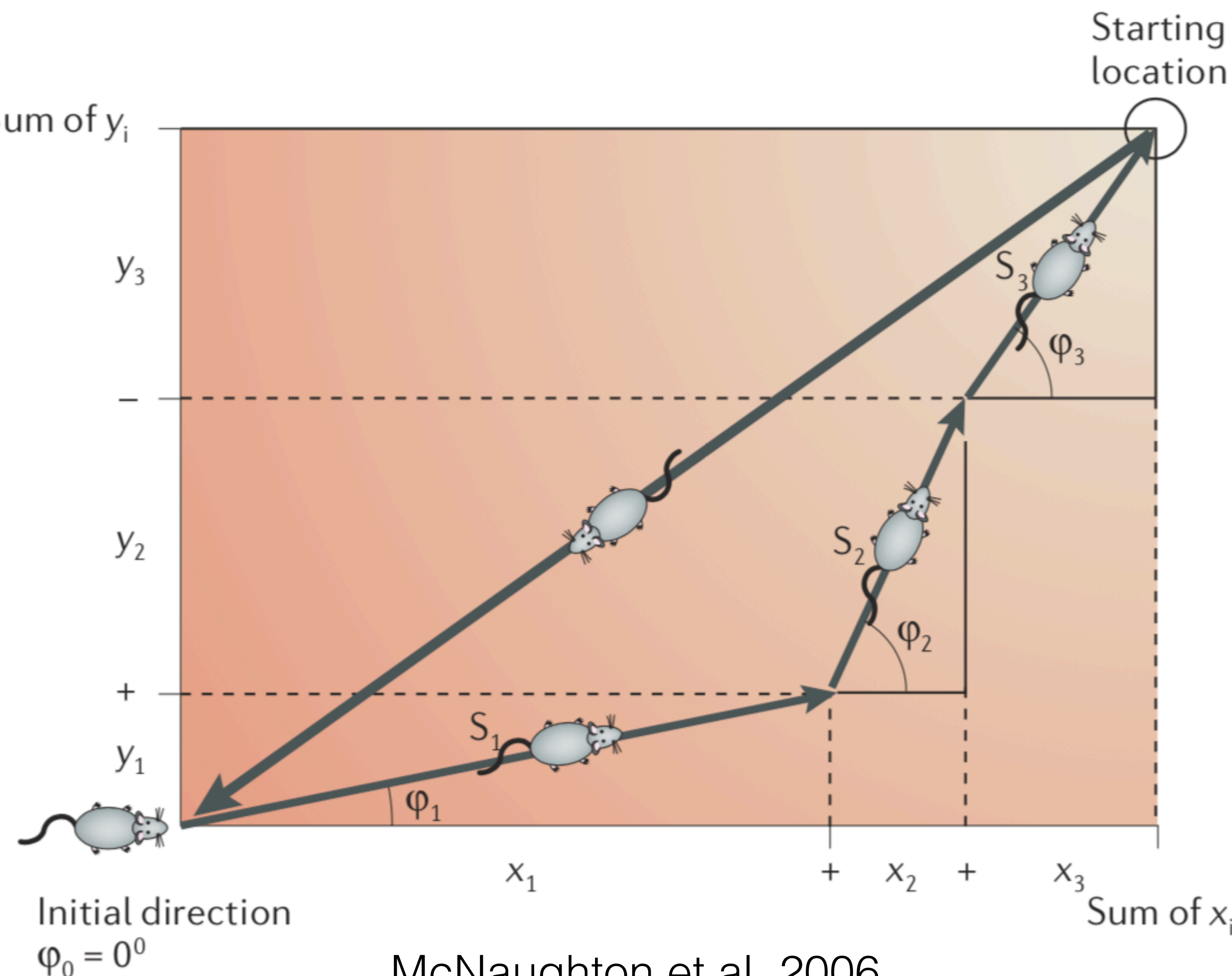
Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks

Aran Nayebi
Neurosciences PhD Candidate
Stanford University

Stanford CNJC
2021.11.17

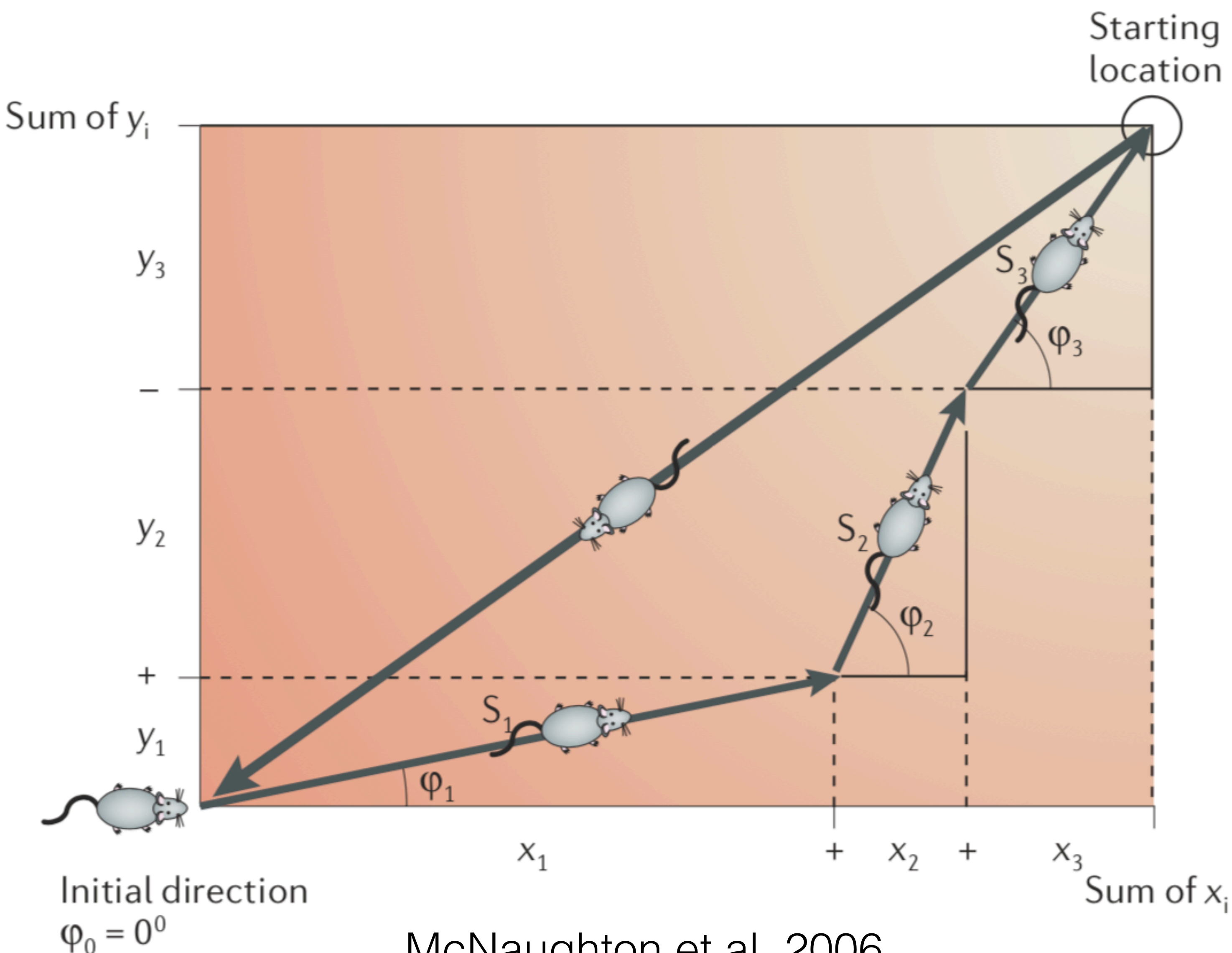
Stanford Neuro AI Lab
Stanford Neuroscience and Artificial Intelligence Lab

Hippocampal-Entorhinal Spatial Map



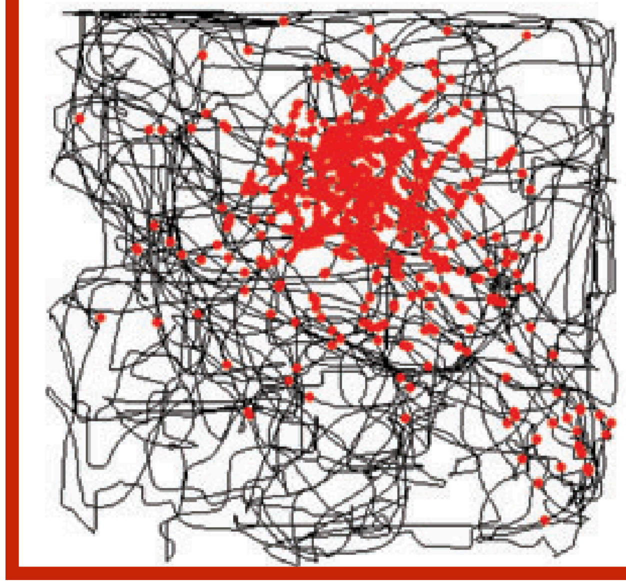
McNaughton et al. 2006

Hippocampal-Entorhinal Spatial Map

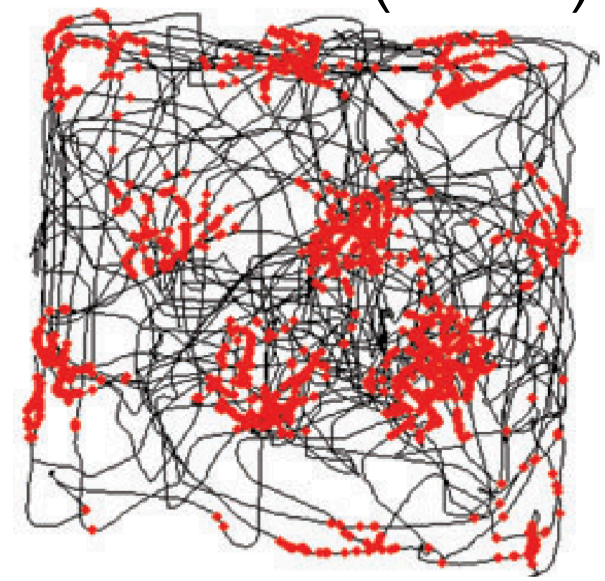


McNaughton et al. 2006

Place Cell (HPC)

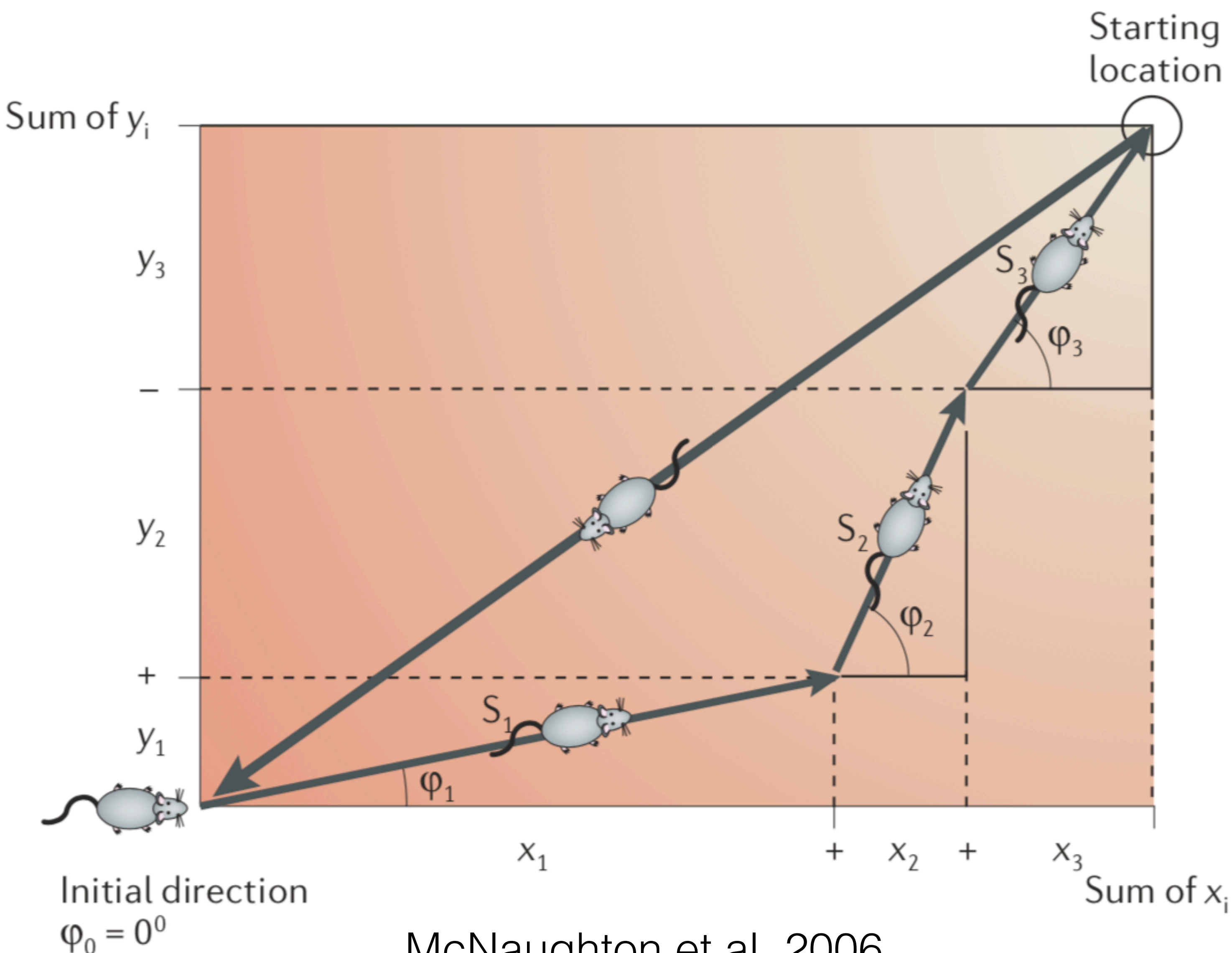


Grid Cell (MEC)



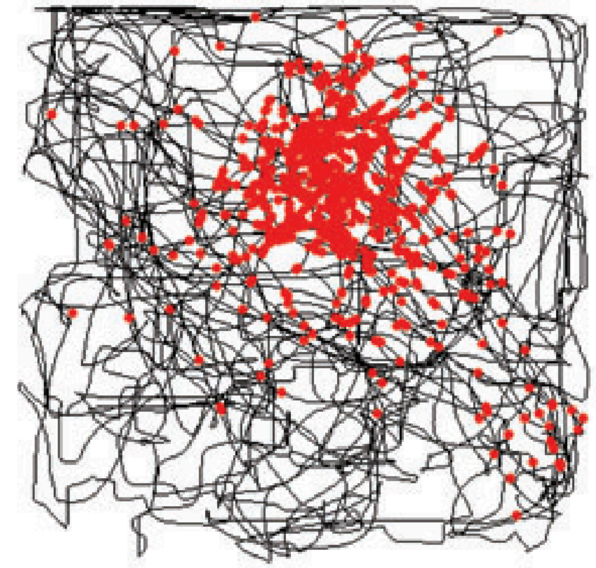
Moser et al. 2008

Hippocampal-Entorhinal Spatial Map

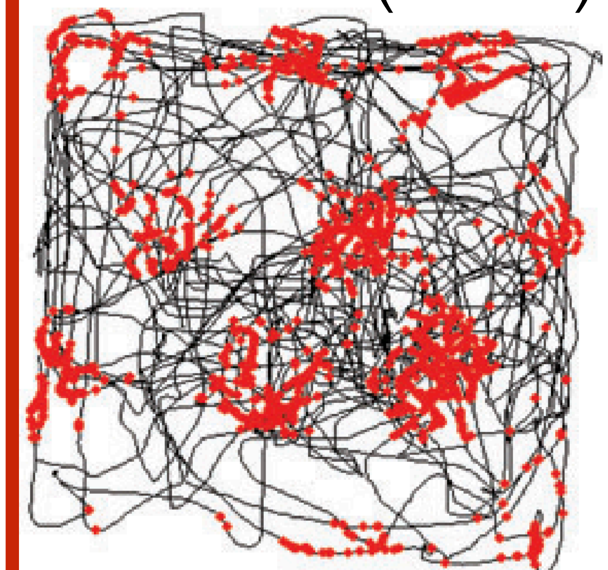


McNaughton et al. 2006

Place Cell (HPC)



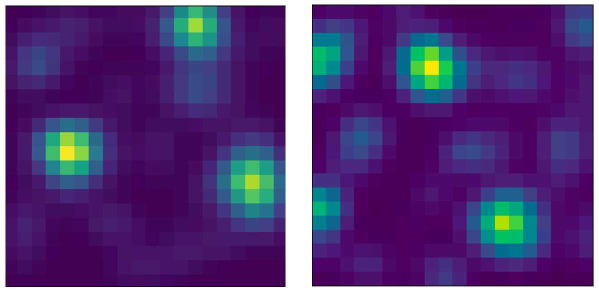
Grid Cell (MEC)

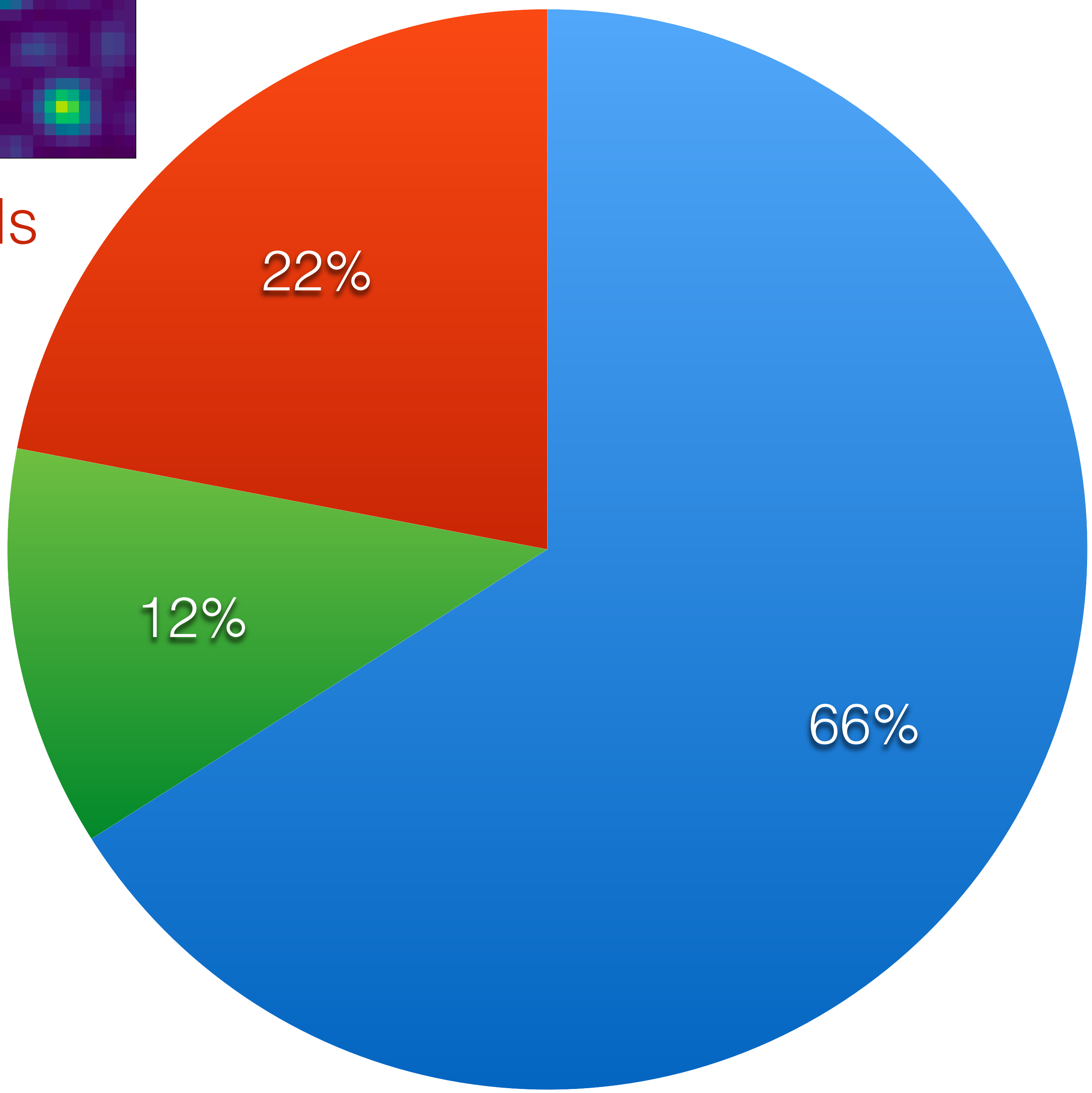


Moser et al. 2008

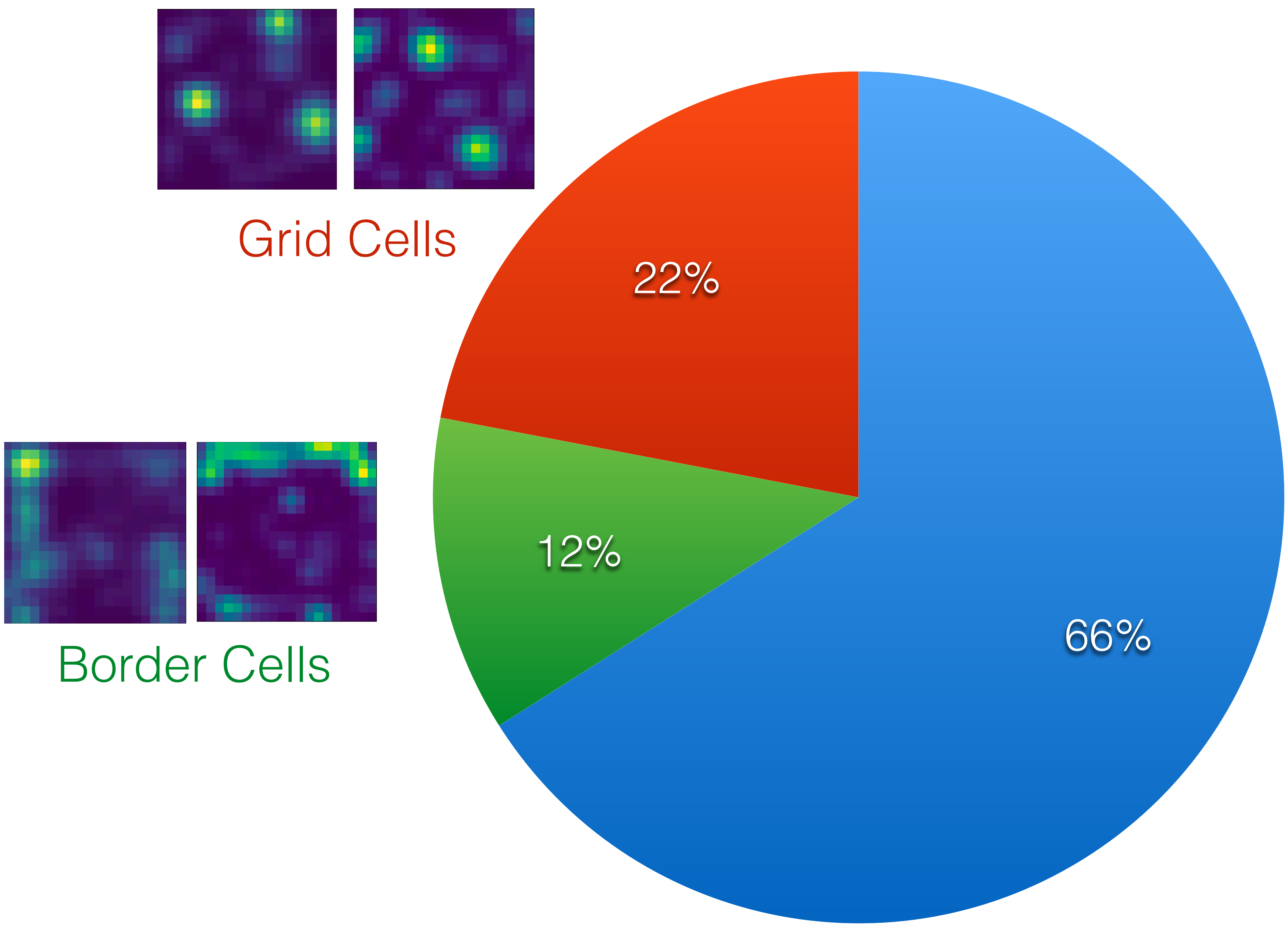
Accounting for heterogeneous code?

Accounting for heterogeneous code?

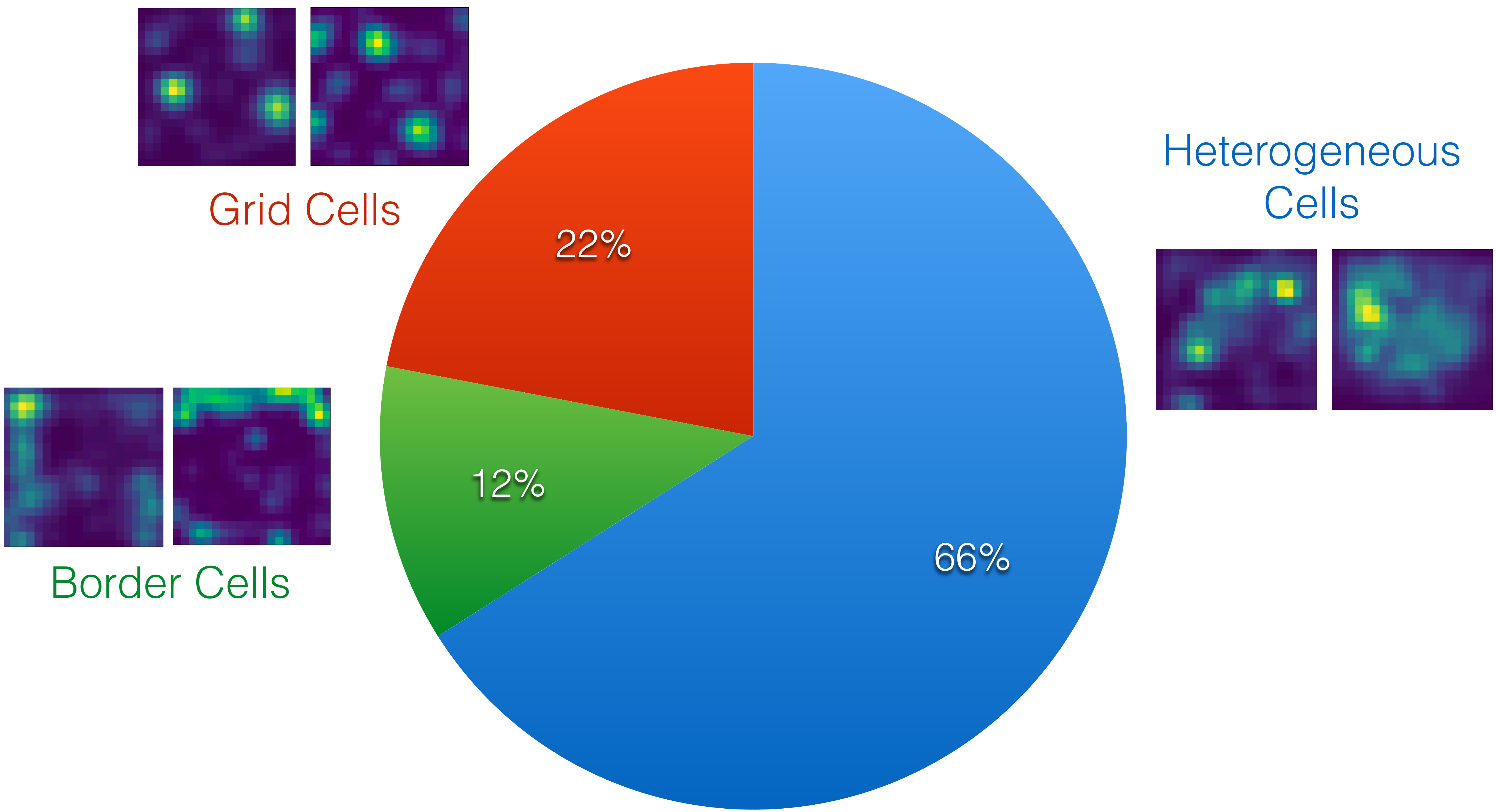




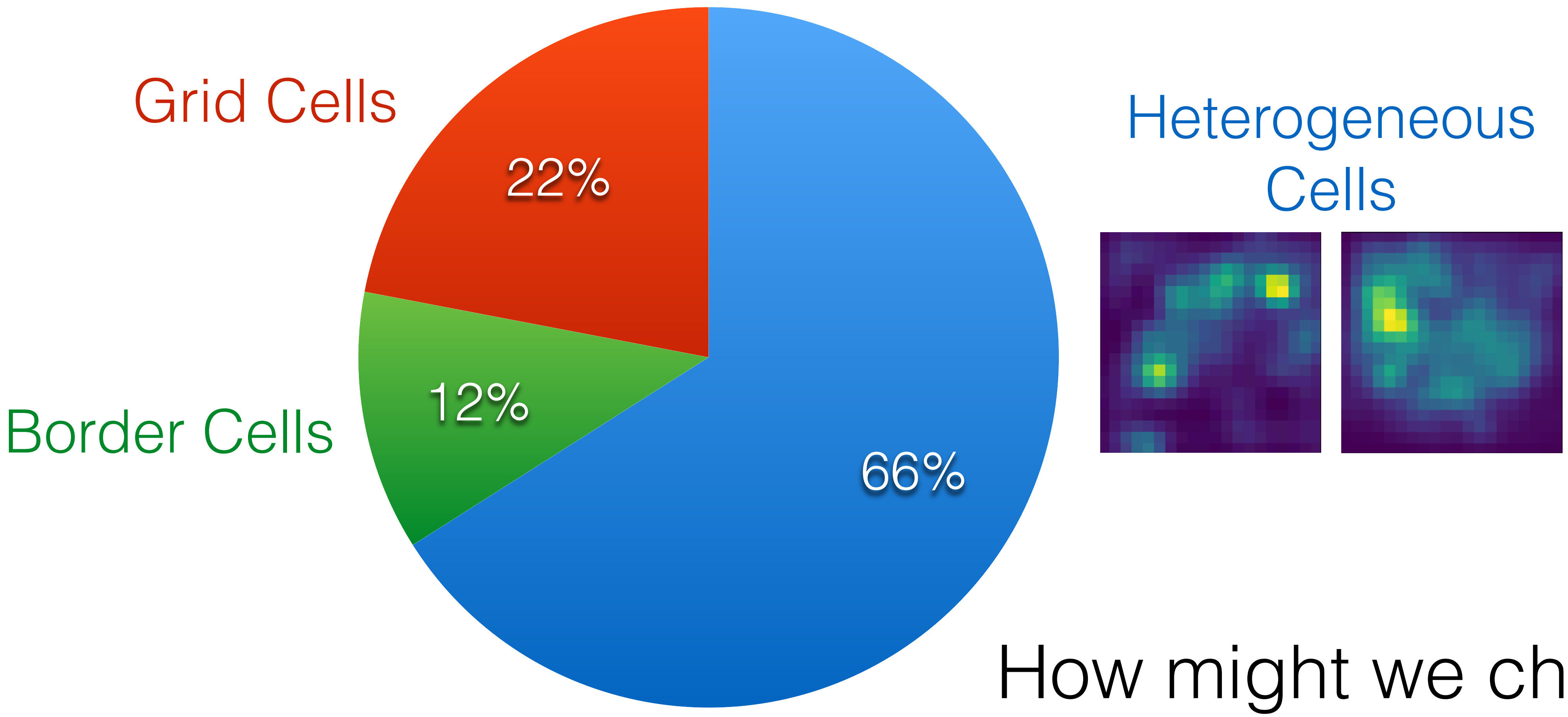
Accounting for heterogeneous code?



Accounting for heterogeneous code?

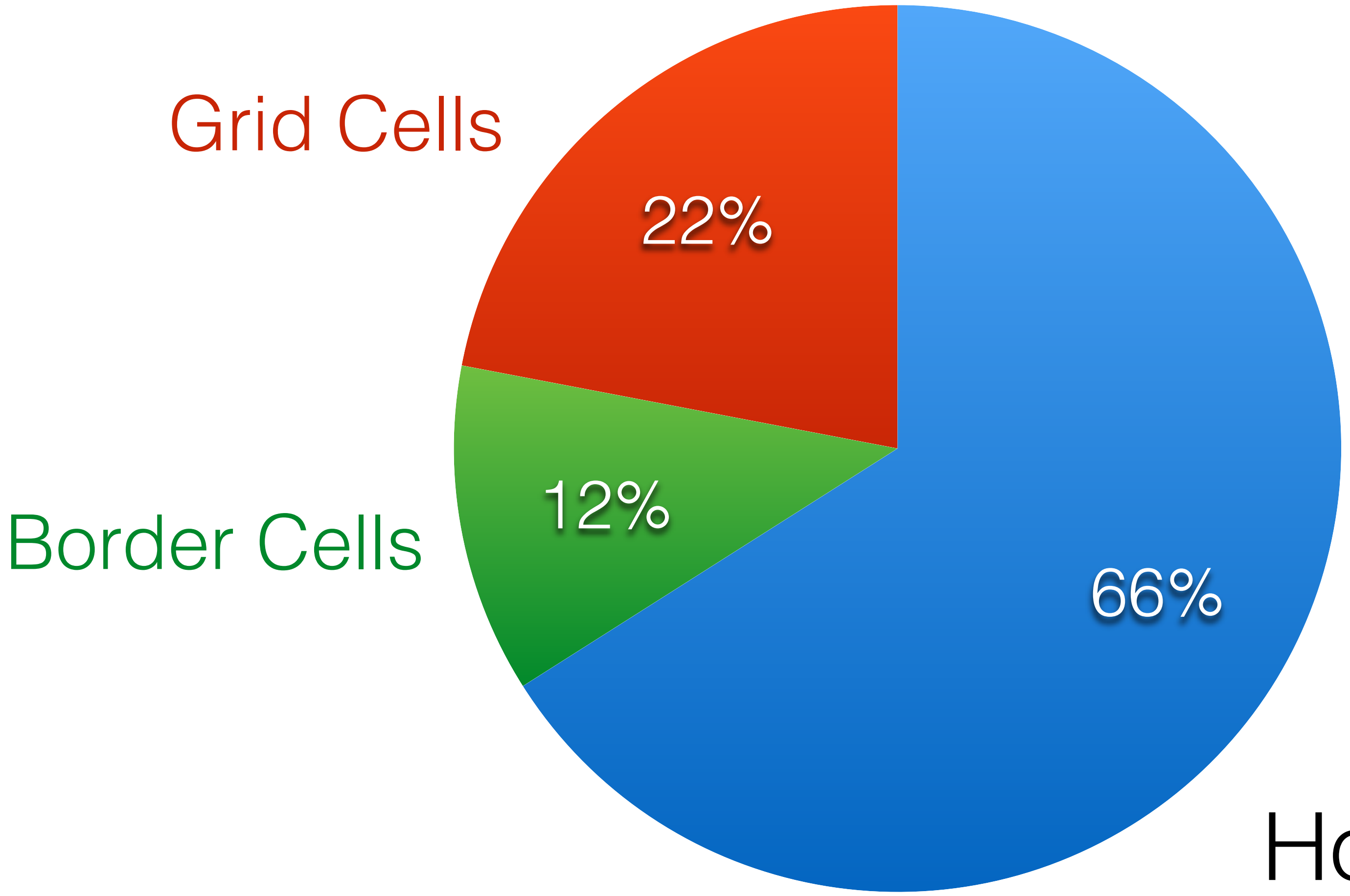


Accounting for heterogeneous code?

