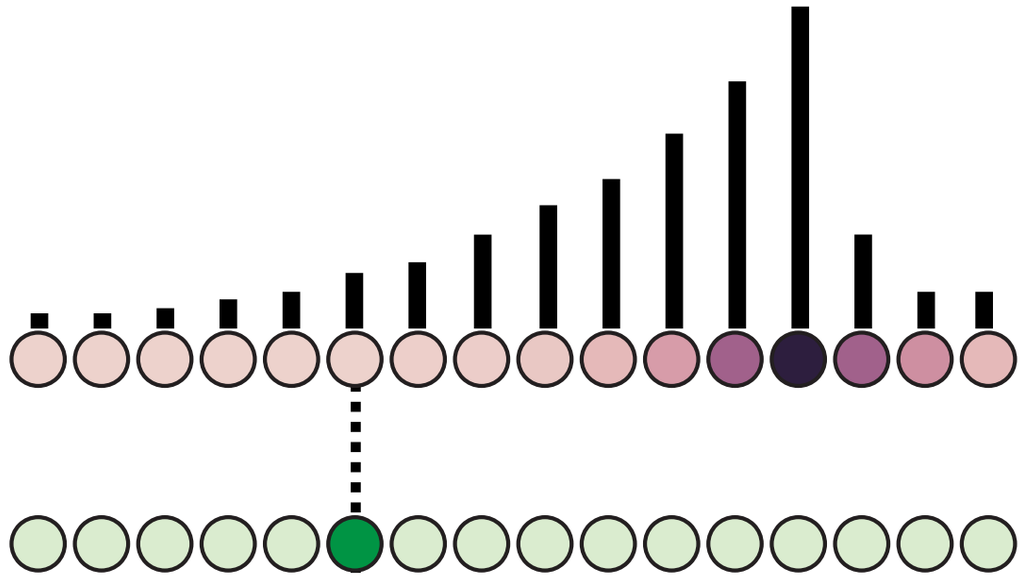
Place Representation



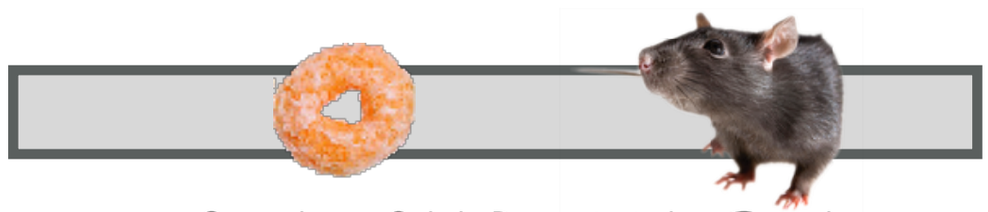
rep x reward =
current reward



Successor Representation



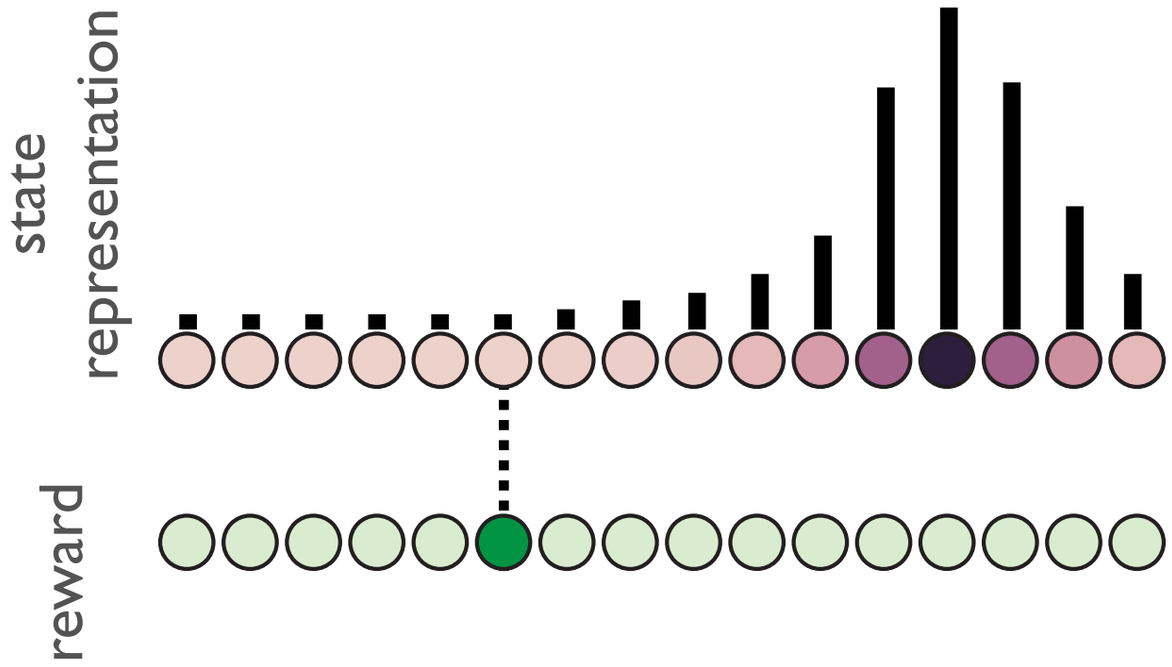
rep x reward =
exp. **future** reward =
Value



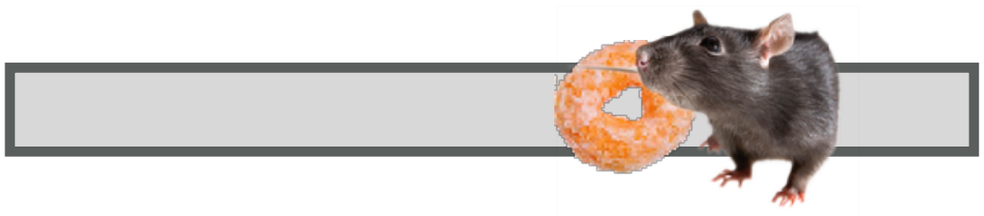
Stachenfeld, Botvinick, Gershman (2017)

Successor representations in hippocampus

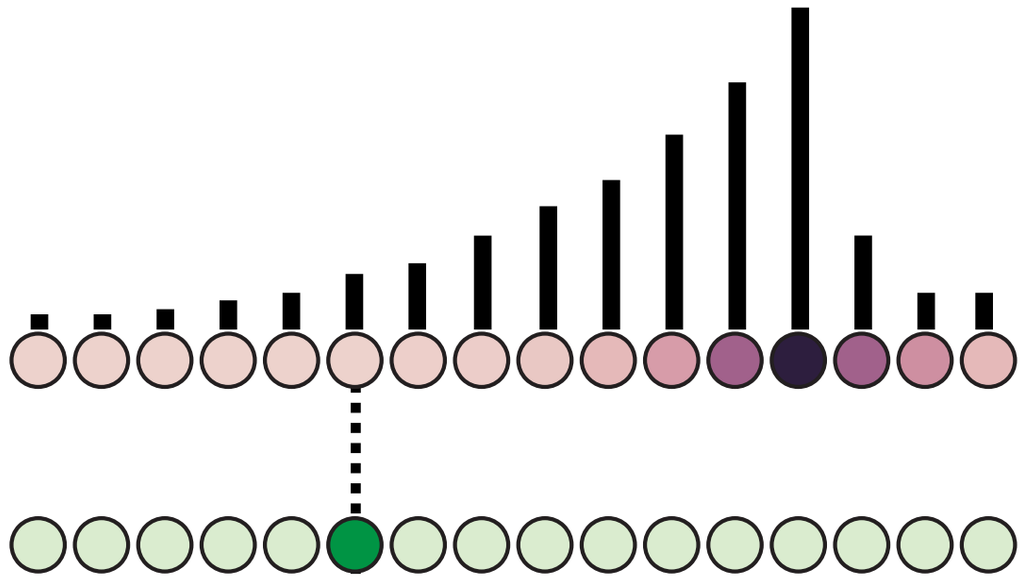
Place Representation



rep x reward =
current reward



Successor Representation



rep x reward =
exp. **future** reward =
Value

Stachenfeld, Botvinick, Gershman (2017)

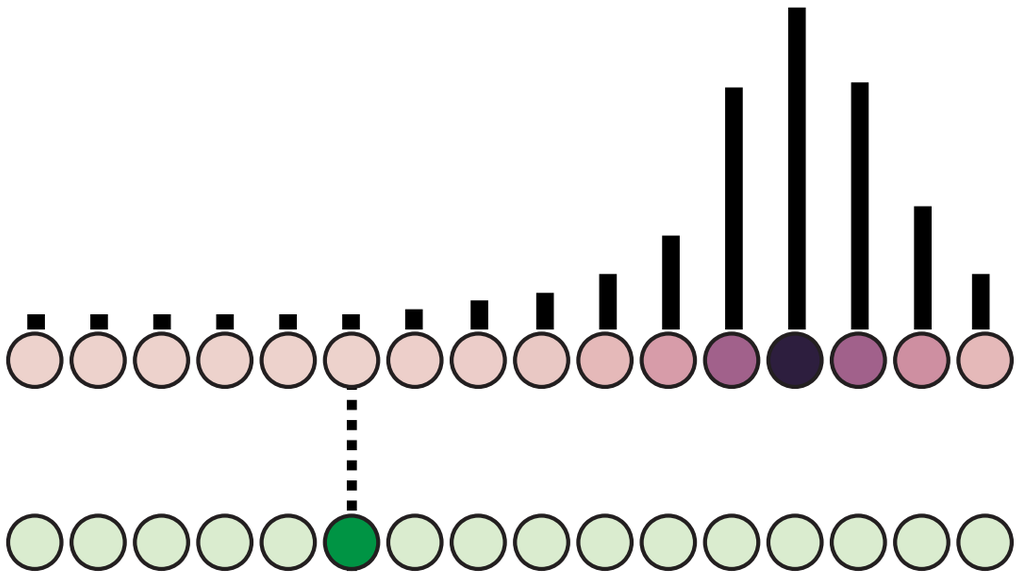
Successor representations in hippocampus

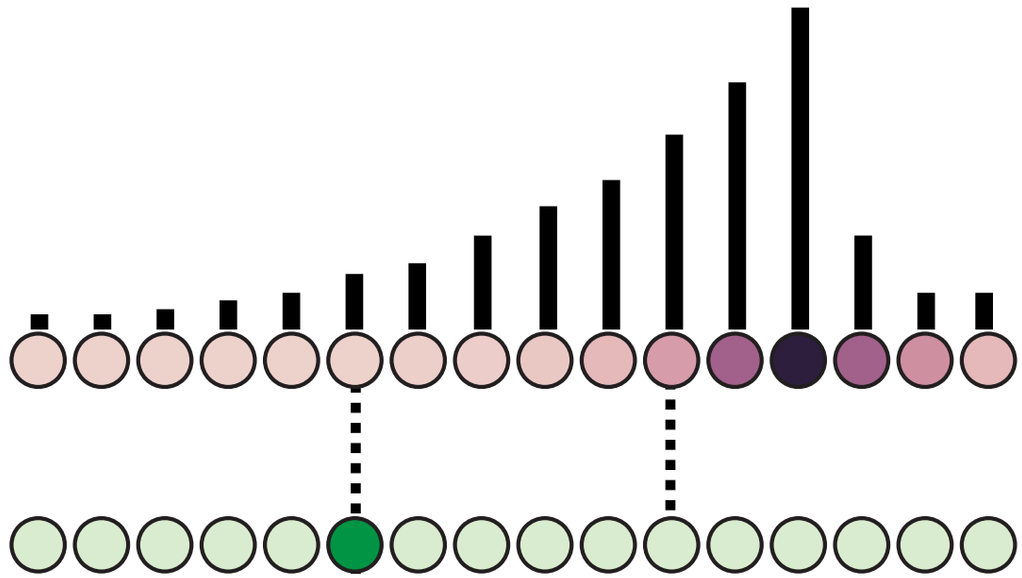
Place Representation

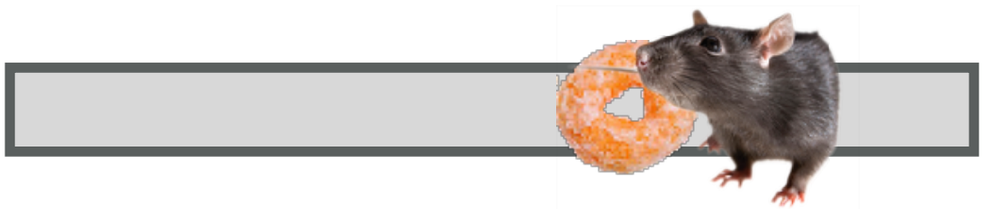
Successor Representation

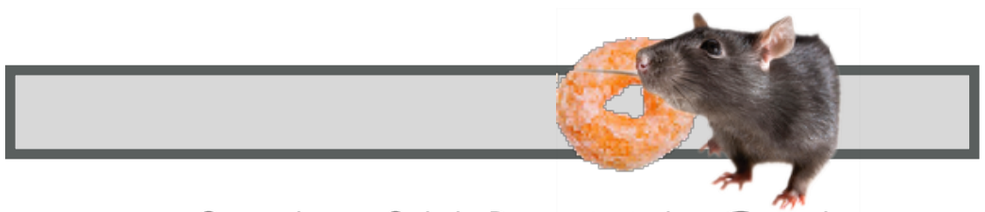
state representation

reward









Stachenfeld, Botvinick, Gershman (2017)

Why is this a good representation?

Why is this a good representation?

- Flexible computation of value when reward changes faster than policy because transition information is preserved in the representation

Why is this a good representation?

- Flexible computation of value when reward changes faster than policy because transition information is preserved in the representation

- **Generalization** over states that predict similar outcomes

Why is this a good representation?

- Flexible computation of value when reward changes faster than policy because transition information is preserved in the representation

- **Generalization** over states that predict similar outcomes

- Model defined with respect to states + temporal adjacency, not spatial adjacency

Why is this a good representation?

- Flexible computation of value when reward changes faster than policy because transition information is preserved in the representation

- **Generalization** over states that predict similar outcomes

- Model defined with respect to states + temporal adjacency, not spatial adjacency

- Can be learned incrementally with TD learning

Why is this a good representation?

- Flexible computation of value when reward changes faster than policy because transition information is preserved in the representation

- **Generalization** over states that predict similar outcomes

- Model defined with respect to states + temporal adjacency, not spatial adjacency

- Can be learned incrementally with TD learning

- **Important limitation:** still not as flexible as model-based planning

Some simulations

Stachenfeld, Botvinick, Gershman (2017)

Some simulations

- Asymmetric experience dependent expansion (Mehta et al 2000)

Stachenfeld, Botvinick, Gershman (2017)

Some simulations

- Asymmetric experience dependent expansion (Mehta et al 2000)
- Obstacle-induced effects (Alvernhe et al 2011)

Stachenfeld, Botvinick, Gershman (2017)

Some simulations

- Asymmetric experience dependent expansion (Mehta et al 2000)
- Obstacle-induced effects (Alvernhe et al 2011)
- Spatiotemporal distance effects (Deuker et al 2016)

Stachenfeld, Botvinick, Gershman (2017)

Some simulations

- Asymmetric experience dependent expansion (Mehta et al 2000)
- Obstacle-induced effects (Alvernhe et al 2011)
- Spatiotemporal distance effects (Deuker et al 2016)
- Temporal clustering effects (Schapiro et al 2013)

Stachenfeld, Botvinick, Gershman (2017)

Some simulations

- Asymmetric experience dependent expansion (Mehta et al 2000)
- Obstacle-induced effects (Alvernhe et al 2011)
- Spatiotemporal distance effects (Deuker et al 2016)
- Temporal clustering effects (Schapiro et al 2013)
- See also: Russek et al (2017), Momennejad et al (2017), Garvert et al (2017), Bellmund et al (in prep)

Stachenfeld, Botvinick, Gershman (2017)

What else is happening with the SR?

- Make it deeper! (Kulkarni et al 2016)
- Make it transfer better! (Barreto et al 2016, “USFAs” Borsa et al 2019, Geerts et al (2019) CCN, Madarasz (2019) Cosyne)
- Integrate it with model-based rollouts and/or replay! (Momennejad et al (2017), Russek et al (2017))
- Make it probabilistic! (Geerts et al (2019) CCN, Vertesz et al (2019) Cosyne)
- Make it multiscale! (Stachenfeld et al 2017, Momennejad & Howard 2018 biorXiv)

Make it transfer better! cont.

There's been a lot of recent work about improving SR transfer when policy changes following a reward change. Here are some quick notes on these (added posthoc). Note that replay/SR-DYNA is also a way to do this — roughly the tradeoff is that replay takes more time to update but the below take more space.

Barreto et al 2016: procedure for learning a whole bunch of successor feature matrices and approximating a new one by linearly combining the old ones as a function of the new reward vector. a successor feature matrix is like a successor representation, except each row/column is a feature, not a state, and each entry says how much each feature predicts each other feature.
https://arxiv.org/abs/1606.05312

Universal Successor Feature Approximators ("USFAs") Borsa et al 2019: basically learn a set of features that can be recombined (linearly or by neural network) to approximate a new successor feature representation as a function of the new goal. Kind of like the previous one, except you're learning the best features to keep rather than just making the best of whatever library of successor feature matrices you've kept.
https://arxiv.org/abs/1812.07626 (about USFAs)
https://arxiv.org/abs/1901.10964 (about using them)
https://arxiv.org/abs/1804.03758 (related: Universal successor representations)

Madarasz (2019) Cosyne): learn a bunch of successor representations corresponding to different rewards, cluster the rewards, use a new reward to infer the appropriate successor representation (similar to USFAs and USRs but set up differently, more probabilistically rather than neural networkily/linearly, and related to hippocampal splitting data).
https://arxiv.org/abs/1906.07663

Geerts et al (2019) CCN abstract <-- done by a student I'm advising, so minor plug: a probabilistic approach that infers a distribution over SRs (rather than just the mean expected SR) using Kalman Filtering. Uses covariance matrix for updates, which can capture both uncertainty about an entry and dependencies between different entries in the matrix, meaning you can update correlated predictions together.

What makes a good representation?

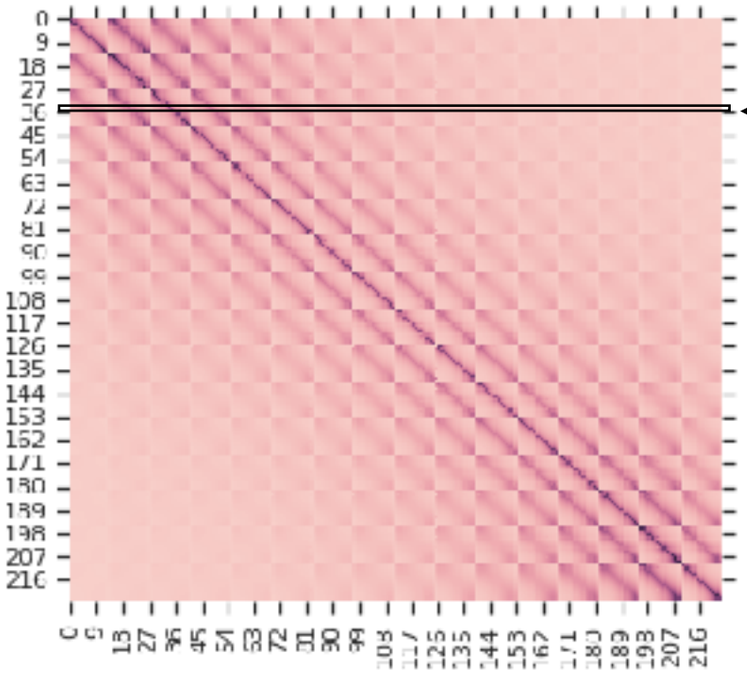
- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”
- **Categorical:** Assign stimuli to discrete object categories
- **Causal:** infer latent variables that cause observations, can be perturbed

What makes a good representation?