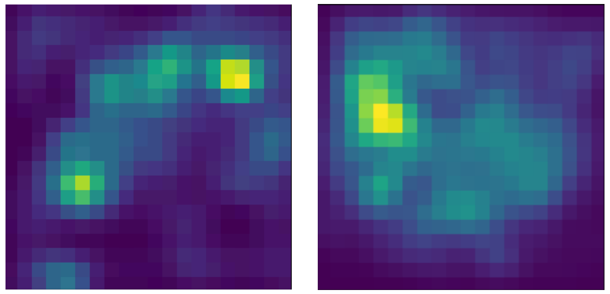


How might we characterize the response patterns of these heterogeneous cells?

Accounting for heterogeneous code?



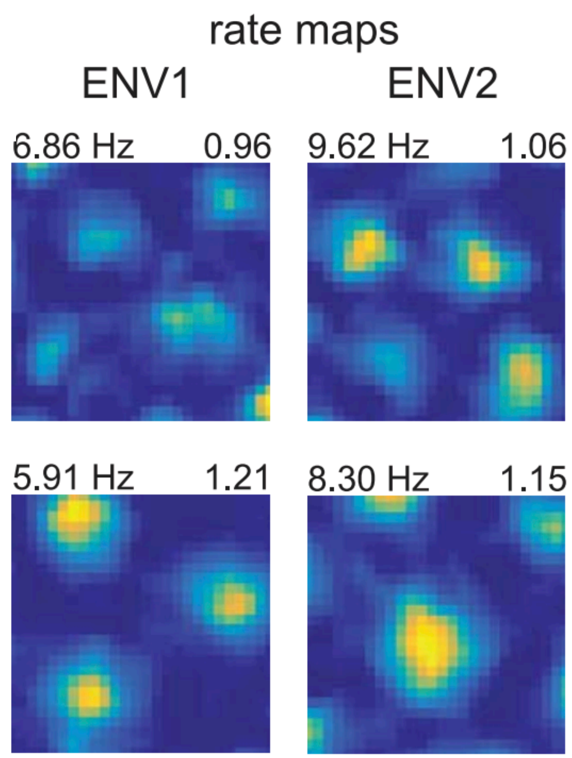
Heterogeneous Cells



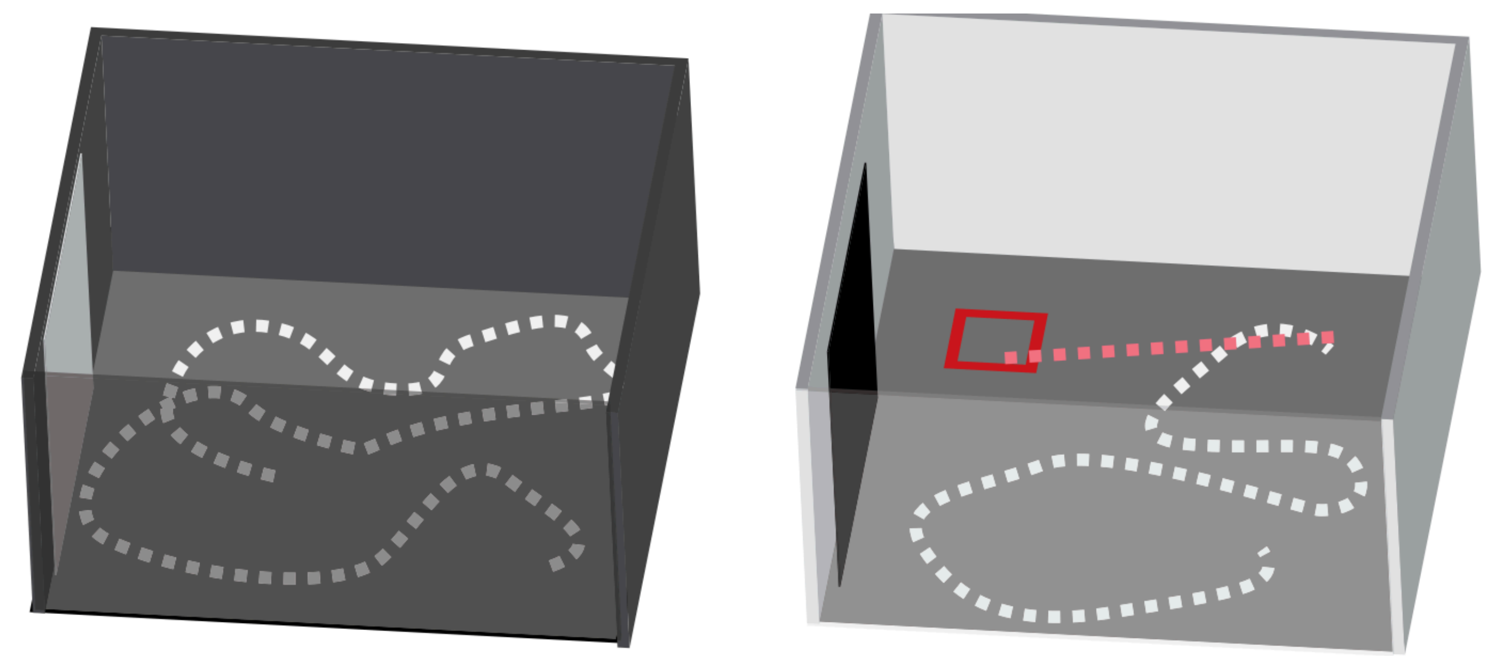
How might we characterize the response patterns of these heterogeneous cells?

What functional role do these cells serve in the circuit, if any?

Accounting for heterogeneous code in the presence of rewards?



Butler*, Hardcastle*, Giocomo 2019



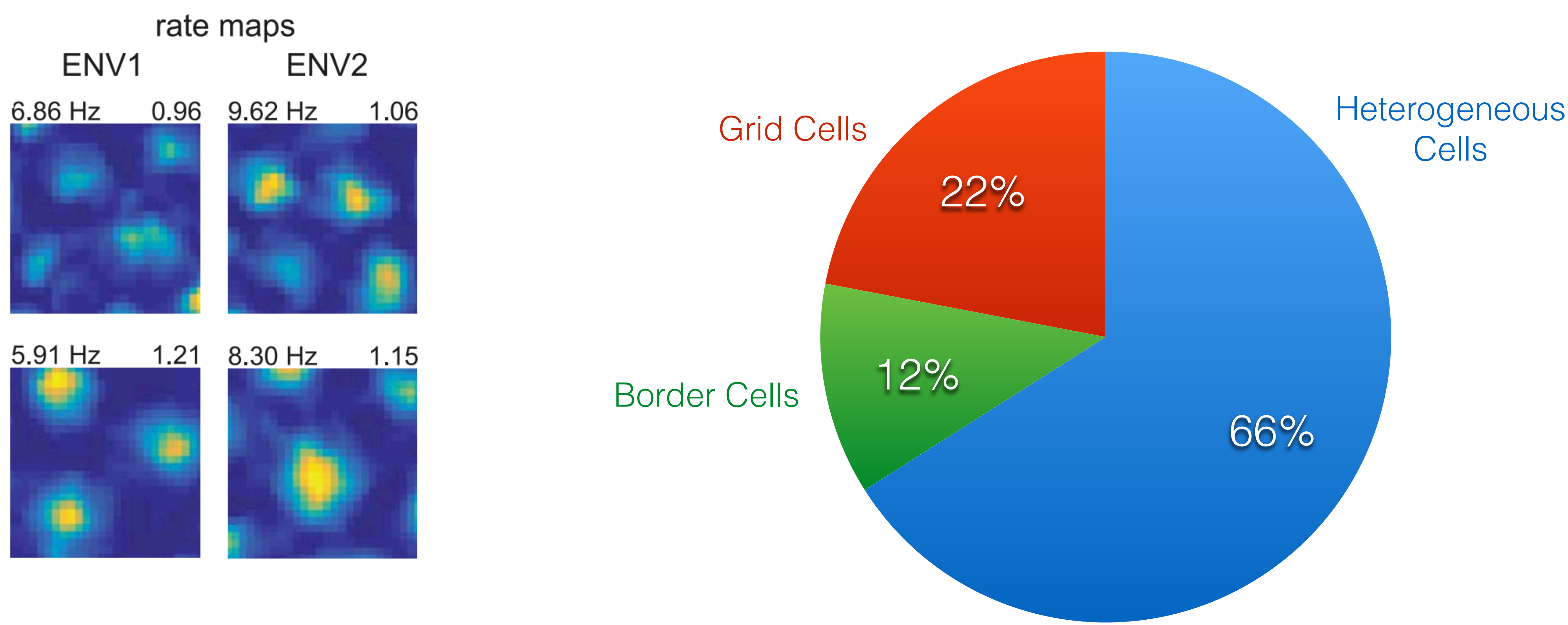
free foraging (ENV1)

spatial task (ENV2)

In fact, MEC remaps in the presence of rewards… so what describes the joint interaction between these heterogeneous cells and reward?

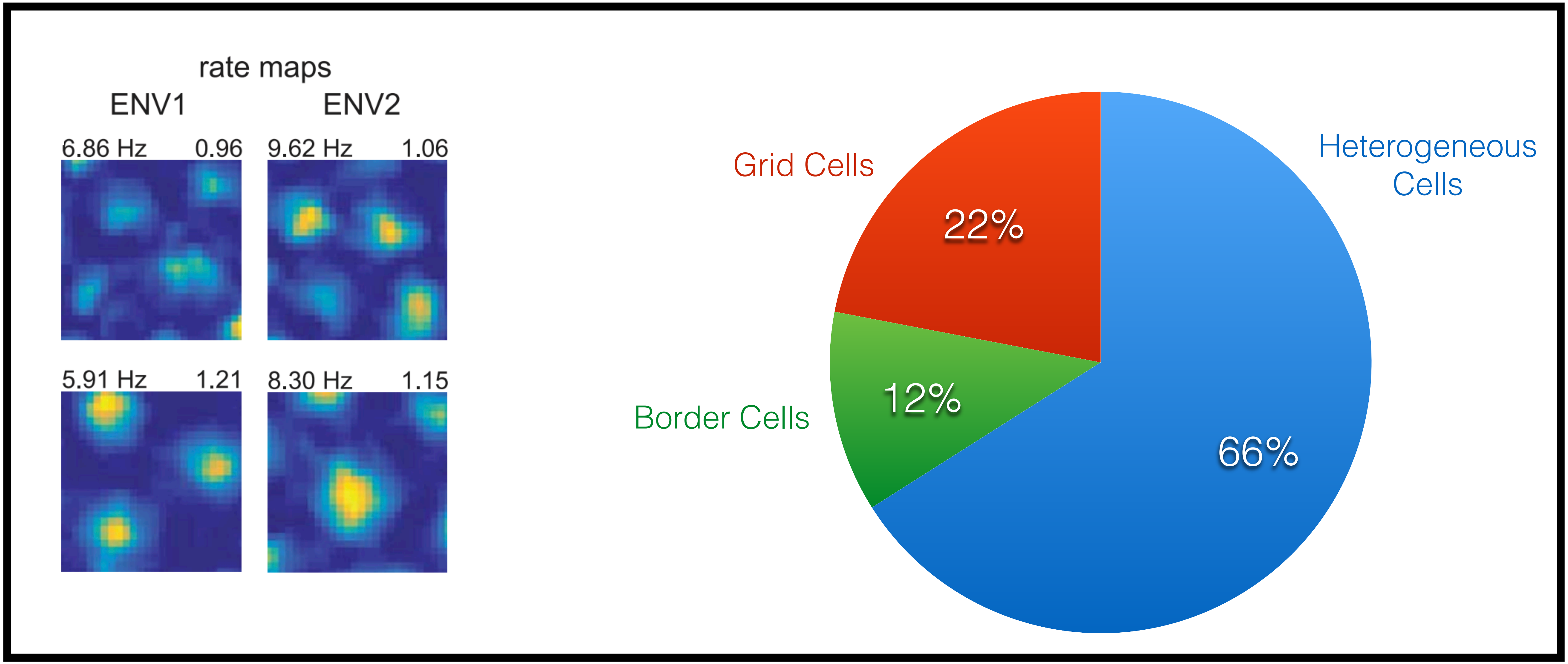
Taking a modeling approach

It would be useful to have a **unified** model that can simultaneously explain different types of neural responses in MEC.



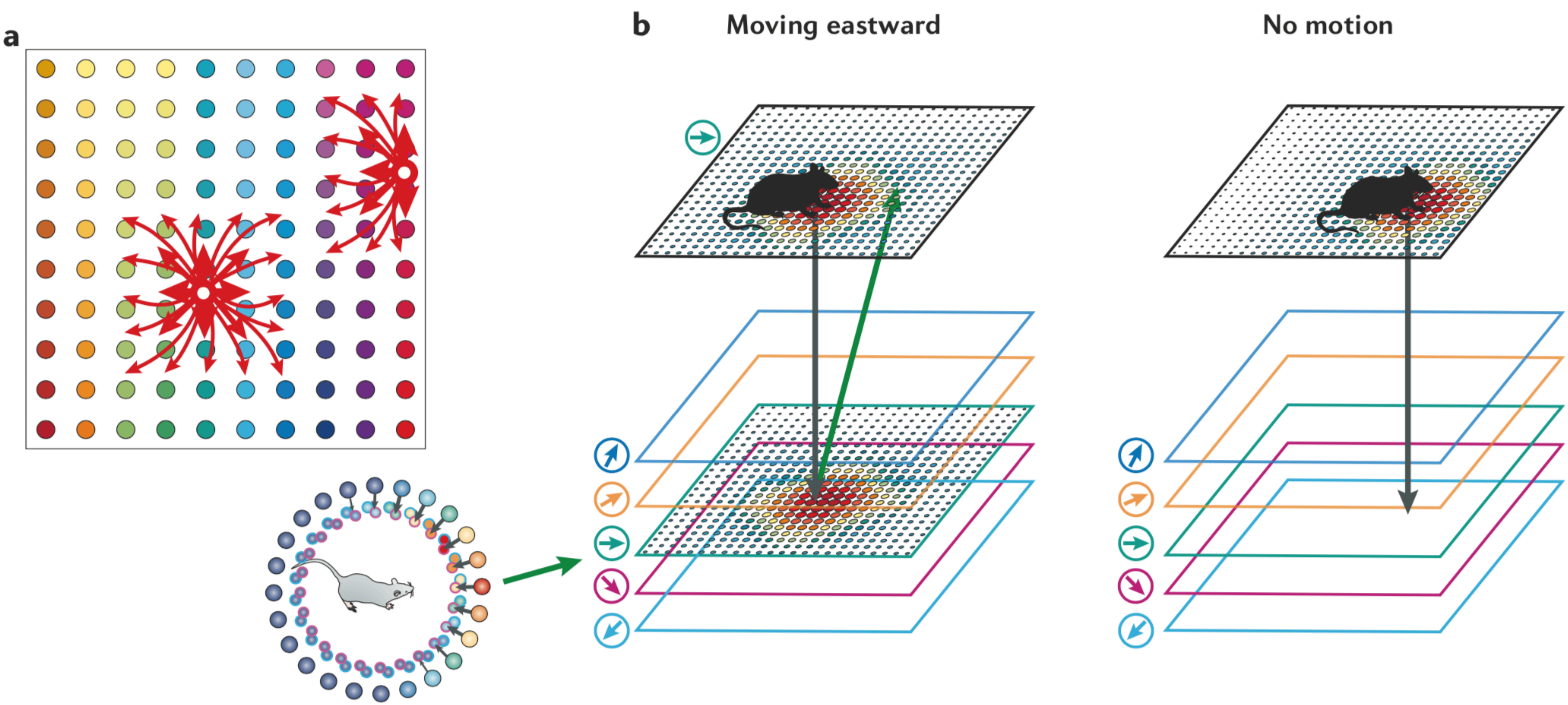
Taking a modeling approach

It would be useful to have a **unified** model that can simultaneously explain different types of neural responses in MEC.



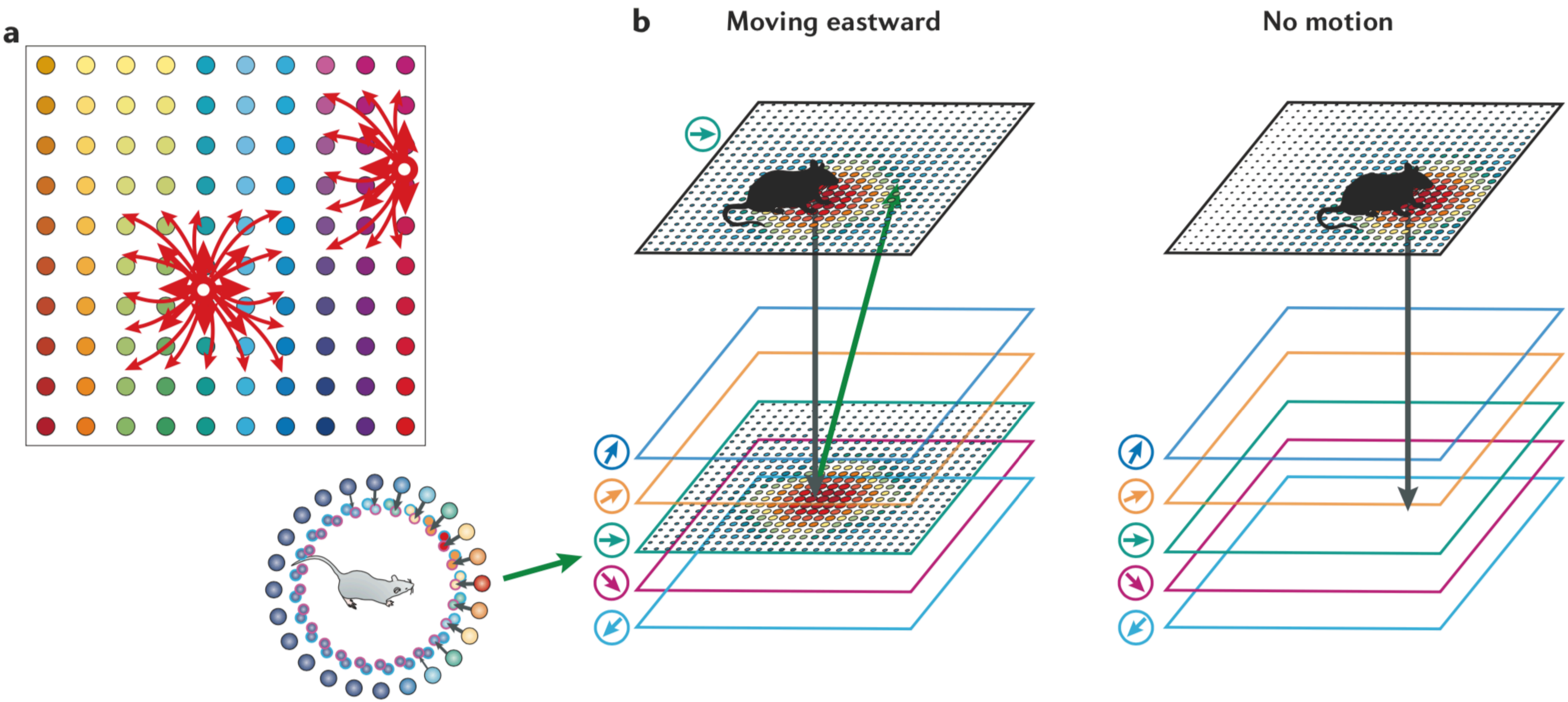
Where do we begin?

“Hand-Tuned” Attractor Models - 2D Case



McNaughton et al. 2006

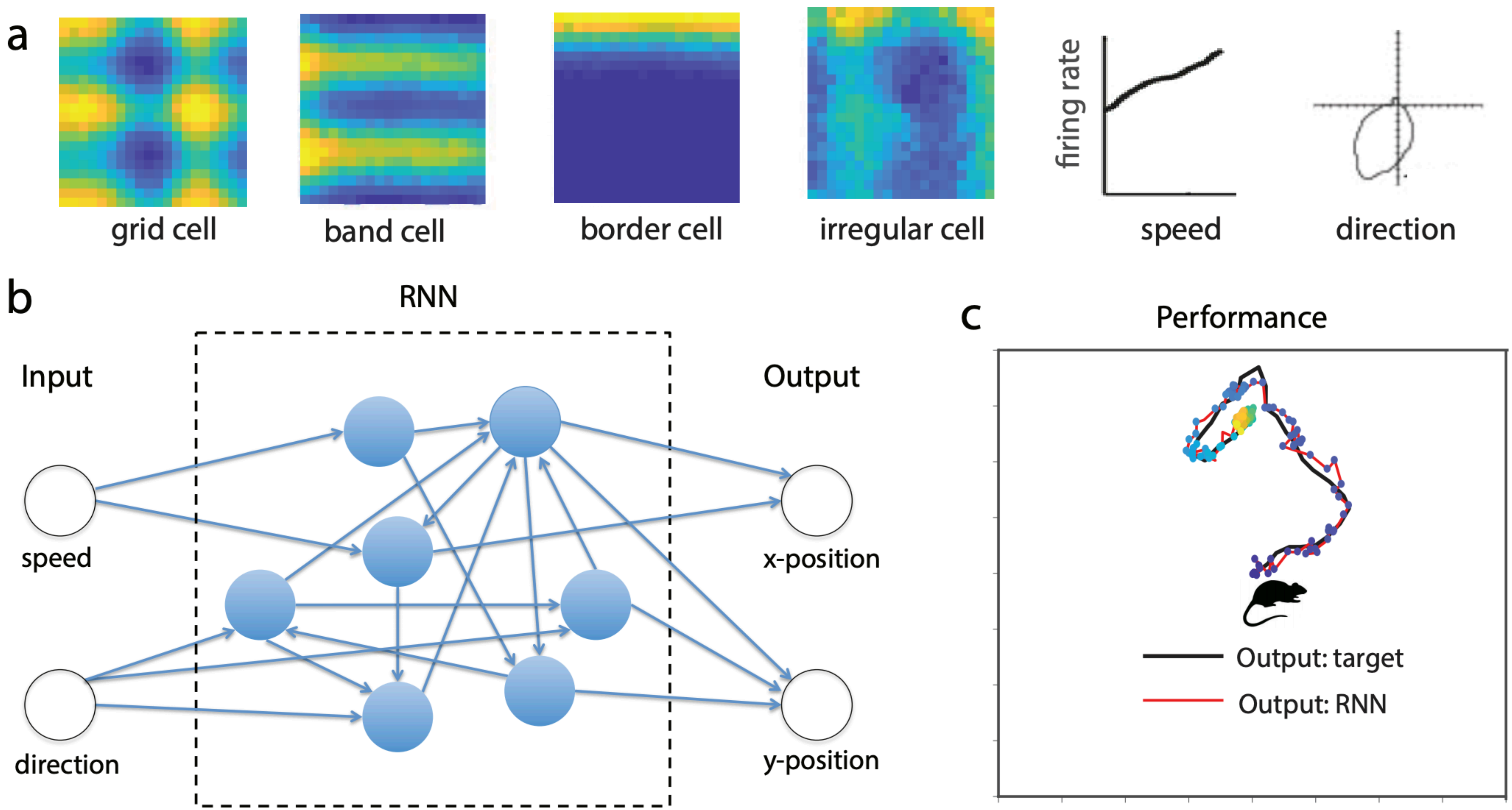
"Hand-Tuned" Attractor Models - 2D Case



McNaughton et al. 2006

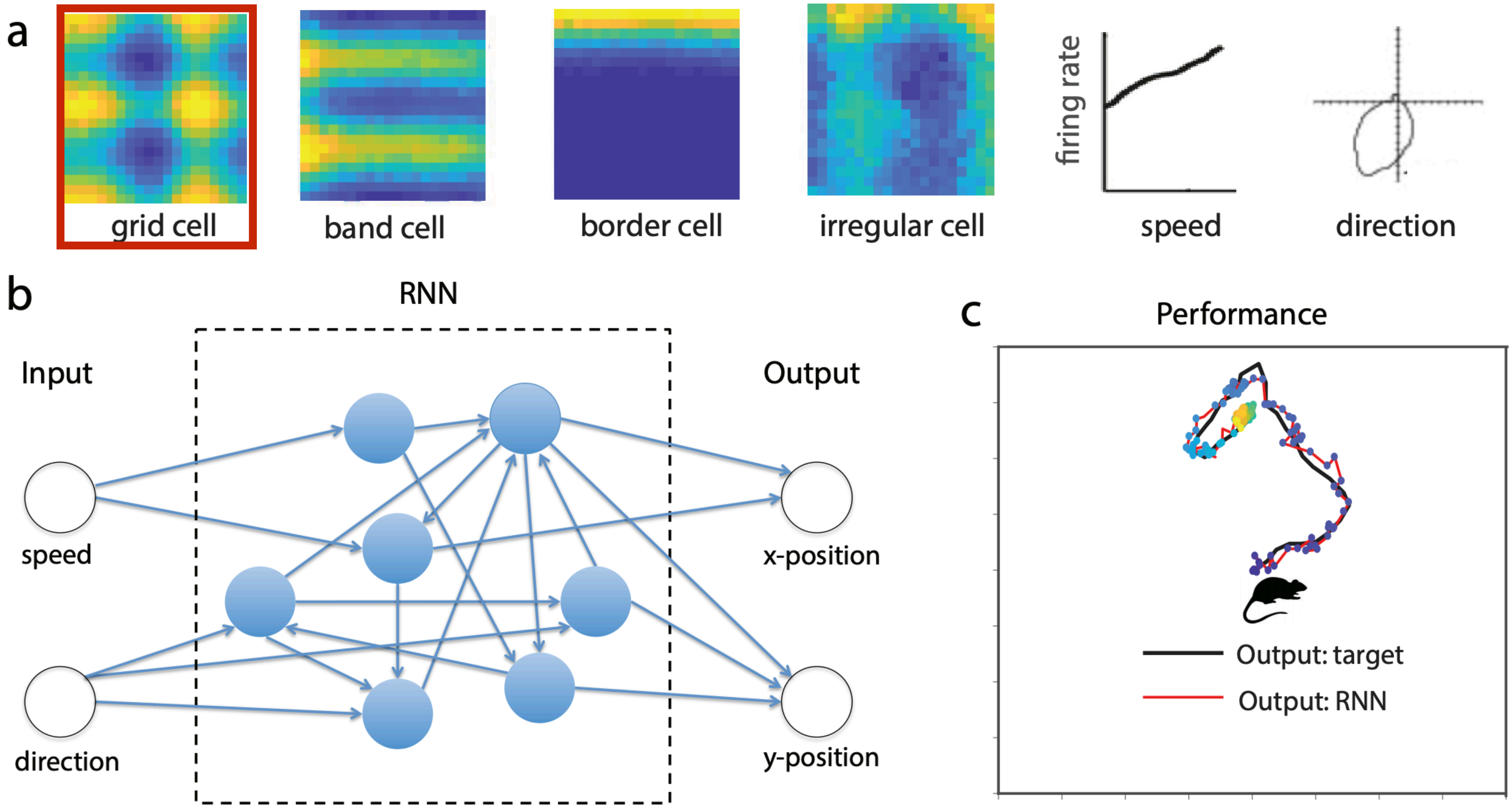
But these hand tuned models capture the properties of stereotypical cell-type classes

But more recently there are neural network models that "develop" these cells…



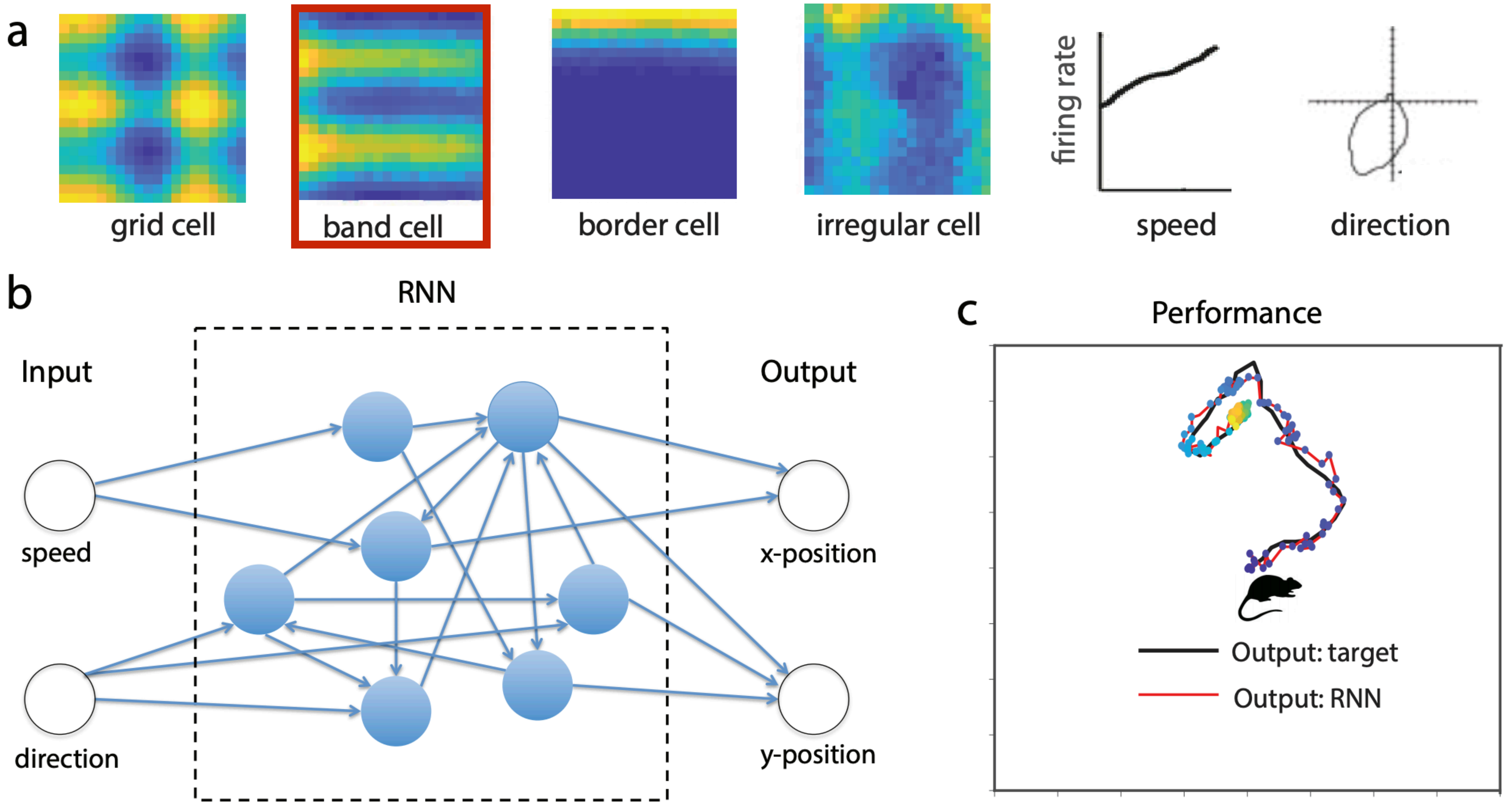
Cueva* & Wei* 2018

But more recently there are neural network models that "develop" these cells…



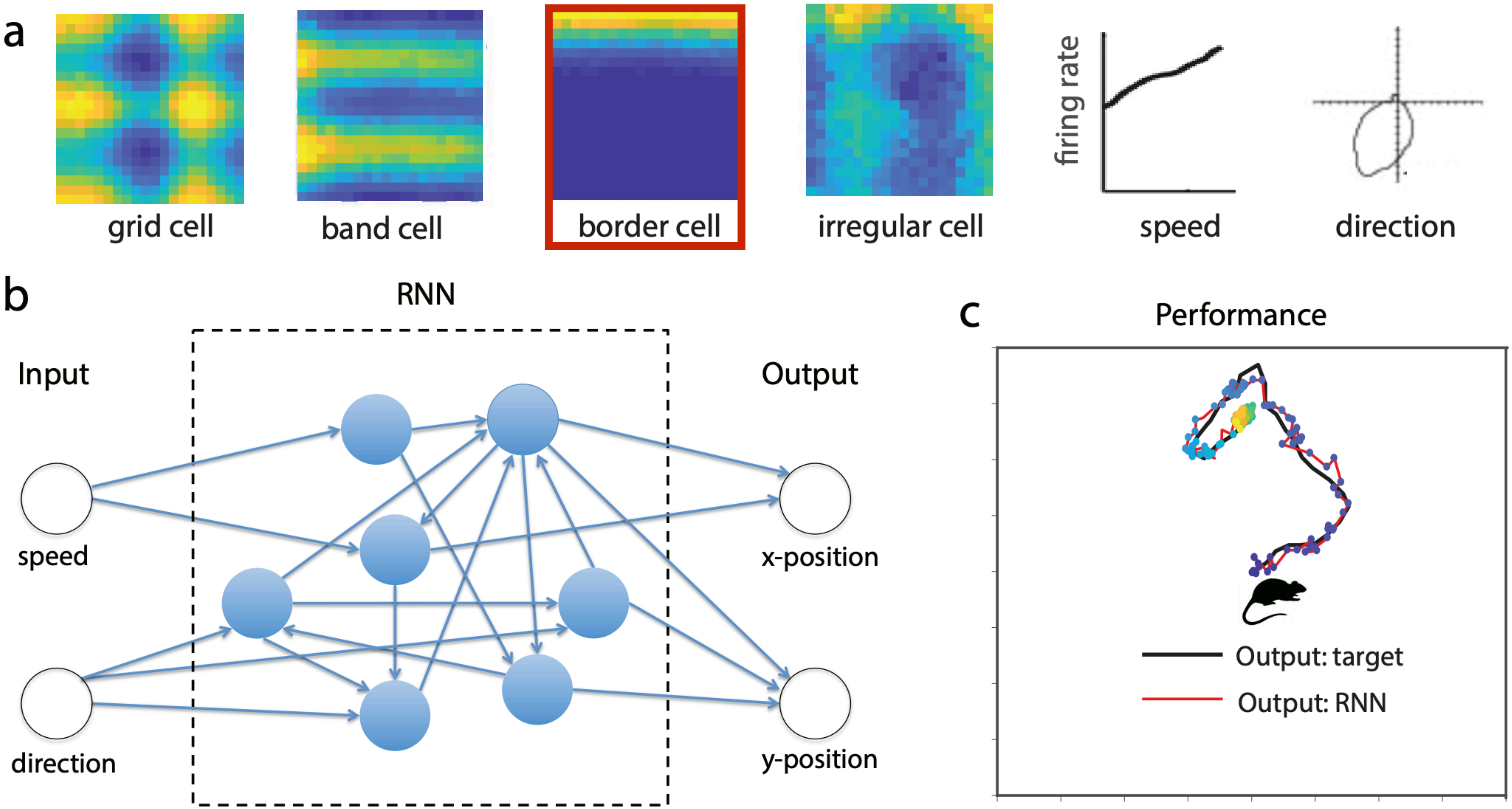
Cueva* & Wei* 2018

But more recently there are neural network models that "develop" these cells…



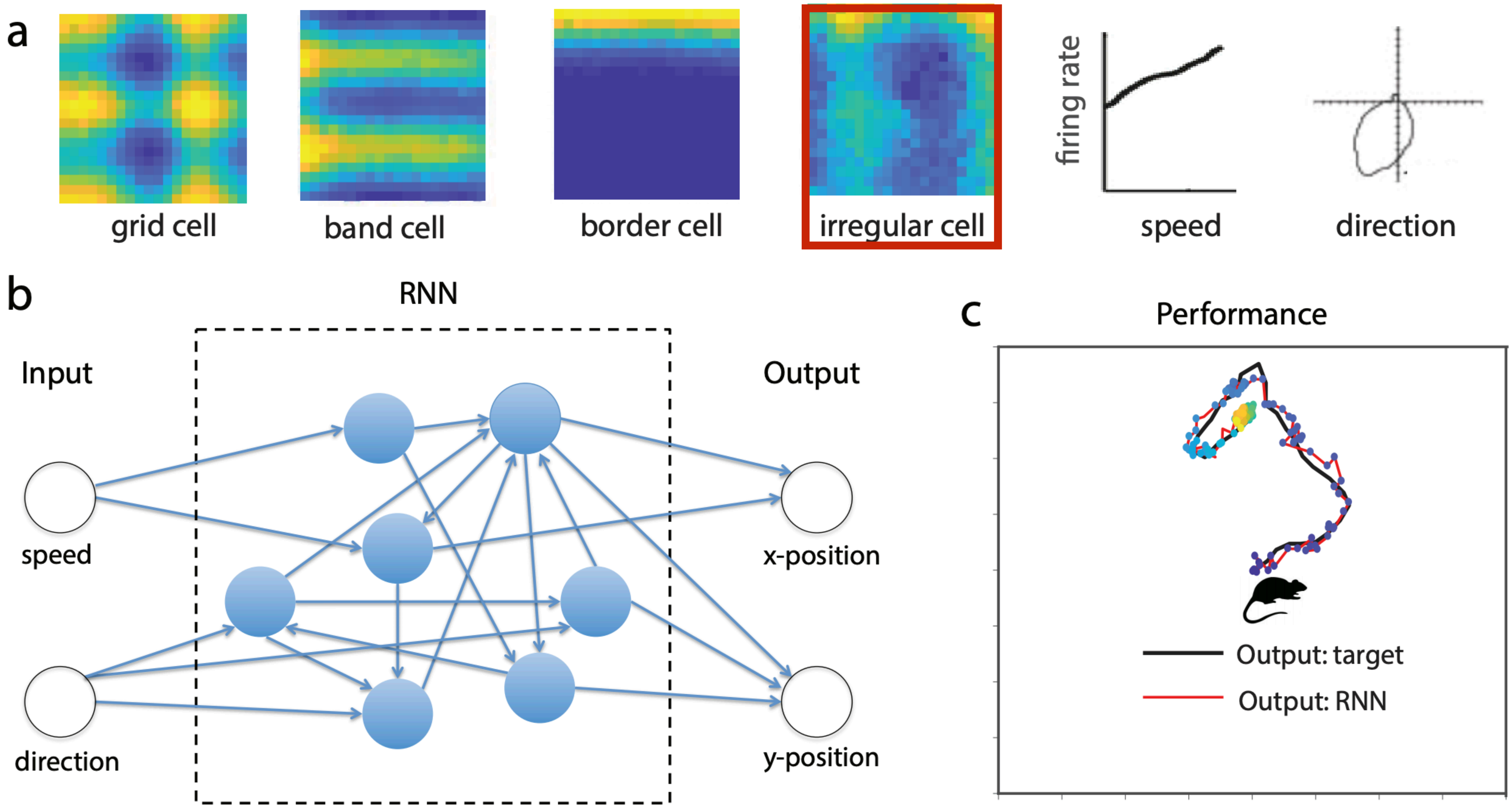
Cueva* & Wei* 2018

But more recently there are neural network models that "develop" these cells…



Cueva* & Wei* 2018

But more recently there are neural network models that "develop" these cells…



Cueva* & Wei* 2018

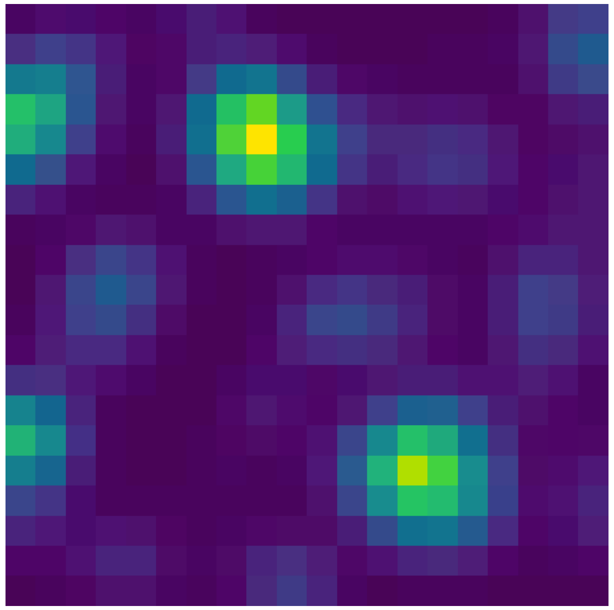
Main Questions

But are they a good ***quantitative*** model of these responses?

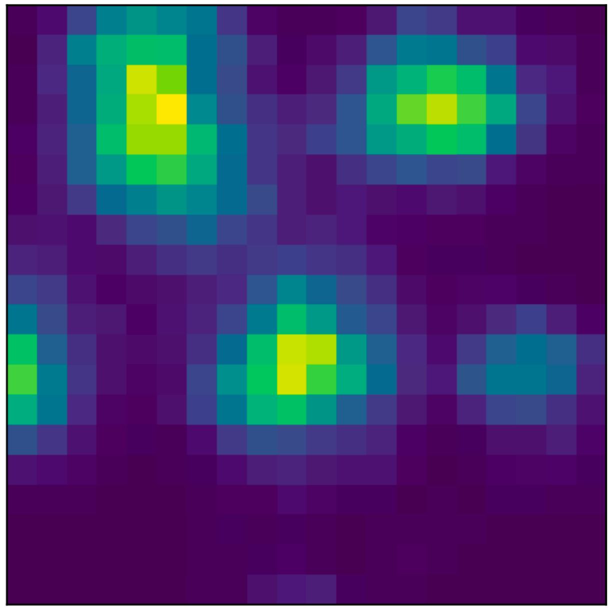
Main Questions

But are they a good ***quantitative*** model of these responses?

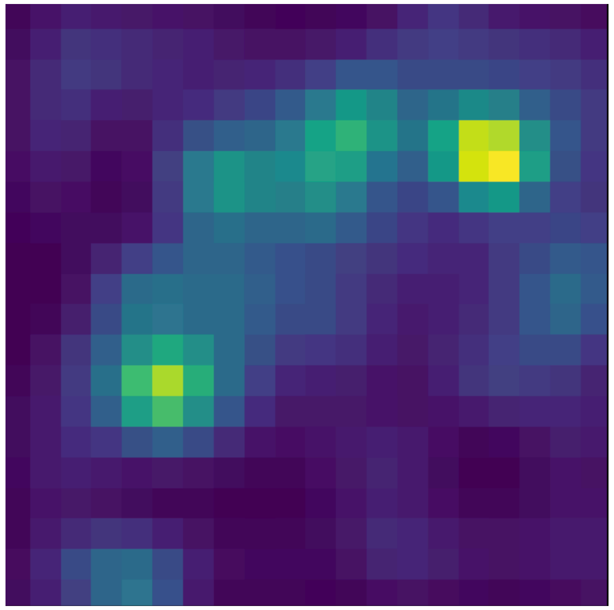
MEC Grid Cell



Model Grid Cell

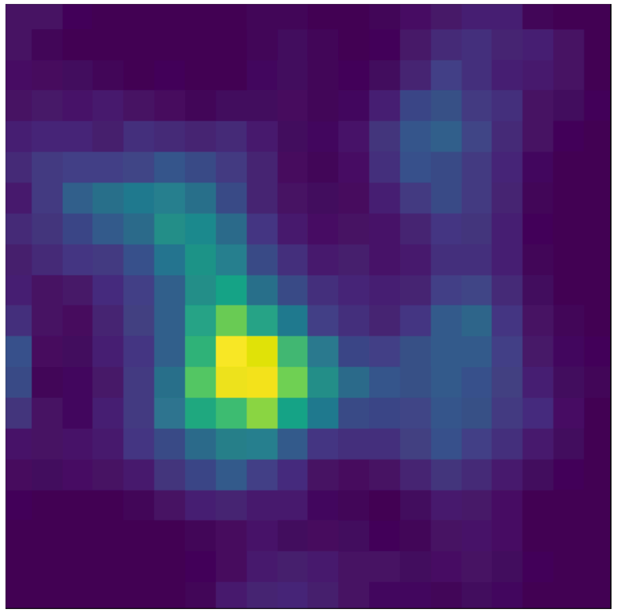


MEC Heterogeneous Cell



?

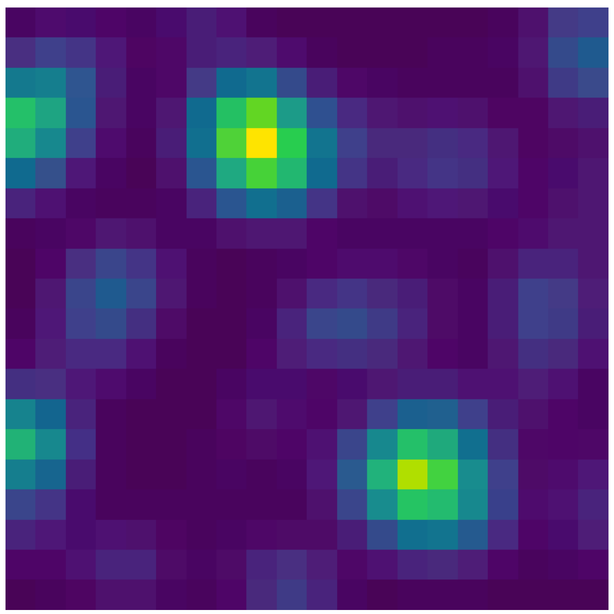
Model Heterogeneous Cell



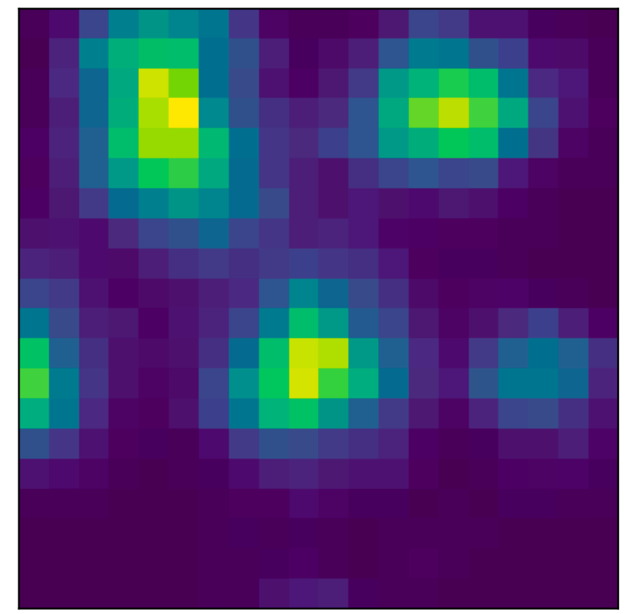
Main Questions

But are they a good ***quantitative*** model of these responses?

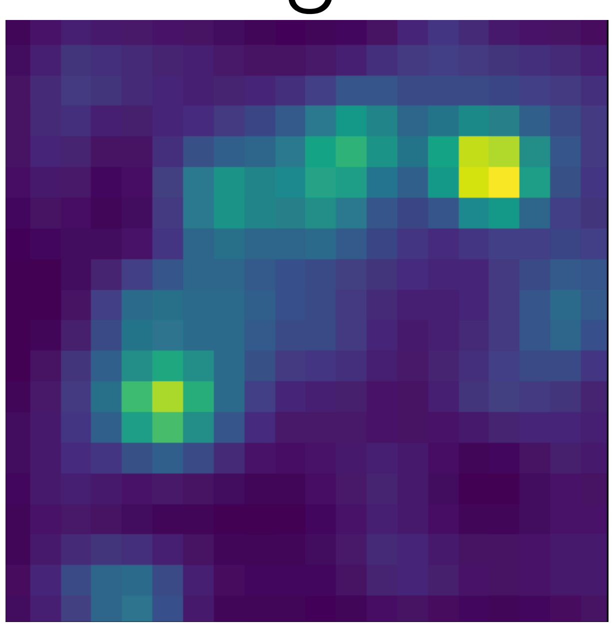
MEC Grid Cell



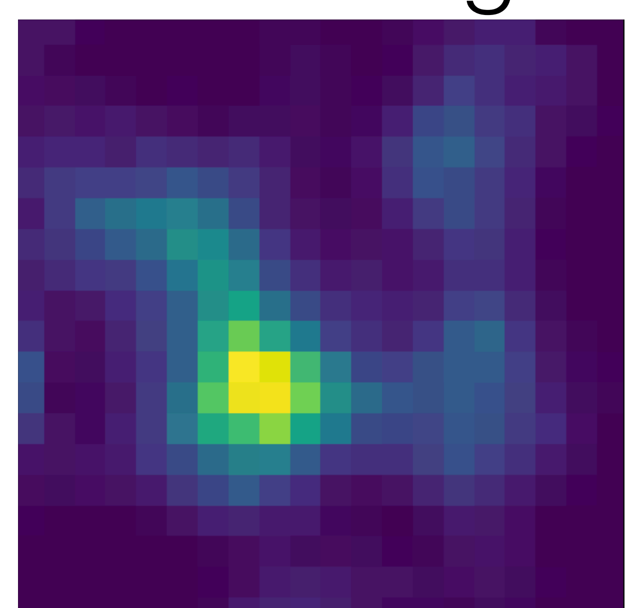
Model Grid Cell



MEC Heterogeneous Cell



Model Heterogeneous Cell

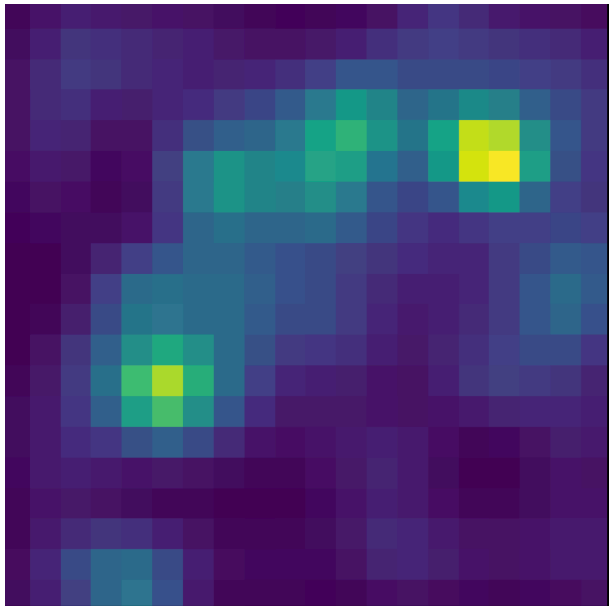


Not all models are equal!

Main Questions

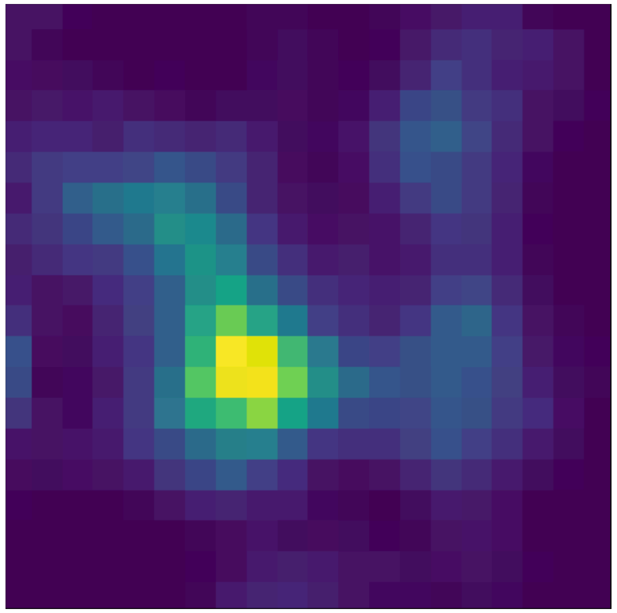
How do we define similarity between sets of heterogeneous responses we can't adequately describe in words?

MEC Heterogeneous Cell



?

Model Heterogeneous Cell



Goal-Driven Approach

"behavior"

1.

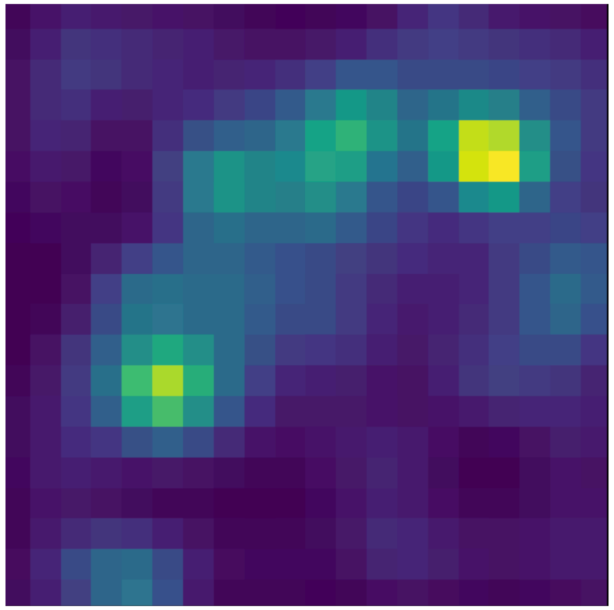
L = *loss function* ***D*** = *dataset*

"circuit"

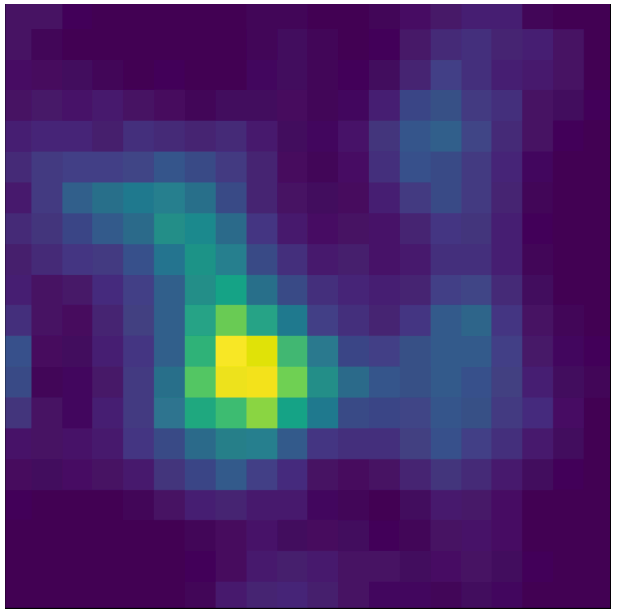
2.

A = *architecture class*

MEC Heterogeneous Cell



Model Heterogeneous Cell



Main Questions

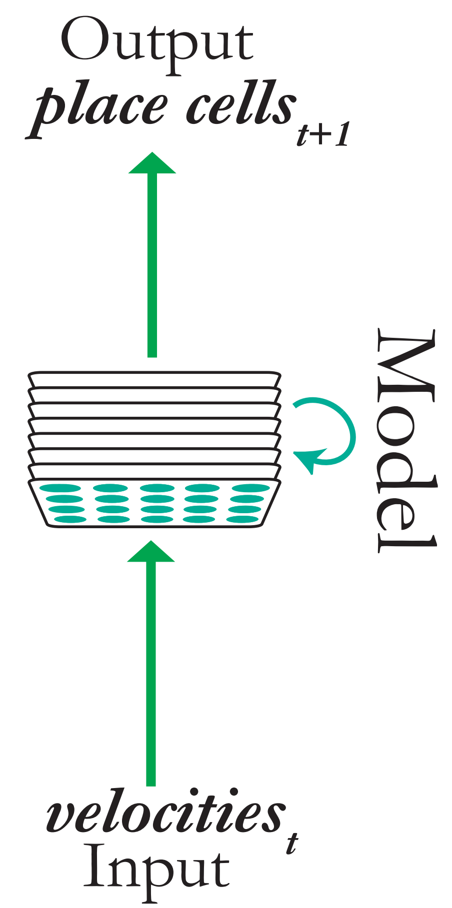
Before we do the goal driven approach, how do we even measure if a model is correct?

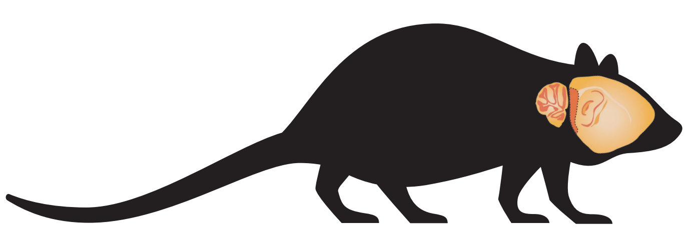
Our approach is that a model should be like the system is unto itself.

Overall procedure

Overall procedure

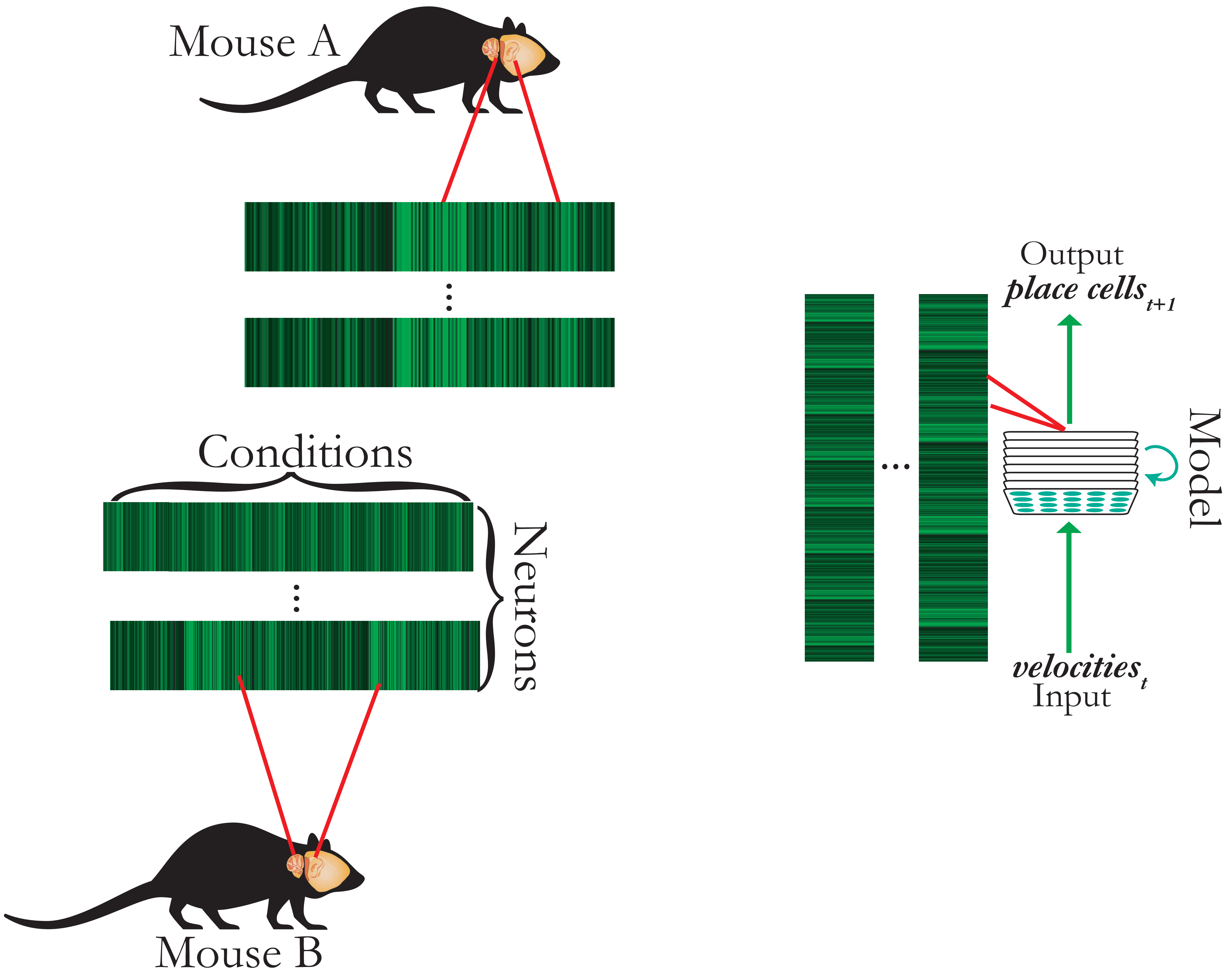
Mouse A



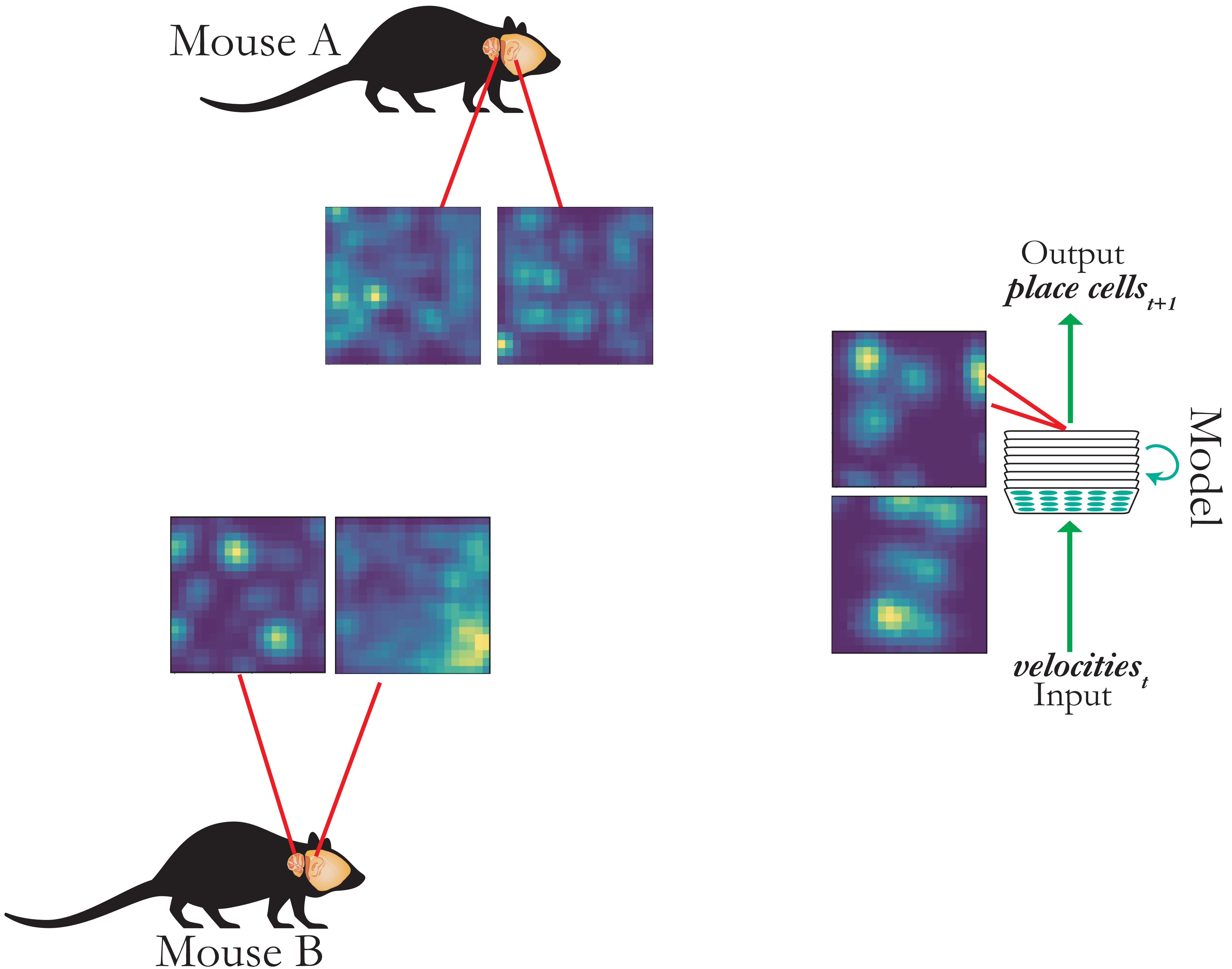


Mouse B

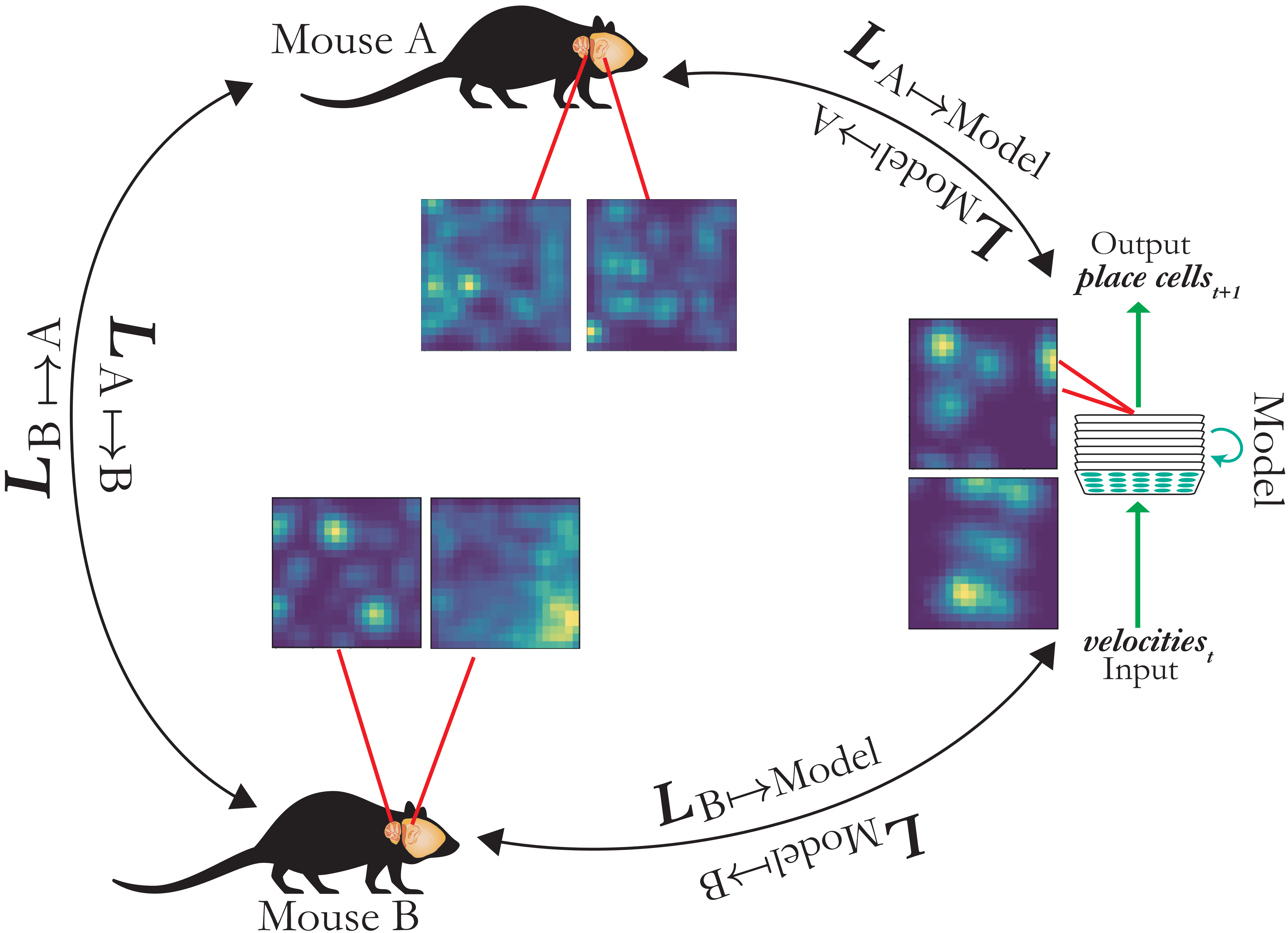
Overall procedure



Overall procedure



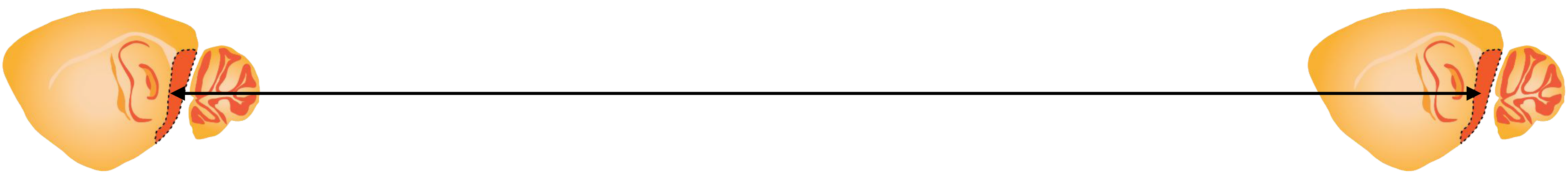
Overall procedure



Spectrum of assumptions

Most Sparse
Source MEC

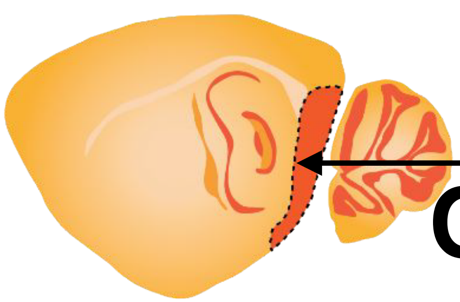
Least Sparse
Target MEC

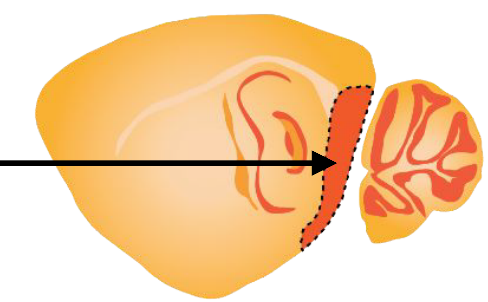


Spectrum of assumptions: One-to-One

Most Sparse

Source MEC





Least Sparse

Target MEC

One-to-One:

Find the most correlated neuron in the source animal to the target neuron

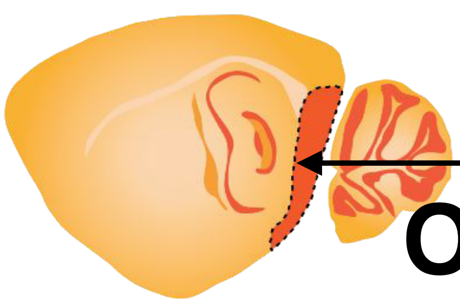
Spectrum of assumptions: Sparse Linear

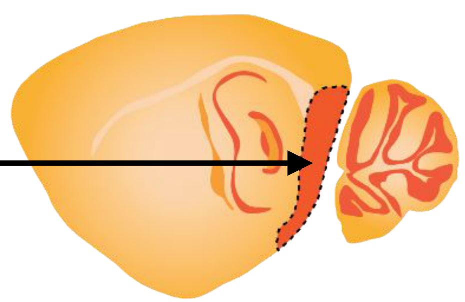
Most Sparse

Source MEC

Least Sparse

Target MEC





One-to-One **Lasso**

Spectrum of assumptions: Sparse Linear

Spectrum of assumptions: “Full” Linear

Most Sparse

Source MEC

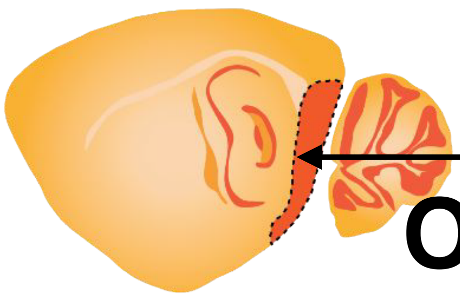
Least Sparse

Target MEC

Spectrum of assumptions: "Full" Linear

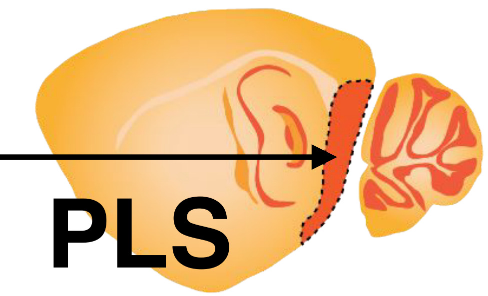
Most Sparse

Source MEC



Least Sparse

Target MEC



One-to-One **Lasso** **ElasticNet** **Ridge** **PLS**

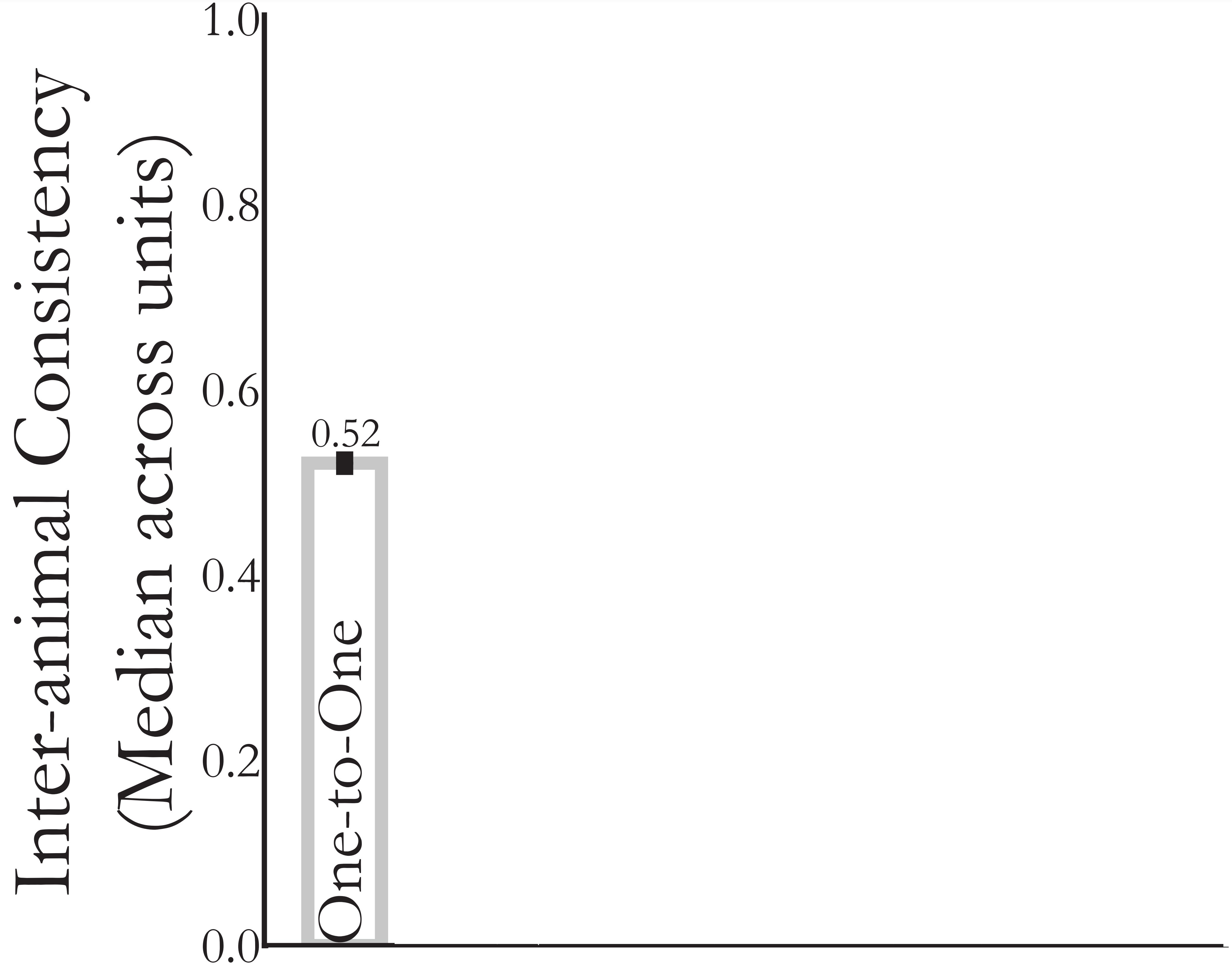
Spectrum of assumptions: "Full" Linear

Regularization constants enforce sparsity

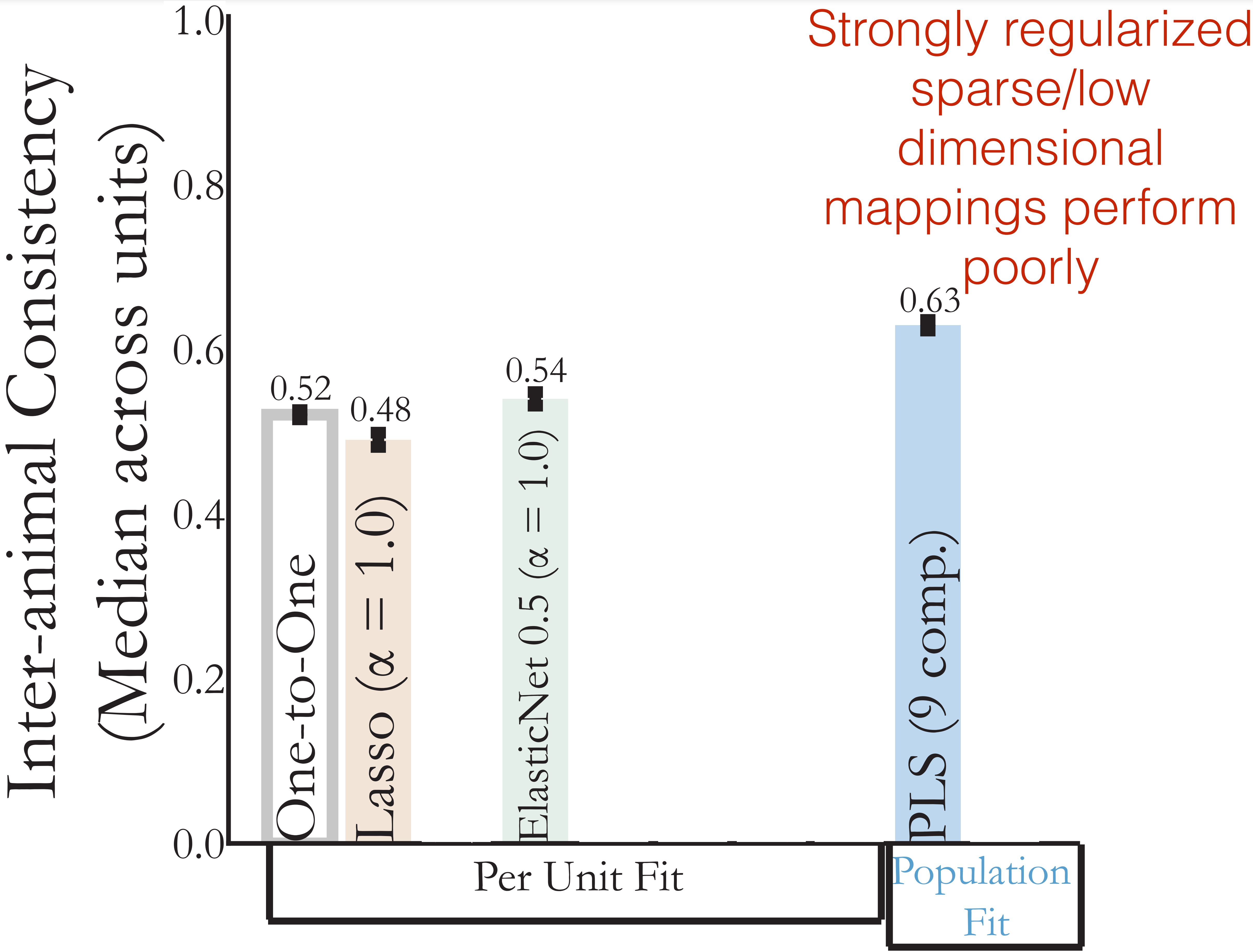
Most Sparse
Source MEC

Least Sparse
Target MEC

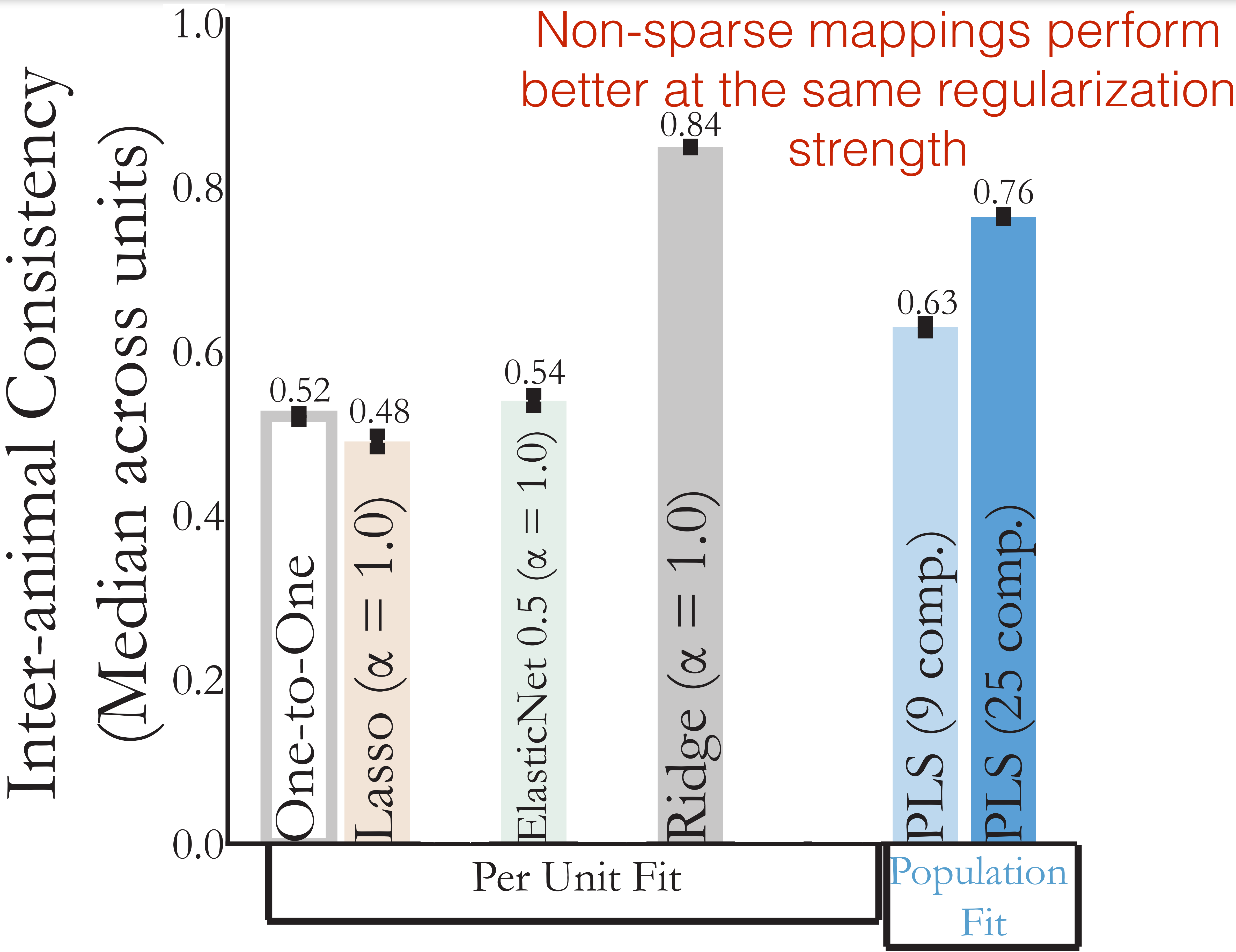
One-to-one is quite bad across animals



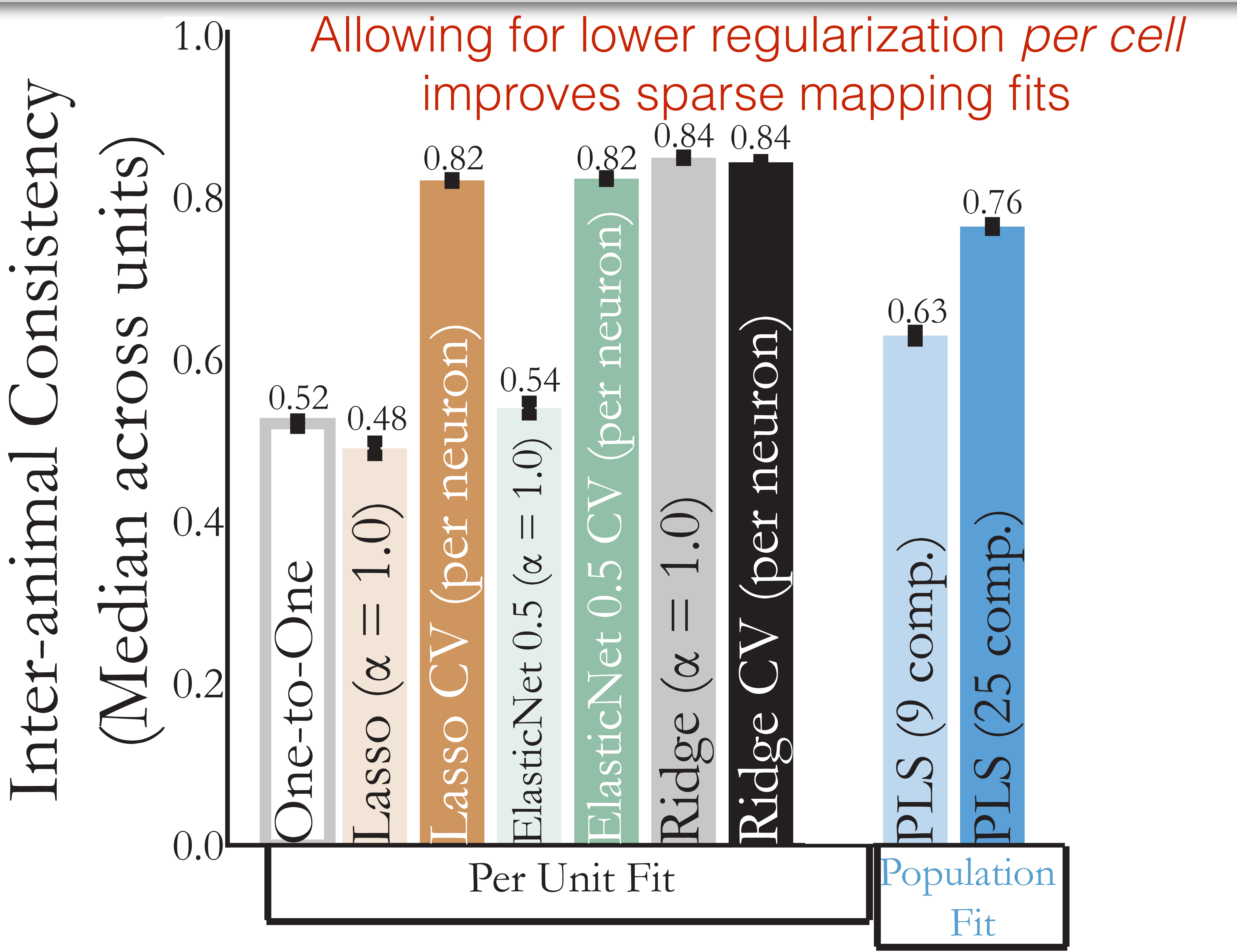
Sparse linear mappings are also quite bad across animals



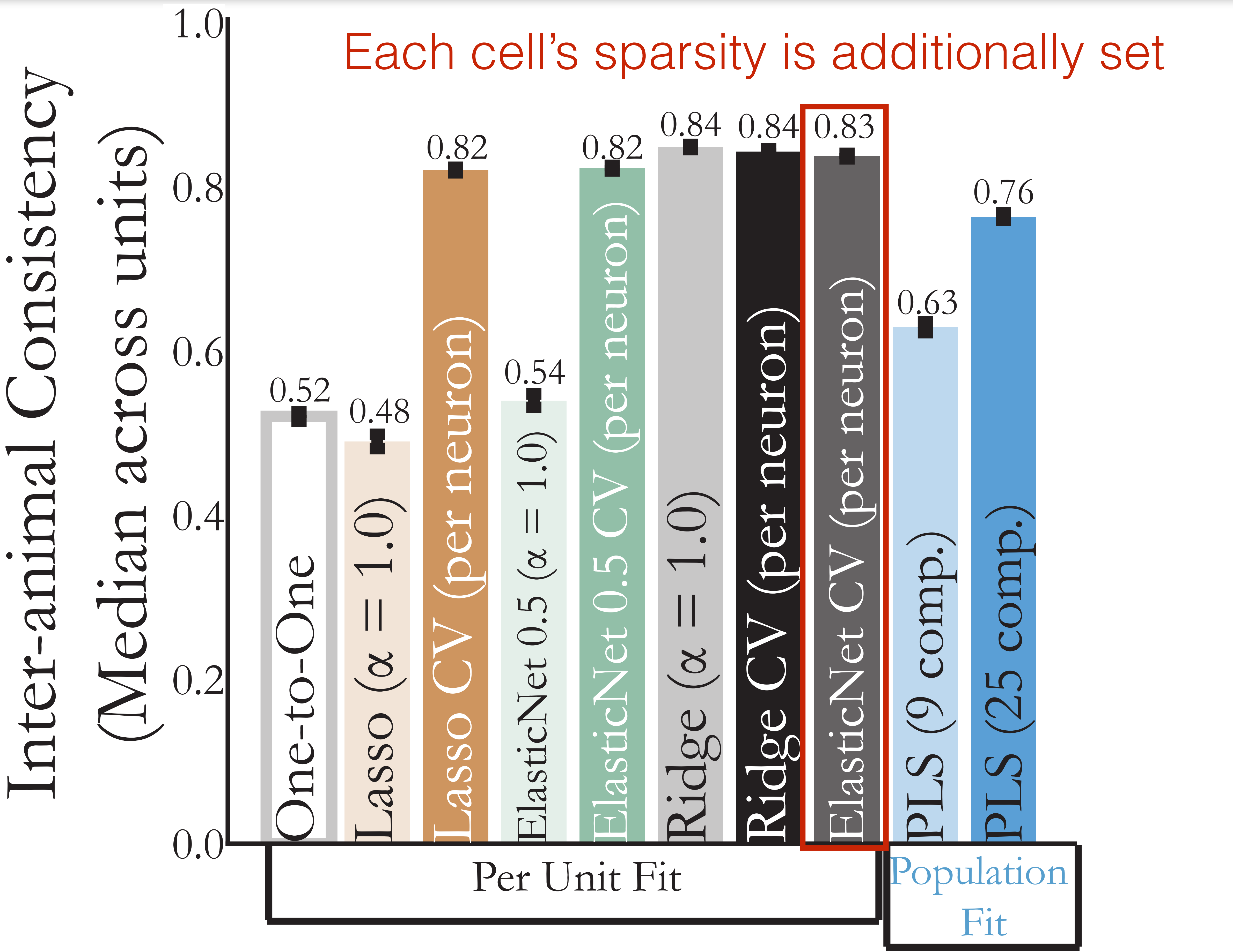
Sparse linear mappings are also quite bad across animals



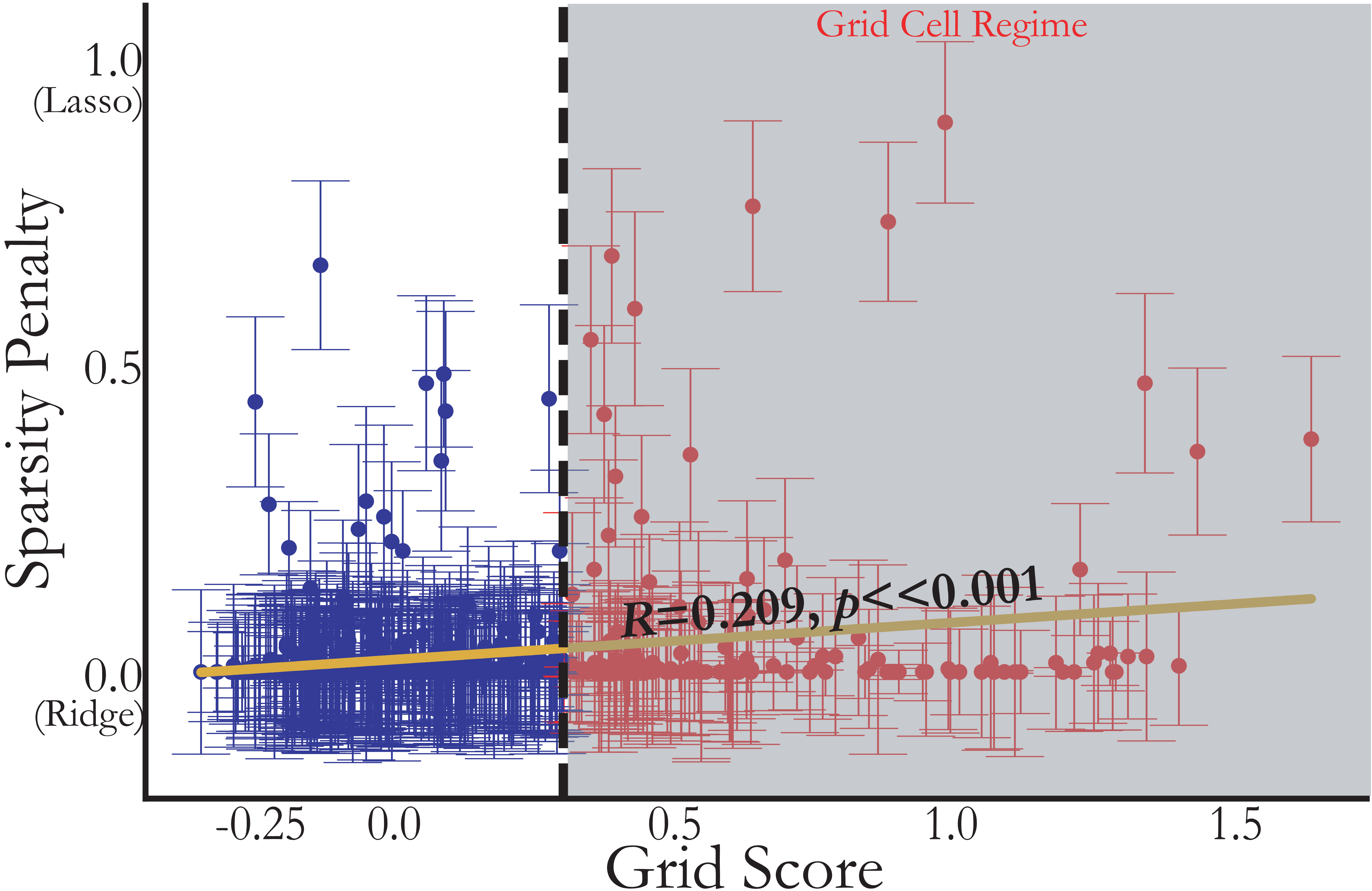
Full linear mappings work best across animals



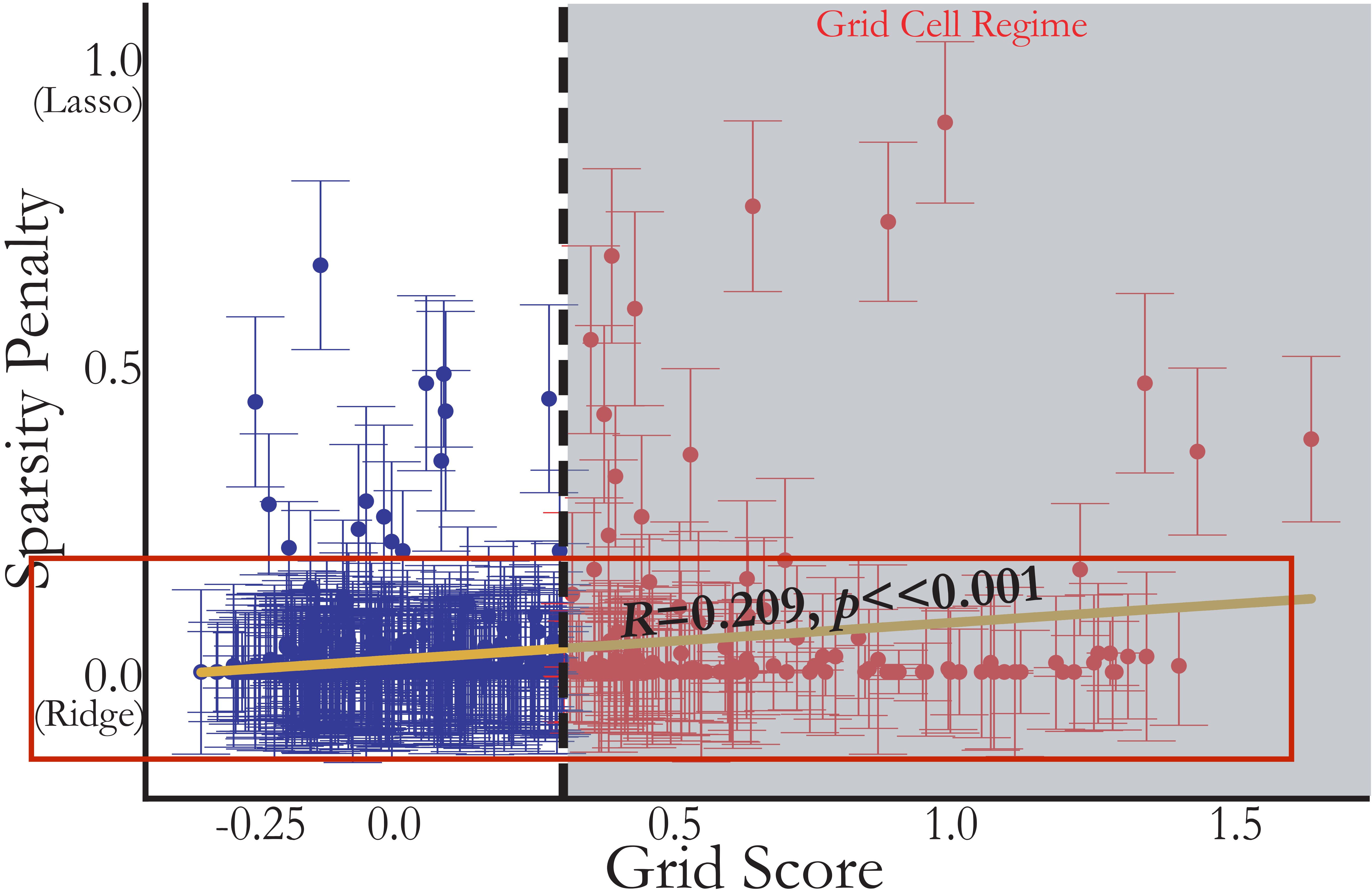
Set each cell's sparsity level



Most cells prefer ridge regression

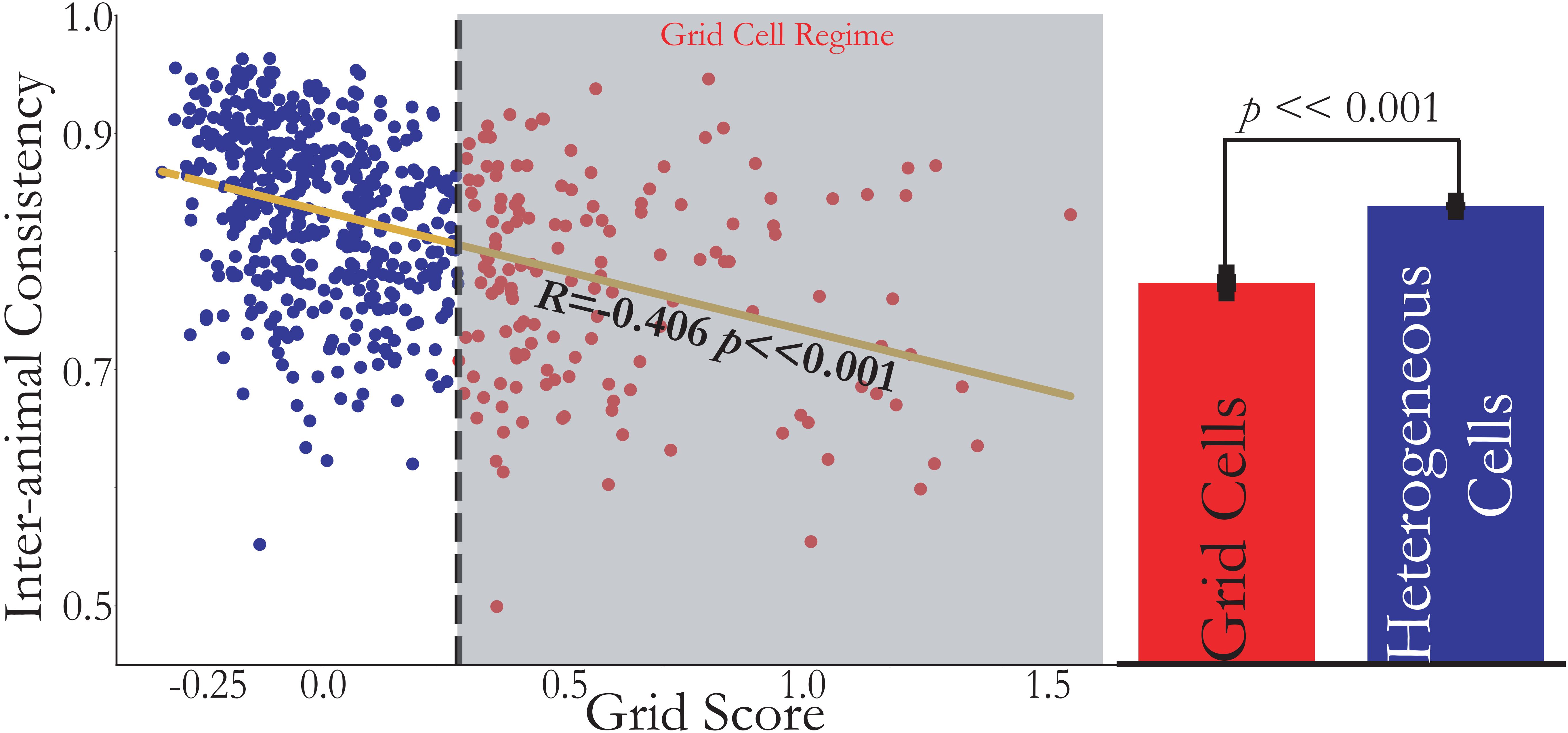


Most cells prefer ridge regression



Heterogeneous cells are reliable targets of explanation

Consistent reliability *across* all cells

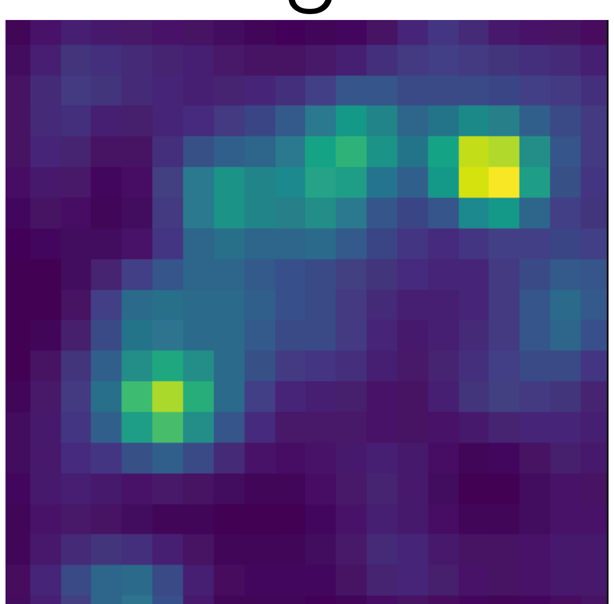


Heterogeneous cells are reliable targets of explanation

So far, we have shown how to measure similarity of the responses of the heterogeneous cells, and that these responses are reliable

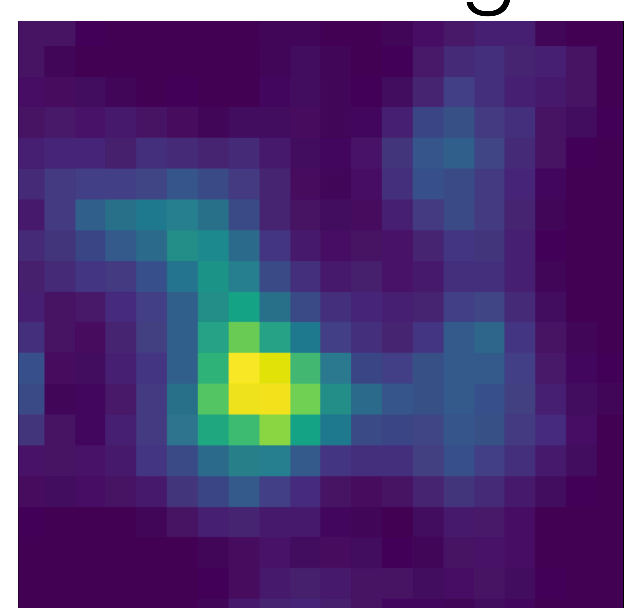
Now, we are going to describe what the constraints are that give rise to these reliable responses

MEC Heterogeneous Cell



?

Model Heterogeneous Cell



Goal-Driven Modeling - Primary Components

“behavior”

1.

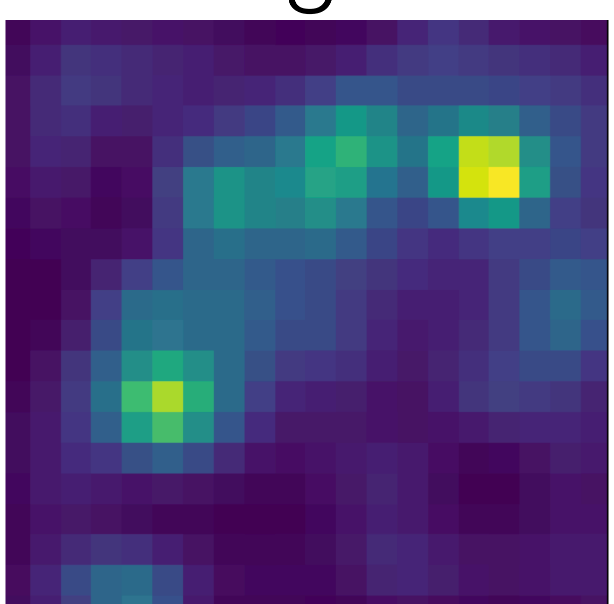
L = *loss function* ***D*** = *dataset*

“circuit”

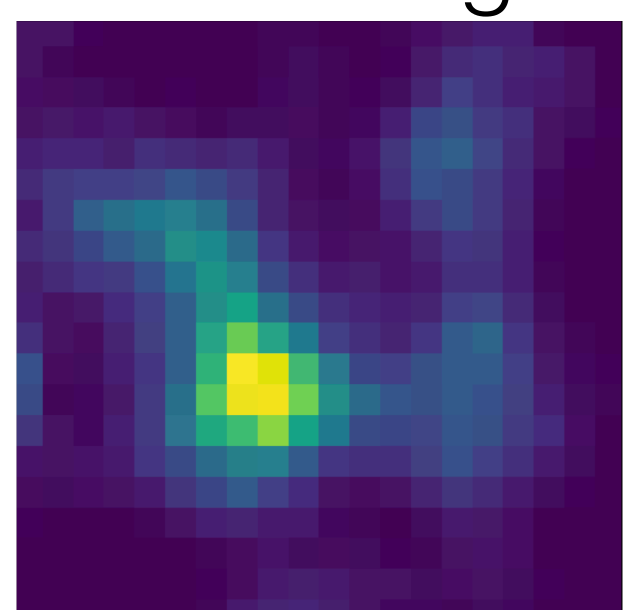
2.

A = *architecture class*

MEC Heterogeneous Cell



Model Heterogeneous Cell



Goal-Driven Modeling - Primary Components

"behavior"

1.