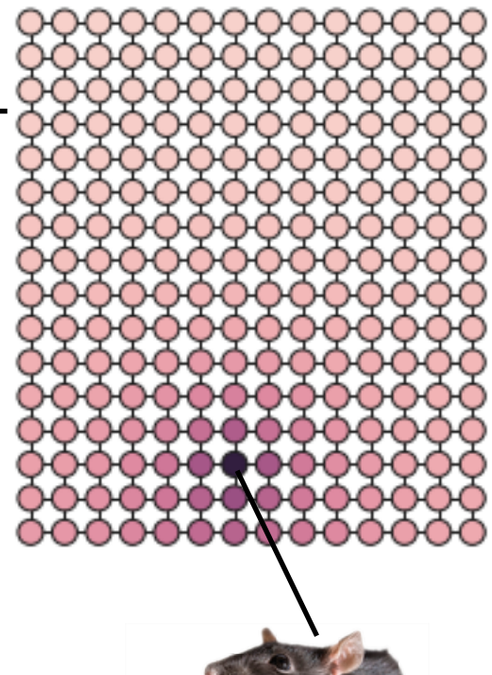
- **Predictive:** cluster states that predict similar outcomes (e.g. successor representation)
- **Low-dimensional:** the fewer dimensions you have to learn about, the better
- **Hierarchical:** option of “coarse-grained” or “fine-grained”
- **Transferrable:** same representation supports lots of different tasks
- **Disentangled/composable:** dimensions that can be reasoned about independently are orthogonal
- **Topological/relational:** Represent task “connectivity”
- **Categorical:** Assign stimuli to discrete object categories
- **Causal:** infer latent variables that cause observations, can be perturbed

Dimensionality reduction

Successor Representation Matrix

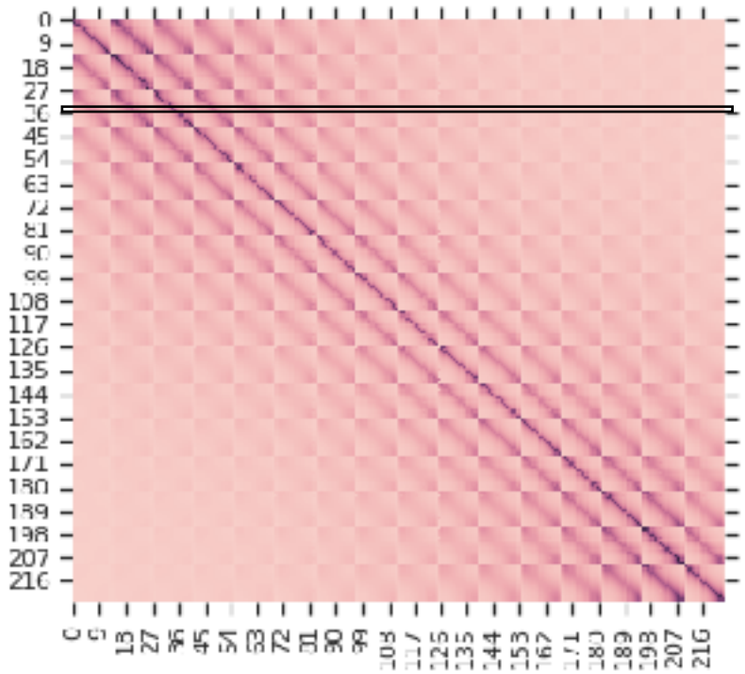


e.g. SR at state 34



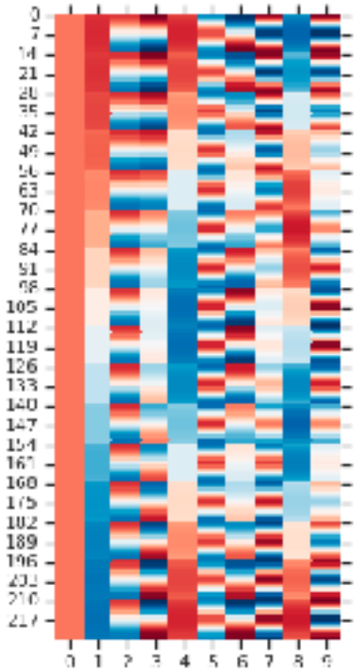
Dimensionality reduction

Successor Representation Matrix M

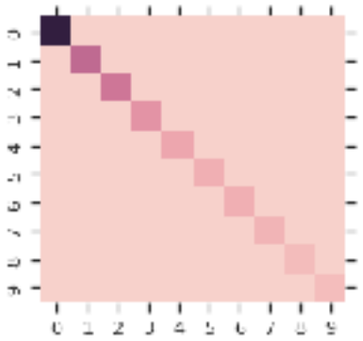


≈

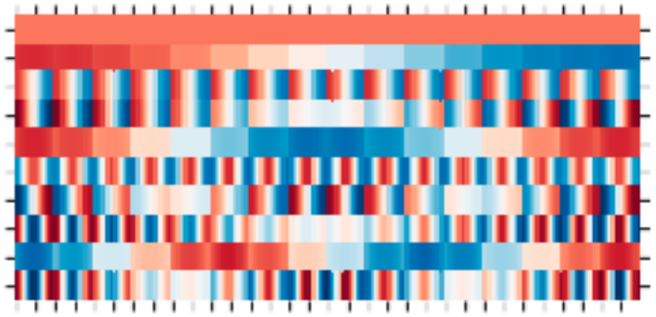
Eigenvectors



Eigenvalues

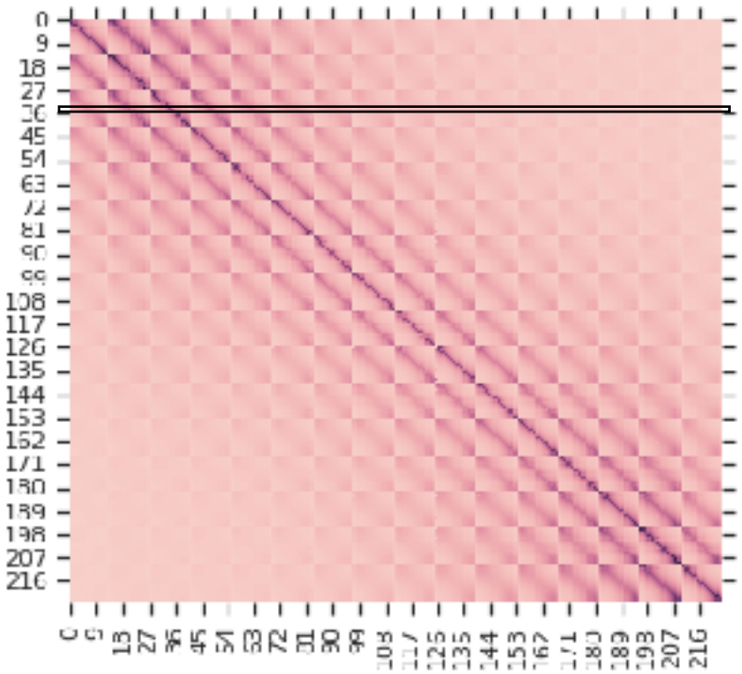


Eigenvectors



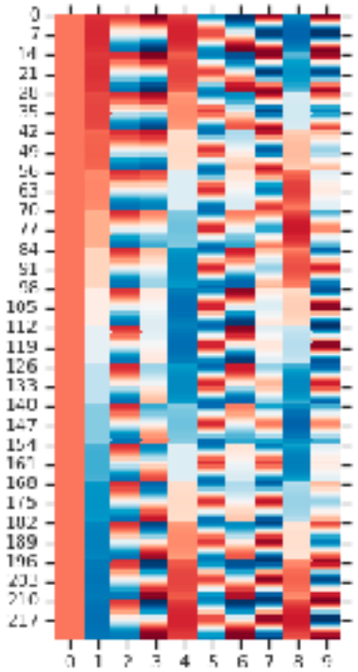
Dimensionality reduction

Successor Representation Matrix M

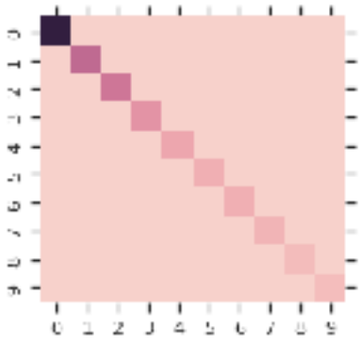


≈

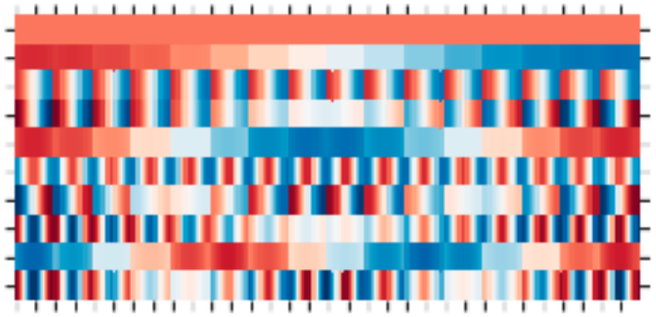
Eigenvectors



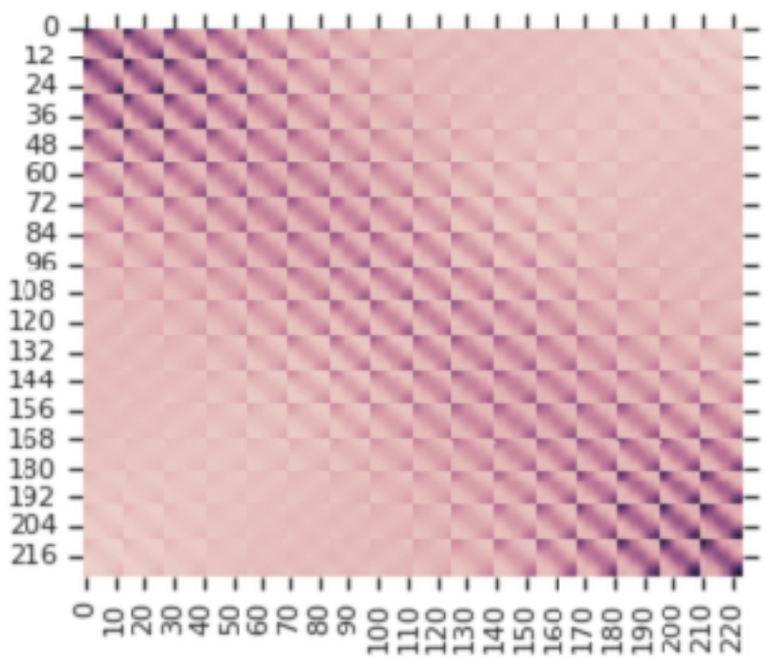
Eigenvalues



Eigenvectors

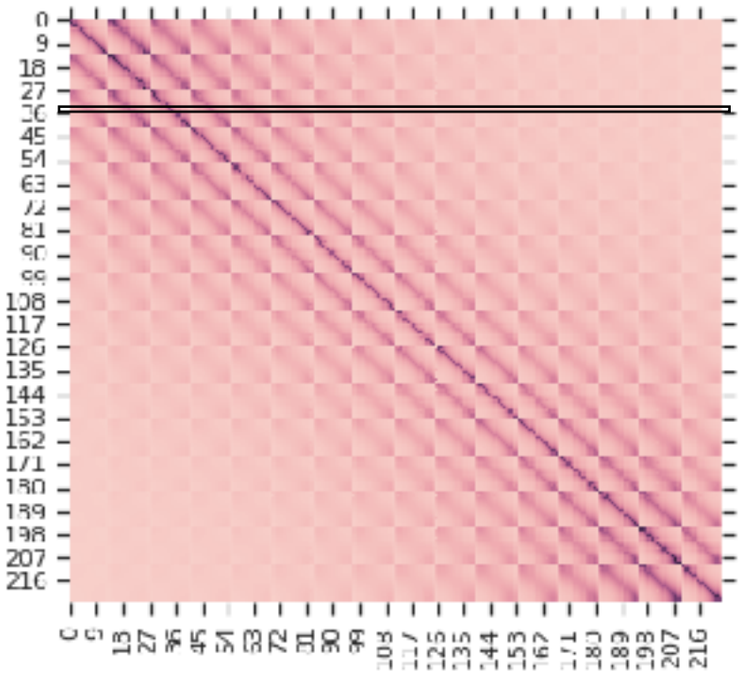


Reconstructed low-rank M



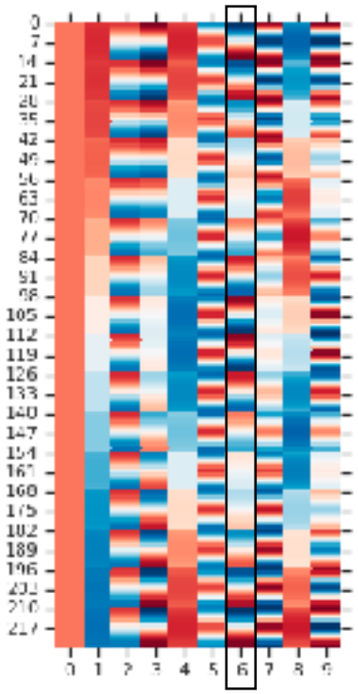
Dimensionality reduction

Successor Representation Matrix M

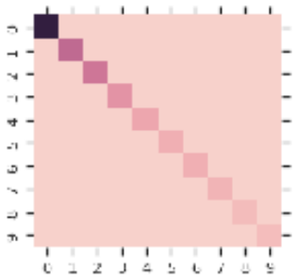


≈

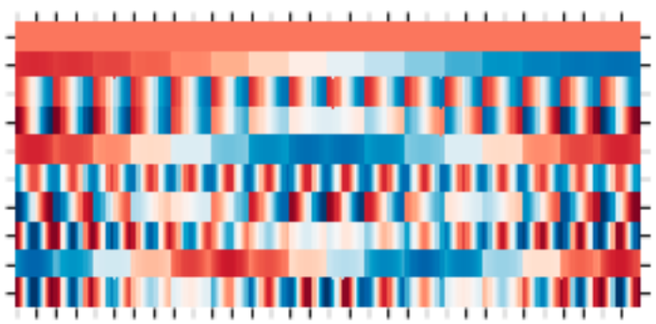
Eigenvectors



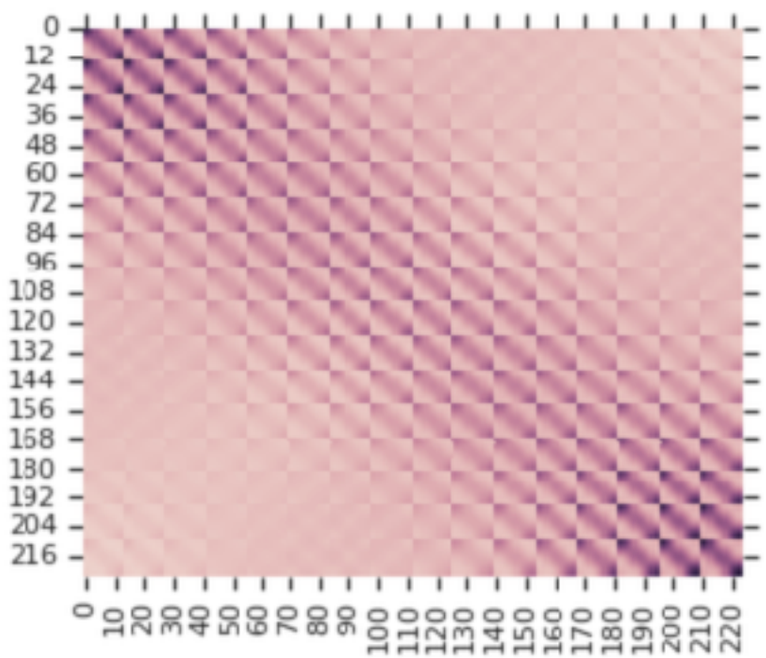
Eigenvalues



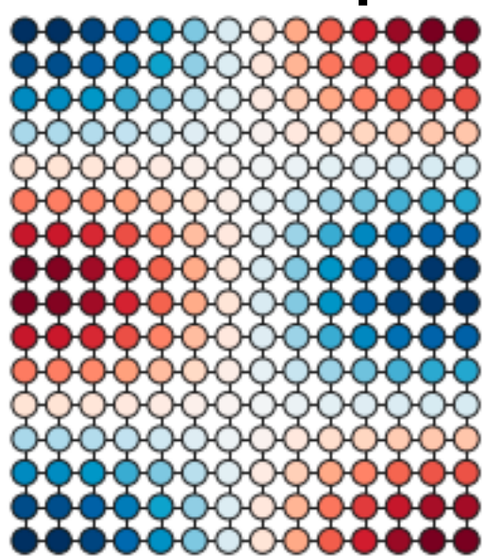
Eigenvectors



Reconstructed low-rank M



Eigenvector “receptive field”

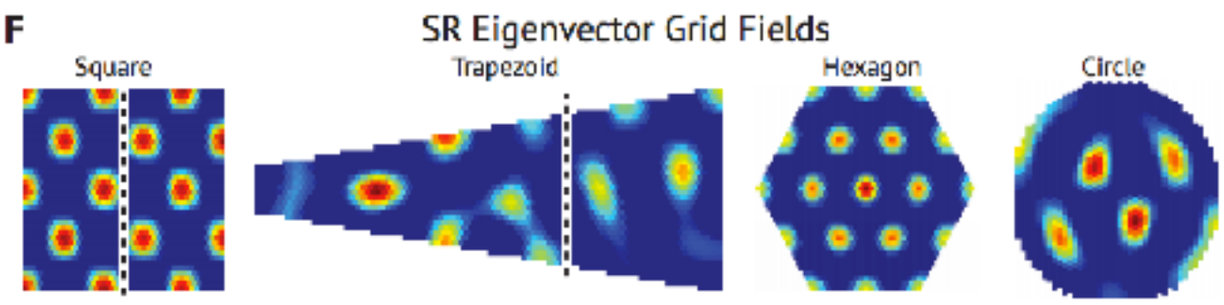


Simulating Grid Cells

Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

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

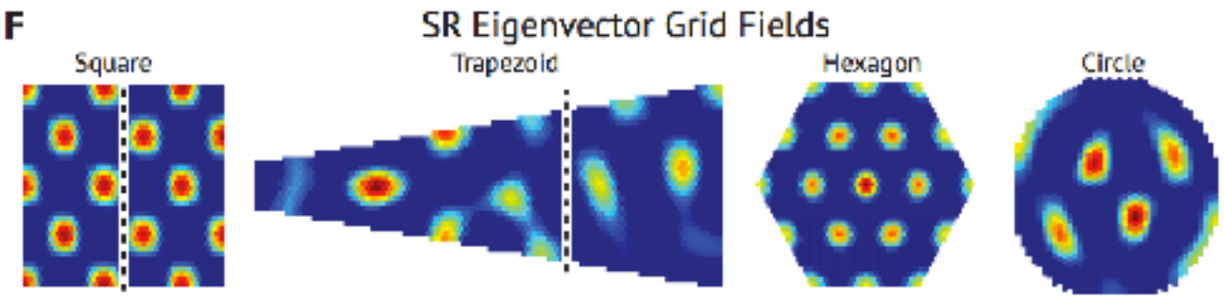


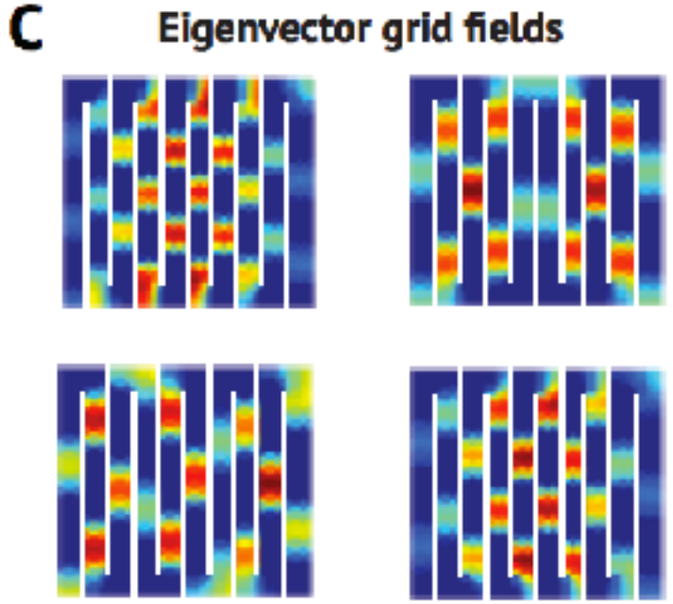
Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

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

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





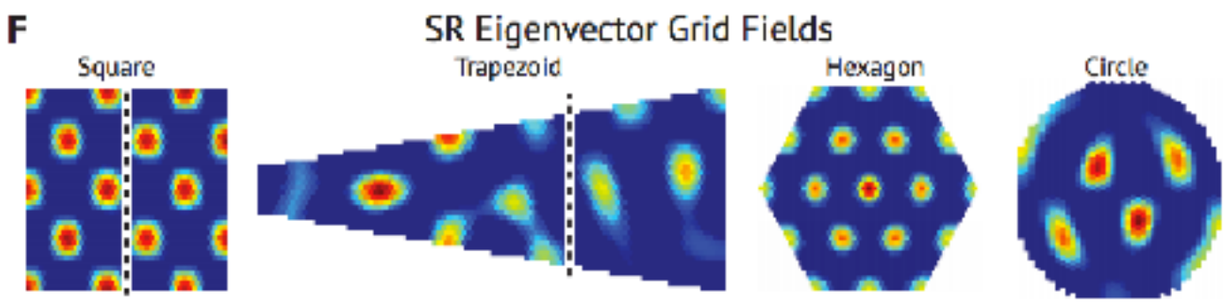
Stachenfeld, Botvinick, Gershman (2017)

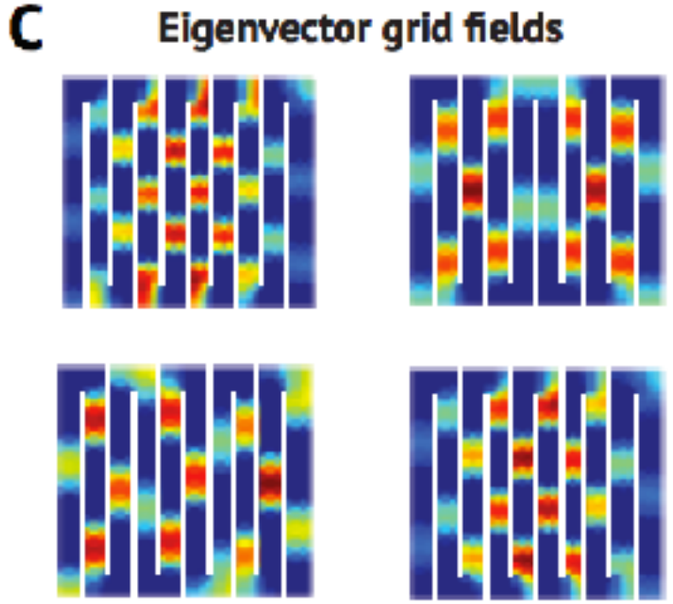
Simulating Grid Cells

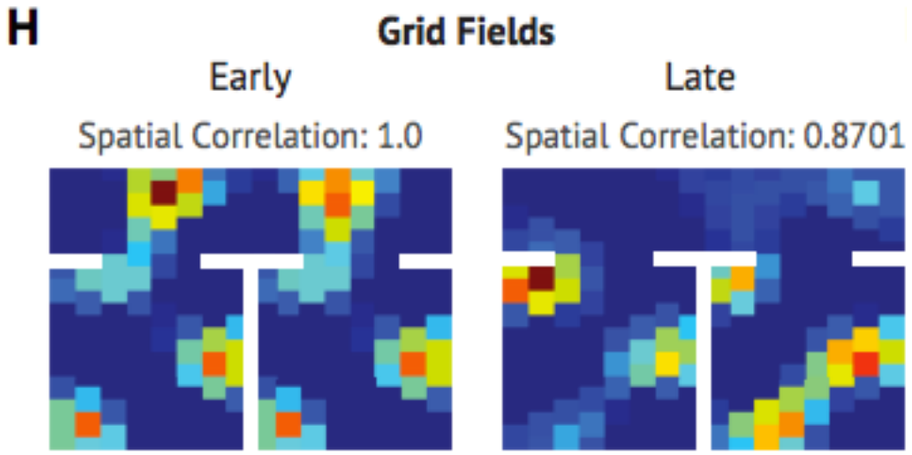
- Grid fields in geometric environments
  Krupic et al. (2013)