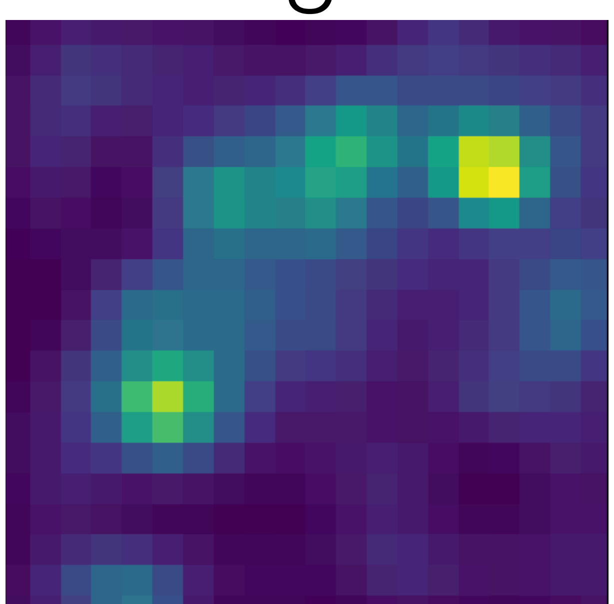
L *= loss function* ***D*** *= dataset*

"circuit"

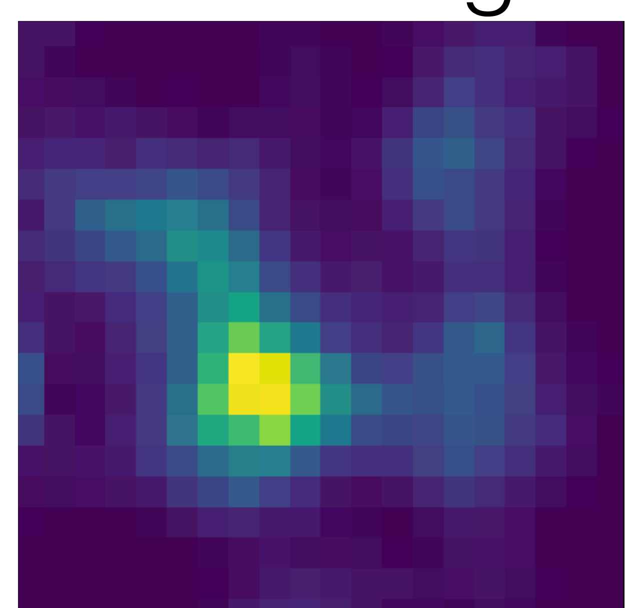
2.

A *= architecture class*

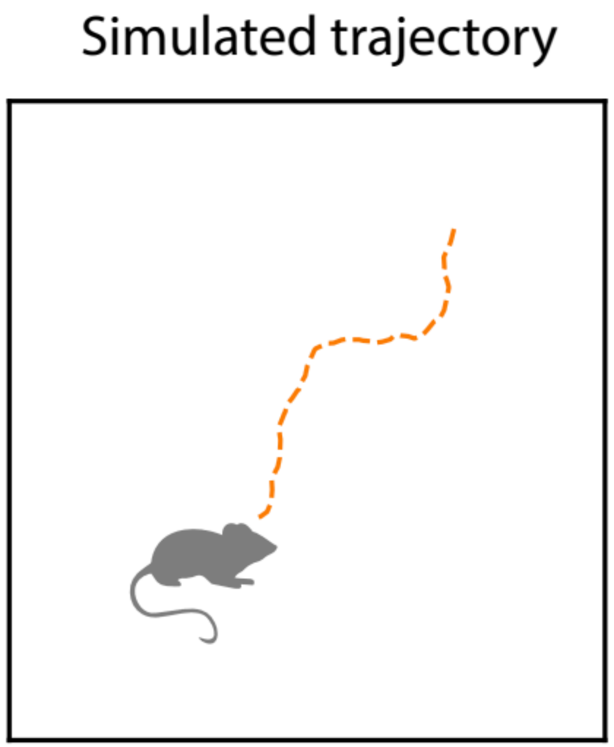
MEC Heterogeneous Cell

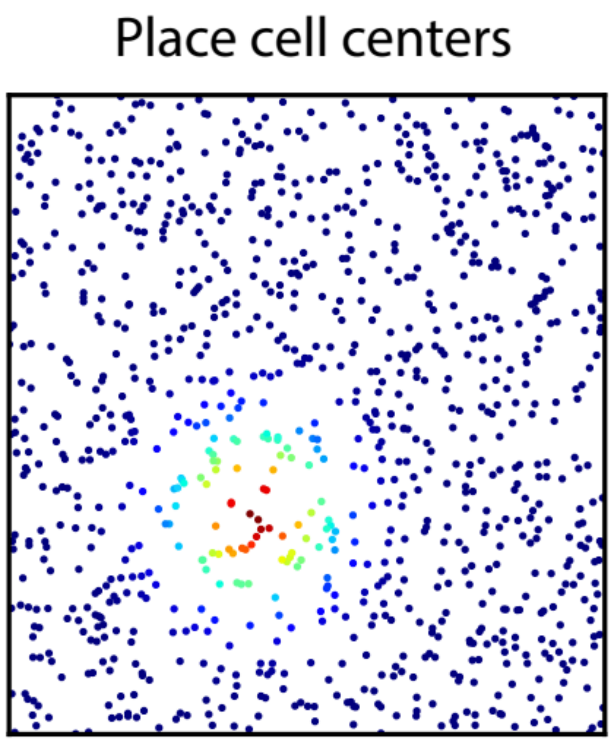


Model Heterogeneous Cell



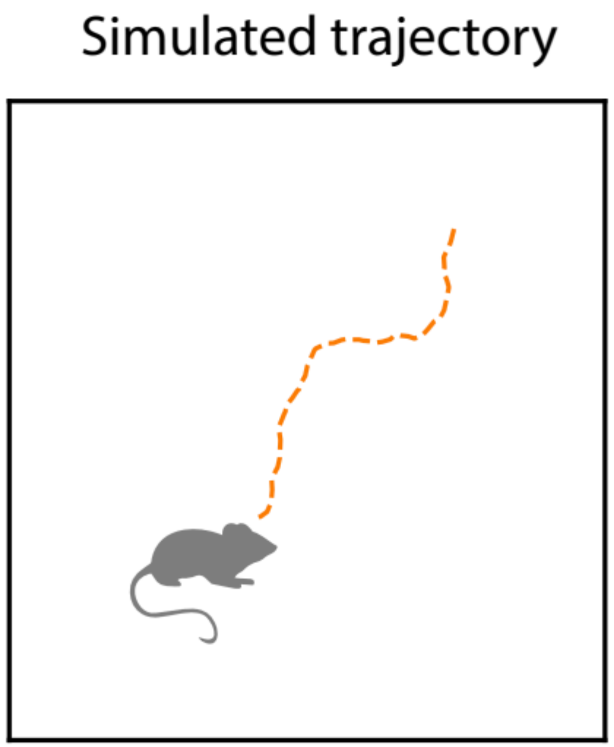
A spectrum of tasks

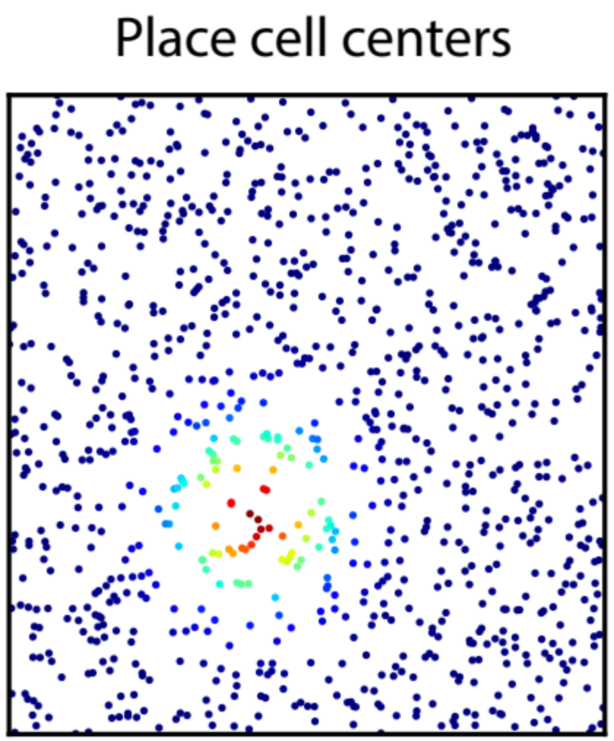




Sorscher, Mel*, Ganguli, Ocko*
***NeurIPS** (2019)*

A spectrum of tasks





Sorscher, Mel*, Ganguli, Ocko*
NeurIPS *(2019)*

Velocity → MEC → Place Cells → Position (x,y)

A spectrum of tasks

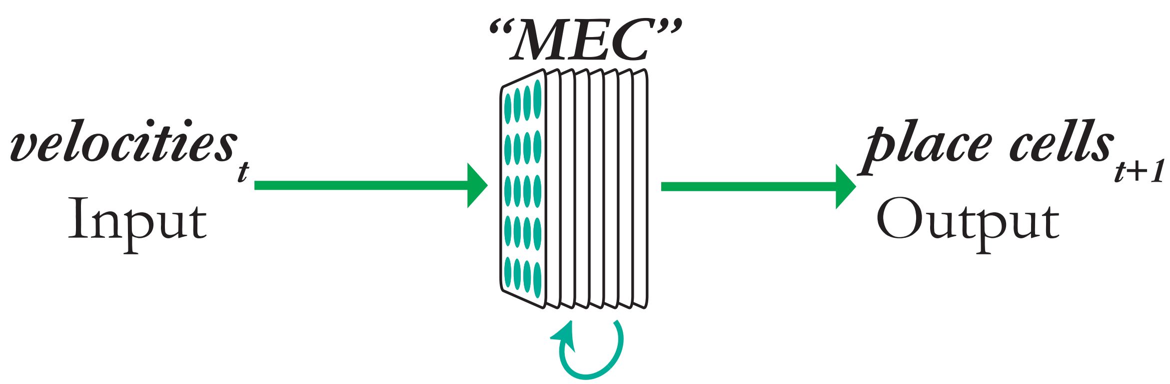
Simplest "model"

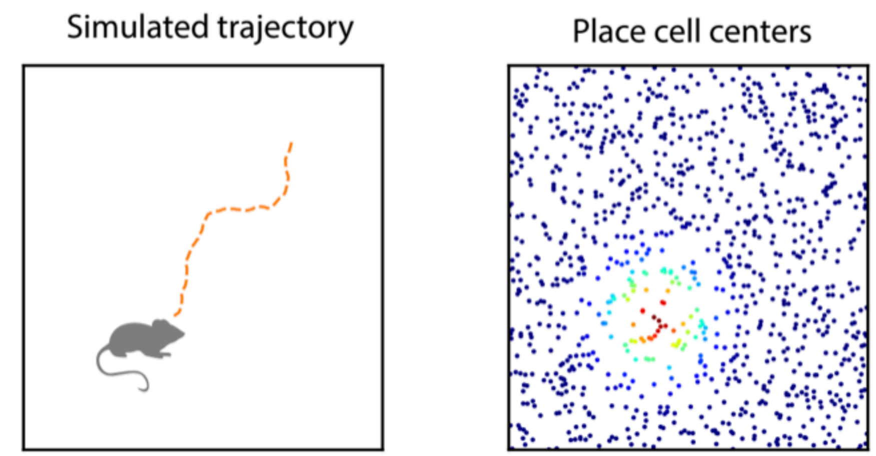
$\boxed{\text{Velocity}} \longrightarrow \text{MEC} \longrightarrow \text{Place Cells} \longrightarrow \text{Position } (x,y)$

A spectrum of tasks

$$\mathcal{L}(\hat{p}, p) := -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_p} p_i^t \log \hat{p}_i^t$$

Banino*, Barry*
et al. 2018



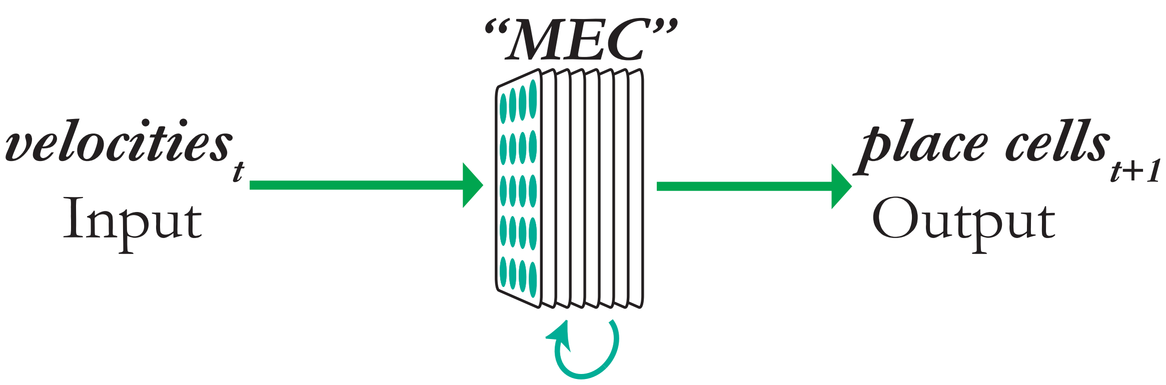


Velocity → MEC → Place Cells → Position (x,y)

A spectrum of tasks

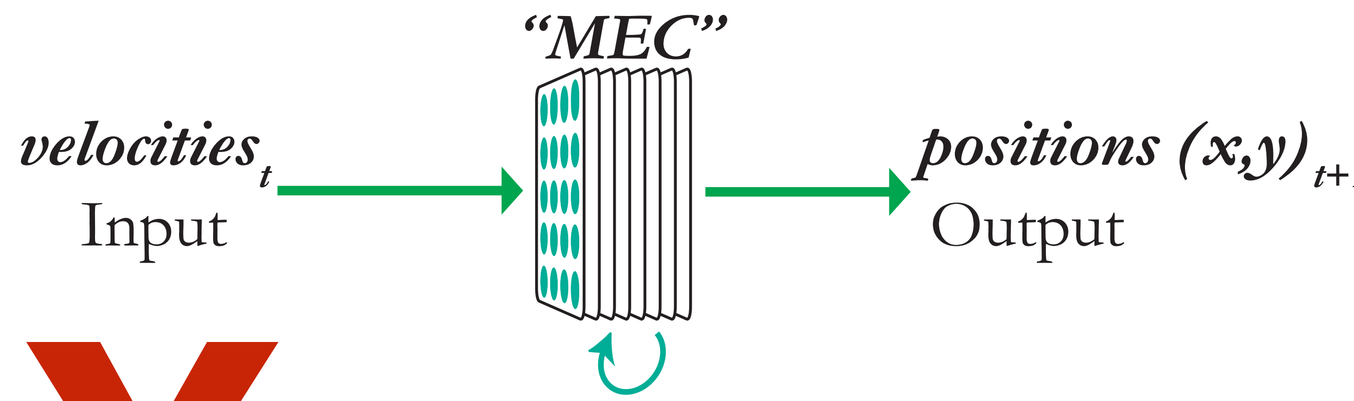
$$\mathcal{L}(\hat{p}, p) := -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_p} p_i^t \log \hat{p}_i^t$$

Banino*, Barry* *et al.* 2018



$$\mathcal{L}(\hat{p}, p) := \frac{1}{2}\frac{1}{T}\sum_{t=1}^{T}\left(\left(p_x^t - \hat{p}_x^t\right)^2 + \left(p_y^t - \hat{p}_y^t\right)^2\right)$$

Cueva* & Wei* 2018

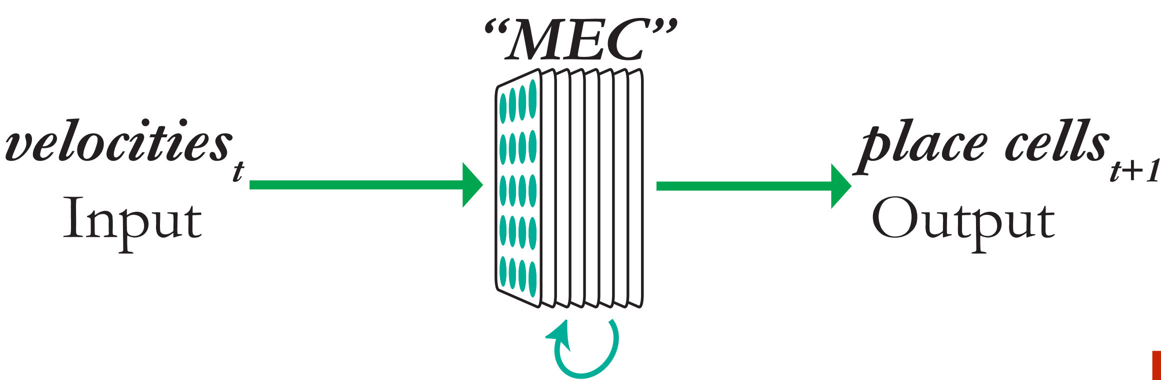


Velocity $\longrightarrow$ MEC $\longrightarrow$ Place Cells $\longrightarrow$ Position (x,y)

A spectrum of tasks

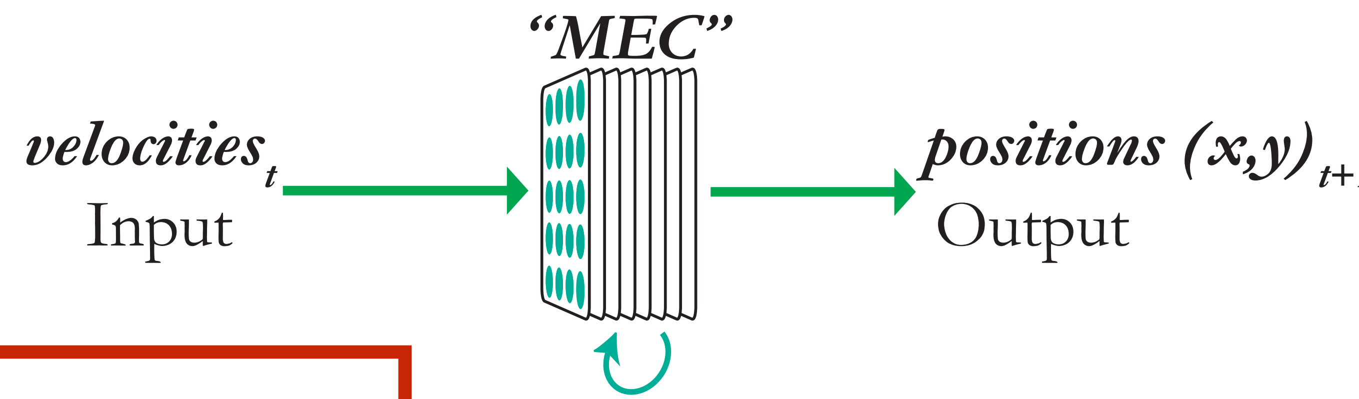
$$\mathcal{L}(\hat{p}, p) := -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_p} p_i^t \log \hat{p}_i^t$$

Banino*, Barry* *et al.* 2018



$$\mathcal{L}(\hat{p}, p) := \frac{1}{2}\frac{1}{T}\sum_{t=1}^{T}\left(\left(p_x^t - \hat{p}_x^t\right)^2 + \left(p_y^t - \hat{p}_y^t\right)^2\right)$$

Cueva* & Wei* 2018



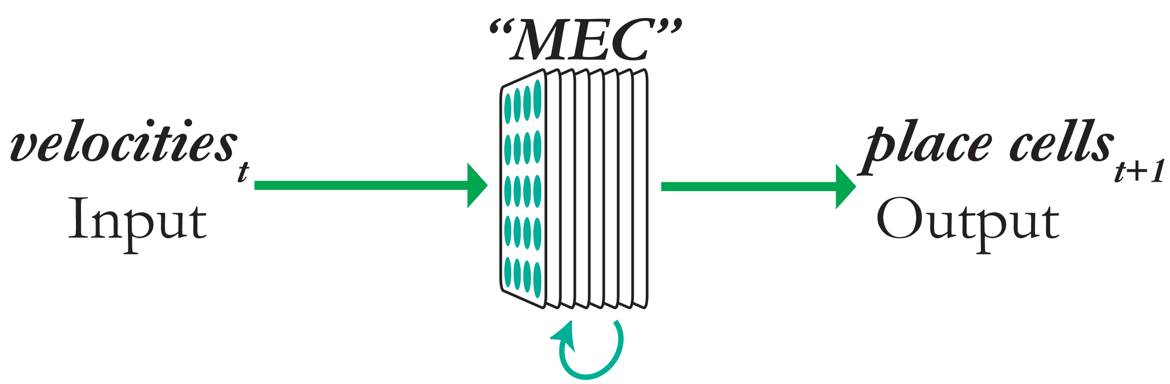
Velocity → MEC → $\boxed{\text{Place Cells}}$ → Position (x,y)

Output-based models

A spectrum of tasks

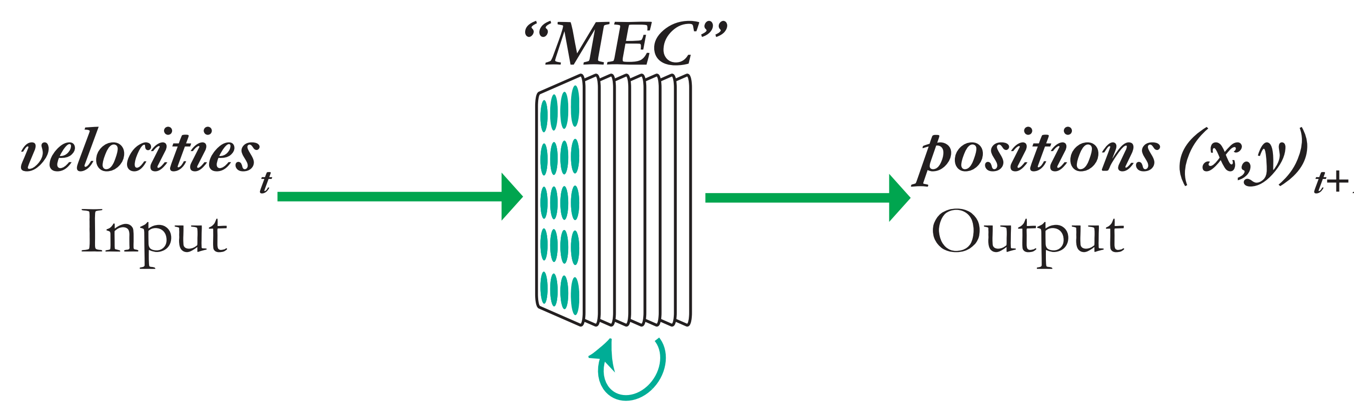
$$\mathcal{L}(\hat{p}, p) := -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_p} p_i^t \log \hat{p}_i^t$$

Banino*, Barry*
et al. 2018



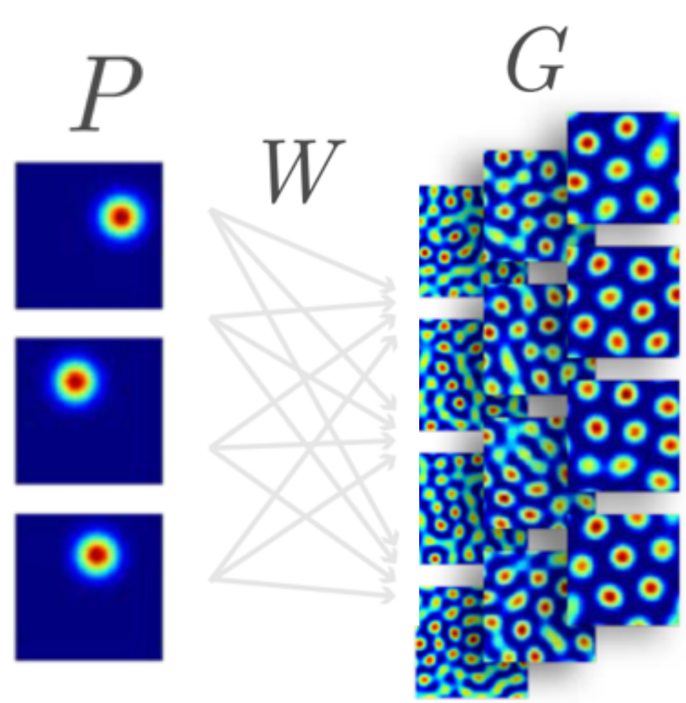
$$\mathcal{L}(\hat{p}, p) := \frac{1}{2}\frac{1}{T}\sum_{t=1}^{T}\left(\left(p_x^t - \hat{p}_x^t\right)^2 + \left(p_y^t - \hat{p}_y^t\right)^2\right)$$

Cueva* &
Wei* 2018



Velocity → MEC → Place Cells → Position (x,y)

NMF
(Place Cell Input)



Goal-Driven Modeling - Primary Components

"behavior"

1.

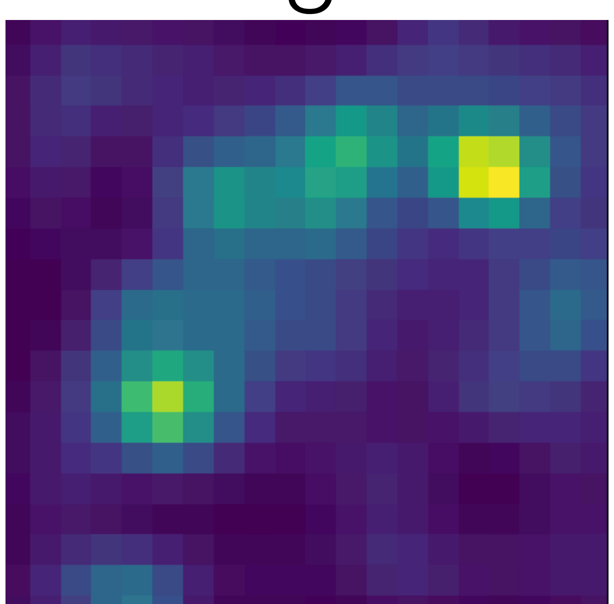
L = *loss function* ***D*** = *dataset*

"circuit"

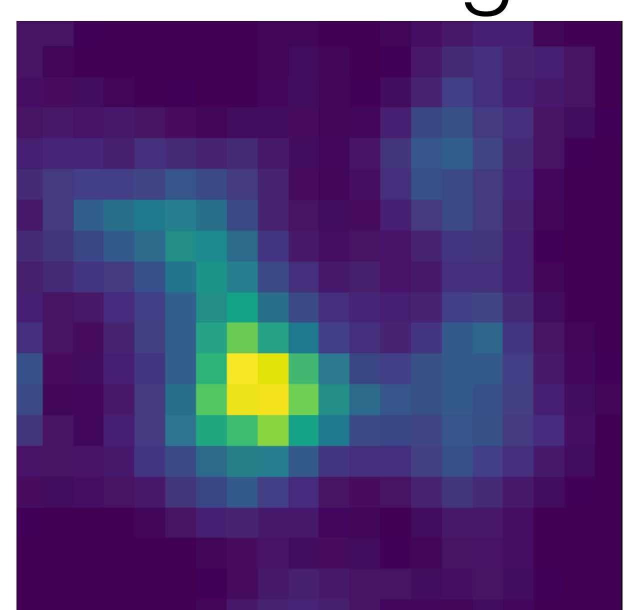
2.

A = *architecture class*

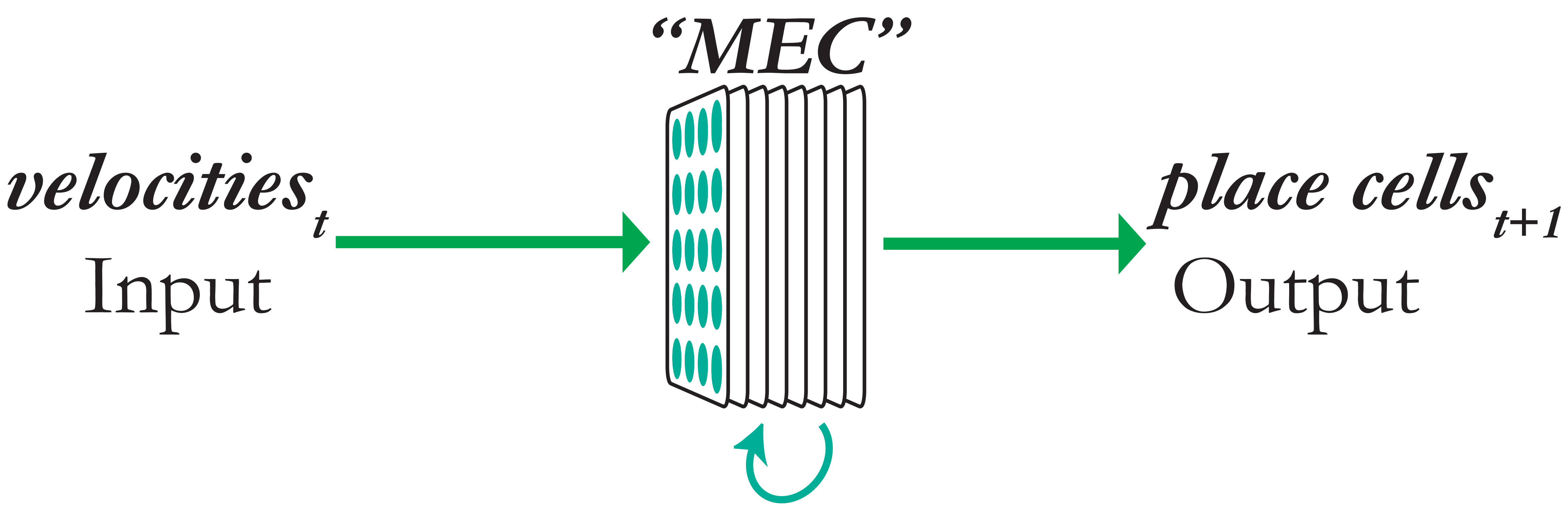
MEC Heterogeneous Cell



Model Heterogeneous Cell

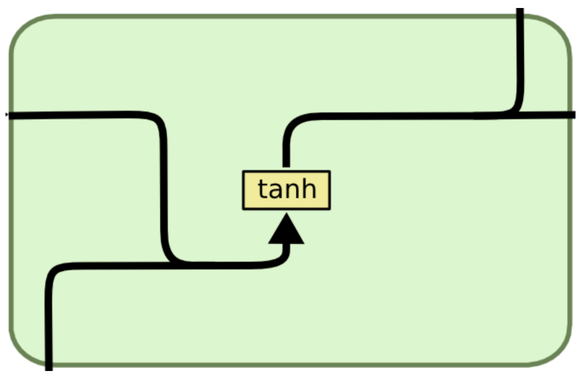


A spectrum of circuits

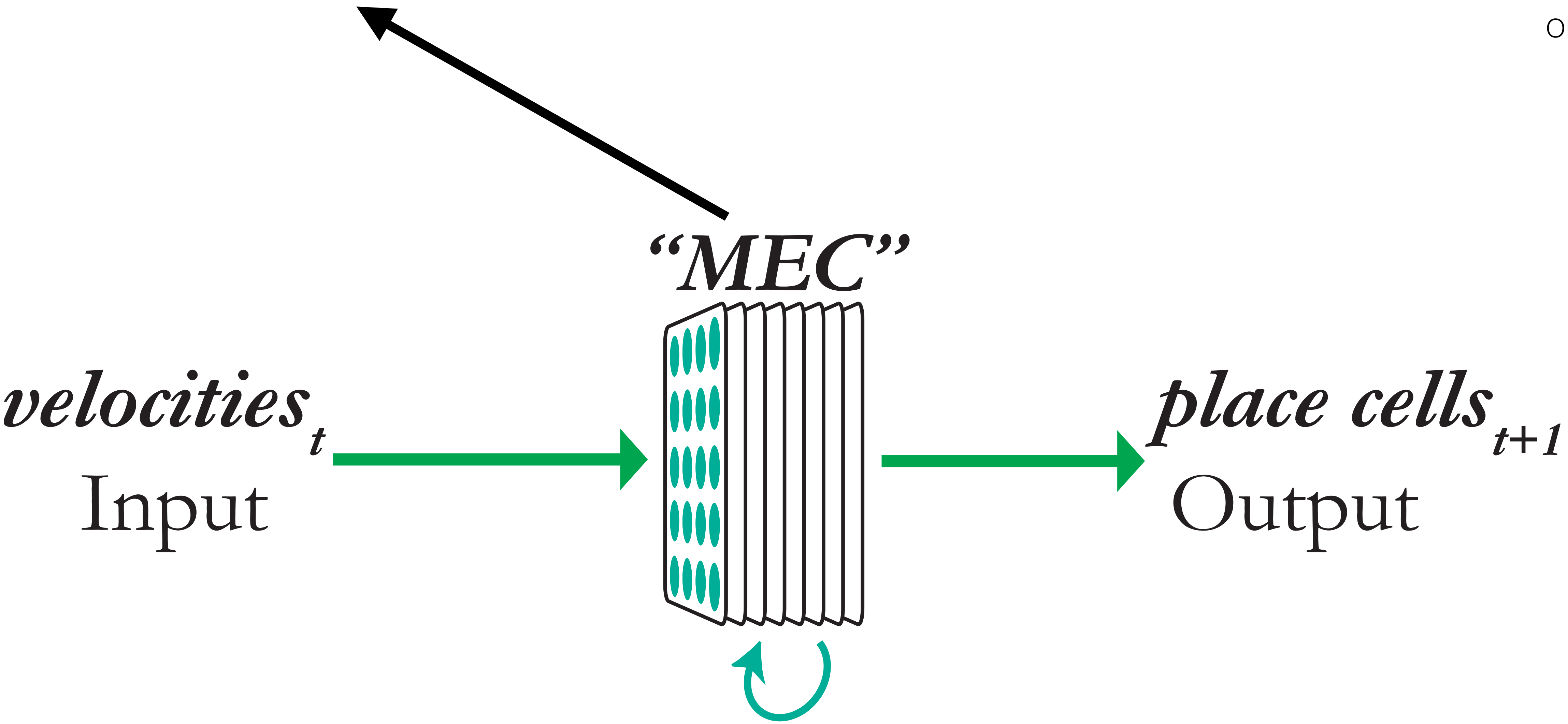


A spectrum of circuits

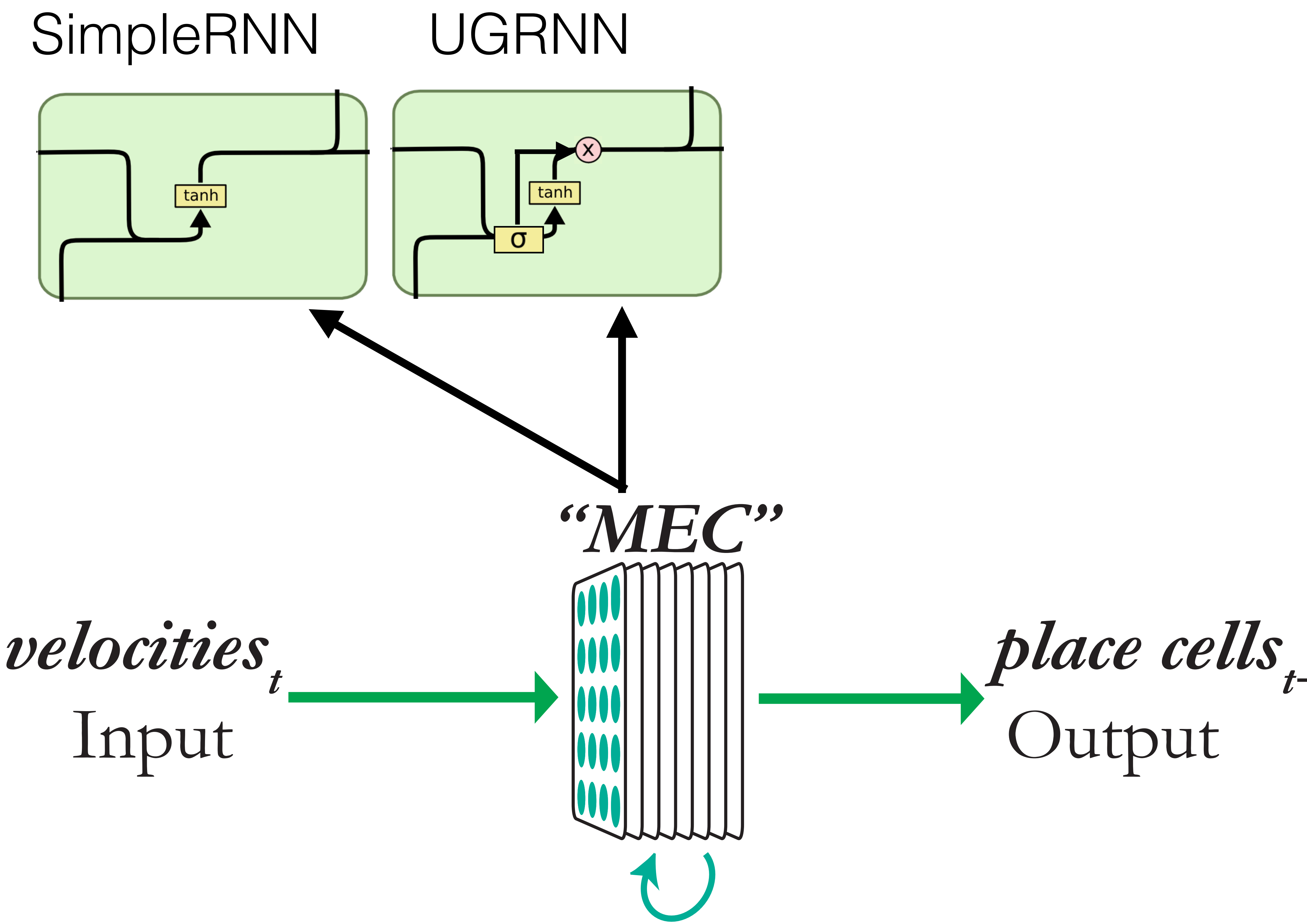
SimpleRNN



Olah 2015

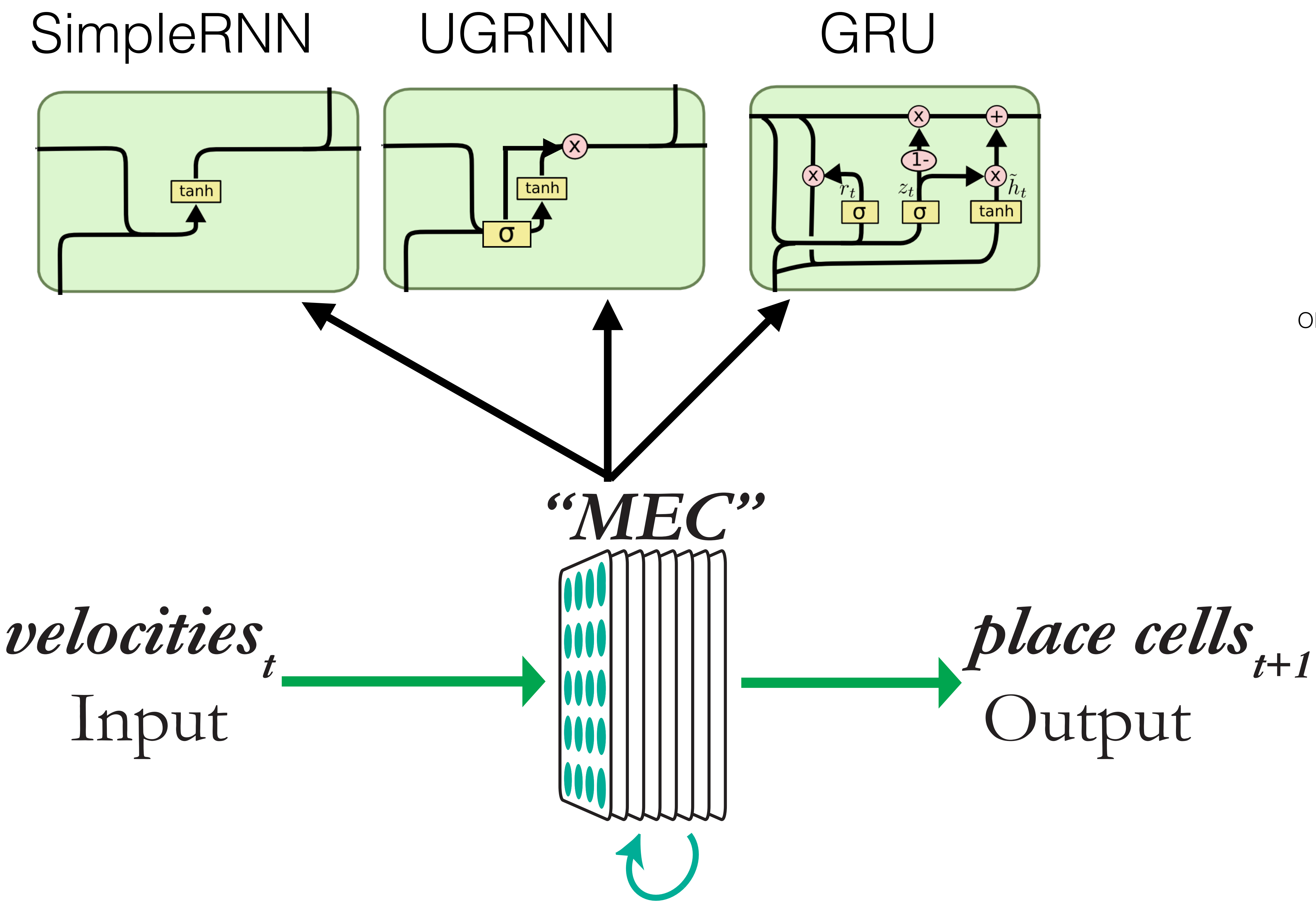


A spectrum of circuits — learnable modulation (“gating”)



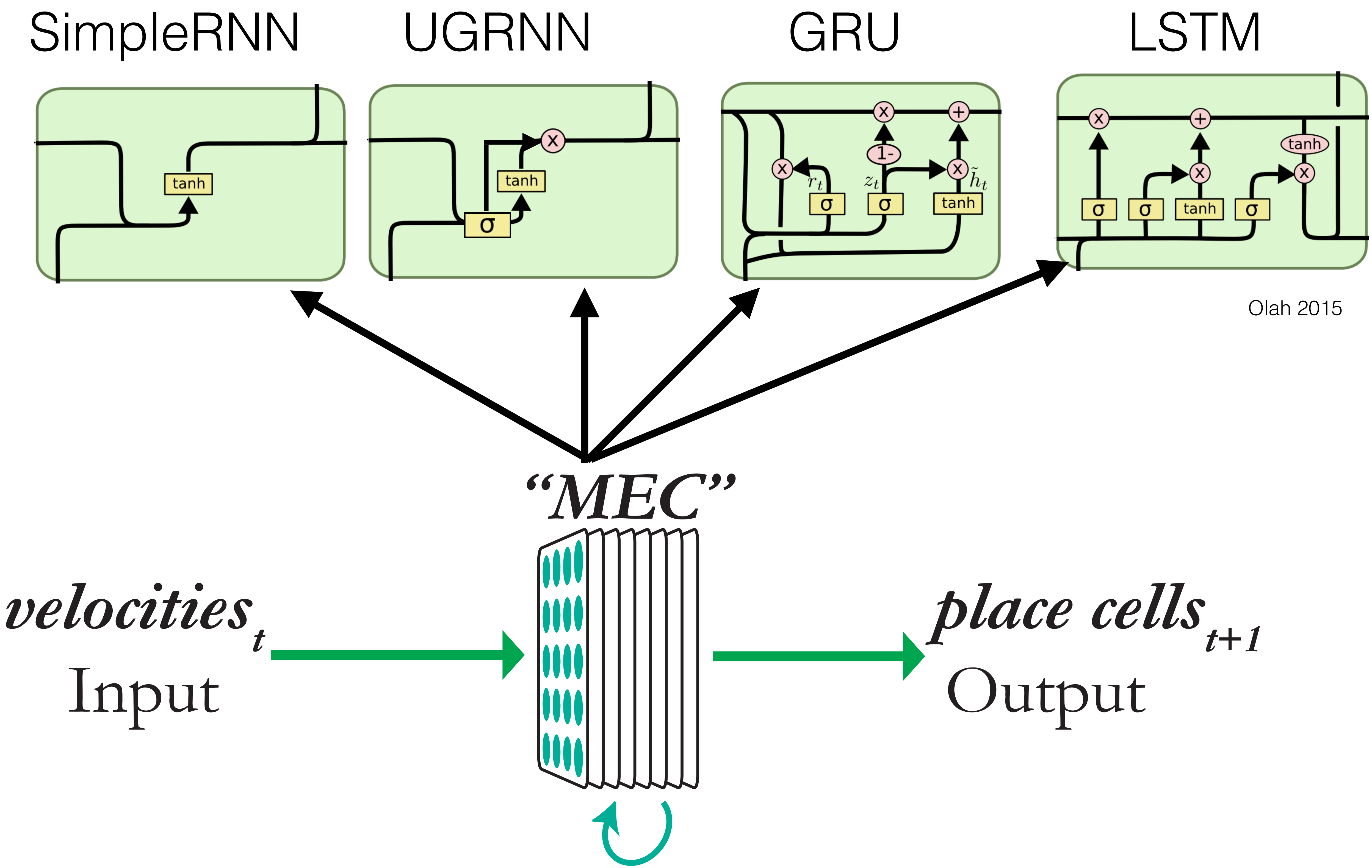
Olah 2015

A spectrum of circuits — learnable modulation (“gating”)

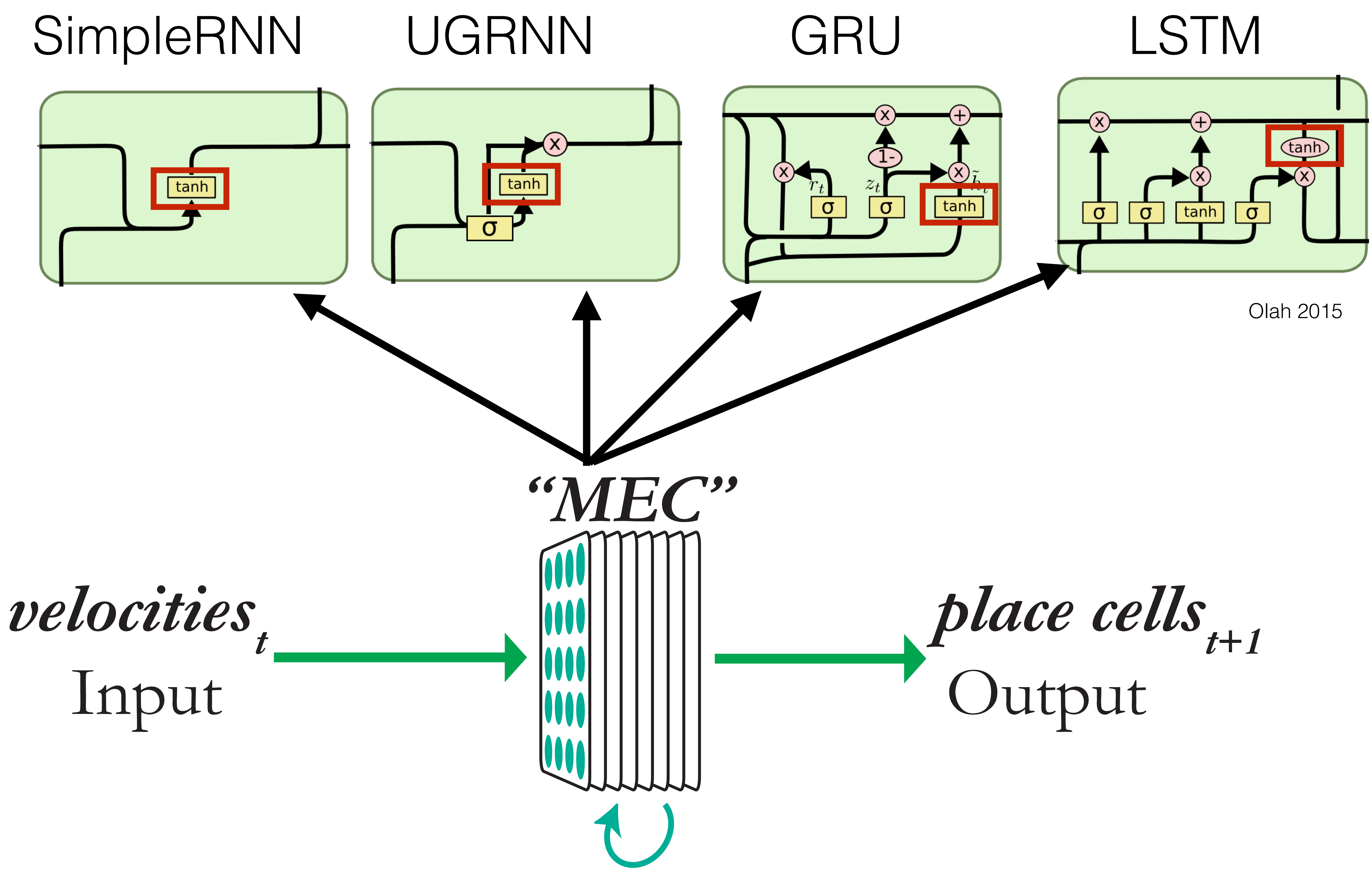


Olah 2015

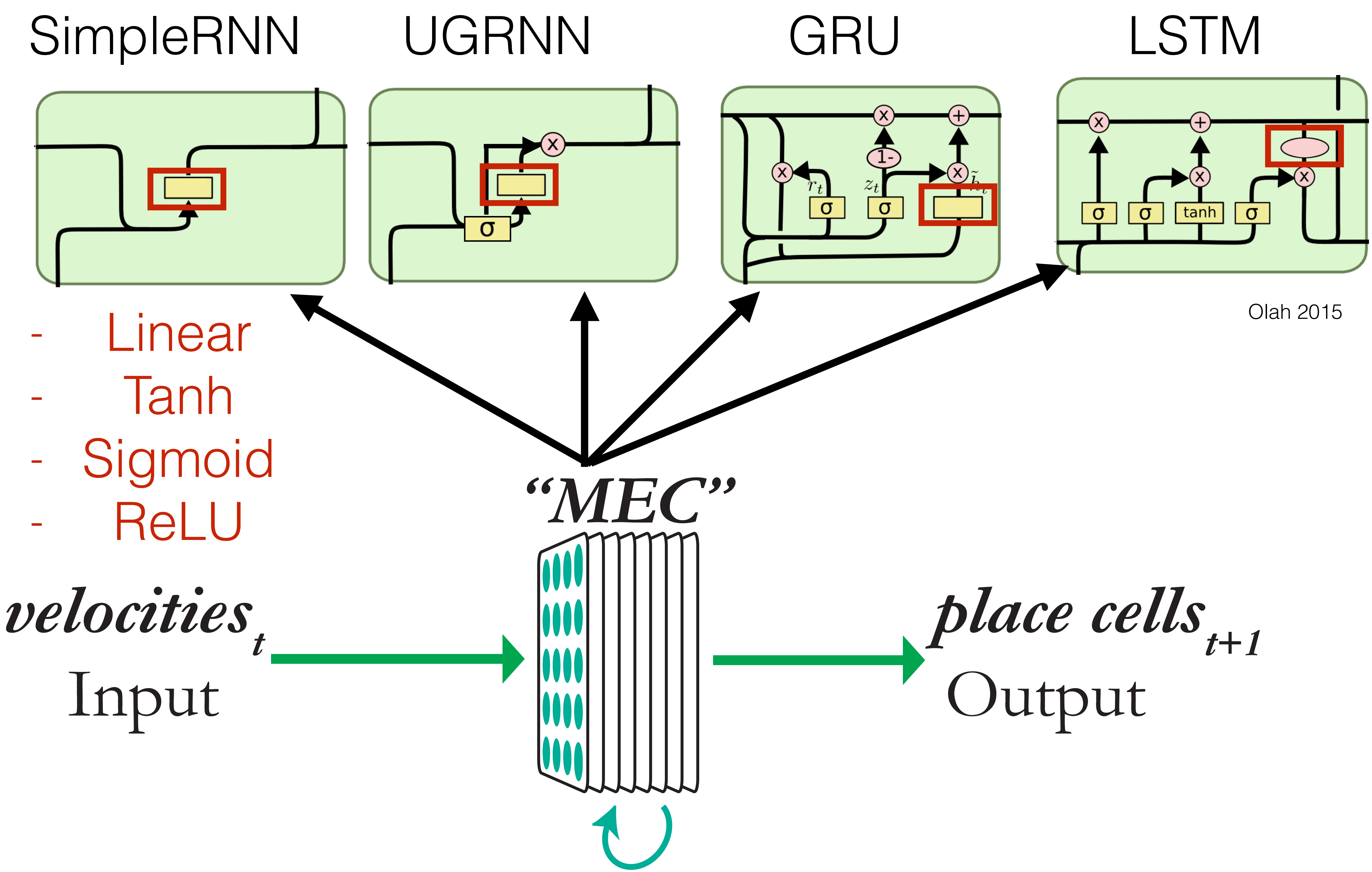
A spectrum of circuits — learnable modulation (“gating”)



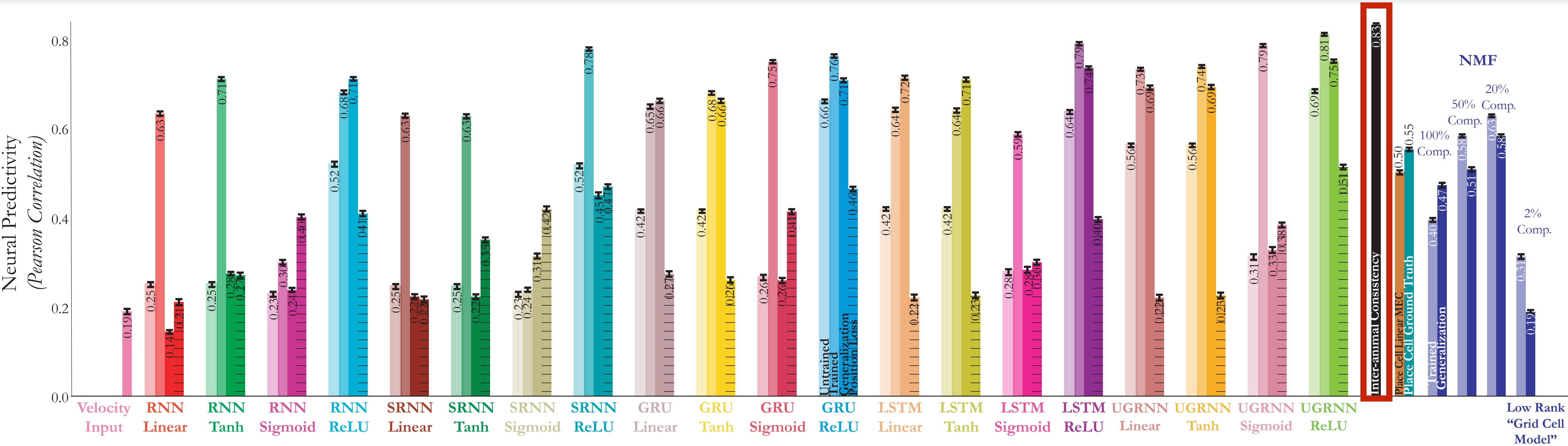
A spectrum of circuits — output nonlinearity



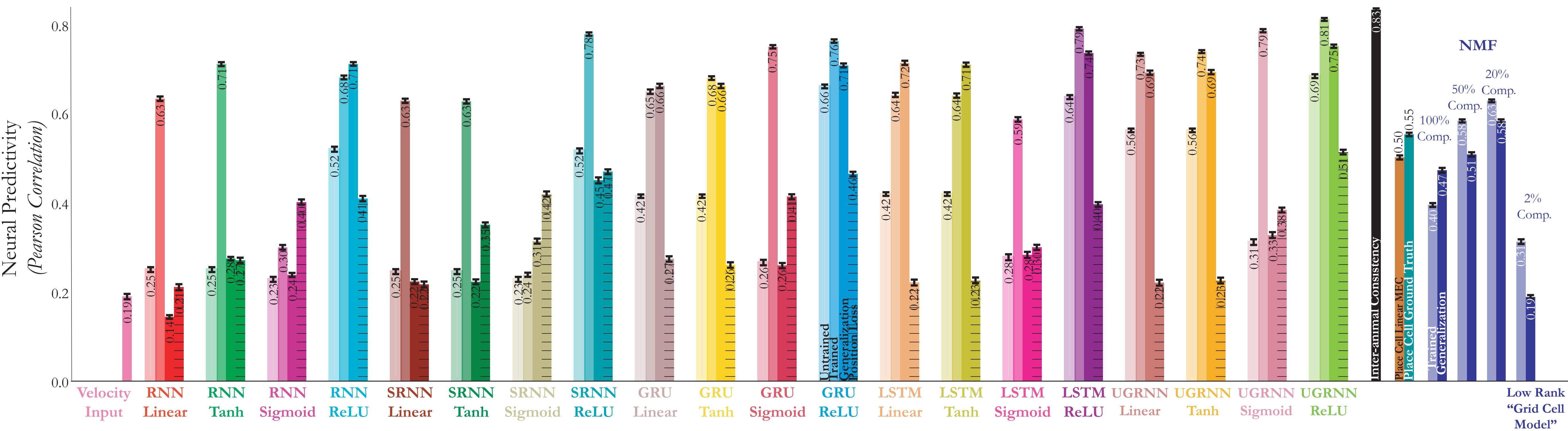
A spectrum of circuits — output nonlinearity



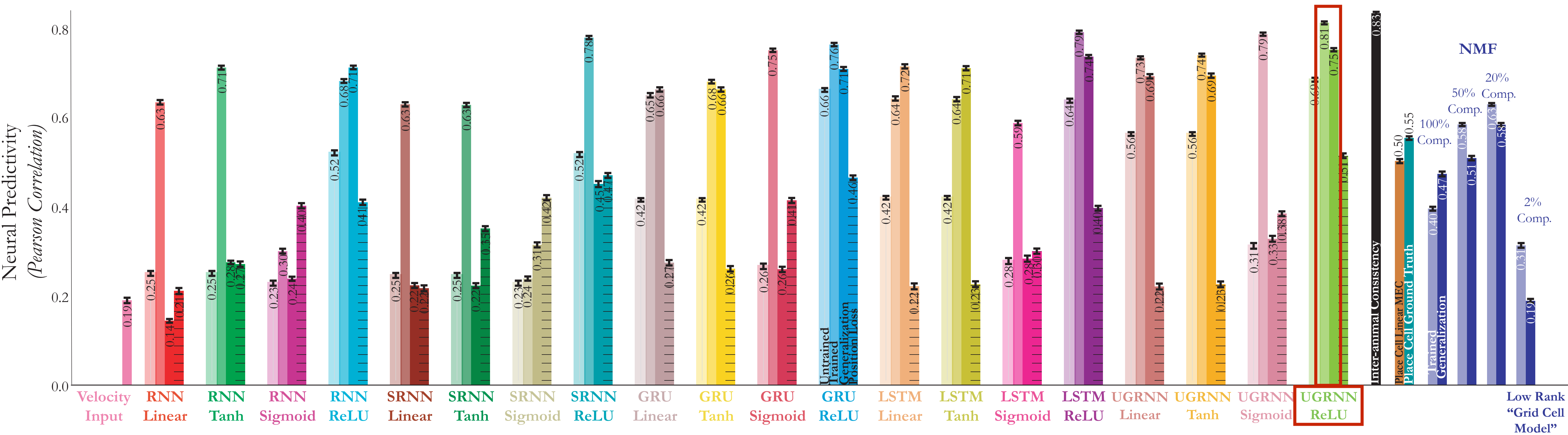
Benchmarking models with the same transform as between animals



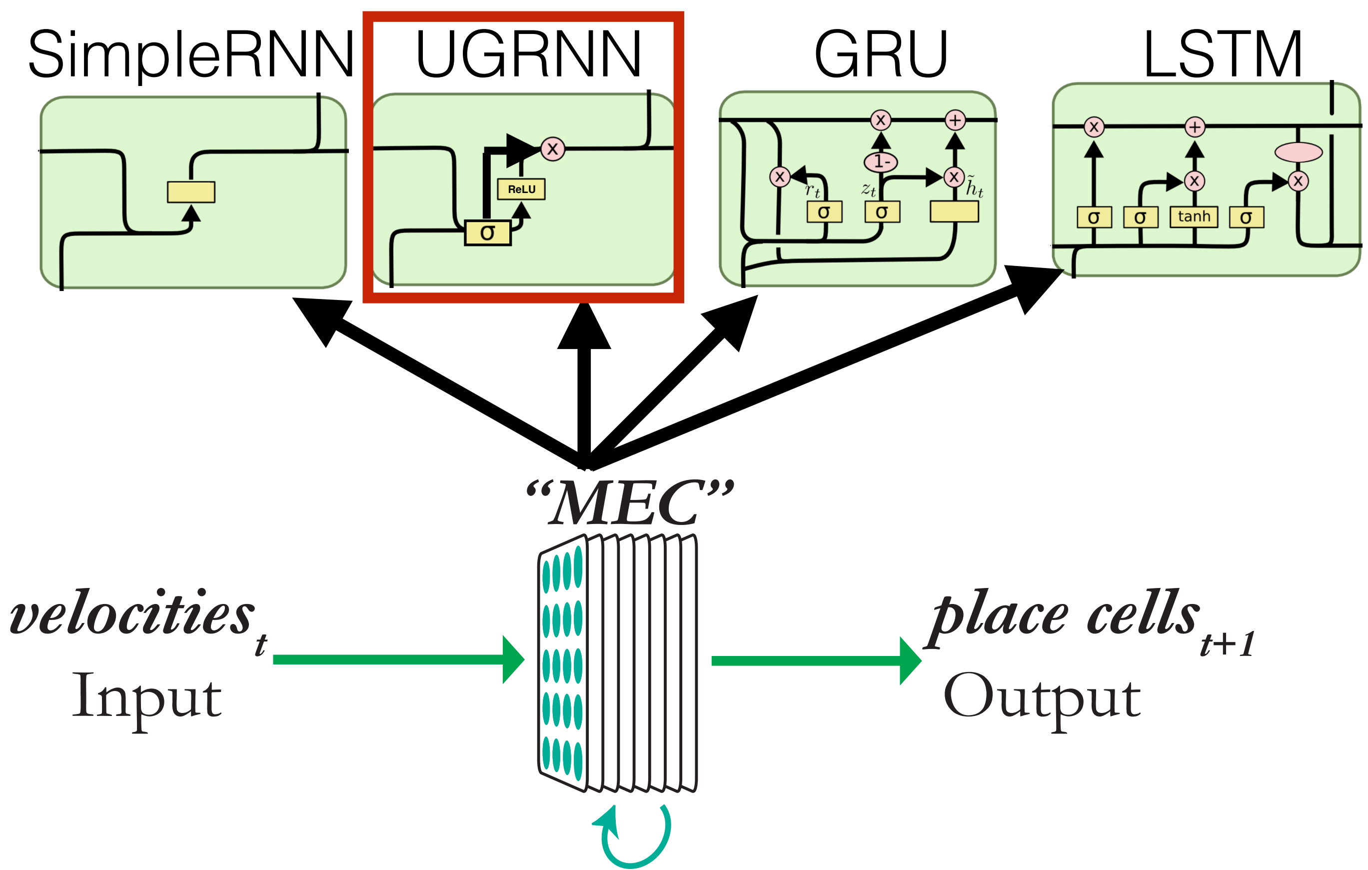
Task-optimized navigational models best predict the entire MEC population



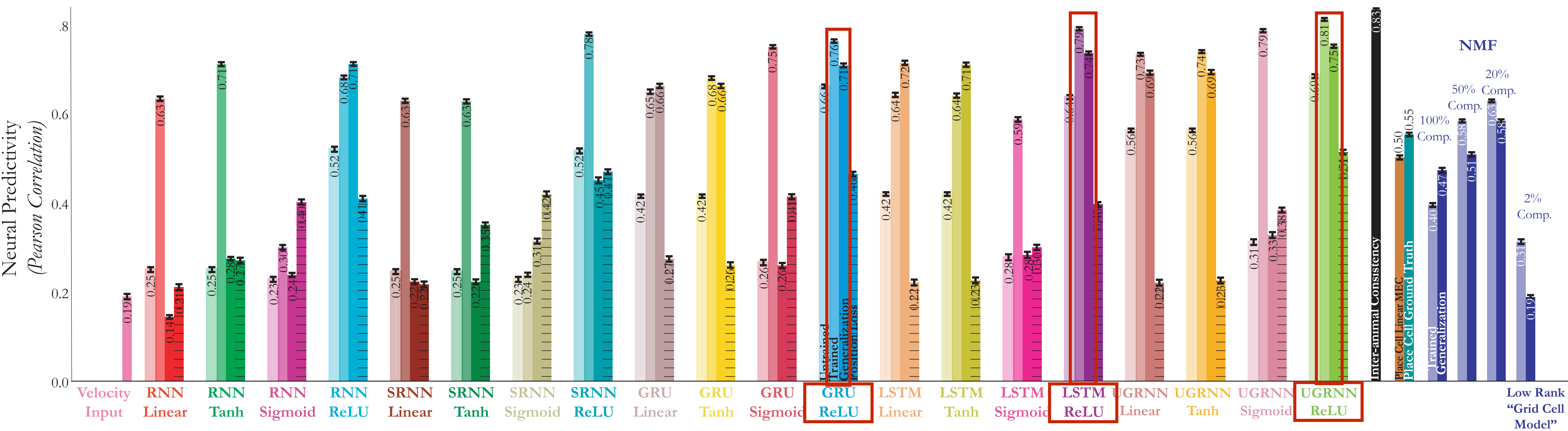
Task-optimized navigational models best predict the entire MEC population



Best task-optimized models "solve" the neurons

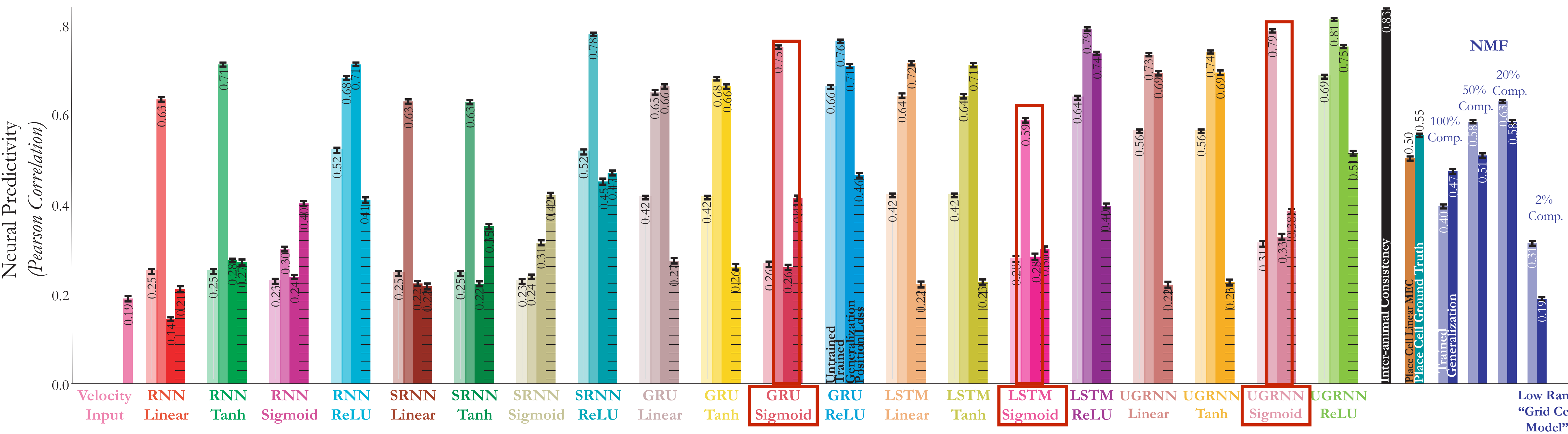


Nonlinearity affects generalization



Nonnegativity constraint + gating aids in generalization across environments

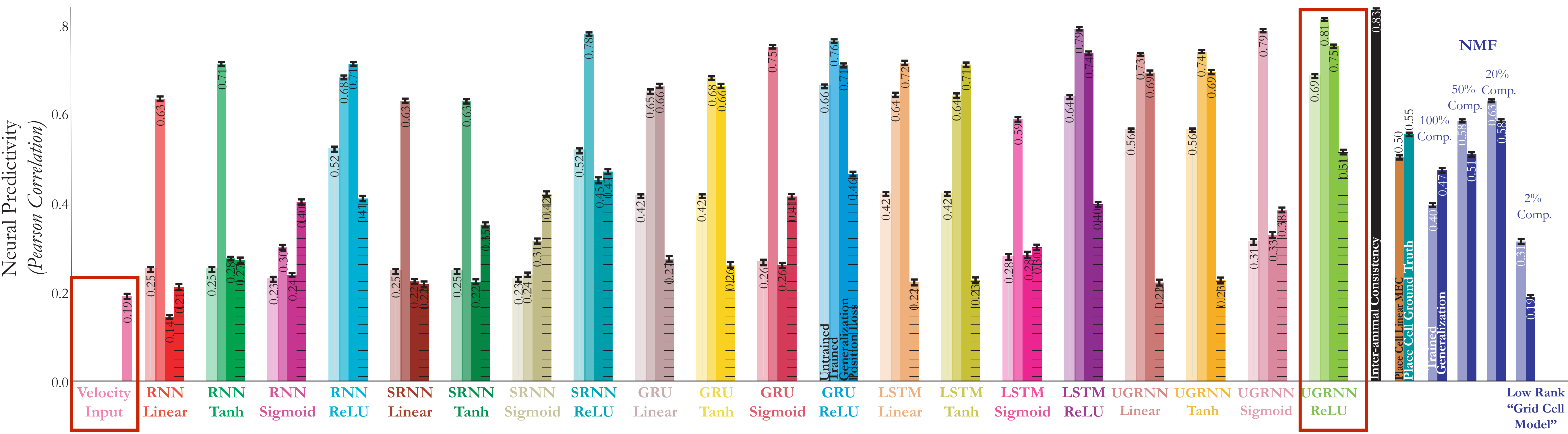
Nonlinearity affects generalization



Nonnegativity constraint + gating aids in generalization across environments

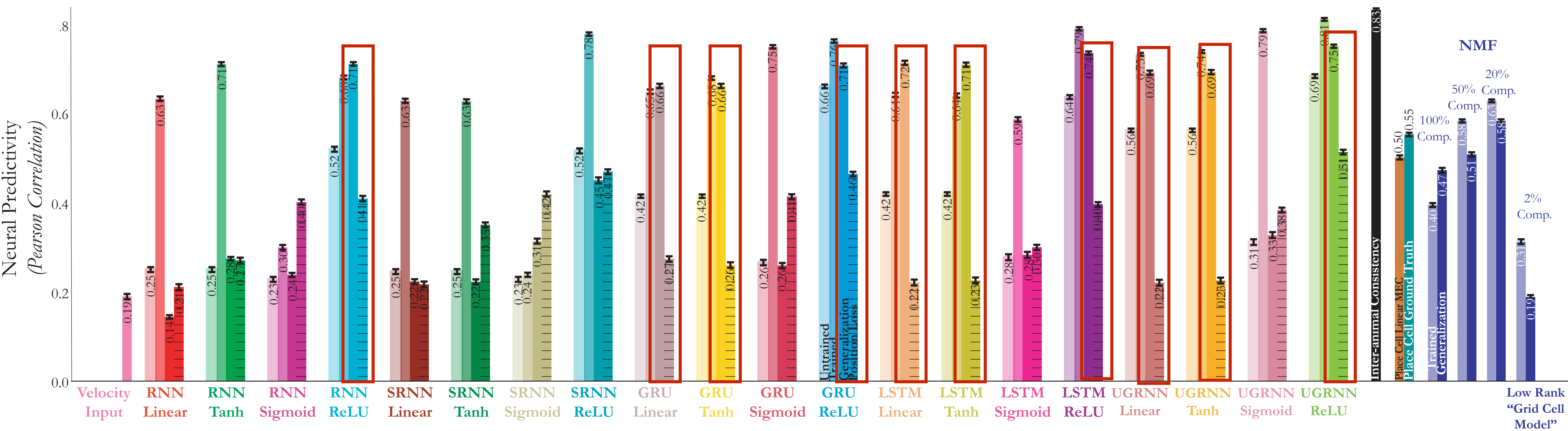
But this nonnegativity constraint must *not* saturate either!