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

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







Stachenfeld, Botvinick, Gershman (2017)

Simulating Grid Cells

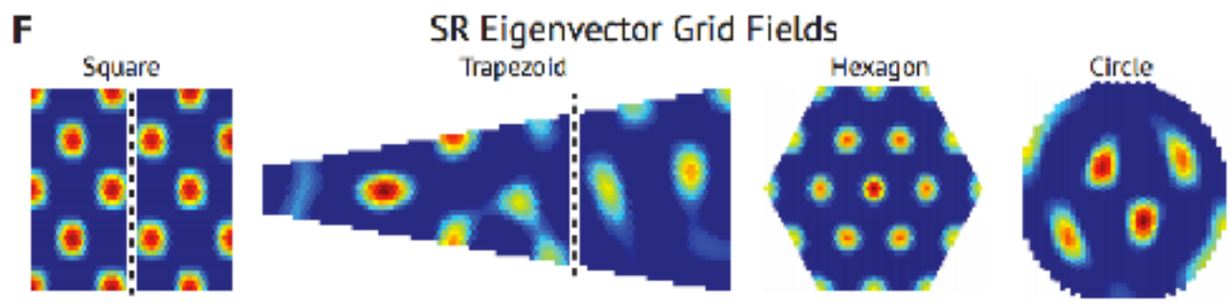
- Grid fields in geometric environments
  Krupic et al. (2013)

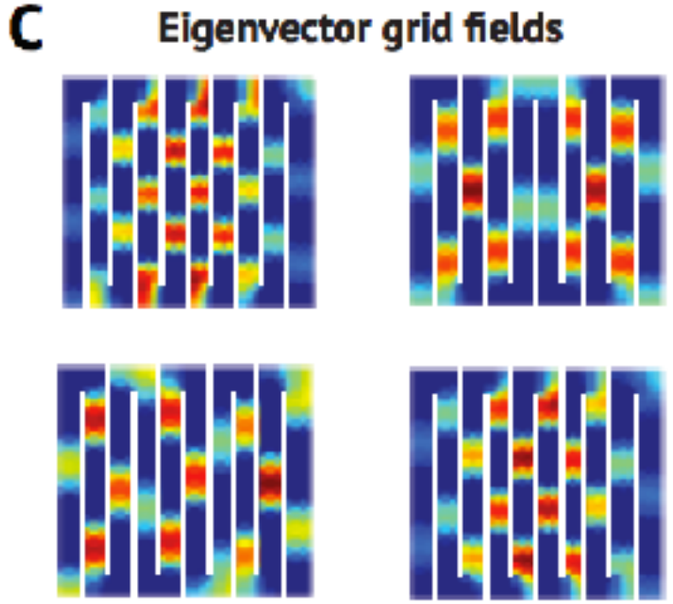
- Fragmentation in Hairpin Maze
  Derdikman et al. (2009)

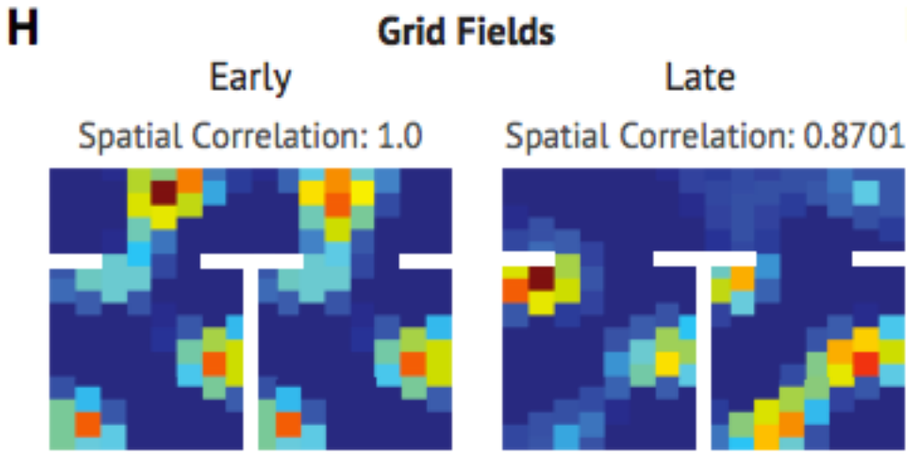
- Grid field learning in a multicompartment environment
  Carpenter et al. (2017)

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

Still missing some constraints



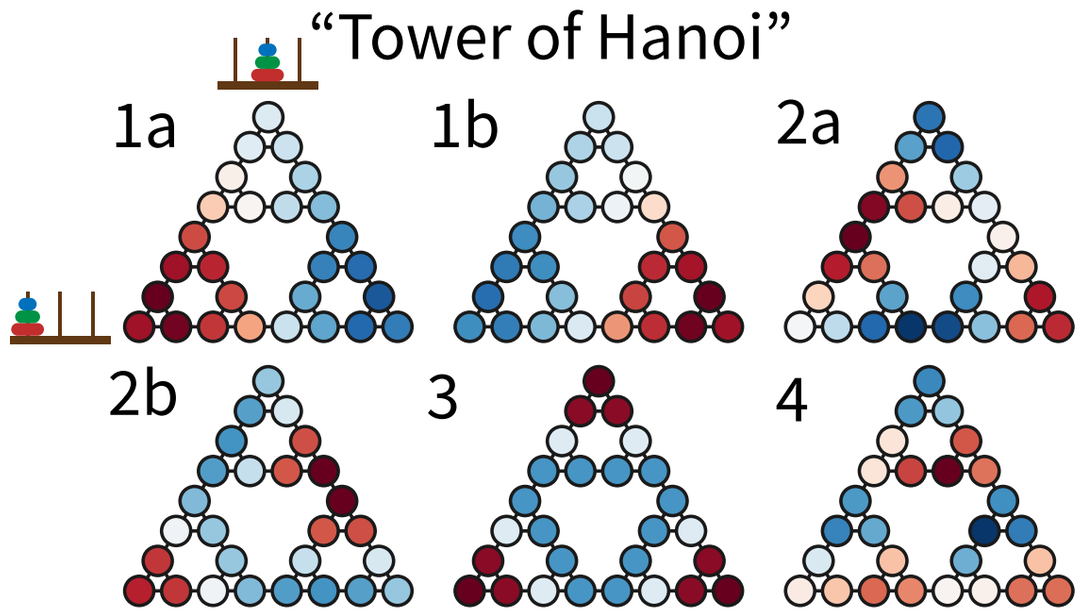


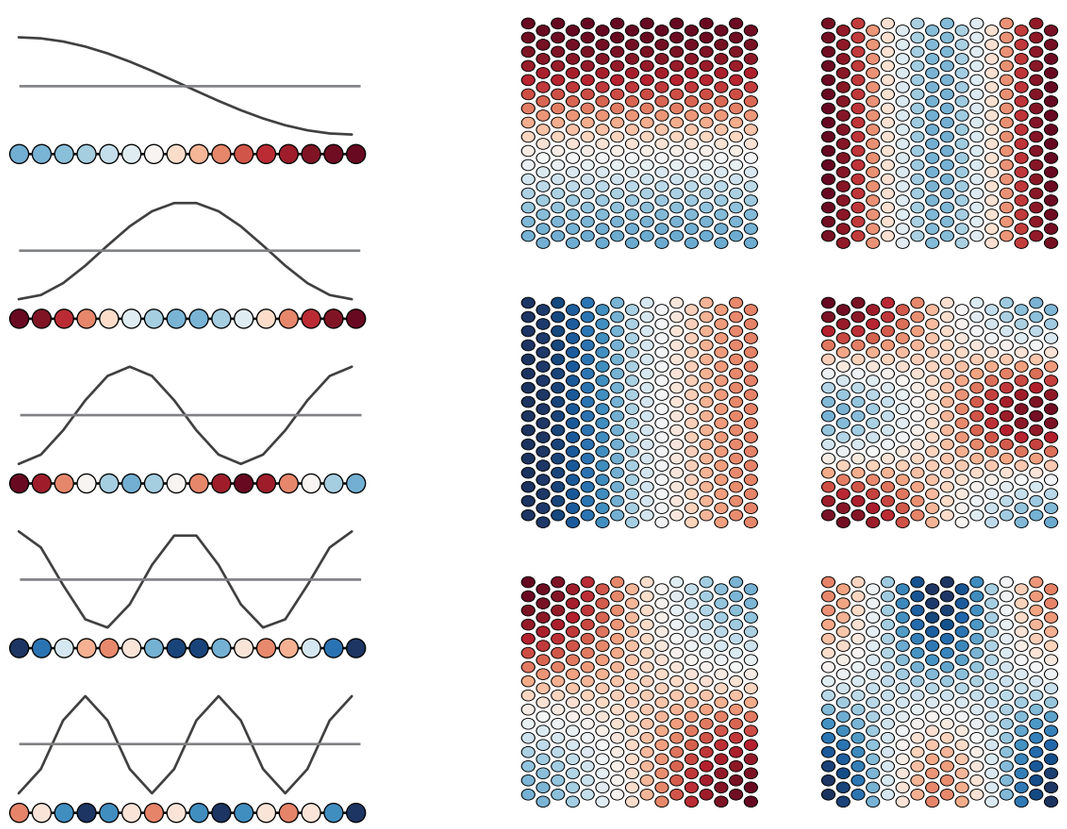


Stachenfeld, Botvinick, Gershman (2017)

These eigs have a lot of nice applications

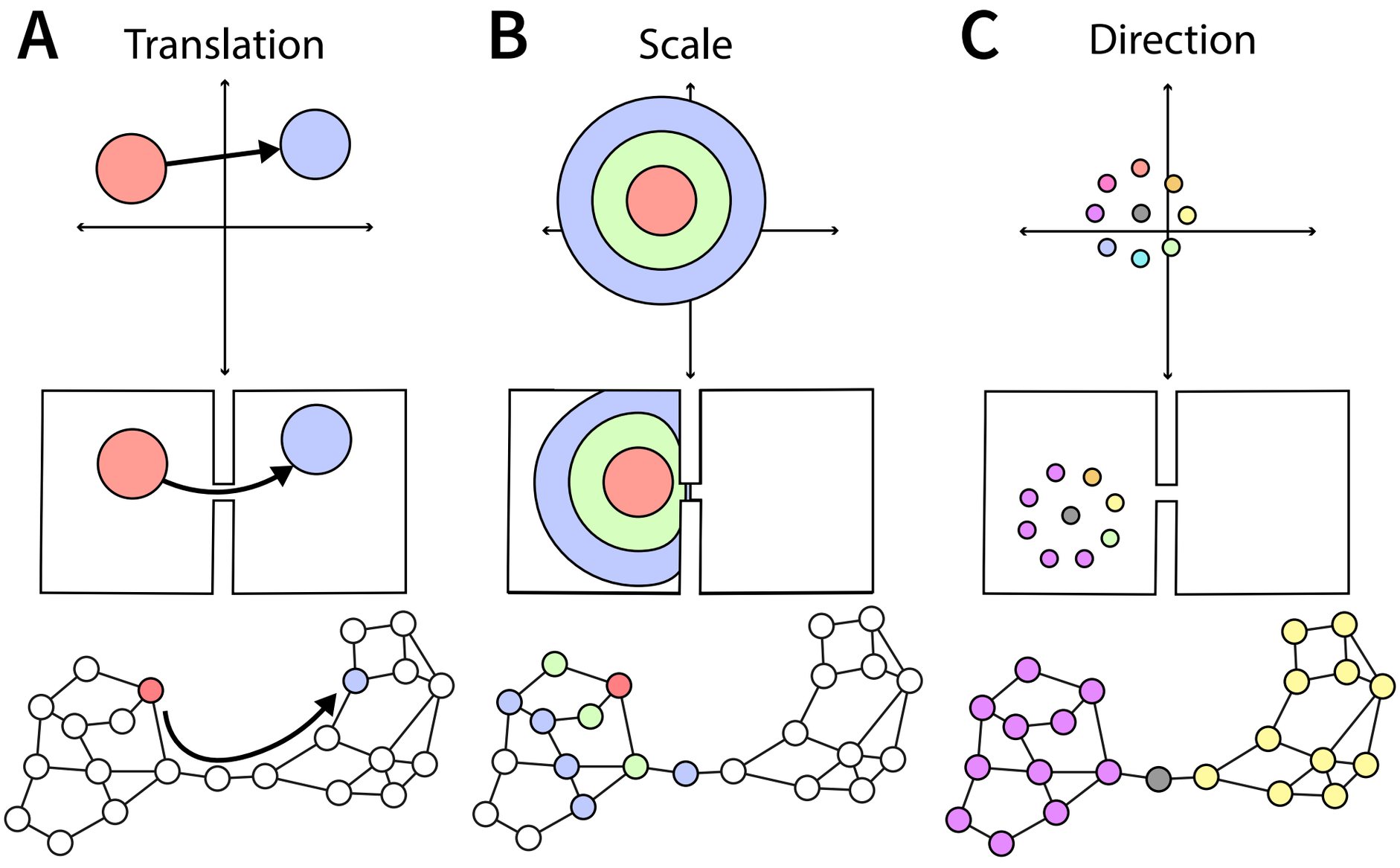
- **Sort of like PCA over transition structure**
  - Can help you smoothly approximate transition structure
- **Generalization of Fourier to graphs**
  Generalize spatial features to arbitrary geometries





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

Generalizing map ideas beyond the plane



Not obvious how these spatial constructs should apply to non-Euclidean spaces

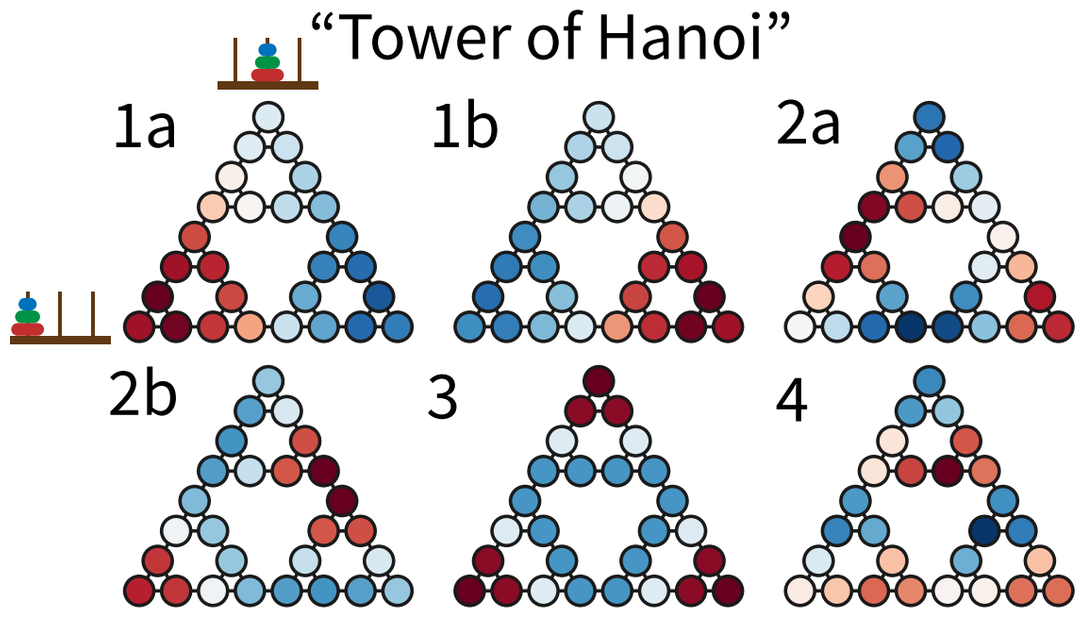
But they do have meanings in Fourier space

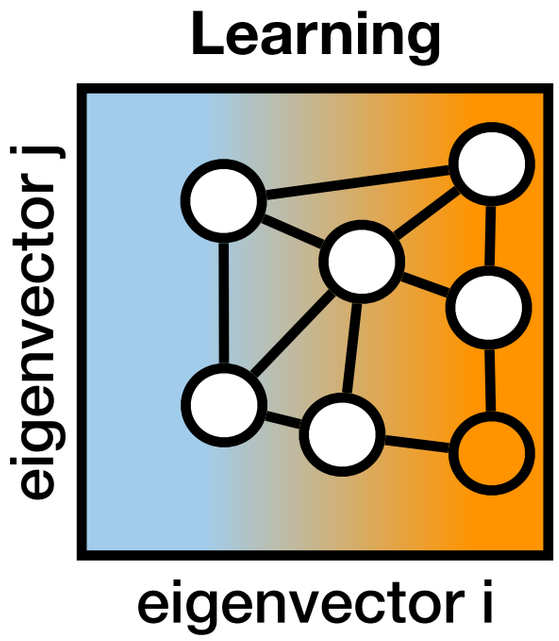
Since eigenvectors = “graph Fourier”, this is a definition that transfers

Stachenfeld, McNamee, Botvinick, Gershman (in prep)

These eigs have a lot of nice applications

- **Sort of like PCA over transition structure**
  - Can help you smoothly approximate transition structure
- **Generalization of Fourier to graphs**
  Generalize spatial features to arbitrary geometries
- **For supporting learning:** “Protovalue functions” (Mahadevan (2007))
  - Learn a value function over Laplacian eigenvectors instead of over states directly

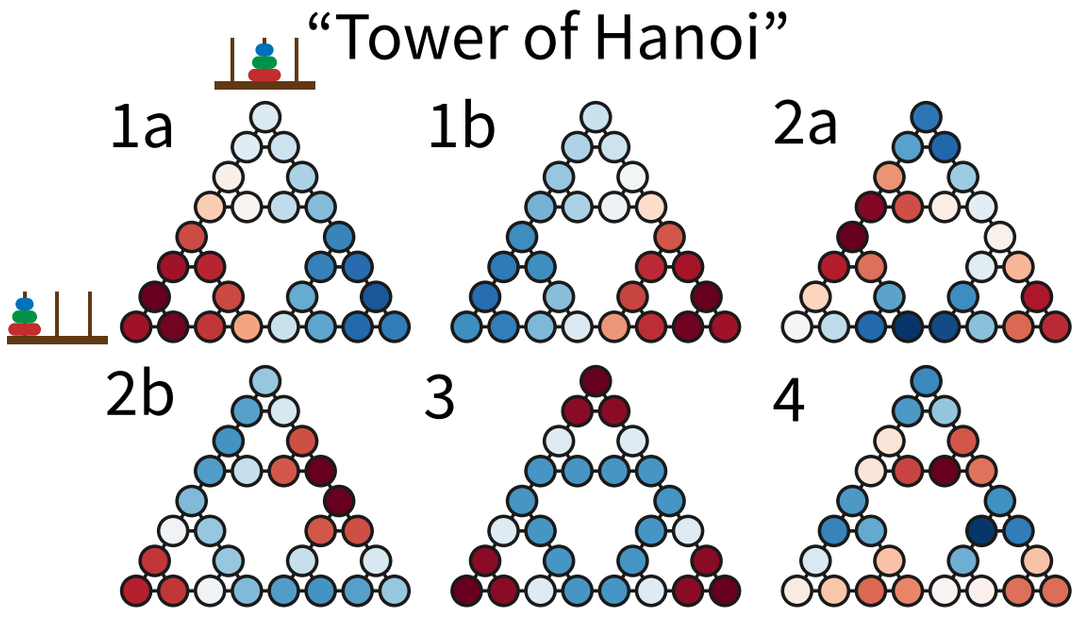


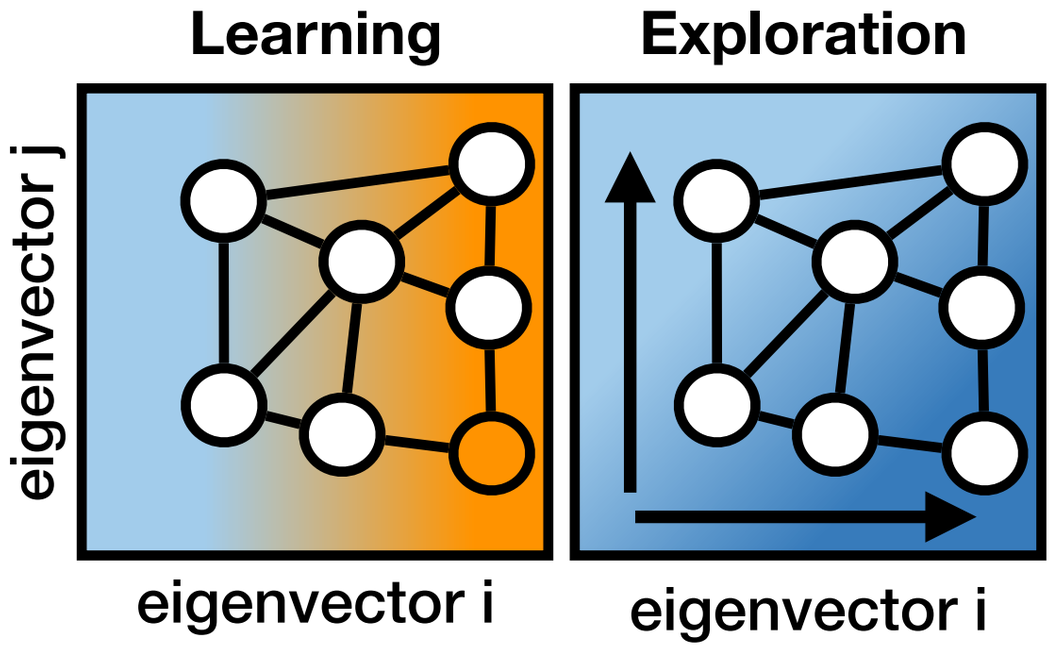


Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

These eigs have a lot of nice applications

- **Sort of like PCA over transition structure**
  - Can help you smoothly approximate transition structure
- **Generalization of Fourier to graphs**
  Generalize spatial features to arbitrary geometries
- **For supporting learning:** “Protovalue functions” (Mahadevan (2007))
  - Learn a value function over Laplacian eigenvectors instead of over states directly
- **For supporting exploration:**
  “Eigenoptions”
  Machado et al (2017a,b—2018)
  - Use Laplacian eigenvectors as options for exploration