Model input is a poor predictor of population



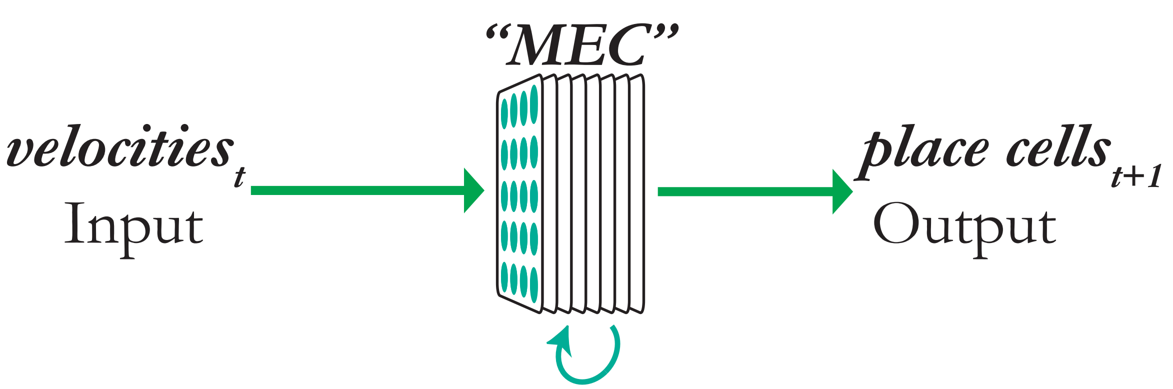
Models add a lot of predictive power to their inputs

Directly supervising on Cartesian coordinates fails to generalize

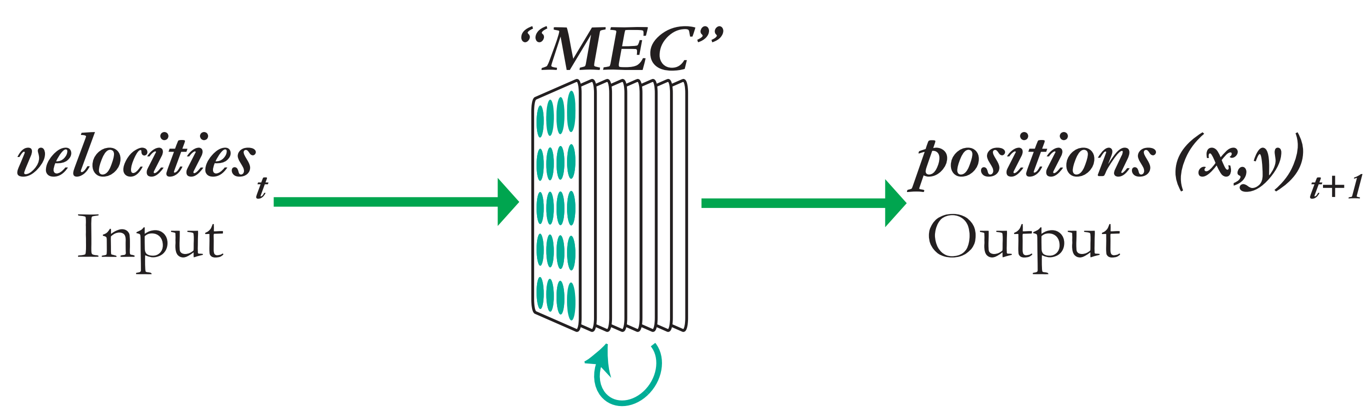


Output place cell supervision provides better generalization over direct supervision of position

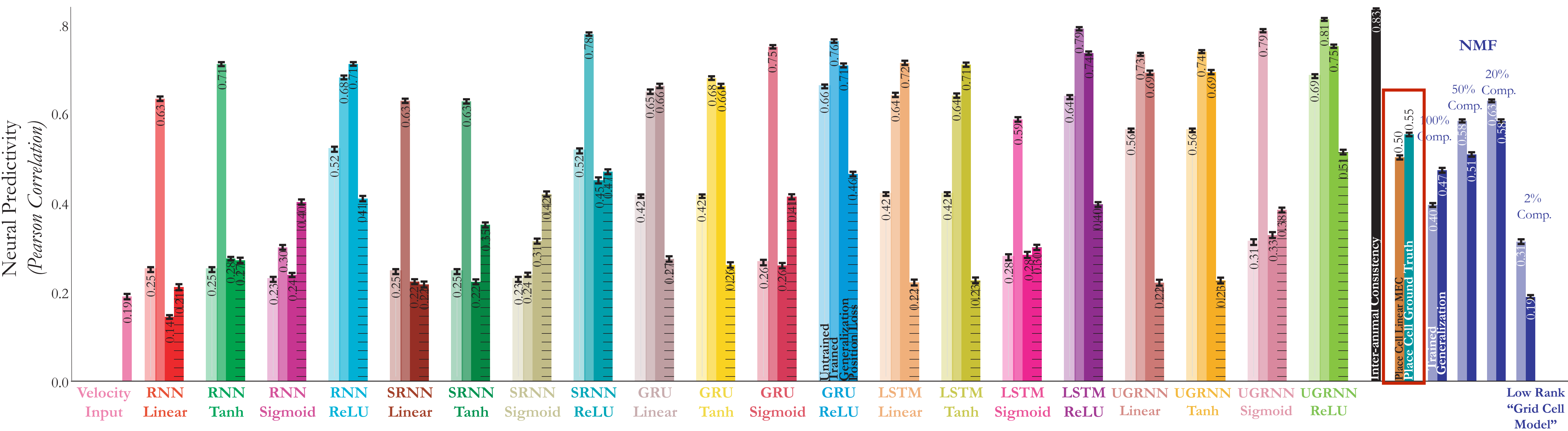
$$\mathcal{L}(\hat{p}, p) := -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_p} p_i^t \log \hat{p}_i^t$$



$$\mathcal{L}(\hat{p}, p) := \frac{1}{2}\frac{1}{T}\sum_{t=1}^{T}\left(\left(p_x^t - \hat{p}_x^t\right)^2 + \left(p_y^t - \hat{p}_y^t\right)^2\right)$$

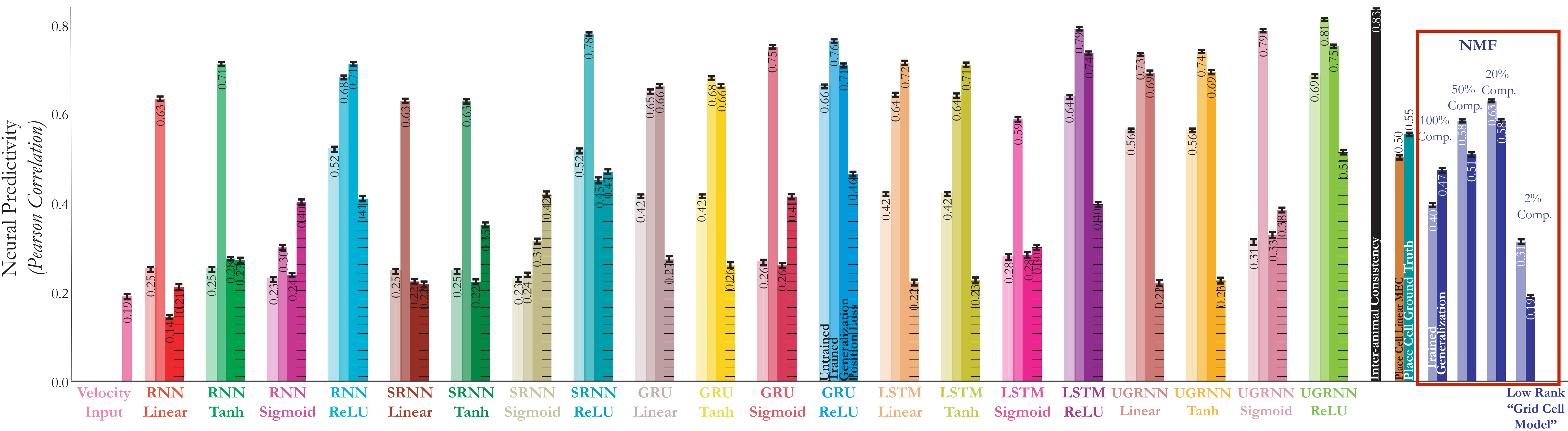


Place cells alone are a poor predictor



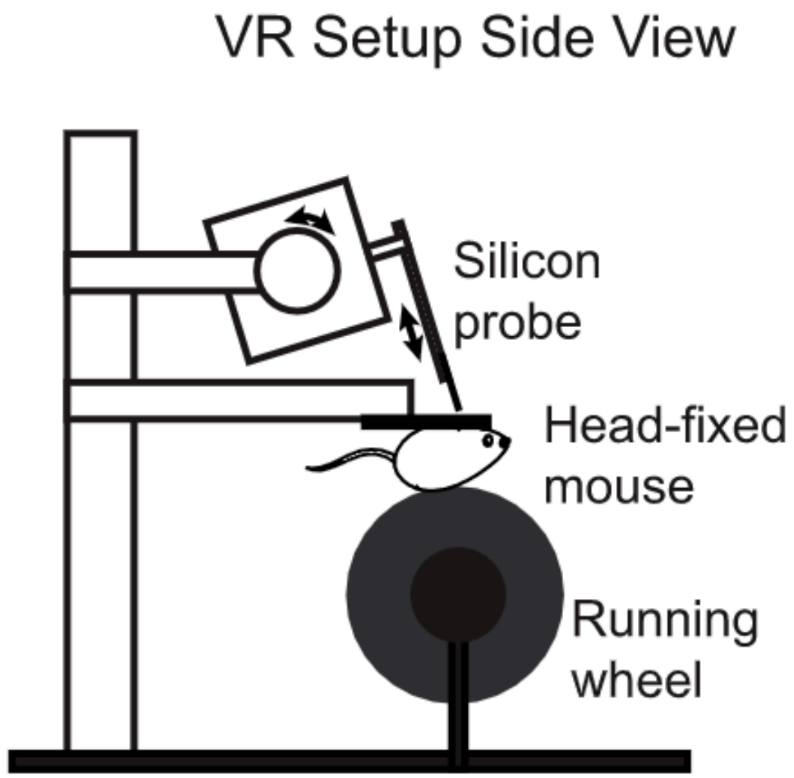
But place cells alone are not a good predictor of MEC (good!)

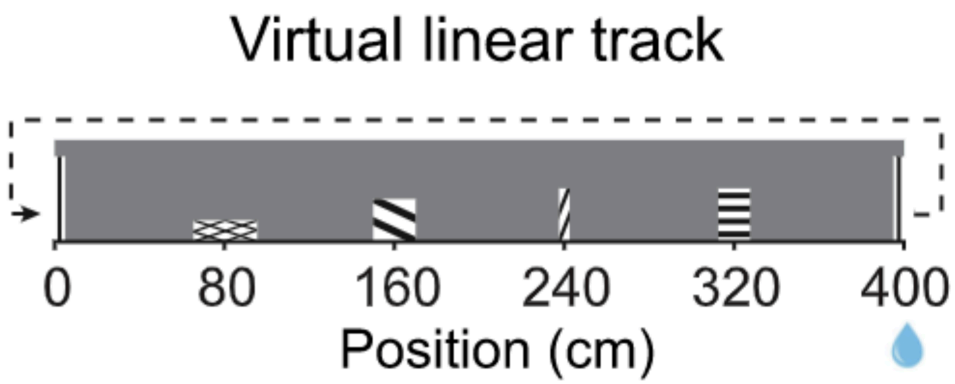
…as is NMF



Dimensionality reduction on place cells is not a good predictor of MEC either

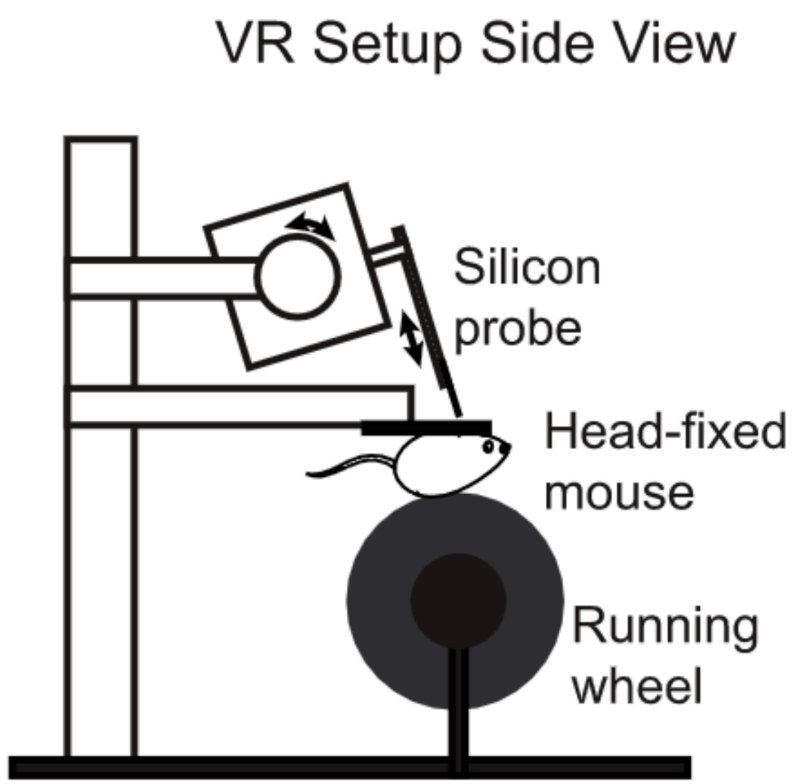
Comparing 2D trained models to 1D data

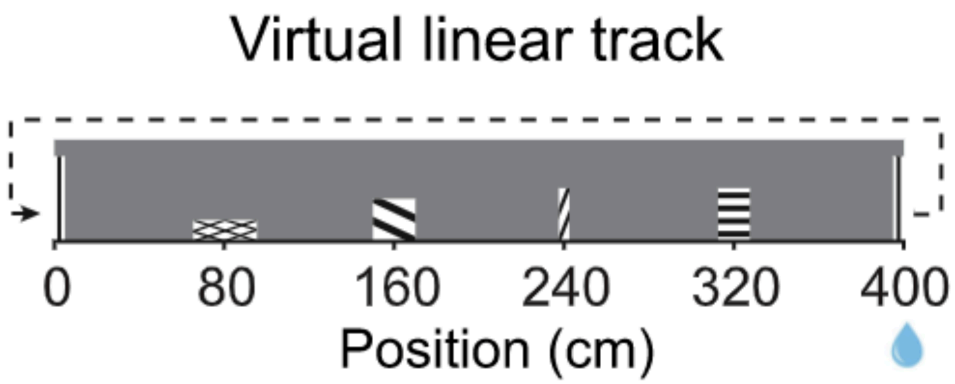




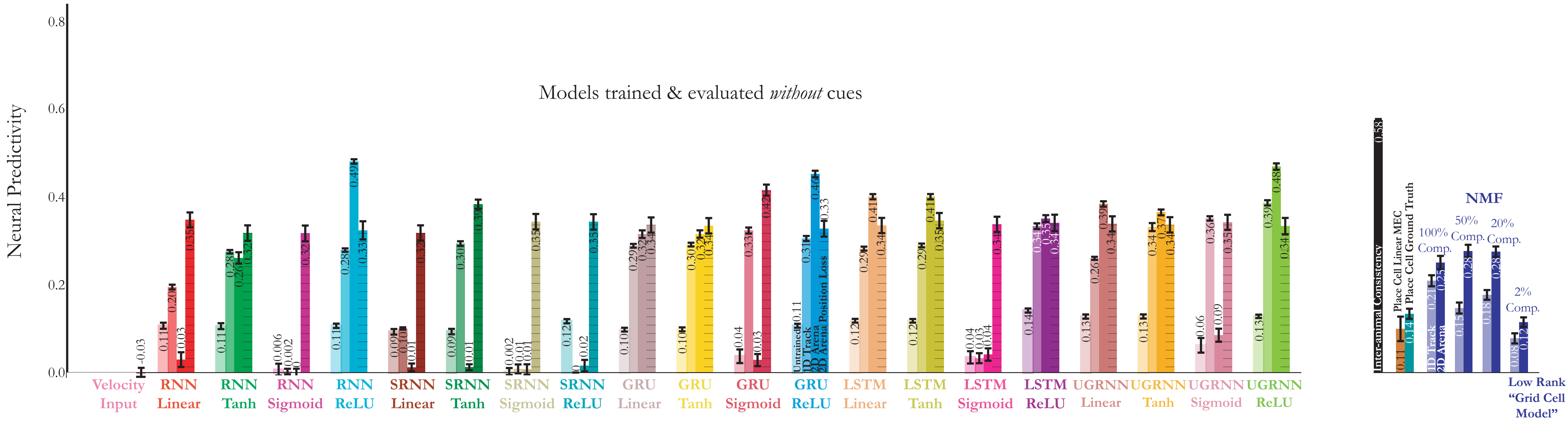
Attinger*, Campbell* et al. 2021

Comparing 2D trained models to 1D data



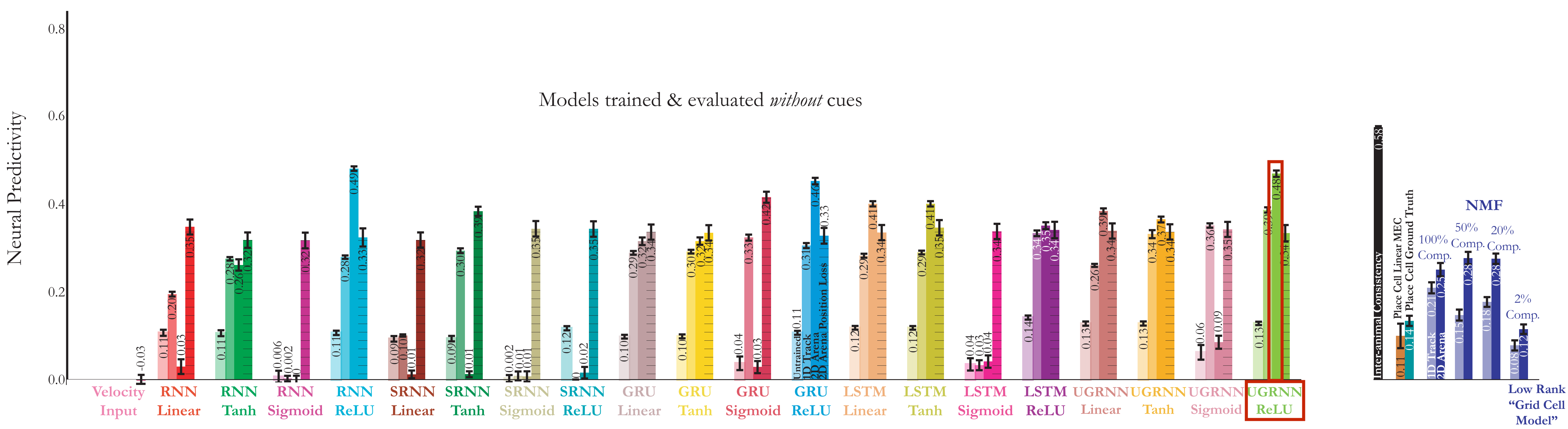


Attinger*, Campbell* et al. 2021

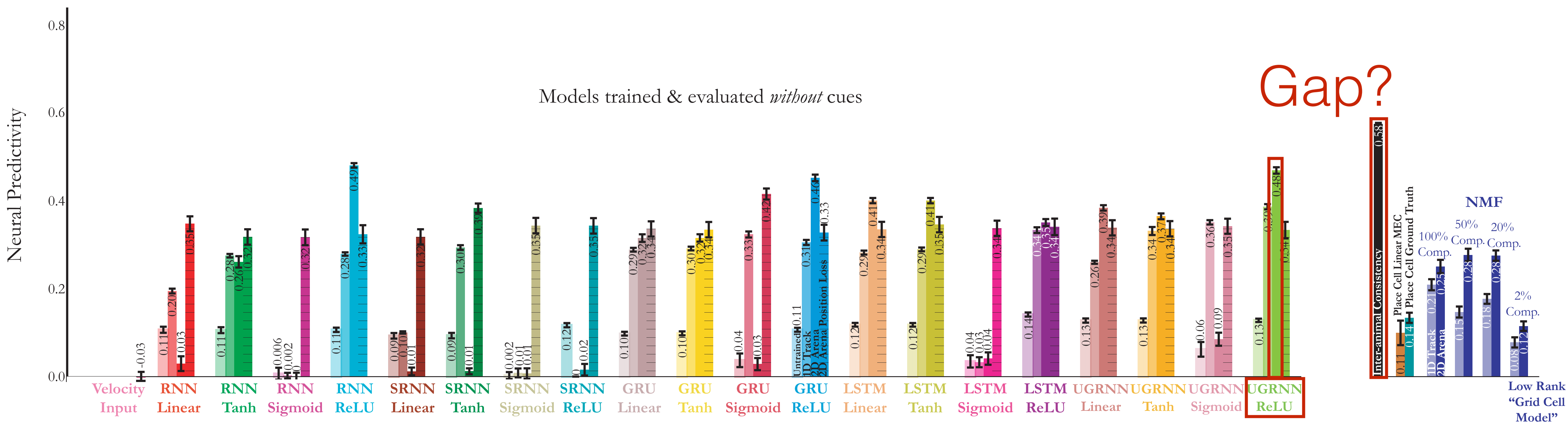


Comparing 2D trained models to 1D data

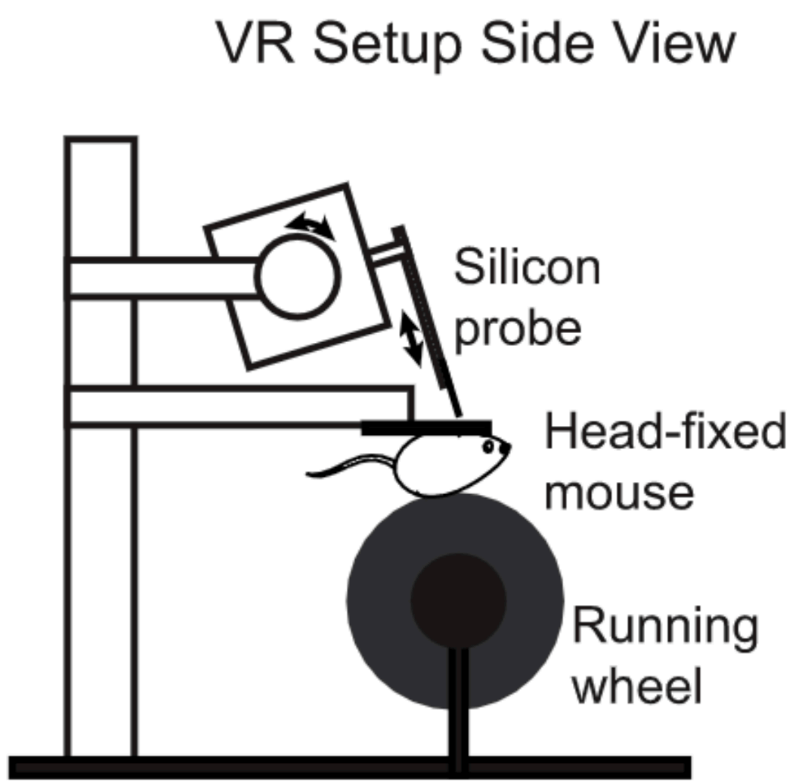
Best model in 2D generalizes to 1D!

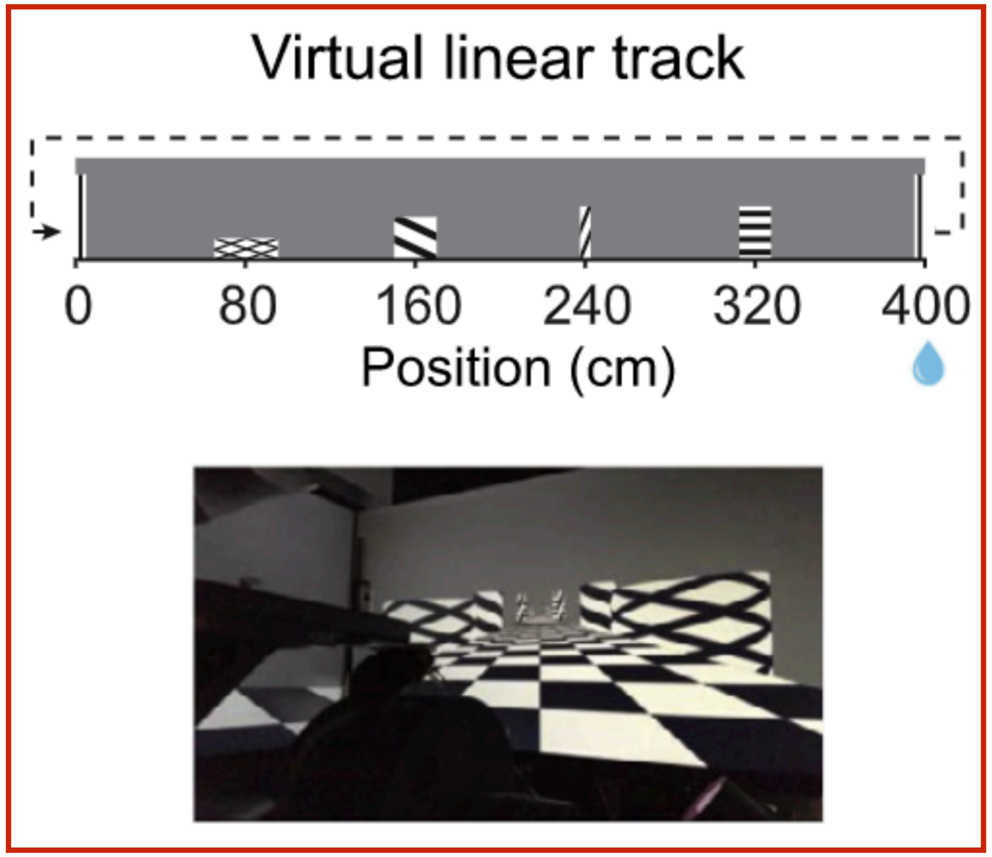


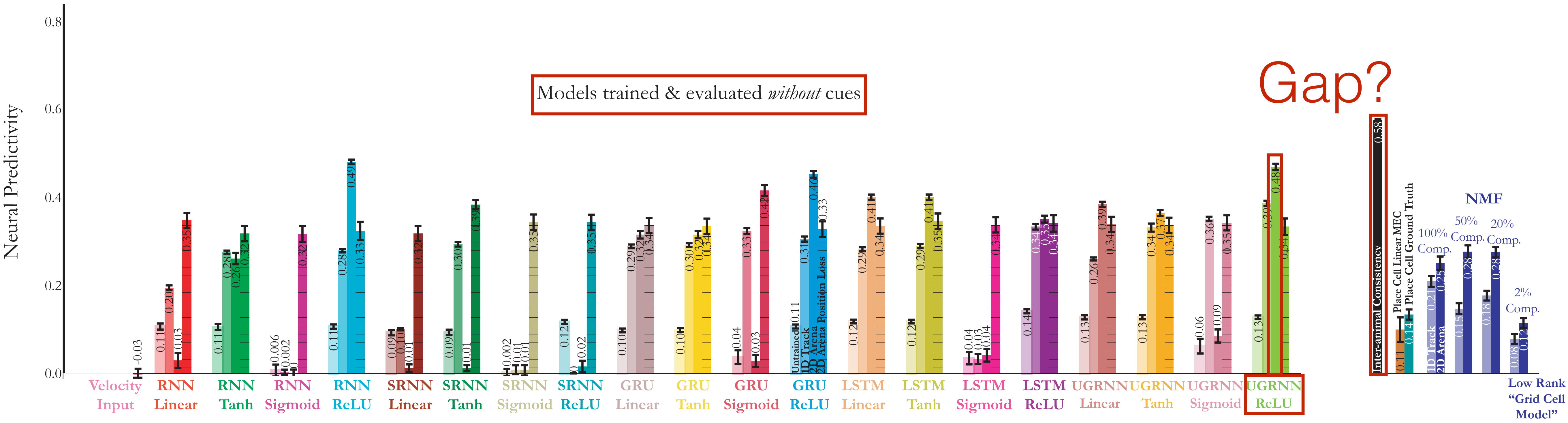
But gap between top models and inter-animal consistency…



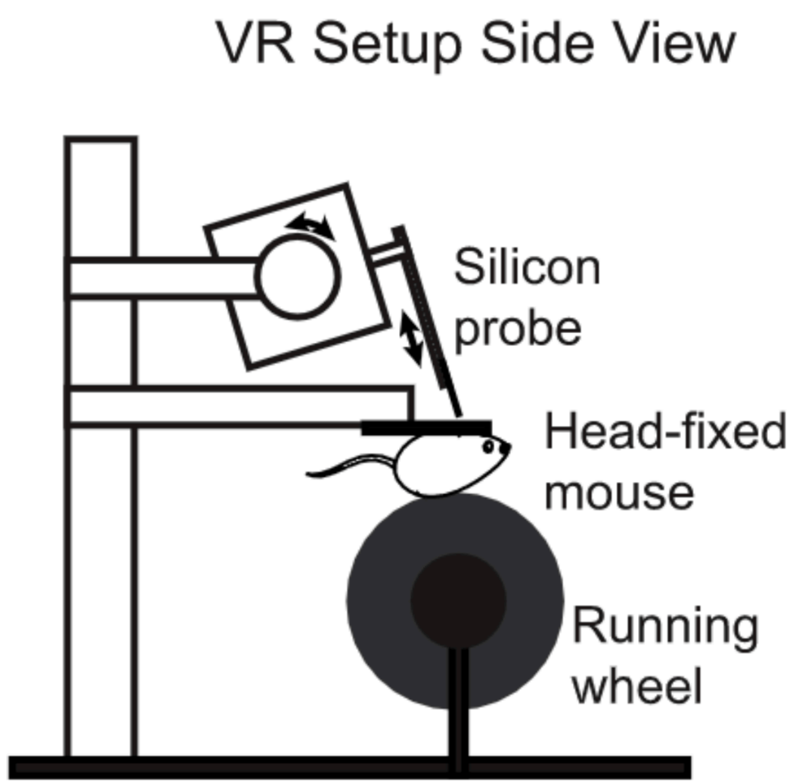
Could this be fixed by making the model sensitive to cues during evaluation?

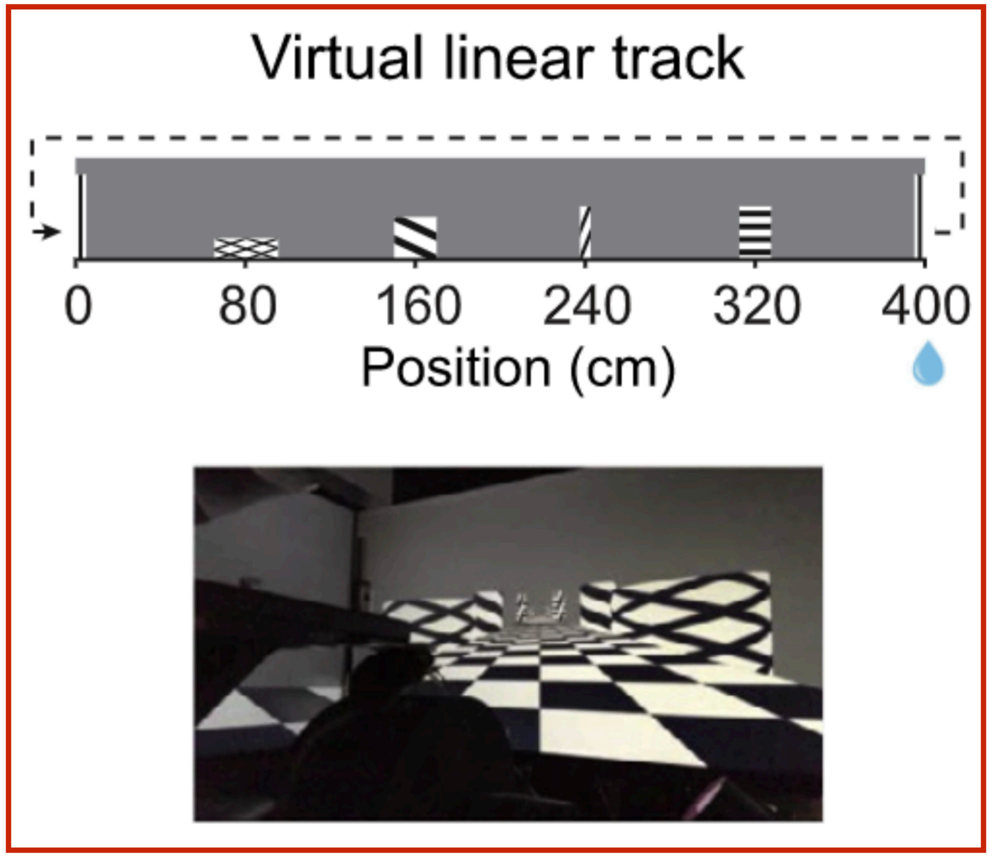




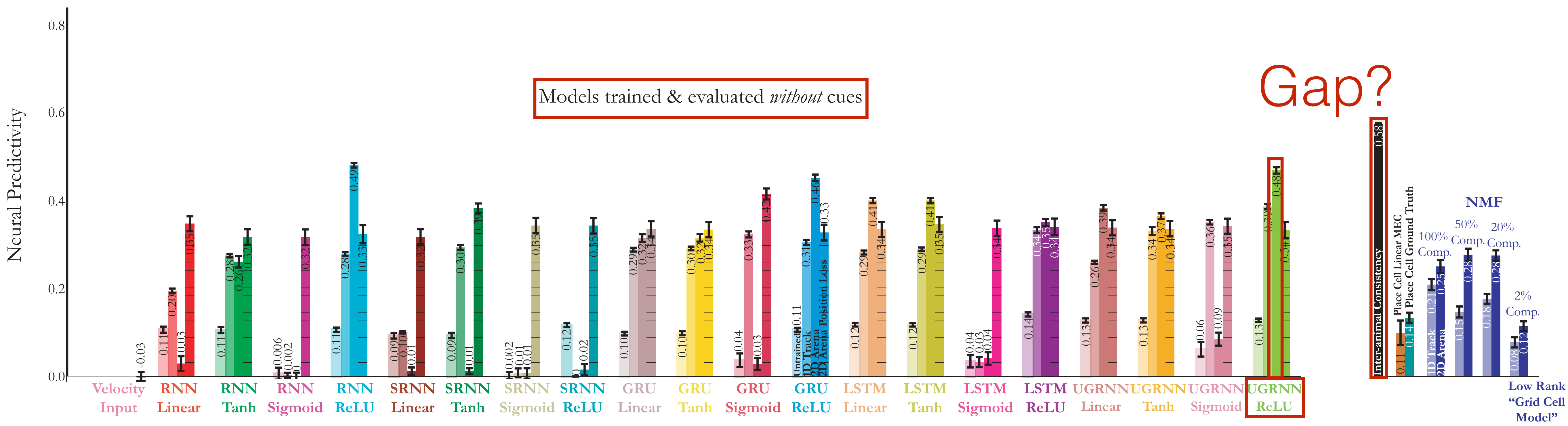


Could this be fixed by making the model sensitive to cues during evaluation?