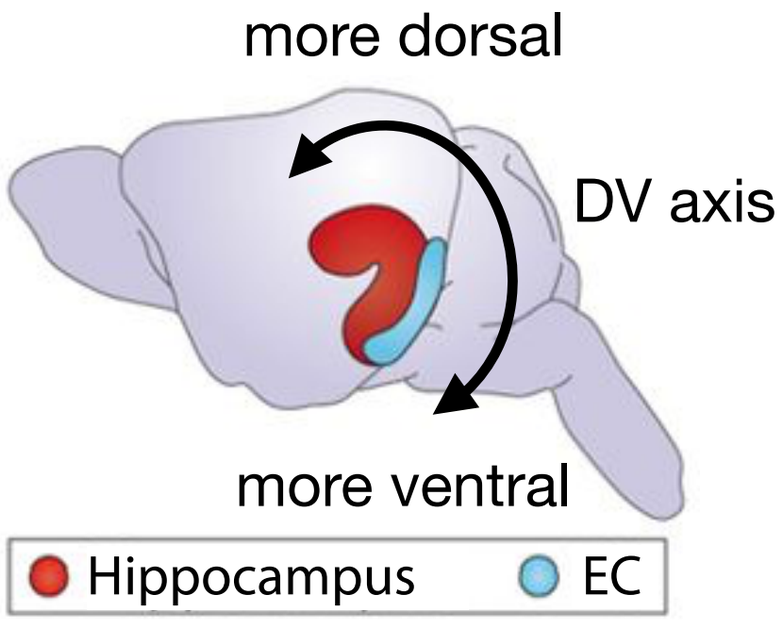




Stachenfeld, McNamee, Behrens, Gershman, Botvinick (review article in prep)

Multi-scale representations

Same eigenvectors support different coarseness

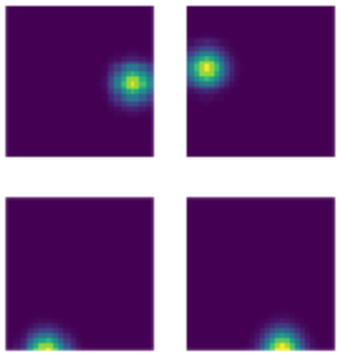


Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

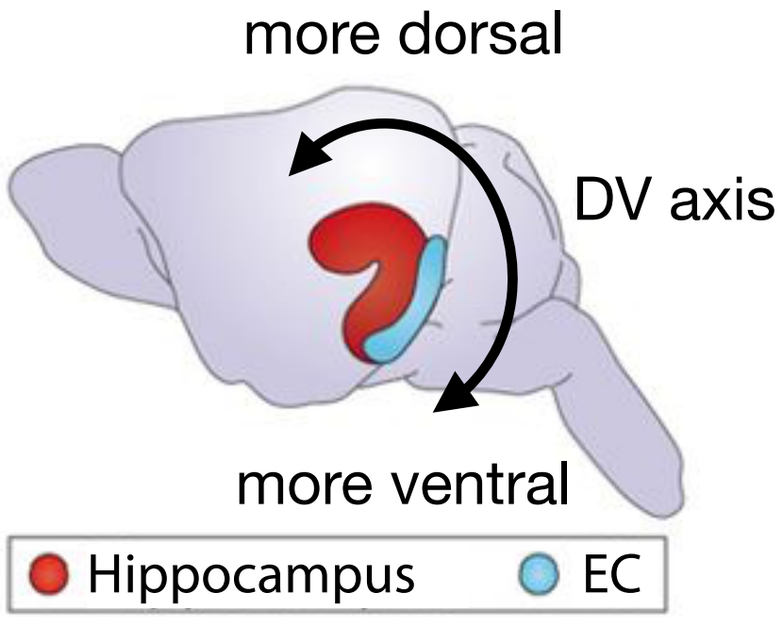
Multi-scale representations

Same eigenvectors support different coarseness

Simulated place fields with different scales



more dorsal
smaller discount factor

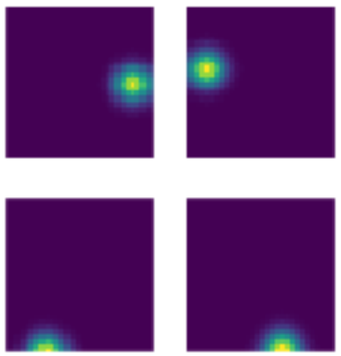


Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

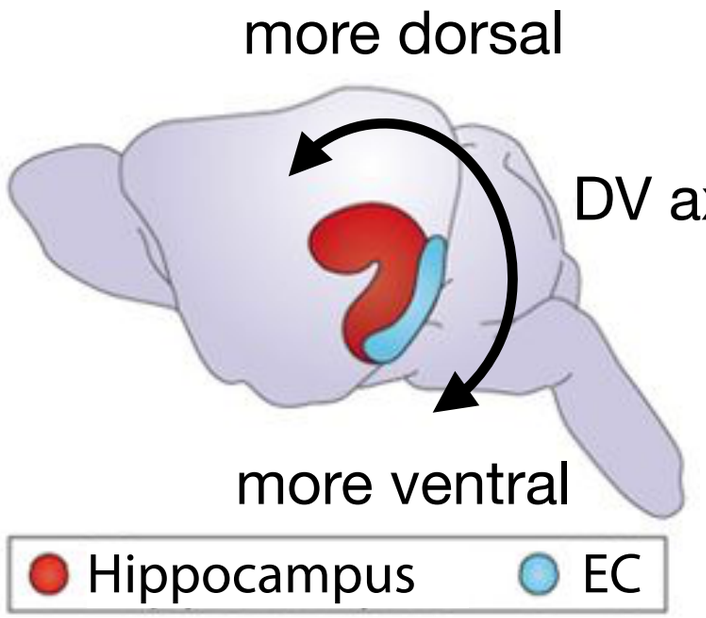
Multi-scale representations

Same eigenvectors support different coarseness

Simulated place fields with different scales

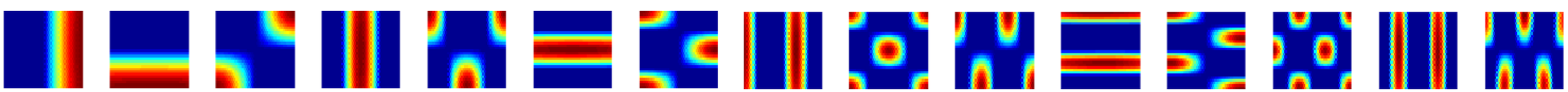


more dorsal
smaller discount factor



Simulated eigenvector grid fields

more ventral



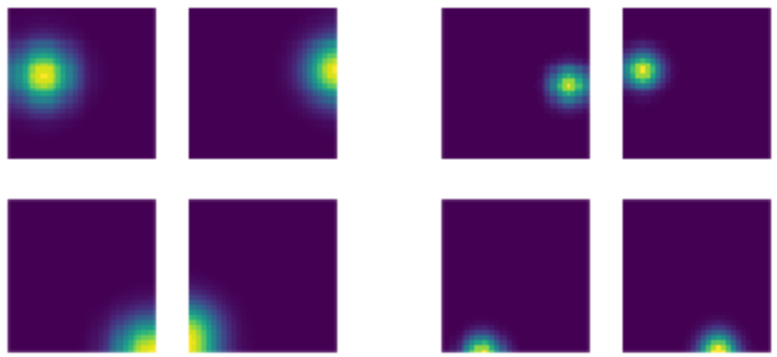
more dorsal

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

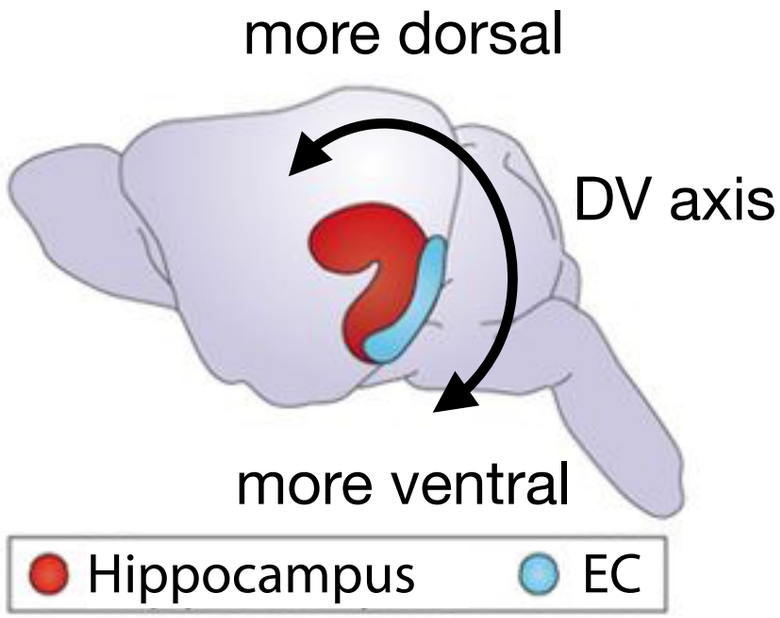
Multi-scale representations

Same eigenvectors support different coarseness

Simulated place fields with different scales

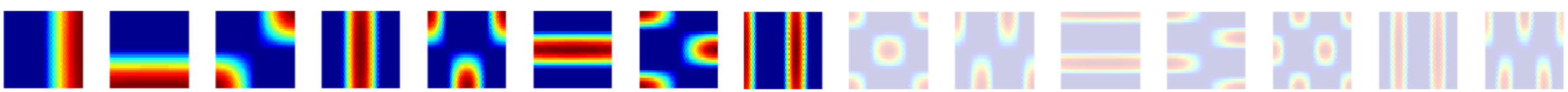


more dorsal
smaller discount factor



Simulated eigenvector grid fields

more ventral



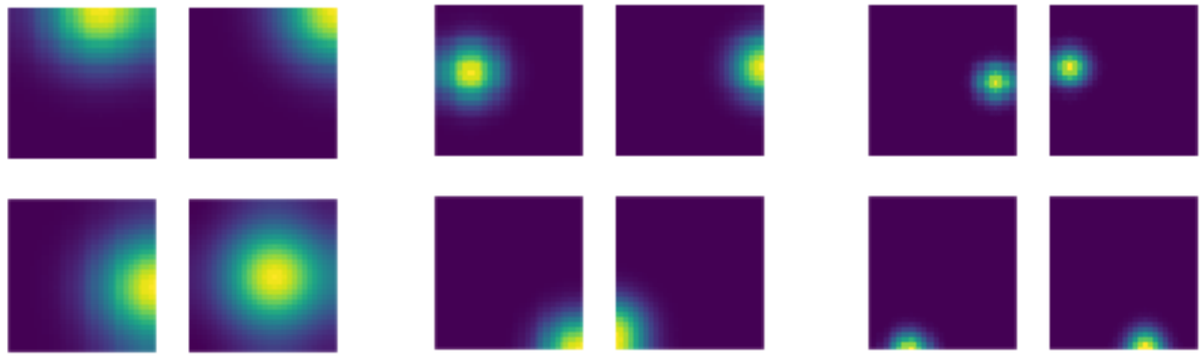
more dorsal

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

Multi-scale representations

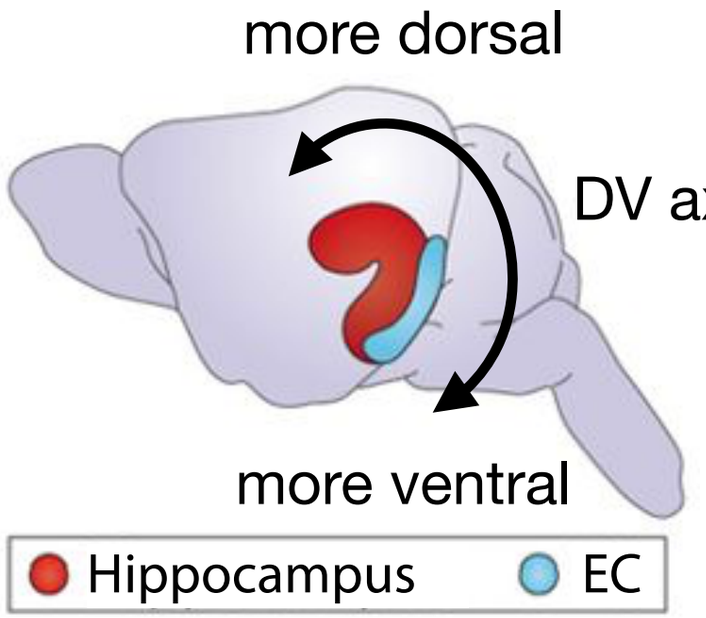
Same eigenvectors support different coarseness

Simulated place fields with different scales



more ventral
larger discount factor

more dorsal
smaller discount factor



Simulated eigenvector grid fields

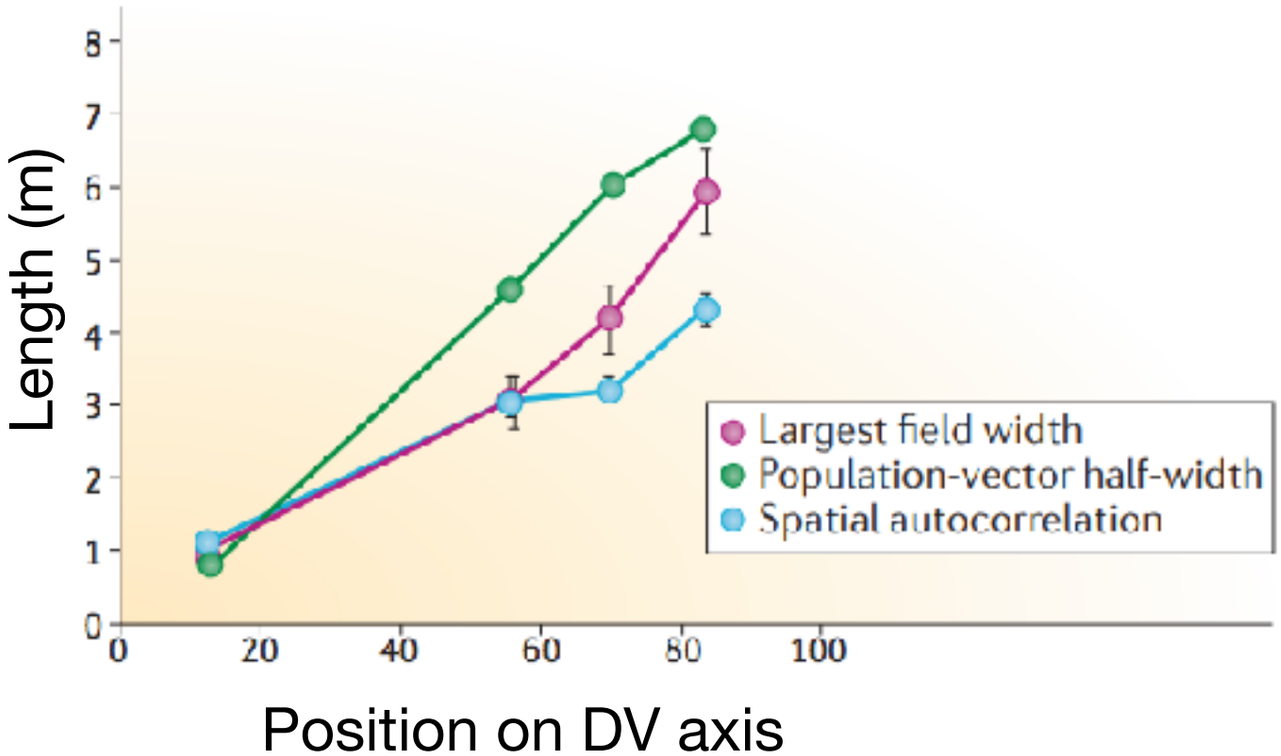
more ventral

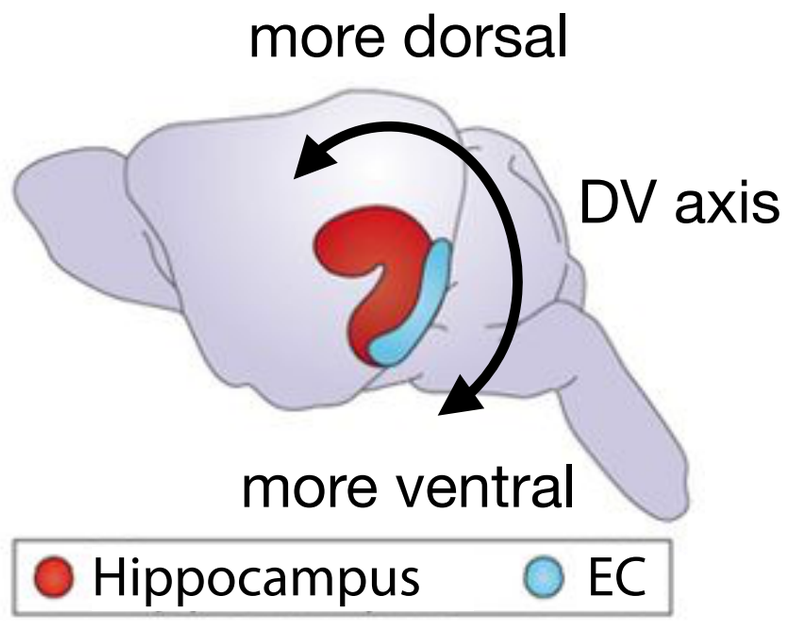
more dorsal

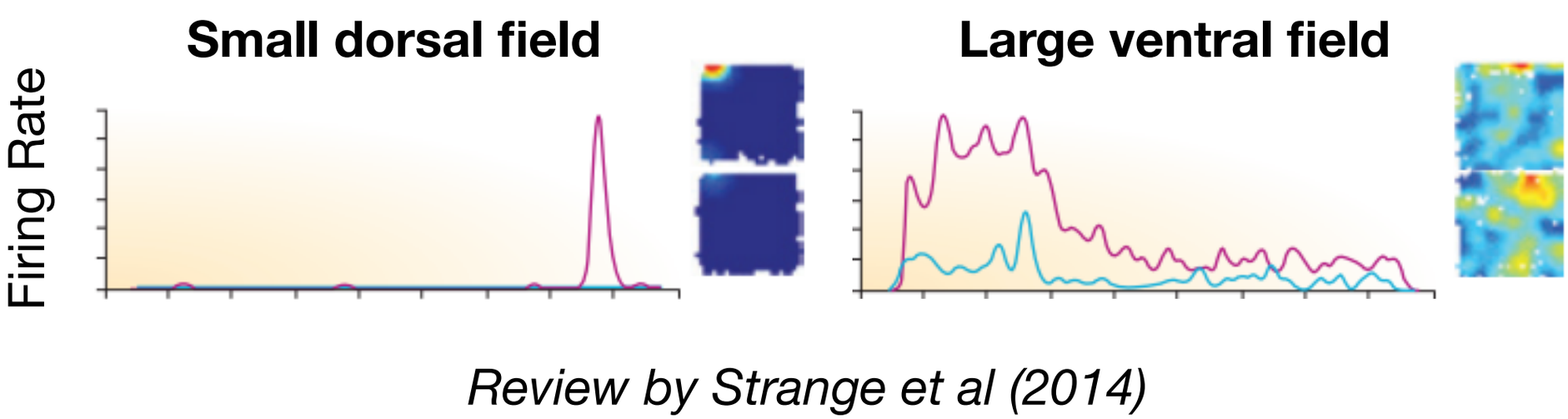
Stachenfeld, Botvinick, Gershman (2017)
also see Momennejad & Howard (2018) for things you can do with multi-scale SRs

Multi-scale representations

Range of spatial scales along longitudinal axis of hippocampus + MEC

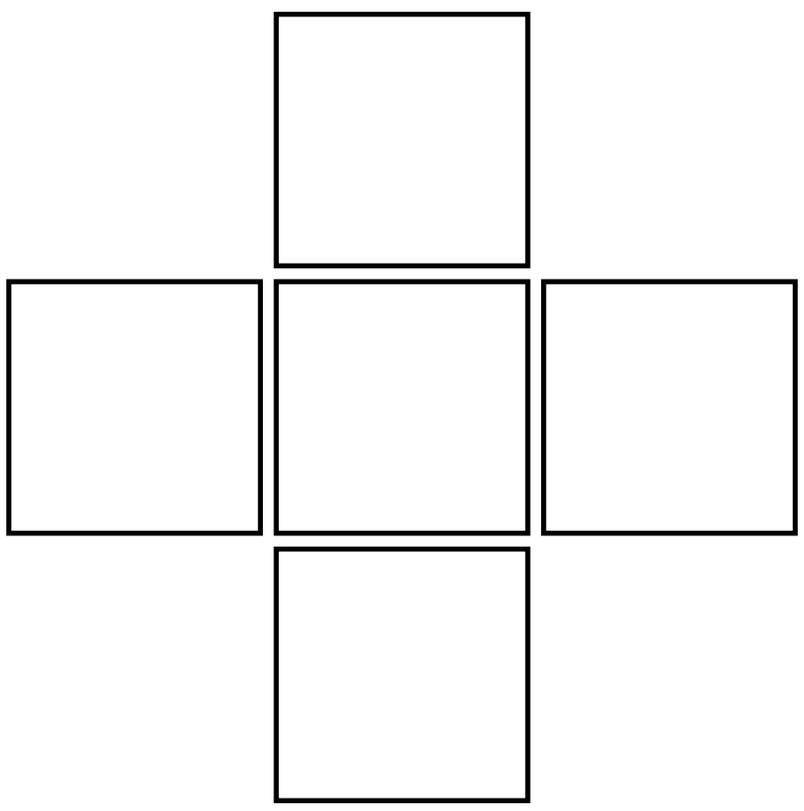






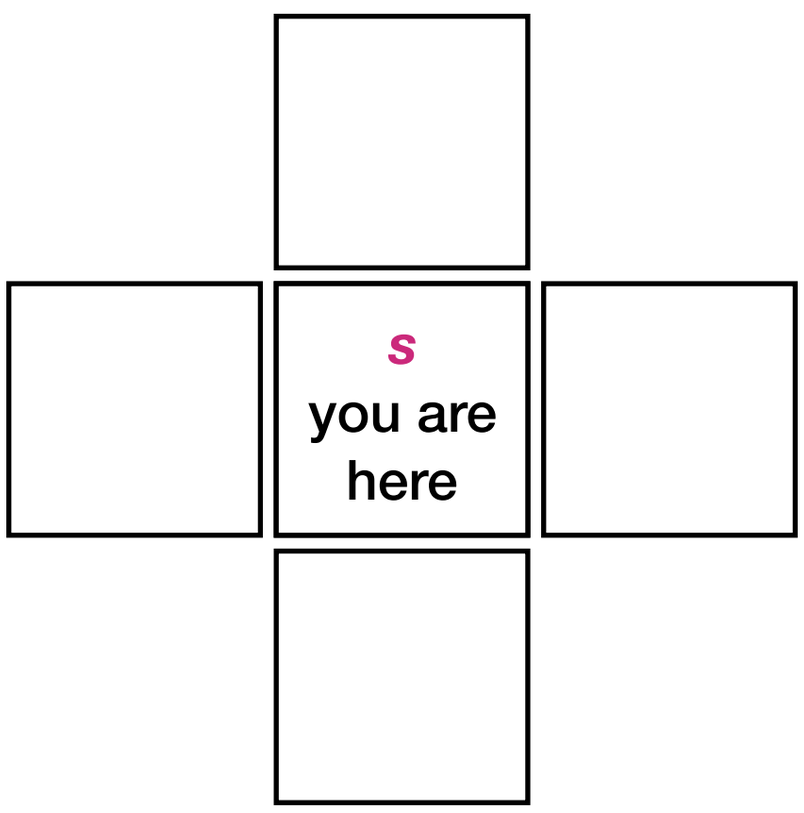
Review by Strange et al (2014)

Exploration with diffusion



McNamee, Stachenfeld, Botvinick, Gershman (in prep)

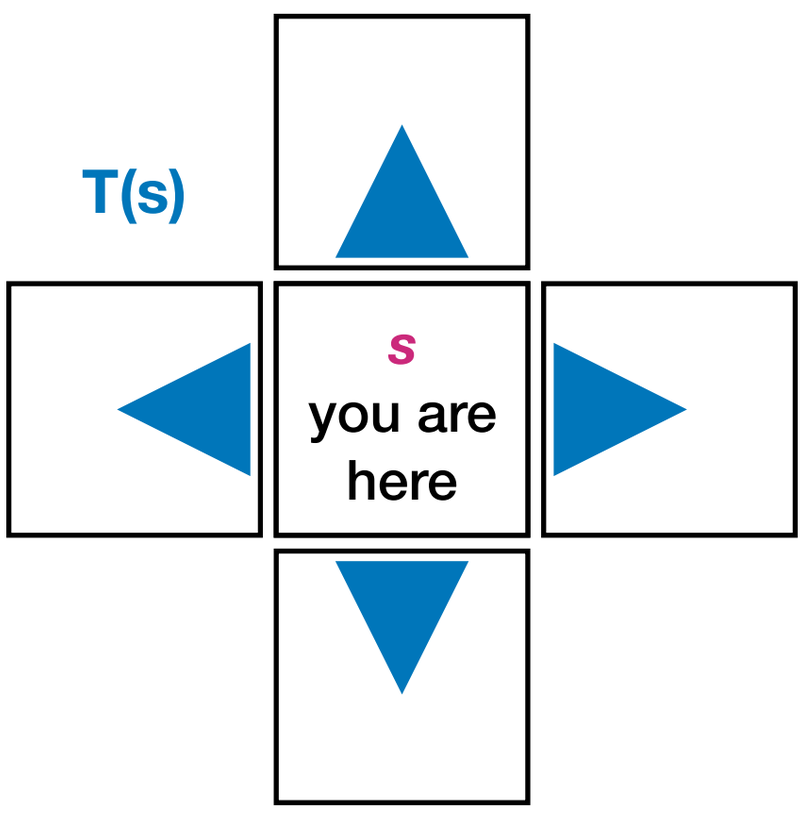
Exploration with diffusion



(1) Start in some state **s**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with diffusion

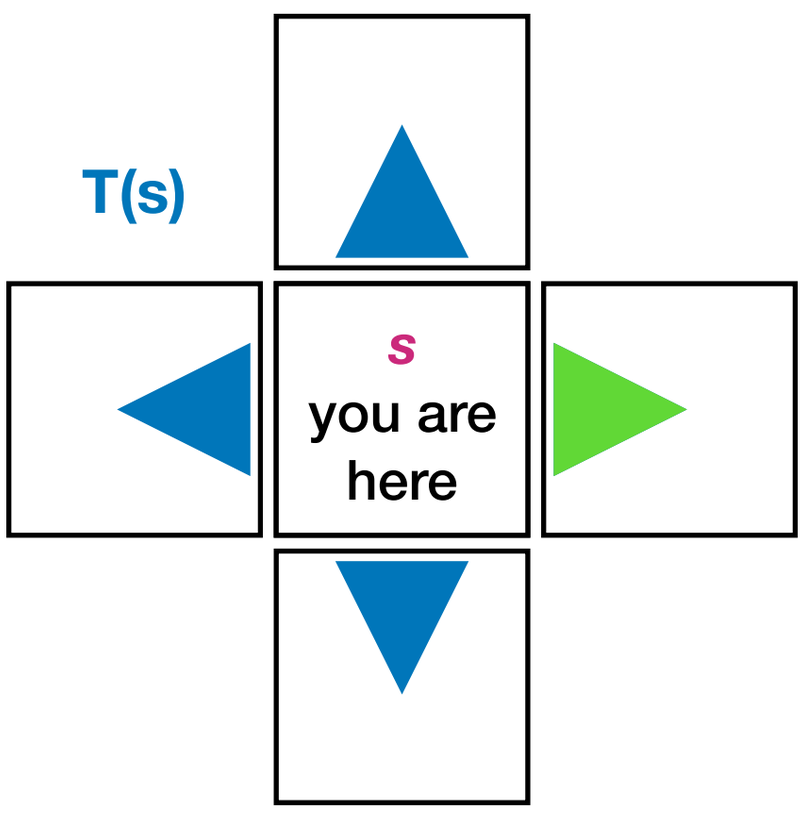


(1) Start in some state **s**

(2) Find which actions are available to you in one-step, **T(s)**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with diffusion



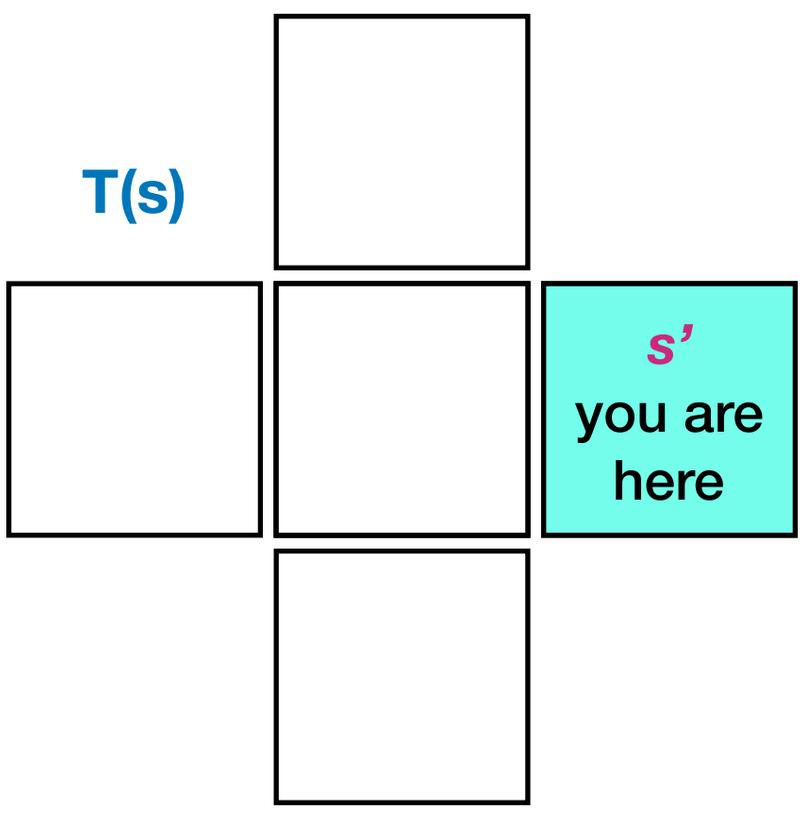
(1) Start in some state **s**

(2) Find which actions are available to you in one-step, **T(s)**

(3) Randomly sample **action** from **T(s)** that takes you to one of your neighbor states, **s'**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with diffusion



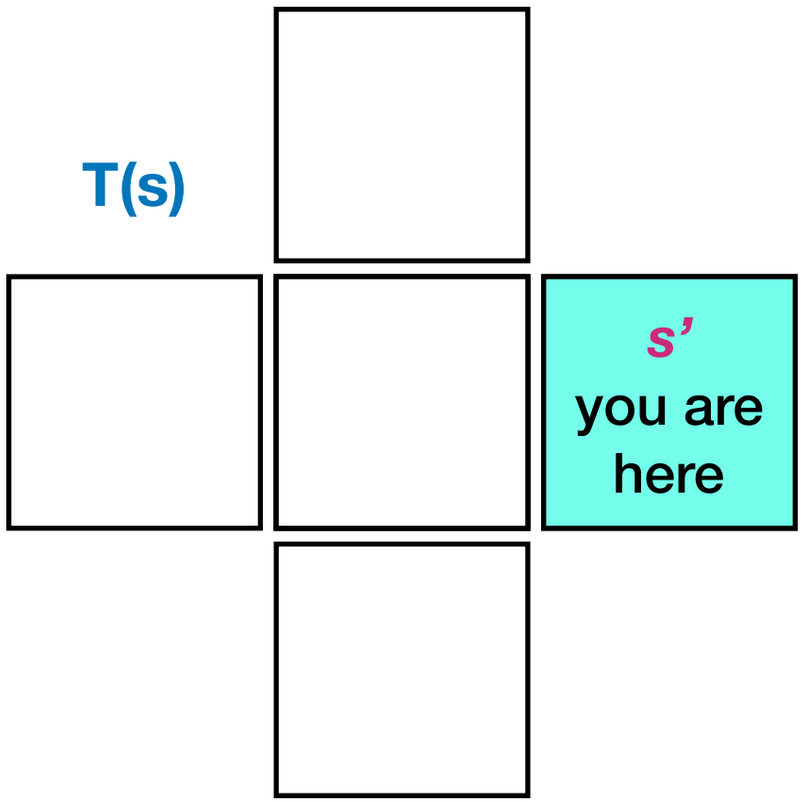
(1) Start in some state **s**

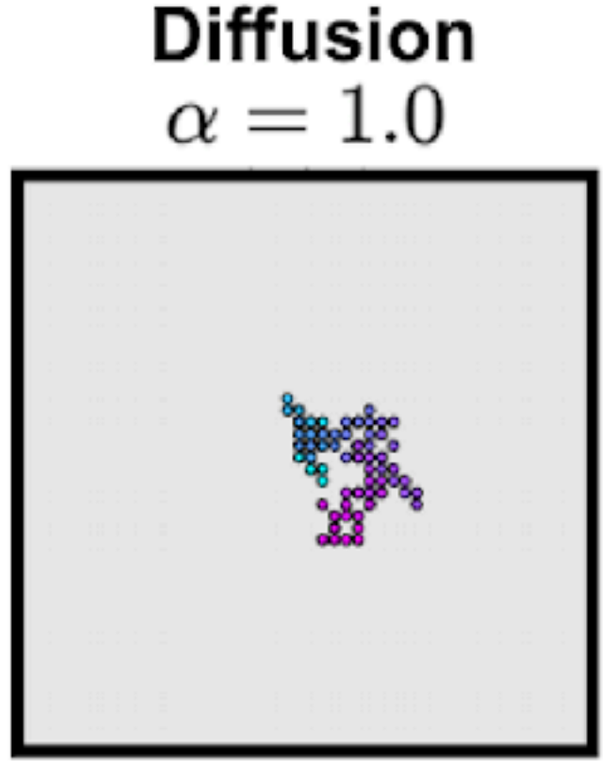
(2) Find which actions are available to you in one-step, **T(s)**

(3) Randomly sample **action** from **T(s)** that takes you to one of your neighbor states, **s'**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with diffusion





Eventually you cover some ground

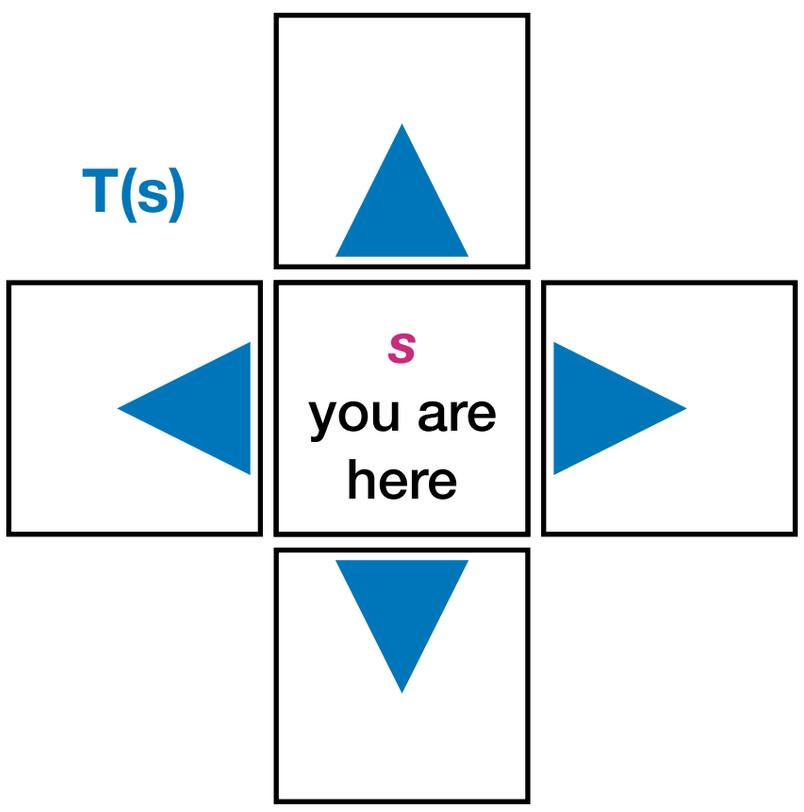
(1) Start in some state **s**

(2) Find which actions are available to you in one-step, **T(s)**

(3) Randomly sample **action** from **T(s)** that takes you to one of your neighbor states, **s'**

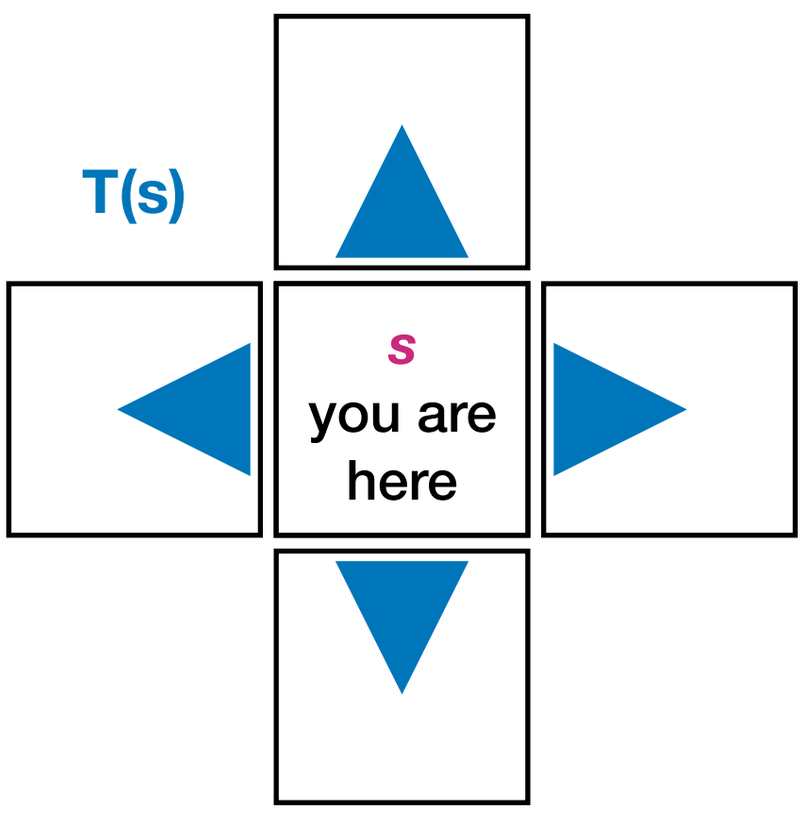
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral diffusion



McNamee, Stachenfeld, Botvinick, Gershman (in prep)

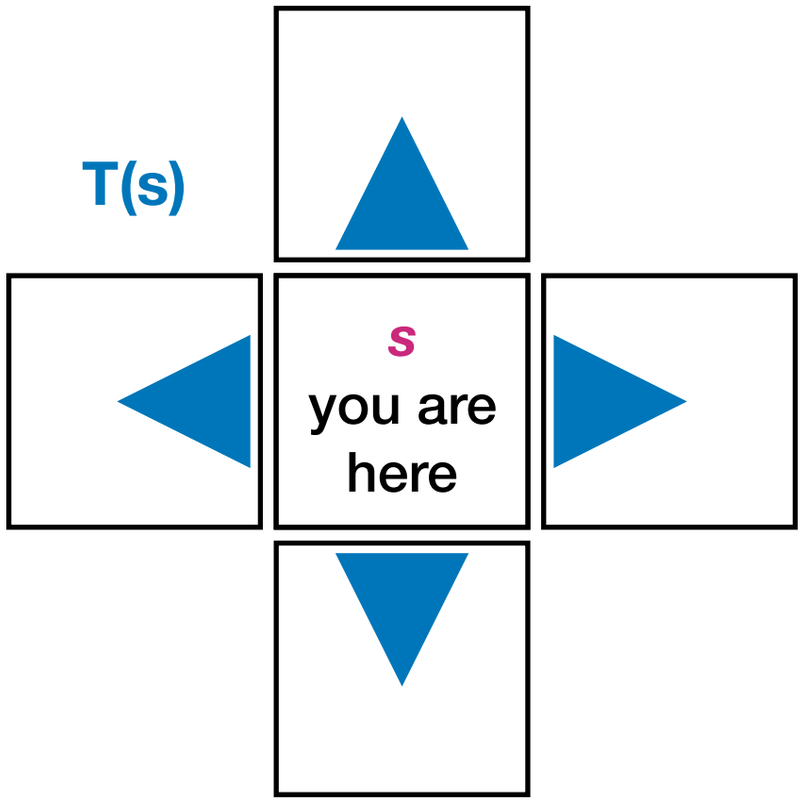
Exploration with spectral diffusion



(1) Start in some state **s** with neighbors **T(s)**

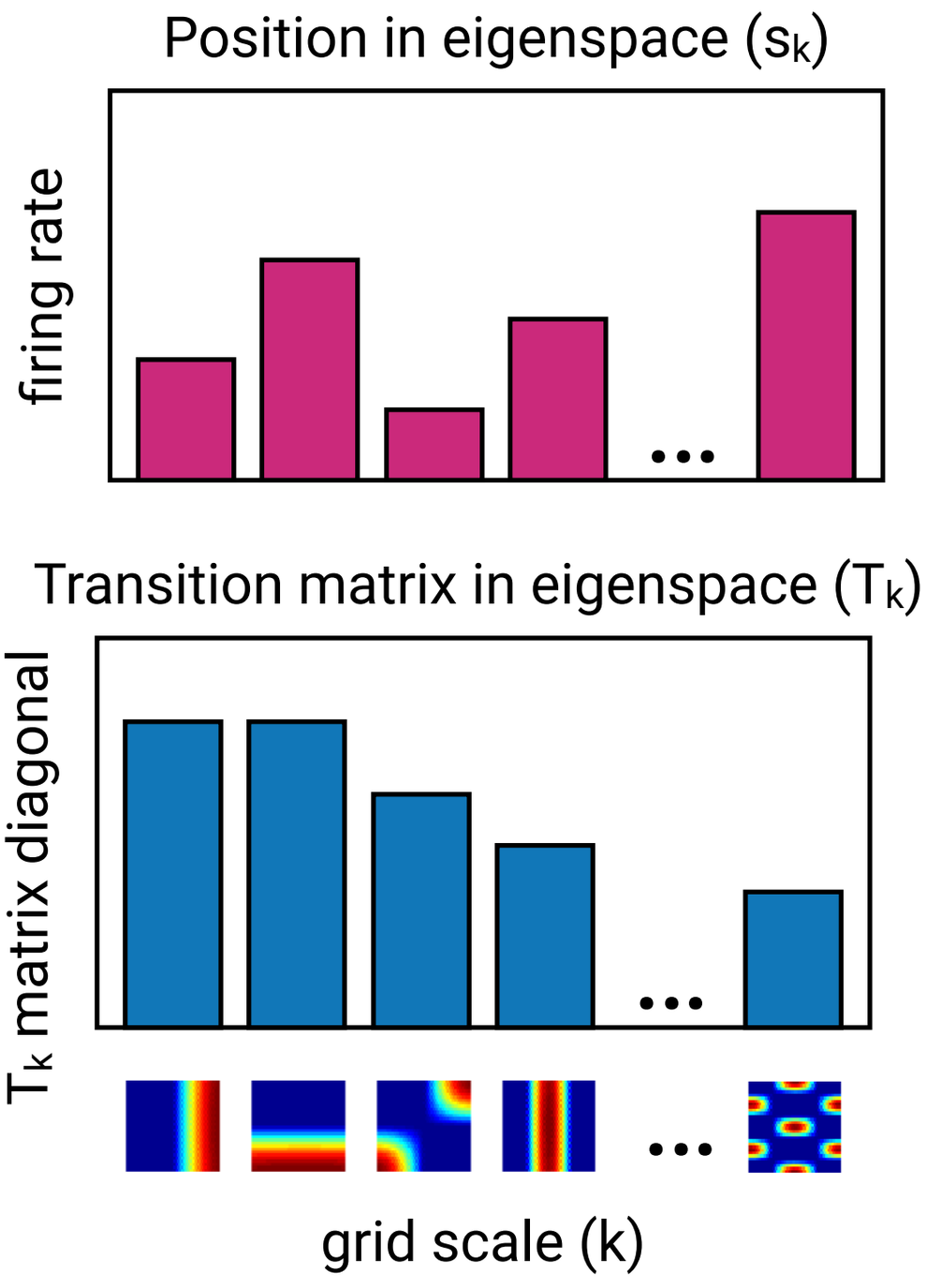
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral diffusion



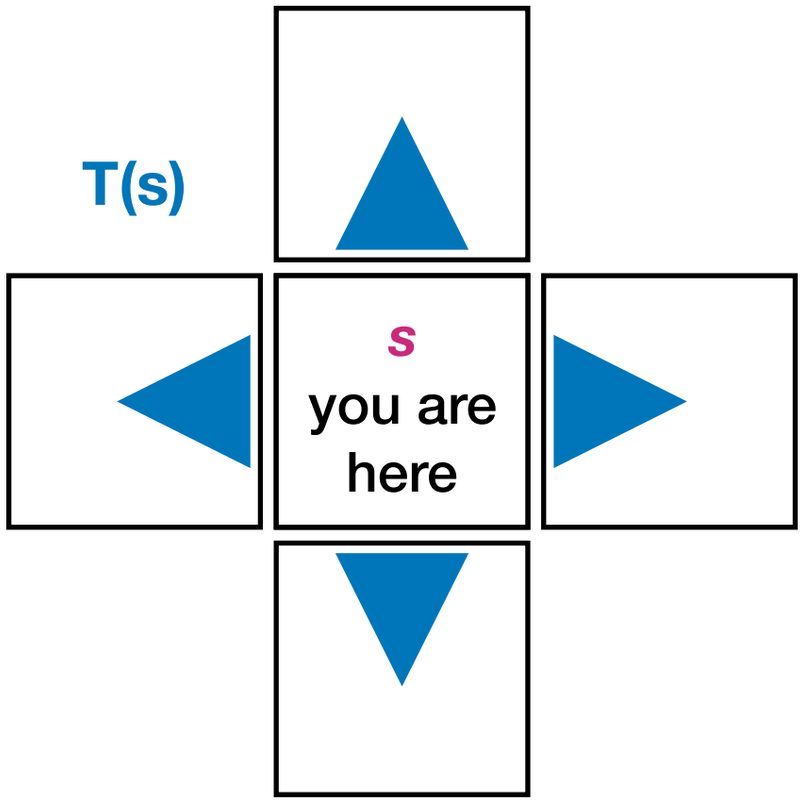
(1) Start in some state **s** with neighbors **T(s)**

(2) Project $\mathbf{s} \rightarrow \mathbf{s_k} = \mathbf{U^T s}$ and $\mathbf{T} \rightarrow \mathbf{T_k} = \mathbf{U^T T U}$



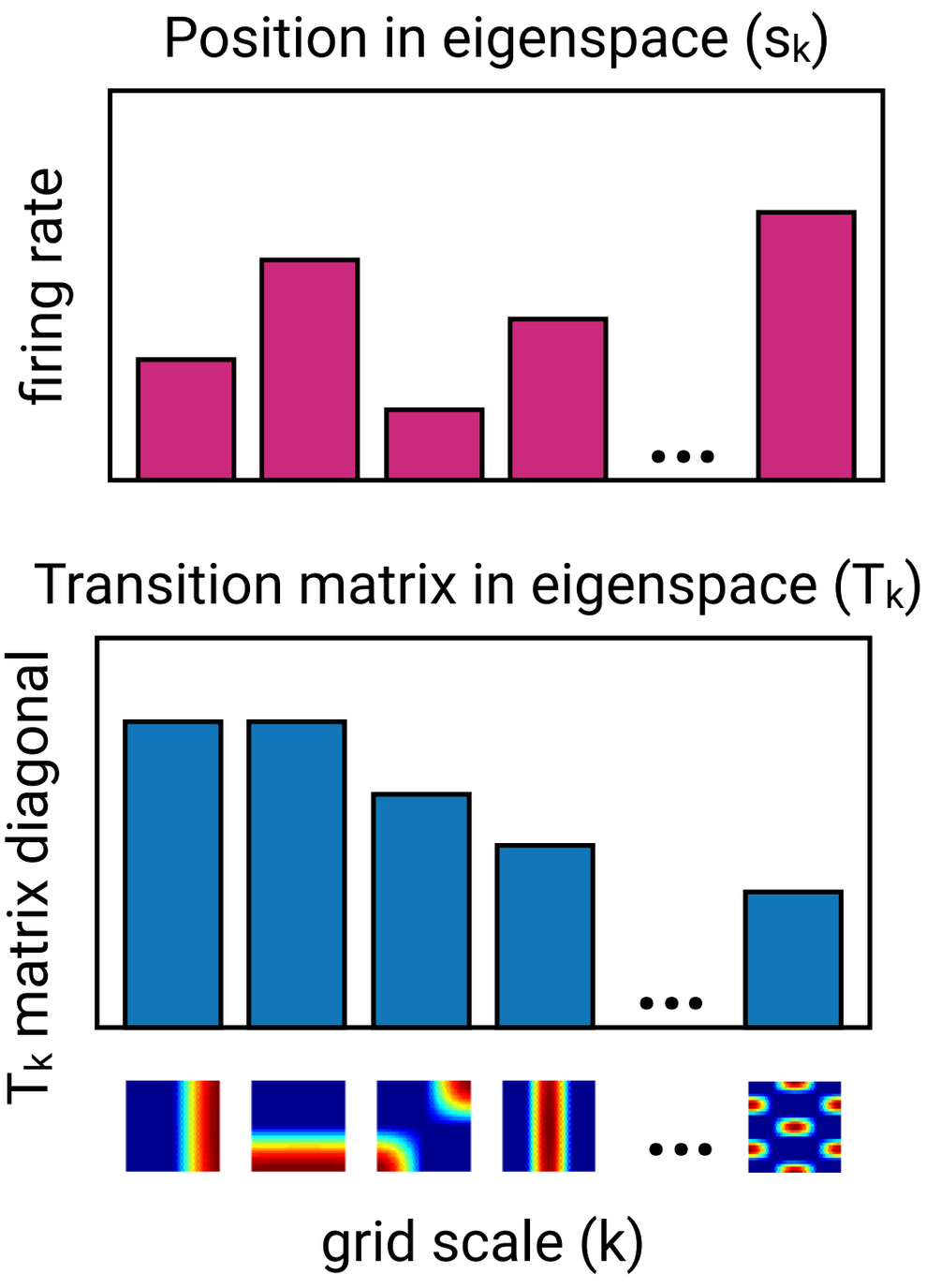
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral diffusion



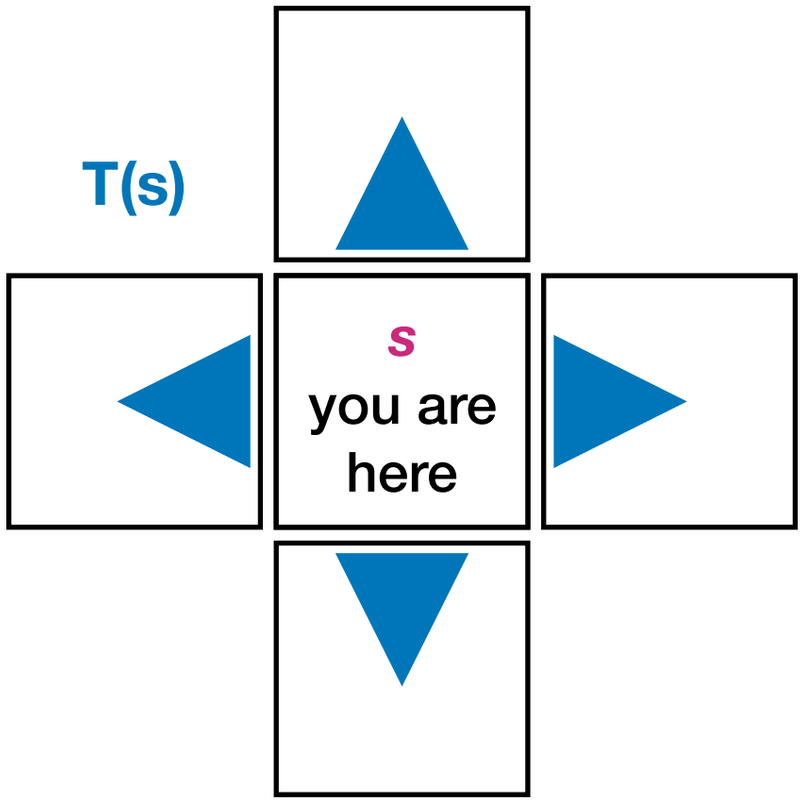
(1) Start in some state **s** with neighbors **T(s)**

(2) Project $\mathbf{s} \rightarrow \mathbf{s_k} = \mathbf{U^T s}$ and $\mathbf{T} \rightarrow \mathbf{T_k} = \mathbf{U^T T U}$



McNamee, Stachenfeld, Botvinick, Gershman (in prep)

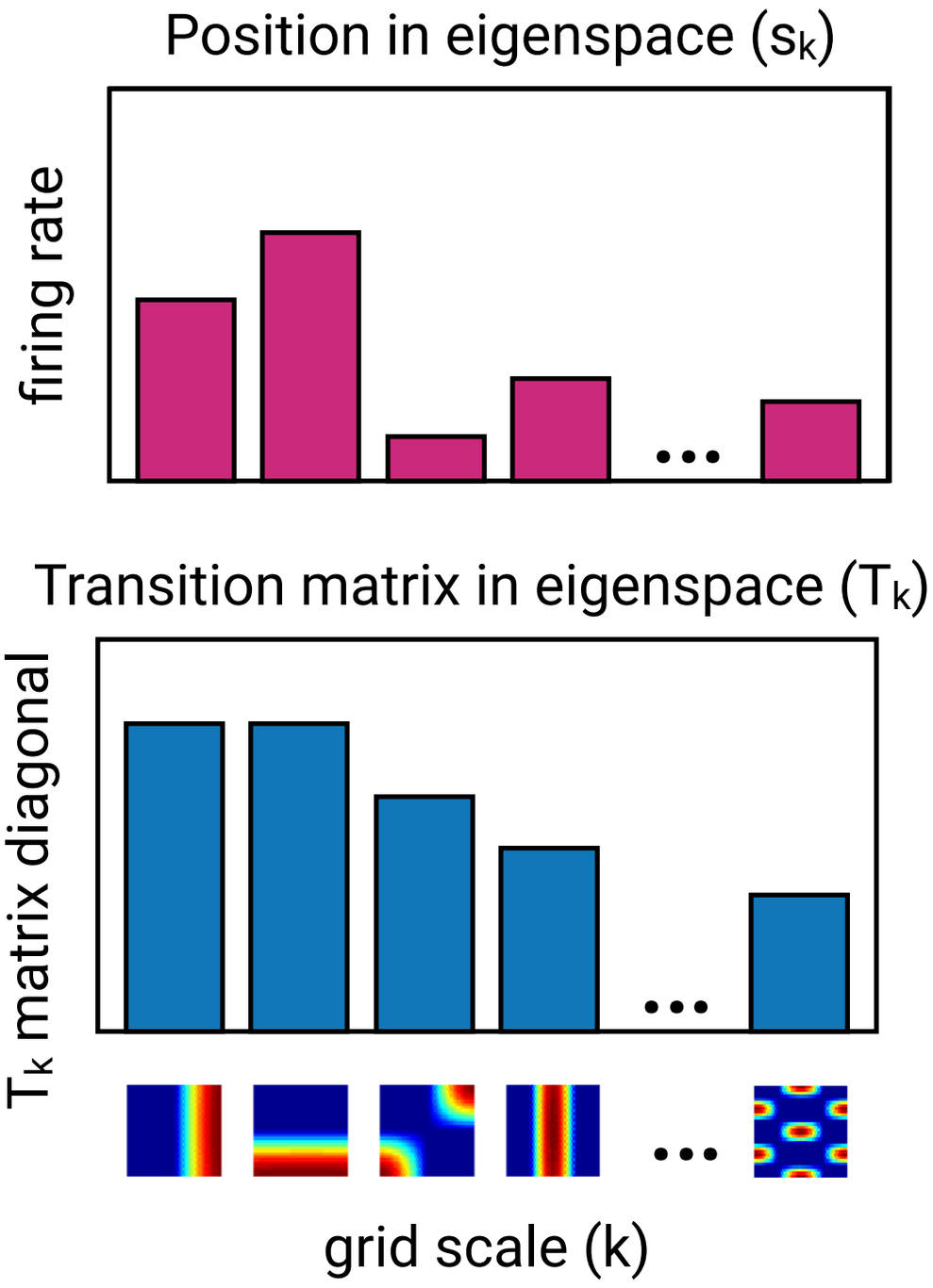
Exploration with <u>spectral</u> diffusion



(1) Start in some state **s** with neighbors **T(s)**

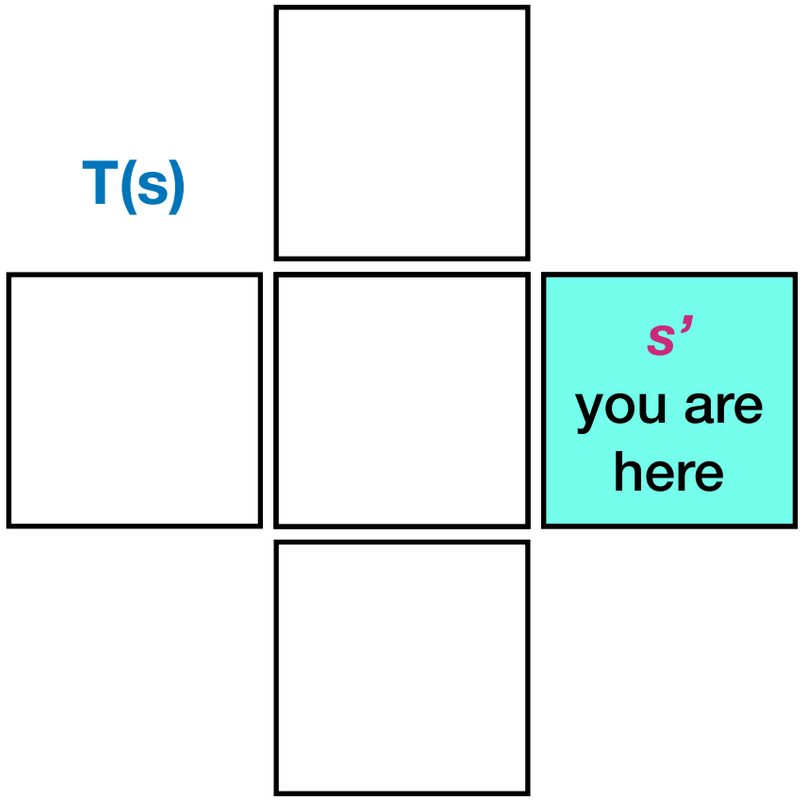
(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

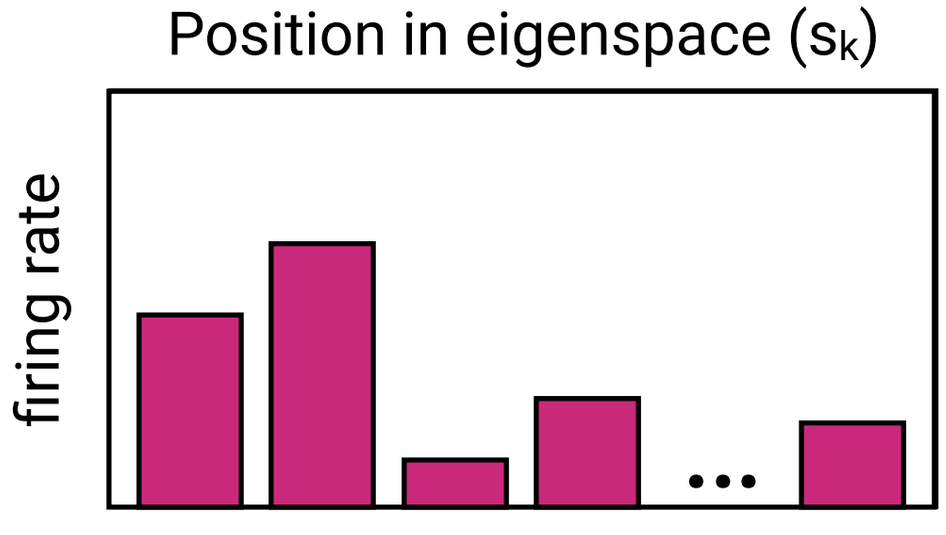
(4) Get probable next state **$p(s_k') = T_k s_k$**

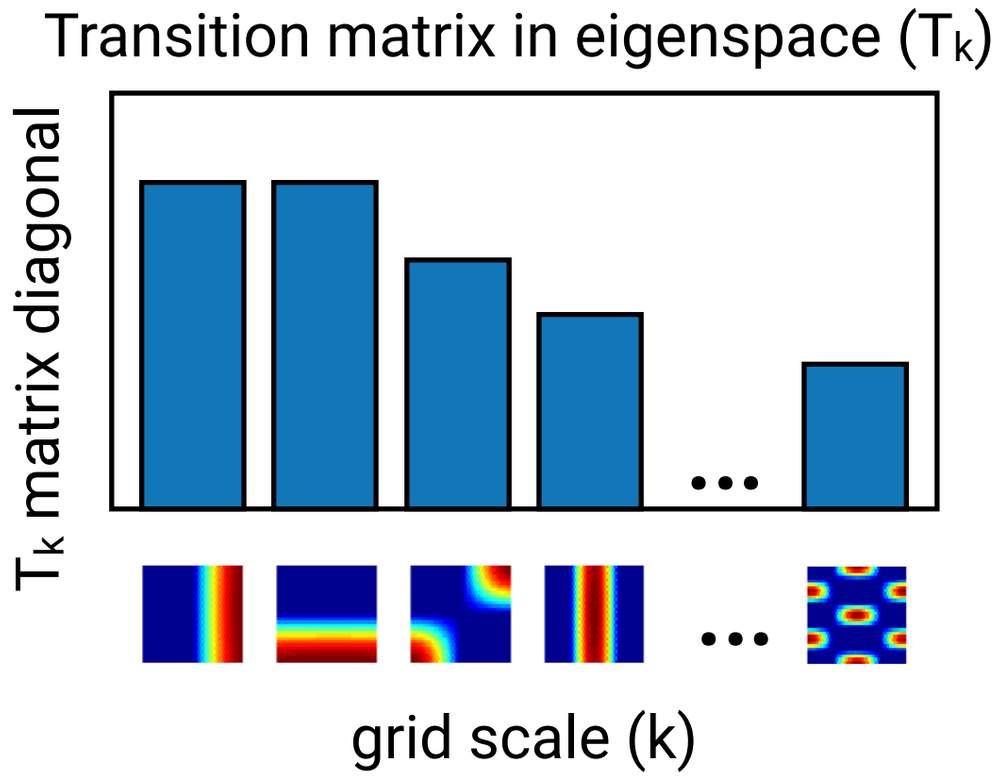


McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral diffusion







(1) Start in some state **s** with neighbors **T(s)**

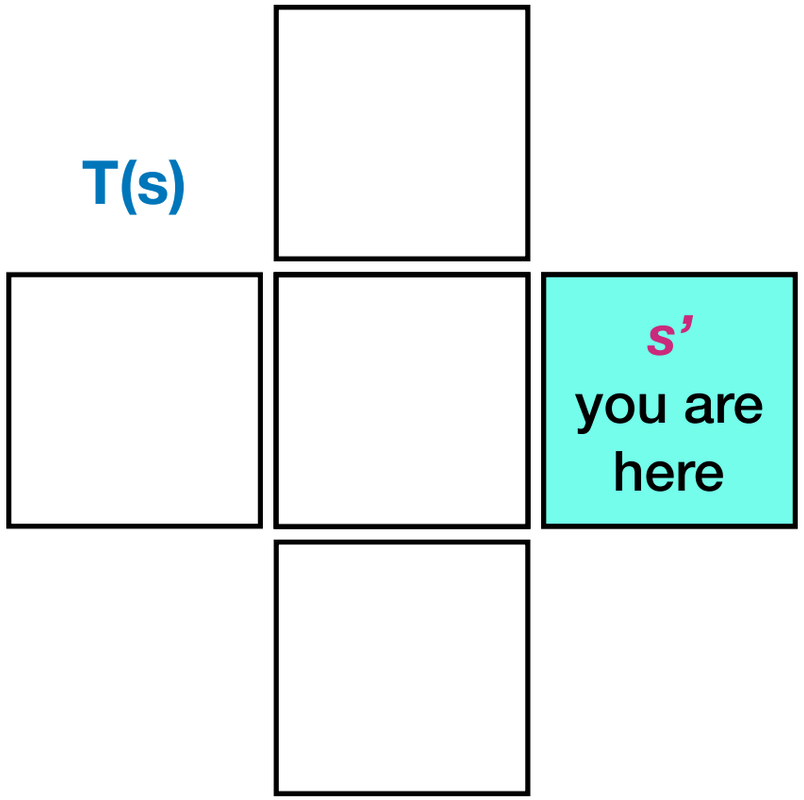
(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

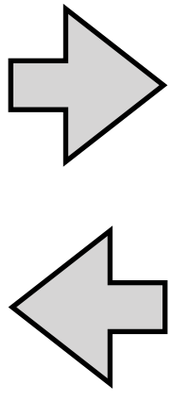
(4) Get probable next state **$p(s_k') = T_k s_k$**

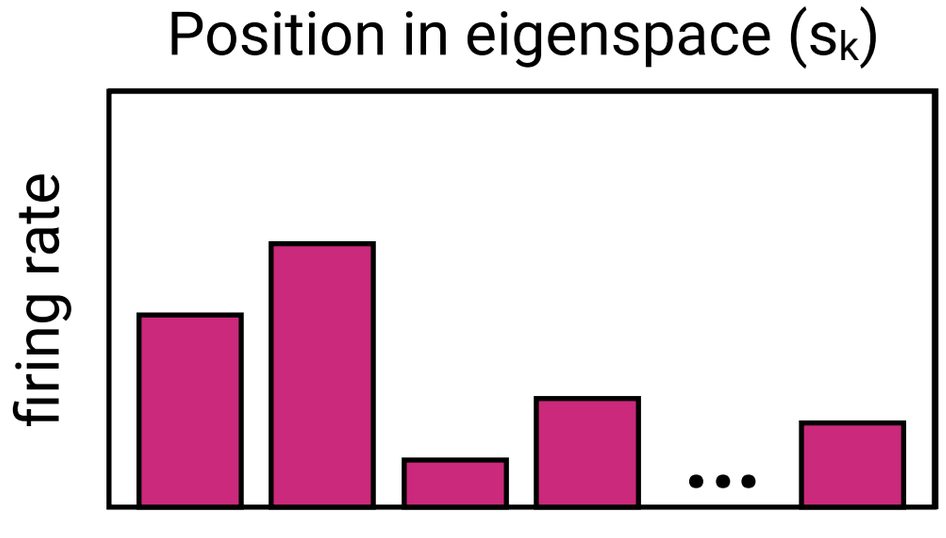
(5) Project back, sample **s'**

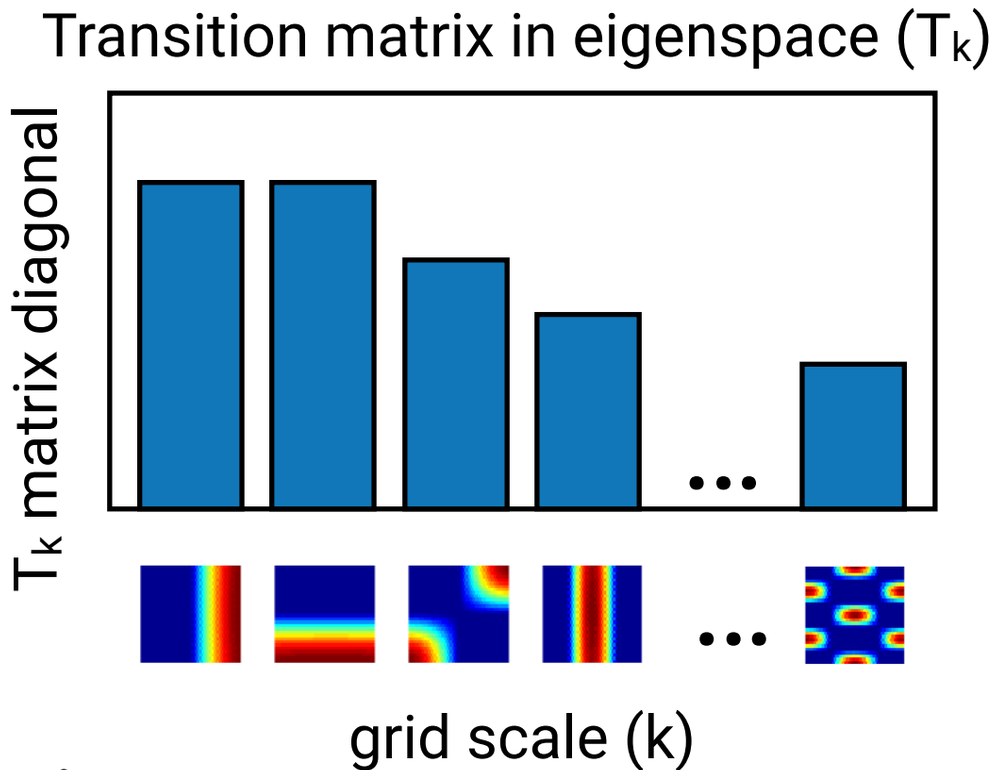
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral diffusion









(1) Start in some state **s** with neighbors **T(s)**

(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

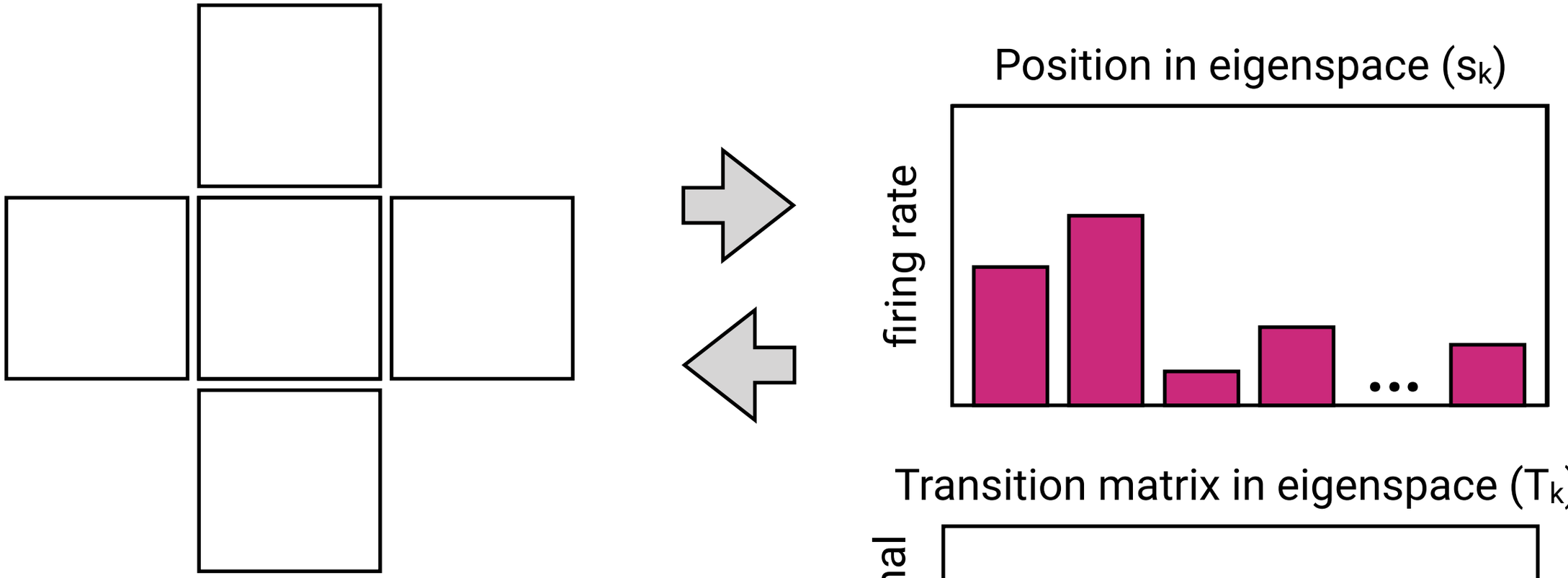
(4) Get probable next state **$p(s_k') = T_k s_k$**

(5) Project back, sample **s'**

Also generates diffusion with most of the same statistics

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with <u>spectral</u> <u>super</u>diffusion



(1) Start in some state **s** with neighbors **T(s)**

(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

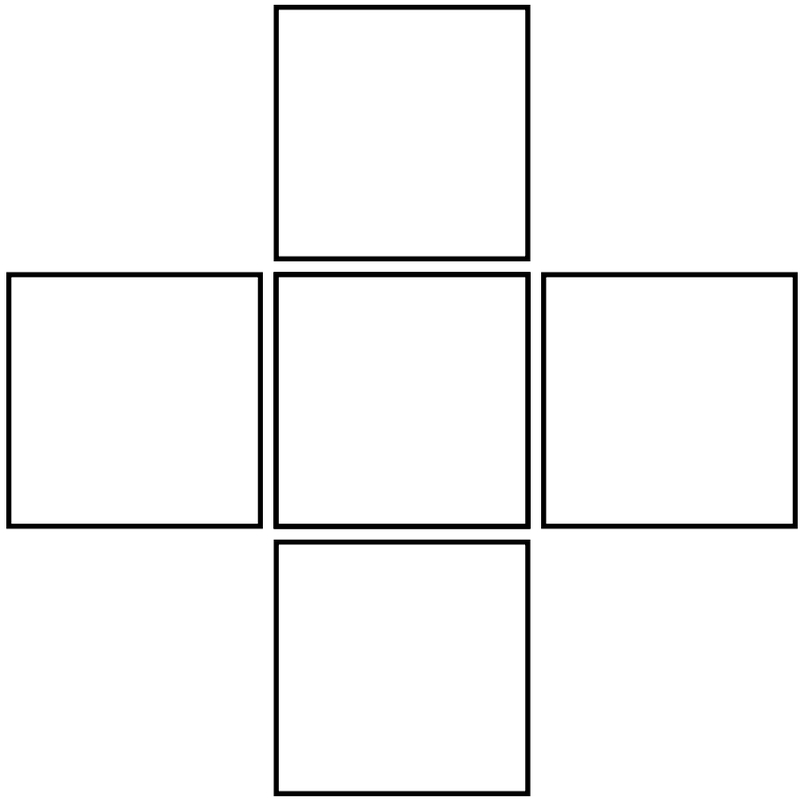
(3) Rescale to T_k to upweight large jump probability

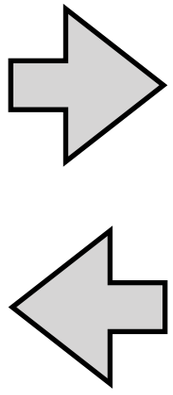
(4) Get probable next state **$p(s_k') = T_k s_k$**

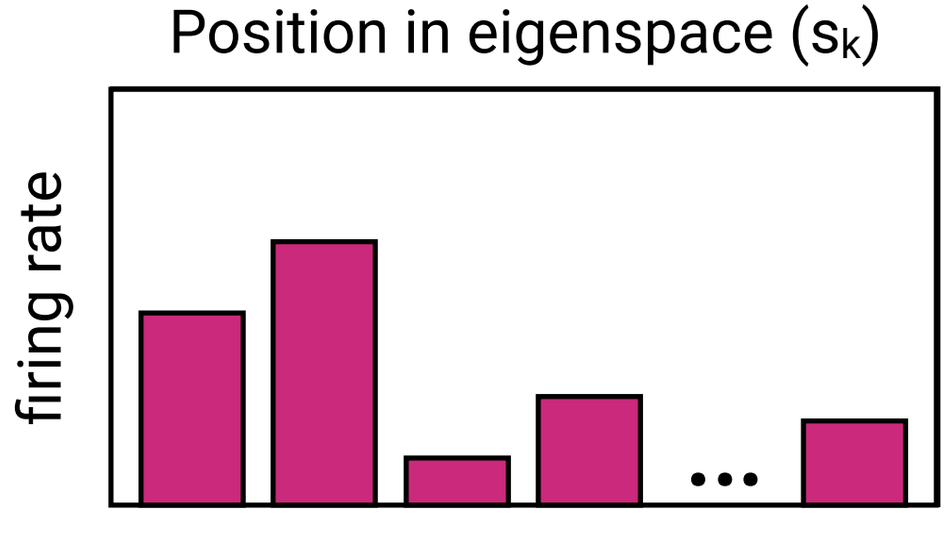
(5) Project back, sample **s'**

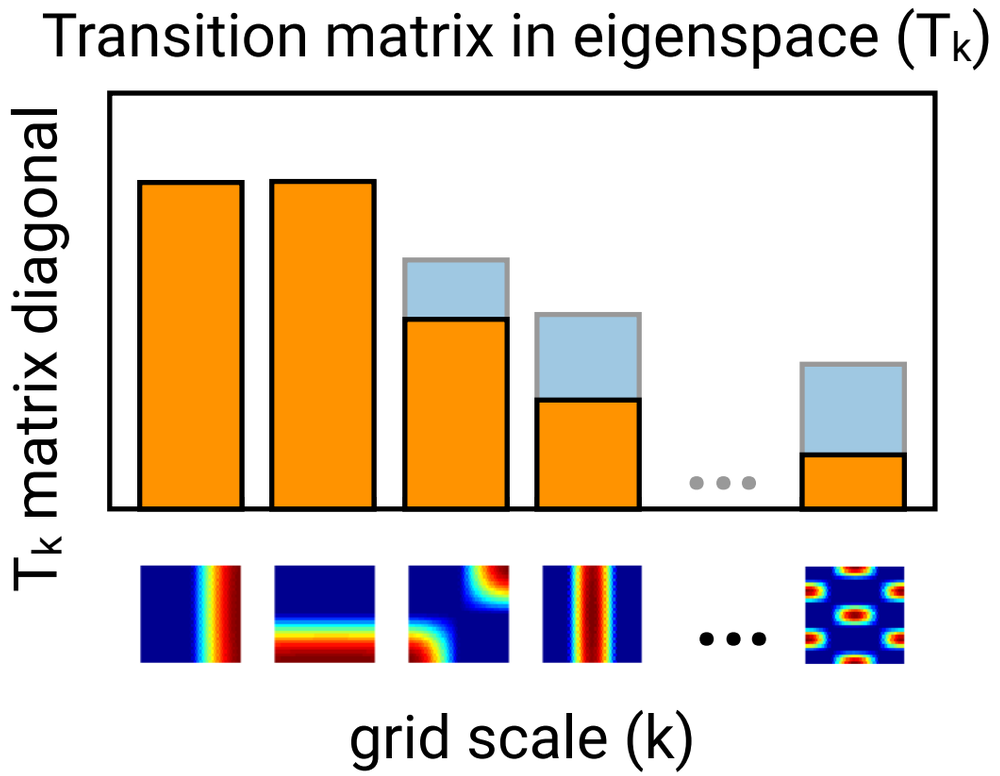
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with <u>spectral</u> <u>super</u>diffusion









(1) Start in some state **s** with neighbors **T(s)**

(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

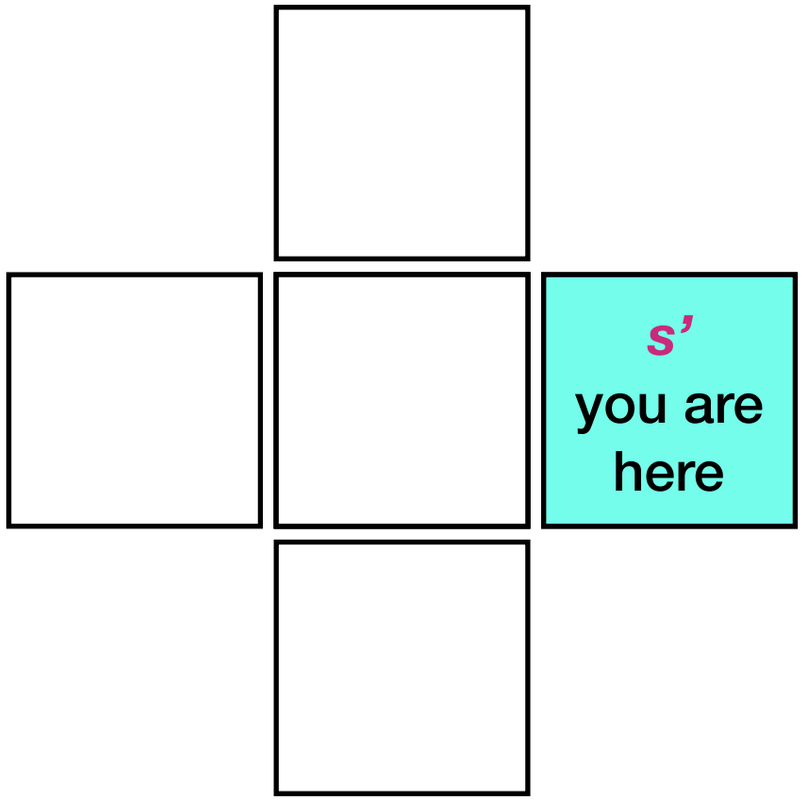
(3) Rescale to T_k to upweight large jump probability

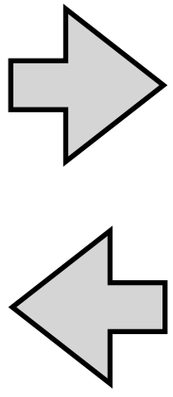
(4) Get probable next state **$p(s_k') = T_k s_k$**

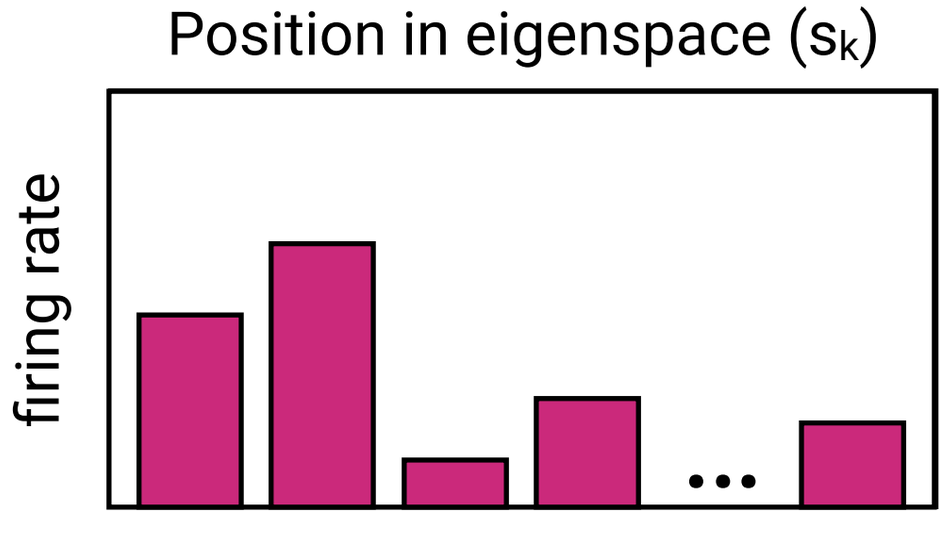
(5) Project back, sample **s'**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with <u>spectral</u> <u>super</u>diffusion







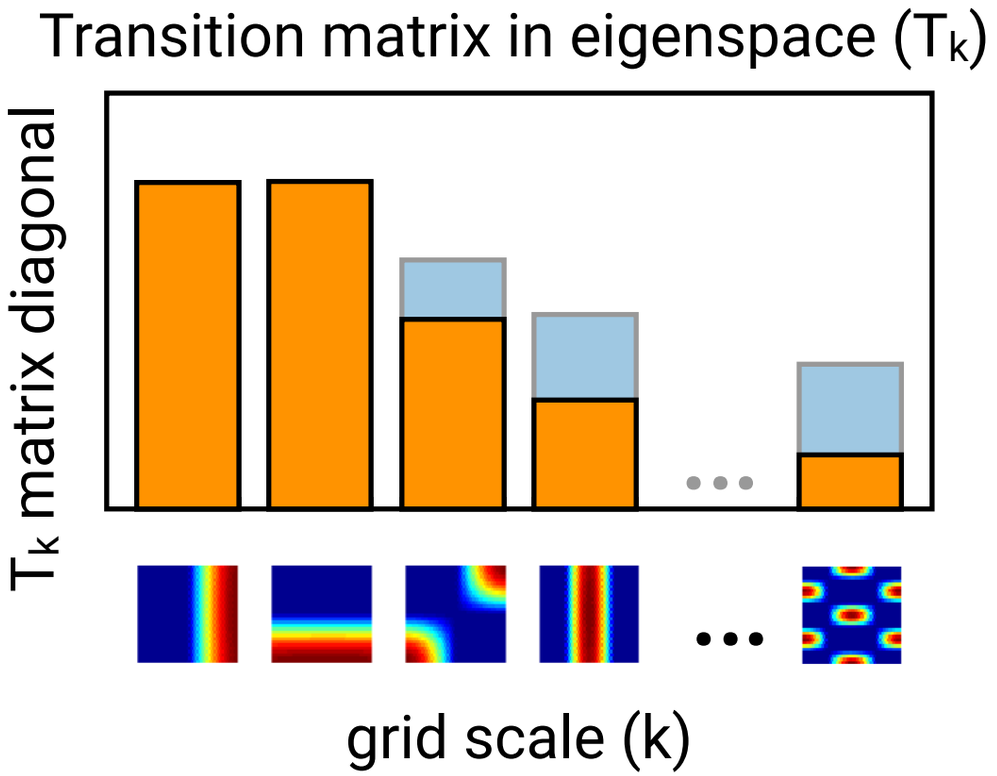
(1) Start in some state **s** with neighbors **T(s)**

(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

(3) Rescale to T_k to upweight large jump probability

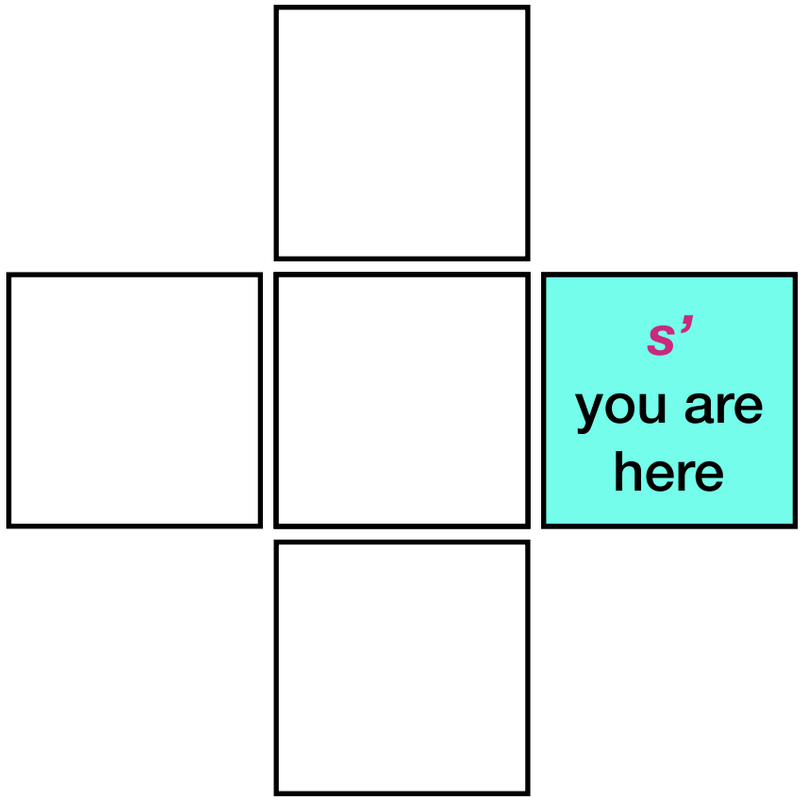
(4) Get probable next state **$p(s_k') = T_k s_k$**

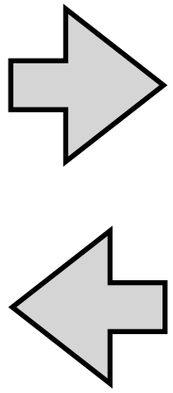
(5) Project back, sample **s'**

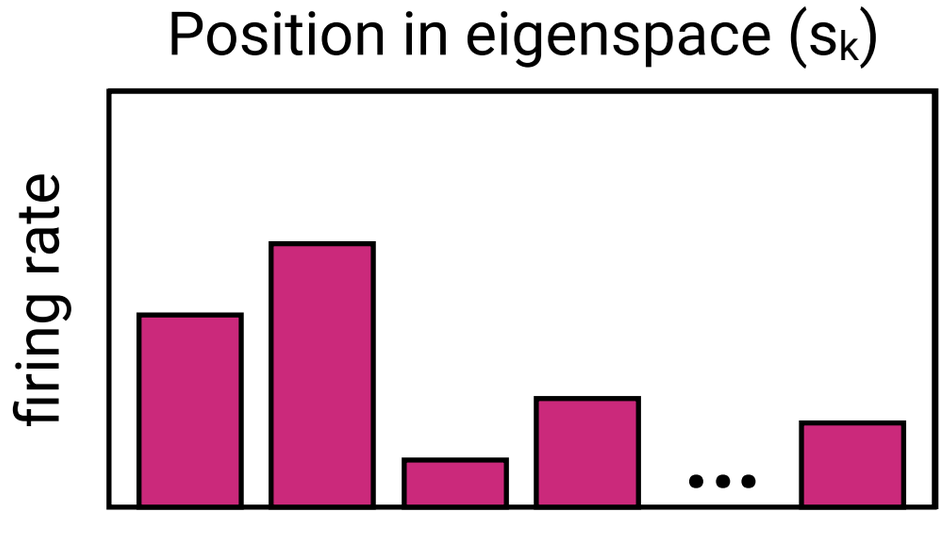


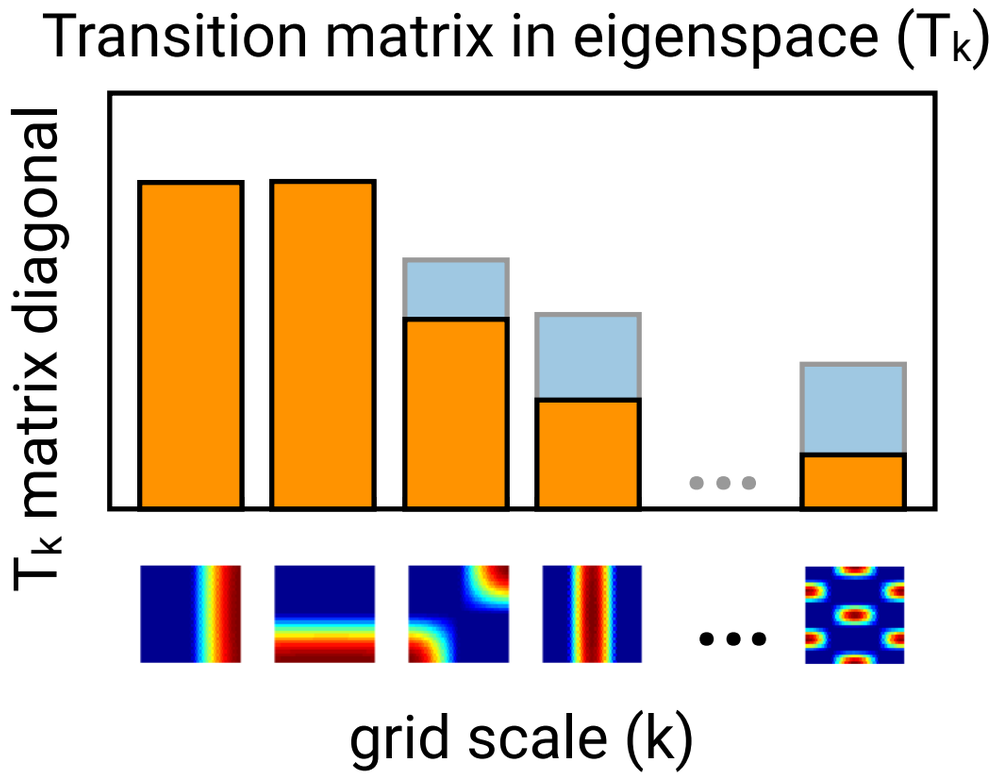
McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral superdiffusion









(1) Start in some state **s** with neighbors **T(s)**

(2) Project **$s \rightarrow s_k = U^T s$** and **$T \rightarrow T_k = U^T T U$**

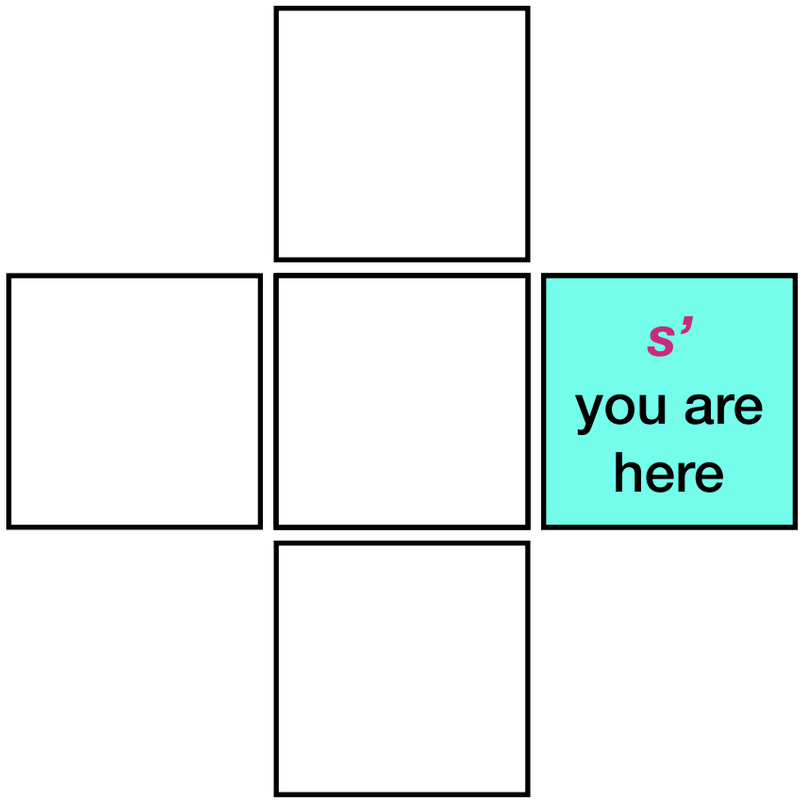
(3) Rescale to T_k to upweight large jump probability

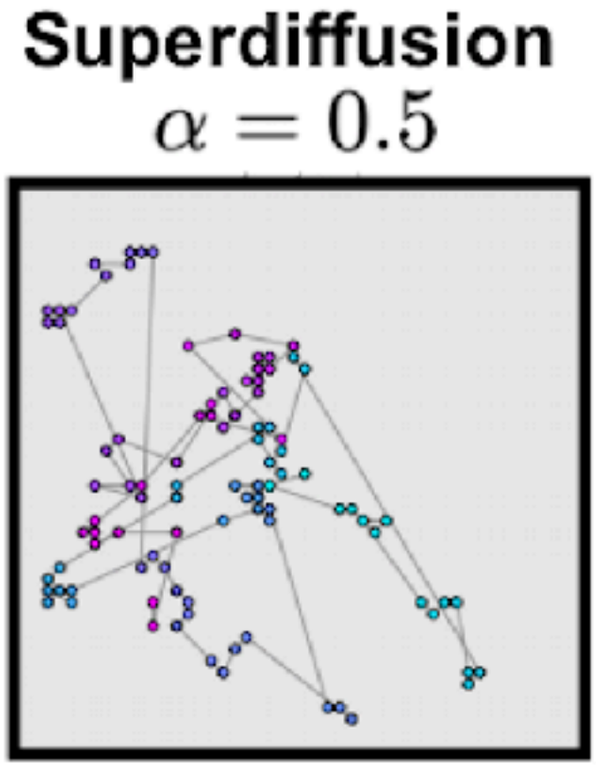
(4) Get probable next state **$p(s_k') = T_k s_k$**

(5) Project back, sample **s'**

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral superdiffusion





(1) Start in some state **s** with neighbors **T(s)**

(2) Project **s -> $s_k = U^T s$** and **T -> $T_k = U^T T U$**

(3) Rescale to T_k to upweight large jump probability

(4) Get probable next state **$p(s_k') = T_k s_k$**

(5) Project back, sample **s'**

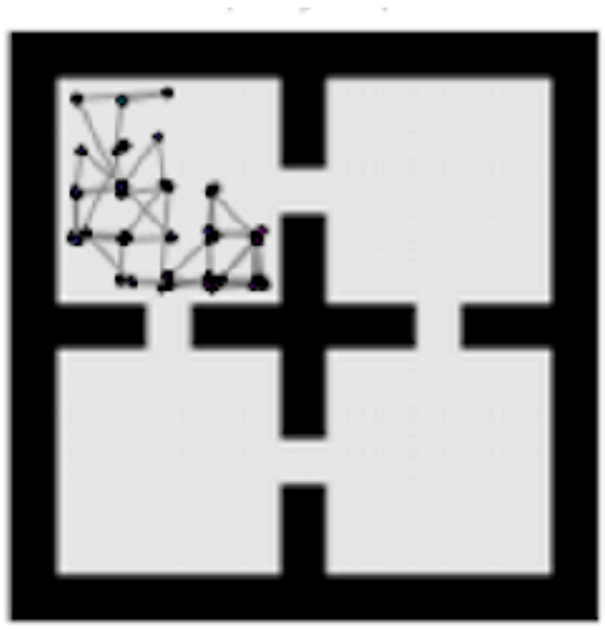
Covers more ground

Factoring into spatial + temporal terms can let you flexibly modulate timescale of exploration

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral superdiffusion in multicompartment environments

Diffusion



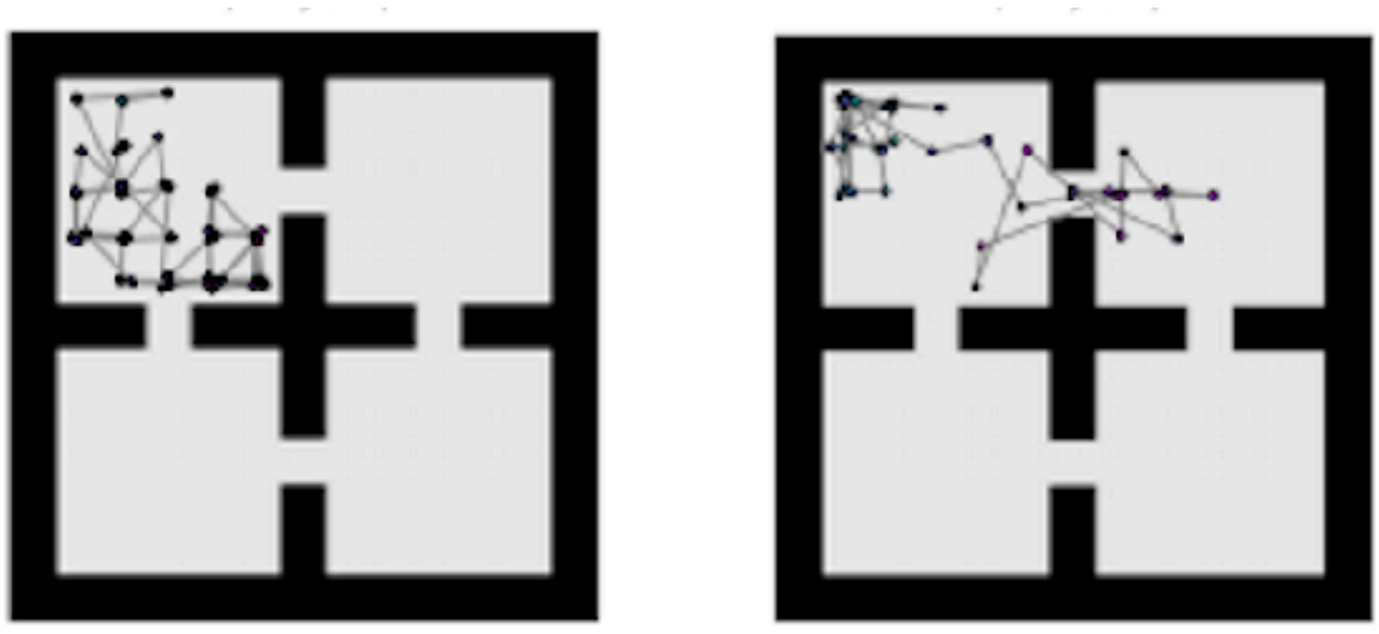
Agent gets stuck in room

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Exploration with spectral superdiffusion in multicompartment environments

Diffusion

Super-diffusion

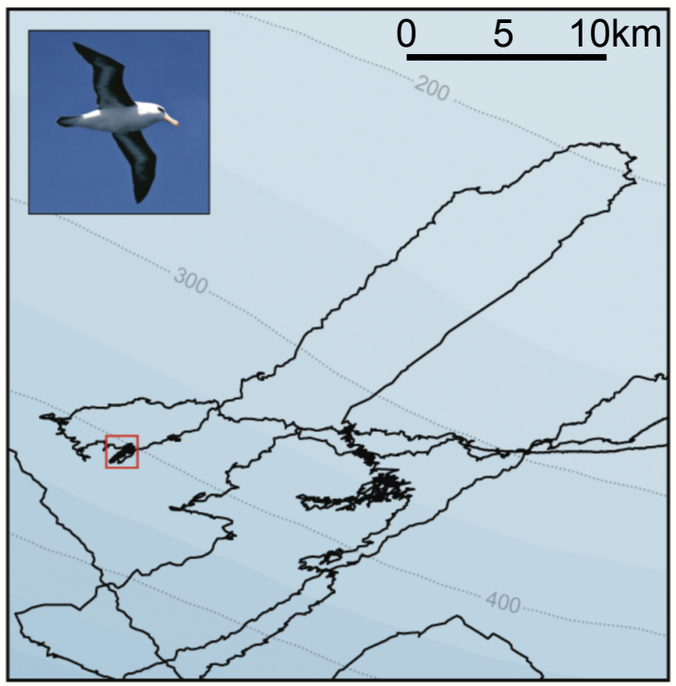


Agent gets stuck in room

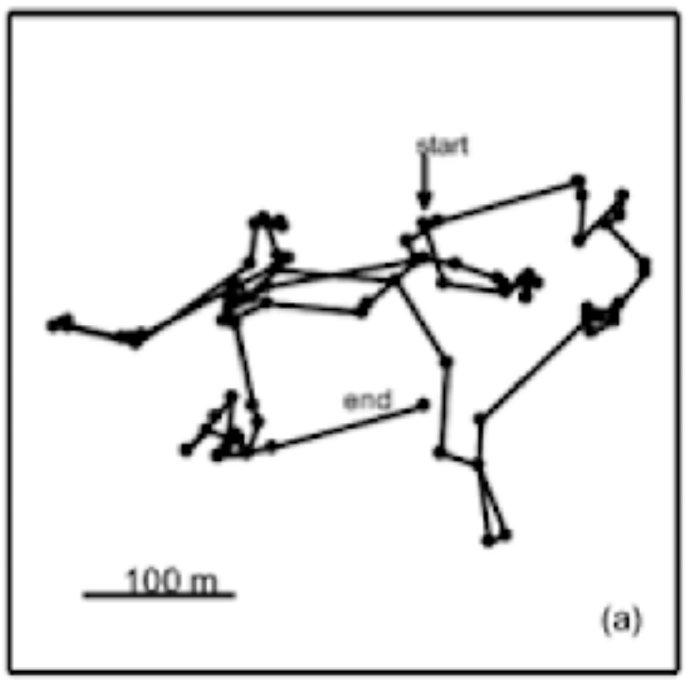
Agent frees itself from room, allowing it to find reward faster

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

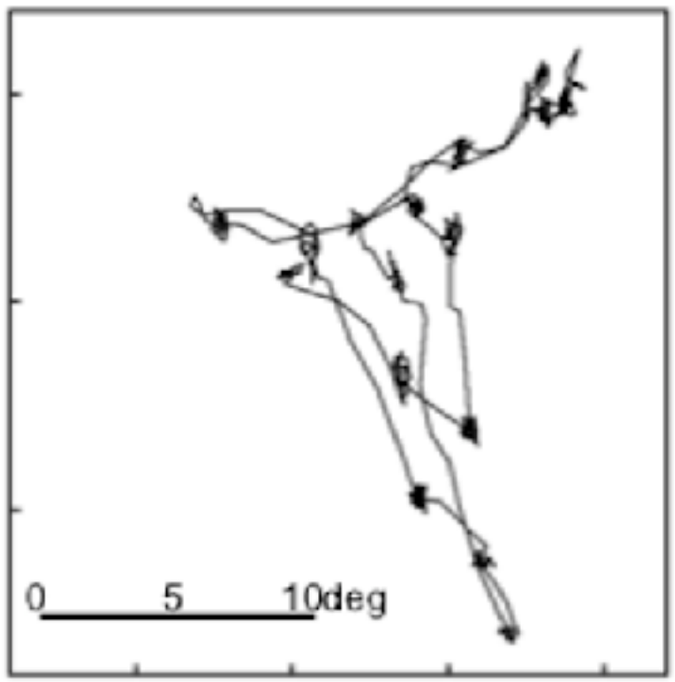
Super diffusion + Lévy Walks



Albatrosses searching for prey
Ornes et al (2013)



Foraging spider monkeys
Ramos-Fernandez et al (2014)



Human view paths for novel fractals
Marlow et al (2015)

Super-diffusion produces Lévy Walks

Ubiquitously observed in animal foraging (right?)

Better coverage time than diffusion

McNamee, Stachenfeld, Botvinick, Gershman (in prep)

Neuron

Available online 25 February 2019

In Press, Corrected Proof ?

Article

Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion

Federico Stella [1], Peter Baracskay [1, 2], Joseph O'Neill [1, 3], Jozsef Csicsvari [1, 4]

Highlights

- Hippocampal replay can represent Brownian diffusion-like random trajectories
- Reactivated trajectories cover positions over wide ranges of spatiotemporal scales
- Replay event statistics are incompatible with actual behavioral trajectories
- Expression dynamics of replayed assemblies was linked to specific oscillatory bands

Neuron

Available online 25 February 2019

In Press, Corrected Proof ?

Article

Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion

Federico Stella [1], Peter Baracskay [1, 2], Joseph O'Neill [1, 3], Jozsef Csicsvari [1, 4]

Highlights

- Hippocampal replay can represent Brownian diffusion-like random trajectories
- Reactivated trajectories cover positions over wide ranges of spatiotemporal scales
- Replay event statistics are incompatible with actual behavioral trajectories
- Expression dynamics of replayed assemblies was linked to specific oscillatory bands

replay has **diffusive** coefficient, behavior has **superdiffusive** coefficient

Neuron

Available online 25 February 2019

In Press, Corrected Proof ?

Article

Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion

Federico Stella [1], Peter Baracskay [1, 2], Joseph O'Neill [1, 3], Jozsef Csicsvari [1, 4]

Highlights

- Hippocampal replay can represent Brownian diffusion-like random trajectories
- Reactivated trajectories cover positions over wide ranges of spatiotemporal scales
- Replay event statistics are incompatible with actual behavioral trajectories

replay has **diffusive** coefficient, behavior has **superdiffusive** coefficient

"the data indicate that hippocampal circuits have **the built-in ability to generate sequences that link assemblies together across the entire cognitive map, on all temporal and spatial scales**."

Take-aways

- Temporally-organized, spatially-extended state representations can turn support efficient learning + exploration
- Intelligent representation + simple learning -> intelligent learning
- We can use principles from representation learning to understand neural representations

Next steps

- Neuroscience: the data part!
- Machine learning: the neural network part!

Lab

- Work with some graphs
- Look at some of these representations
- See how they generalise differently

Acknowledgements

Dan McNamee

Matt Botvinick

Sam Gershman

Tim Behrens

Jesse Geerts

Neil Burgess

Andrea Banino
Jess Hamrick
Raphael Koster
Kevin Miller
Sam Ritter
Peter Battaglia
Martin Chadwick
Zeb Kurth-Nelson
Ida Mommenejad
Neil Rabinowitz
Francis Song
Dharsh Kumaran
Nathaniel Daw
Chris Summerfield

Tufts UNIVERSITY

PRINCETON Neuroscience INSTITUTE

DeepMind

Contact Info

stachenfeld@google.com

www.neurokim.com