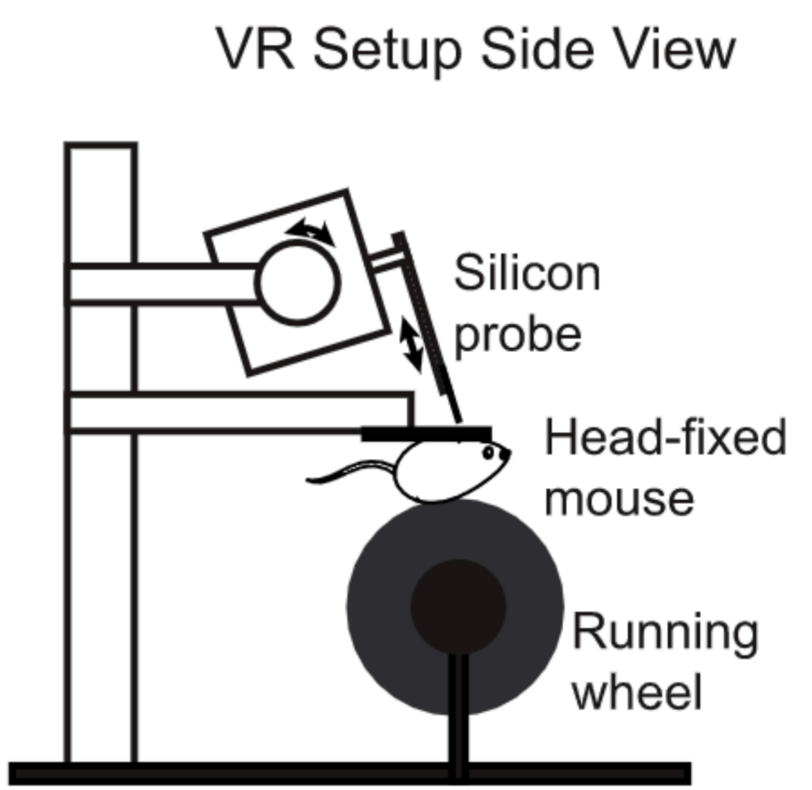


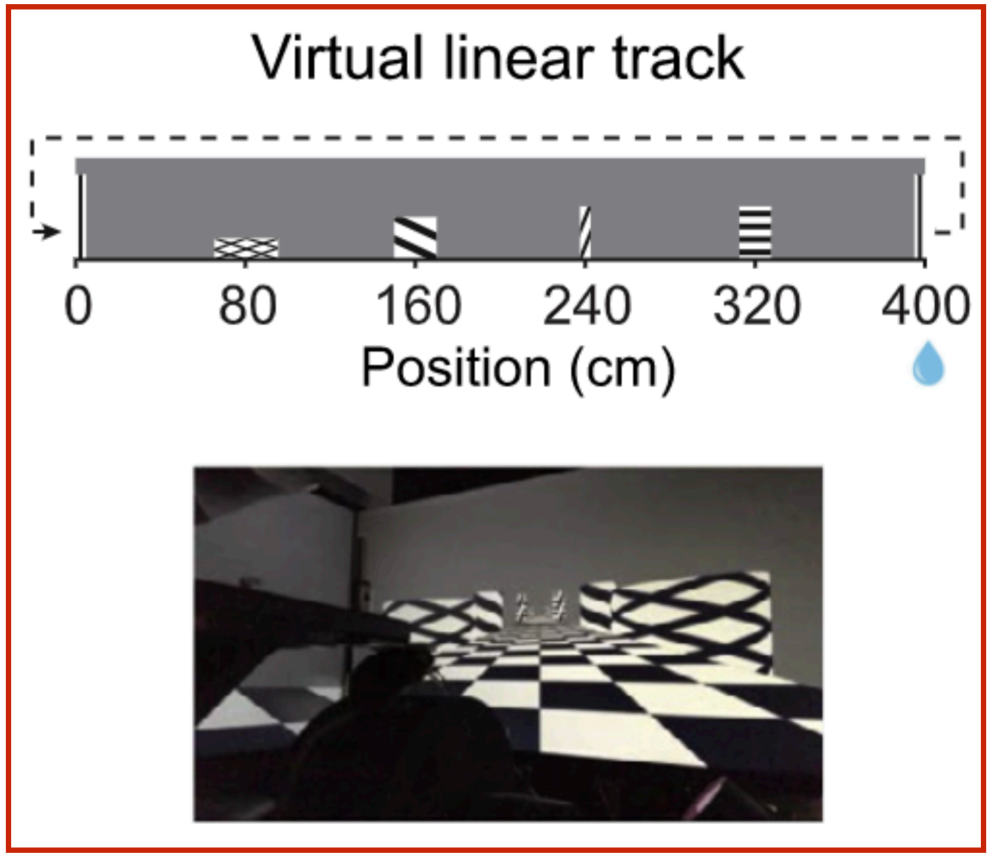


We encode visual cues as Euclidean distance to each one

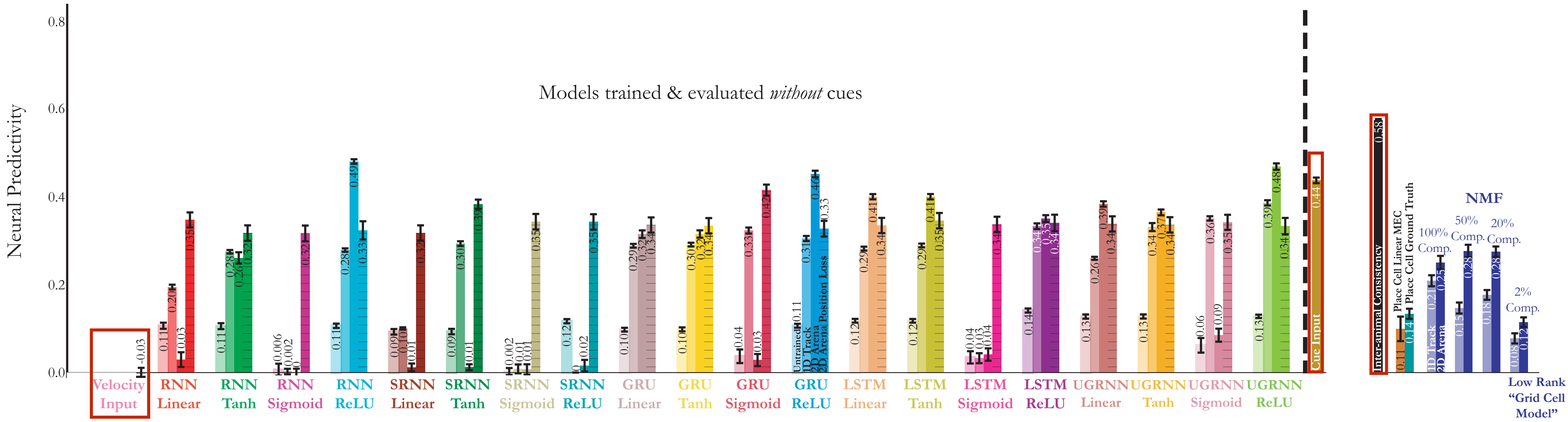


Cue input is a strong predictor of the population responses

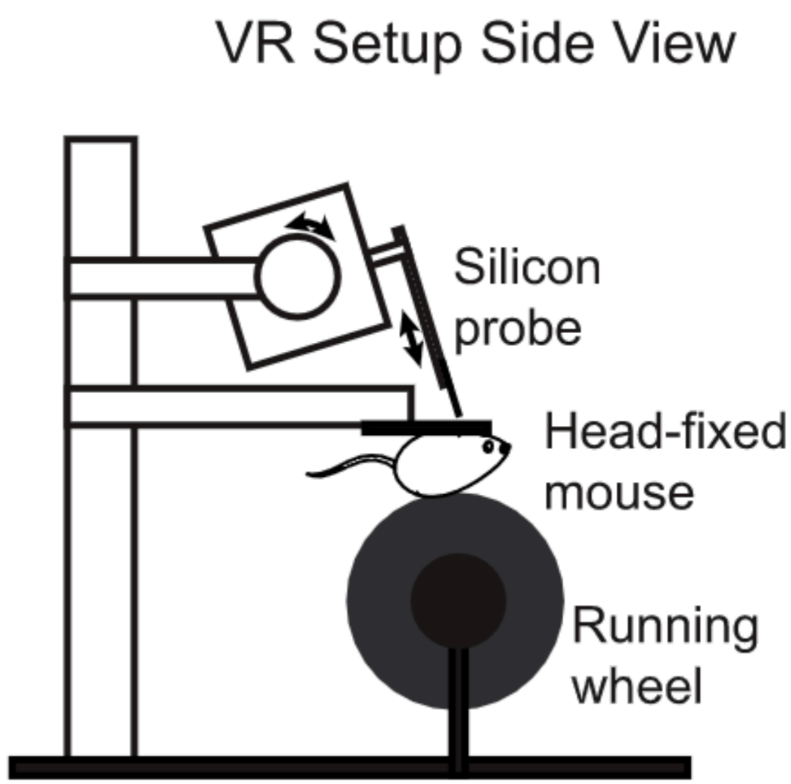


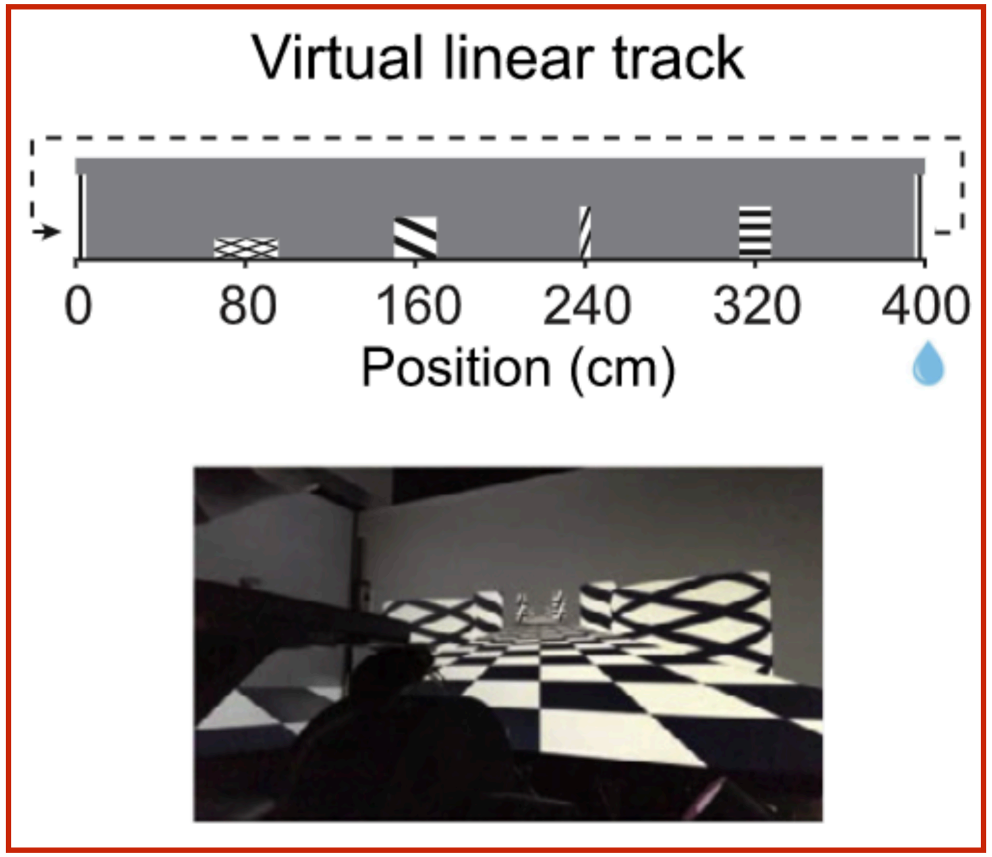


Cues drive population response variability …unlike velocity!

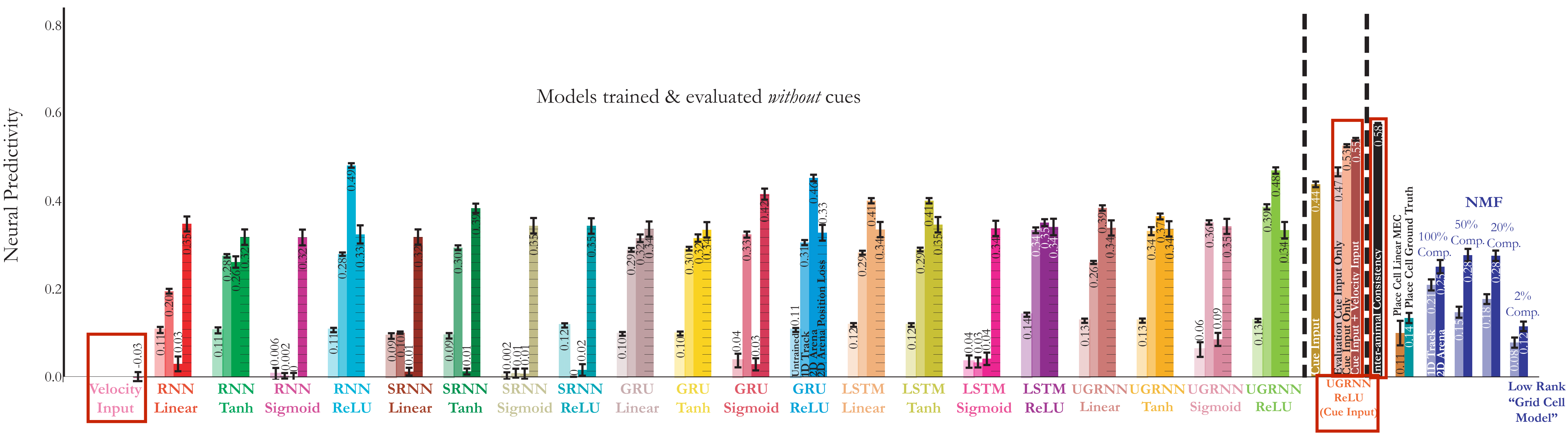


Training UGRNN ReLU with place cell loss + cue input closes gap



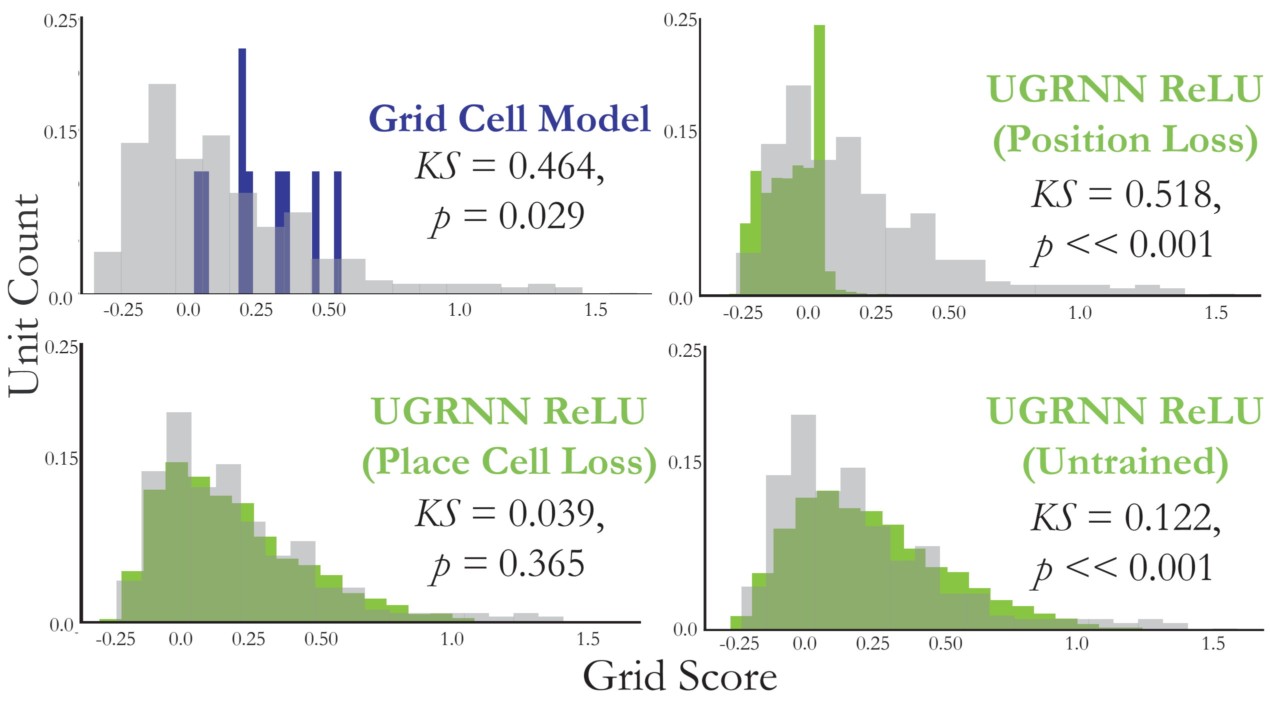


Gap closed by training with cues + velocity input (in 2D)

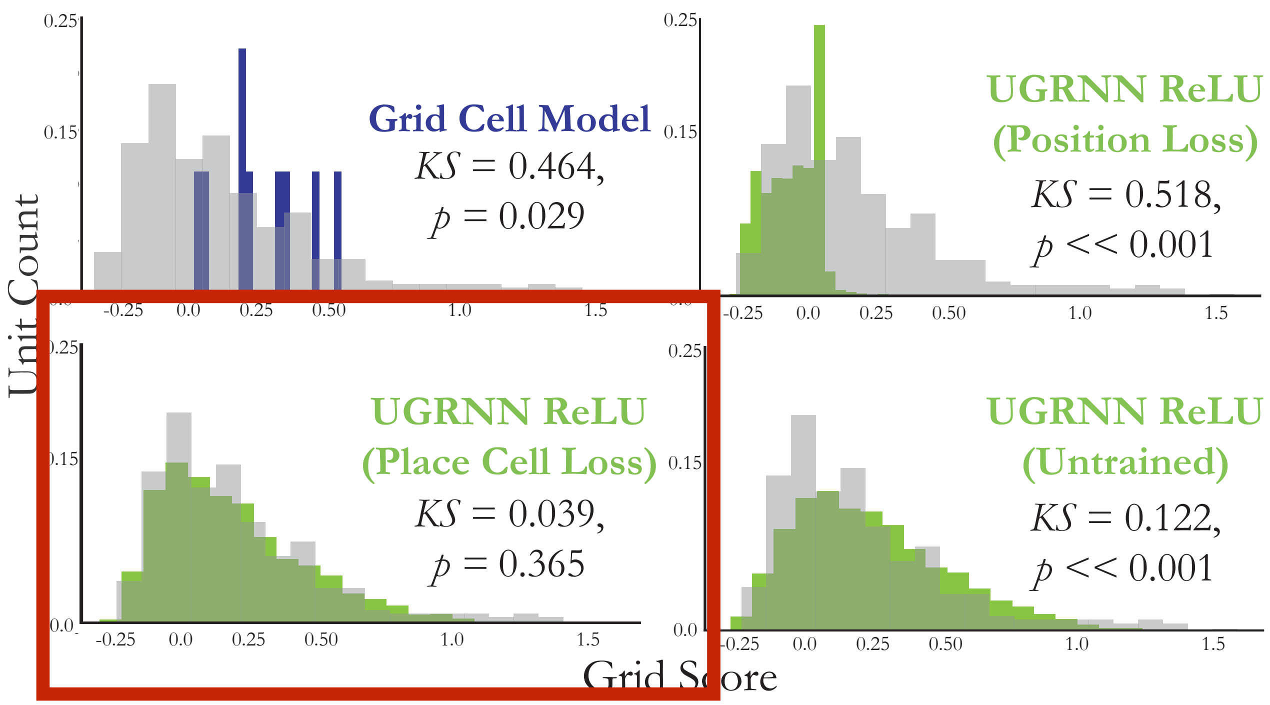


Task-optimized navigational models best predict the entire MEC population

Grid score distribution does not require any parameter fitting

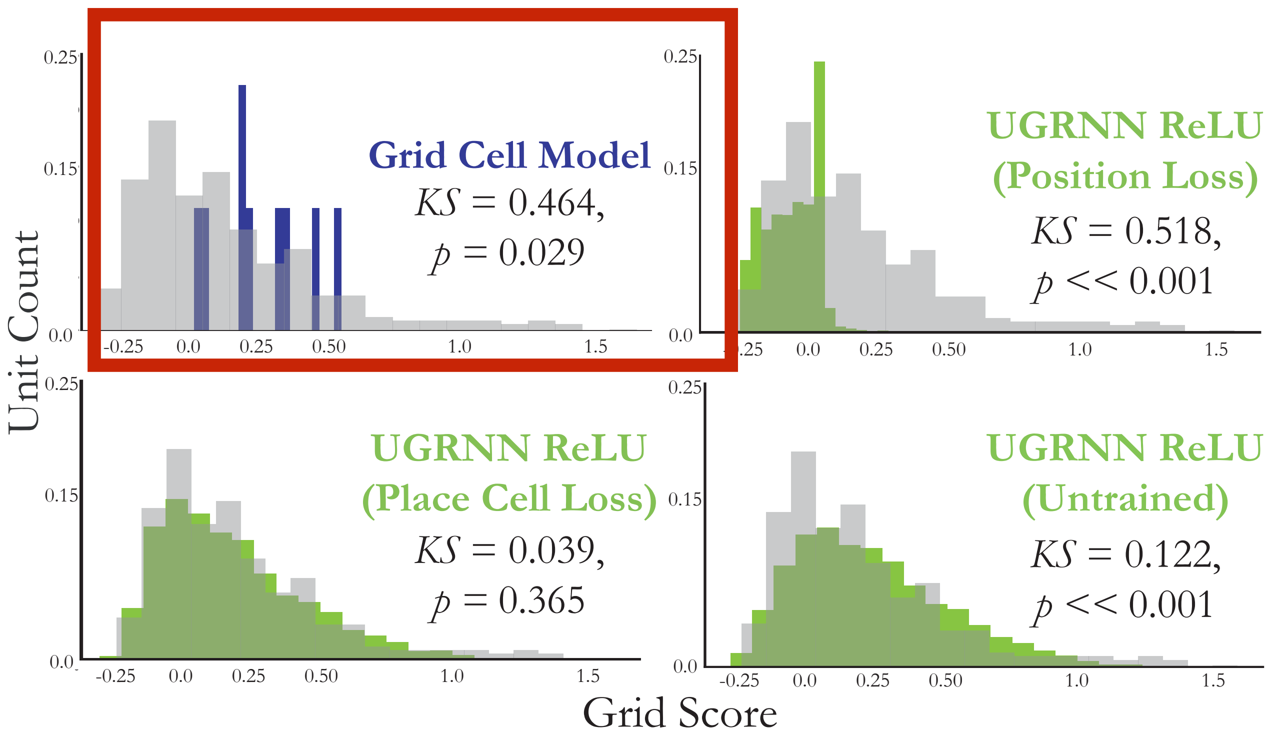


Task-optimized navigational models best predict the entire MEC population



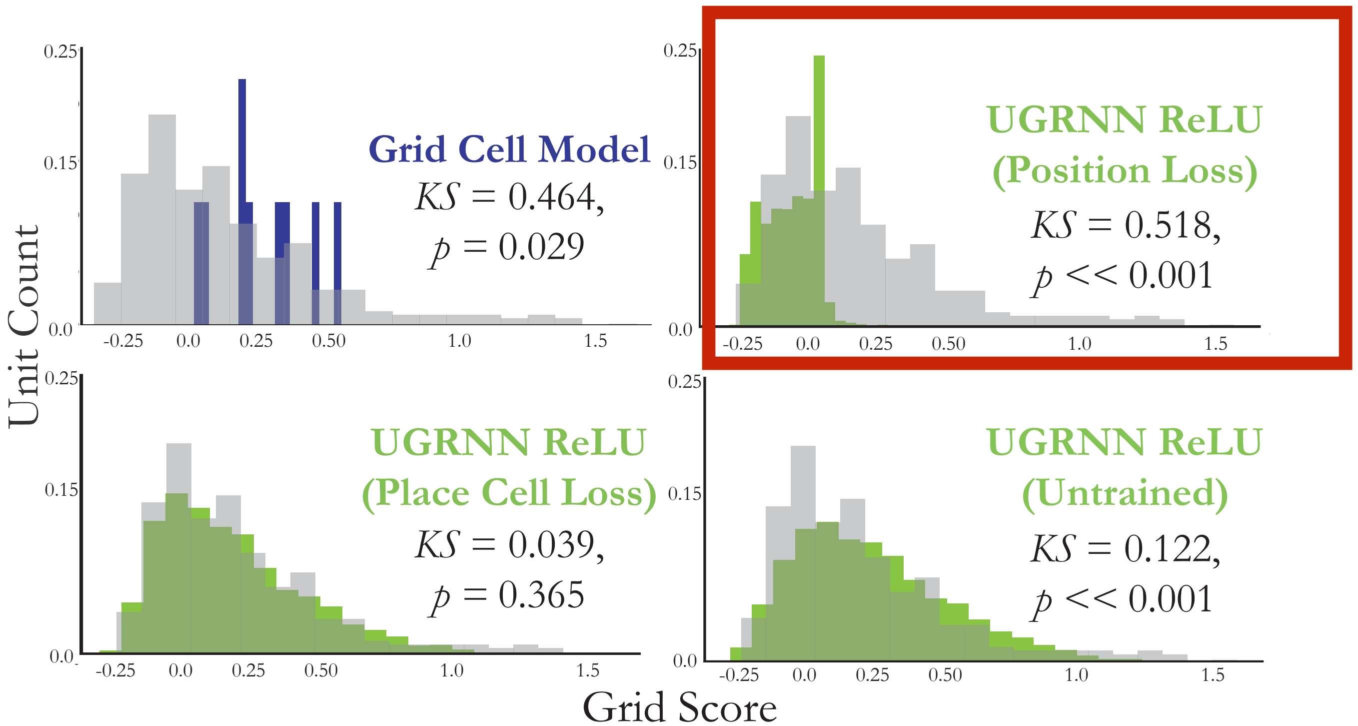
Best model class in terms of neural predictivity also matches grid score distribution in its own synthetic population

Task-optimized navigational models best predict the entire MEC population



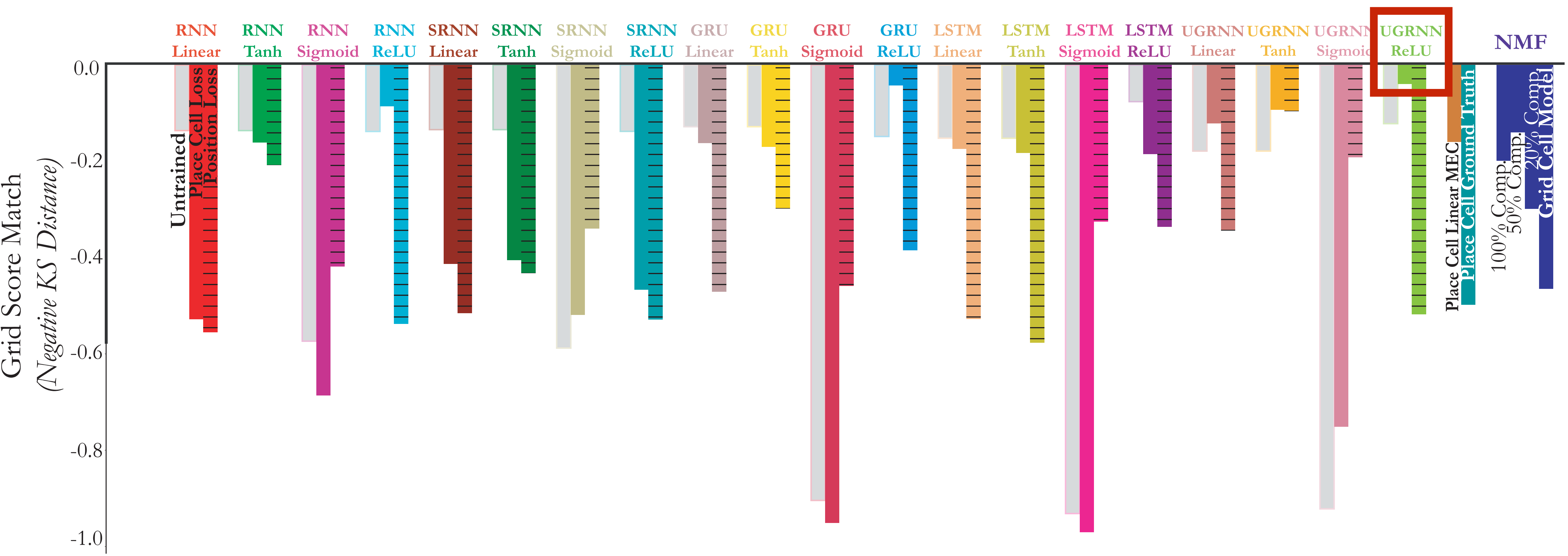
Low-rank model is too biased towards grid-like units

Task-optimized navigational models best predict the entire MEC population



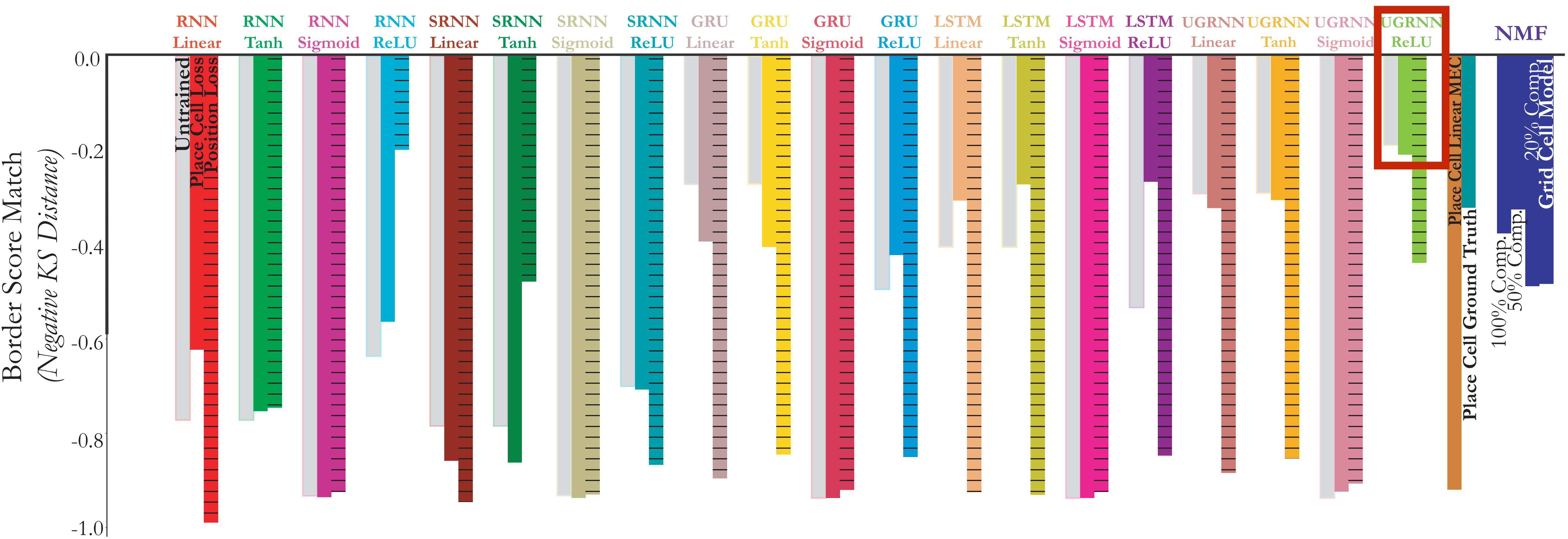
Without place cell representation, the model is too biased towards *non* grid-like units

More fine-grained unit matching metrics



Best model matches the data's grid score distribution in its own synthetic population

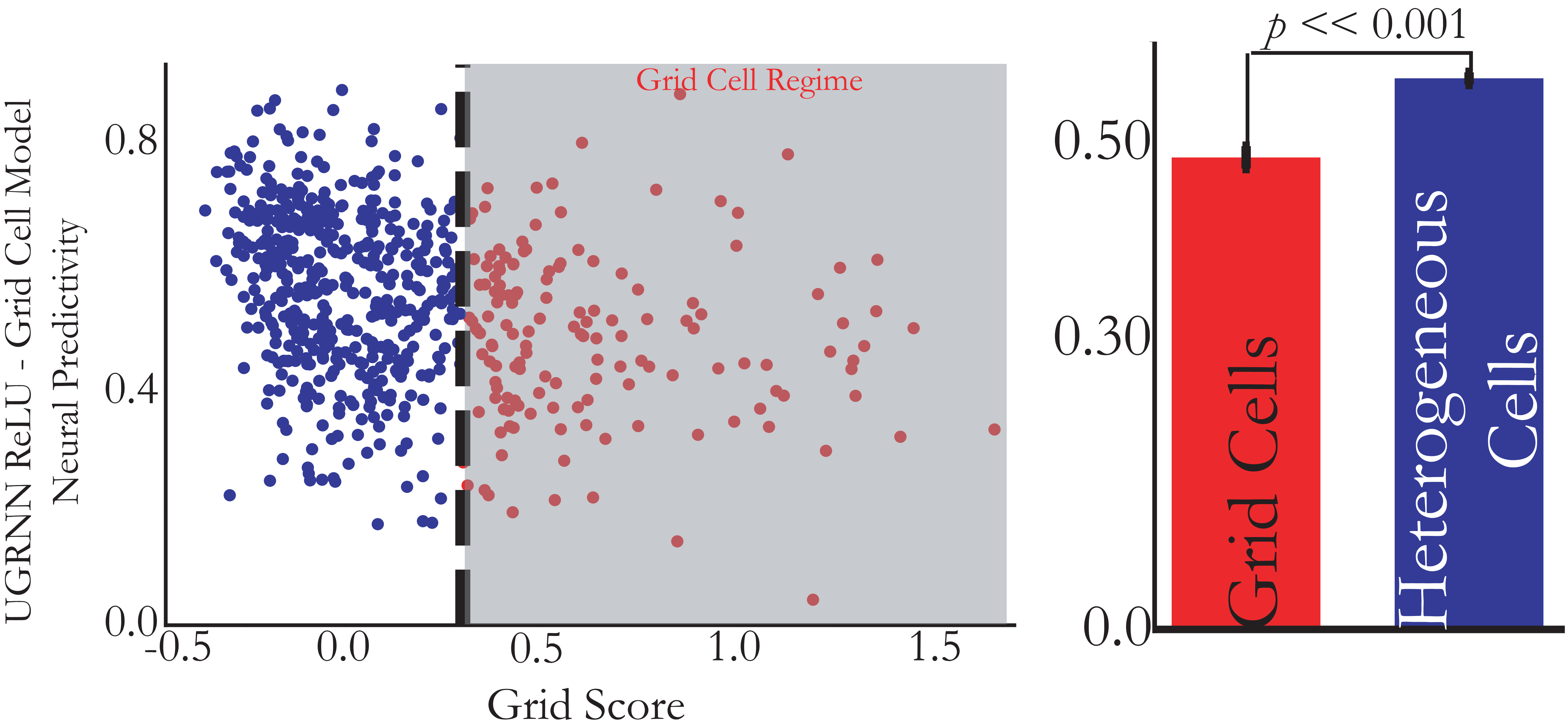
More fine-grained unit matching metrics



Best model also matches the data's *border* score distribution in its own synthetic population

Neural network model better predicts heterogeneous cells

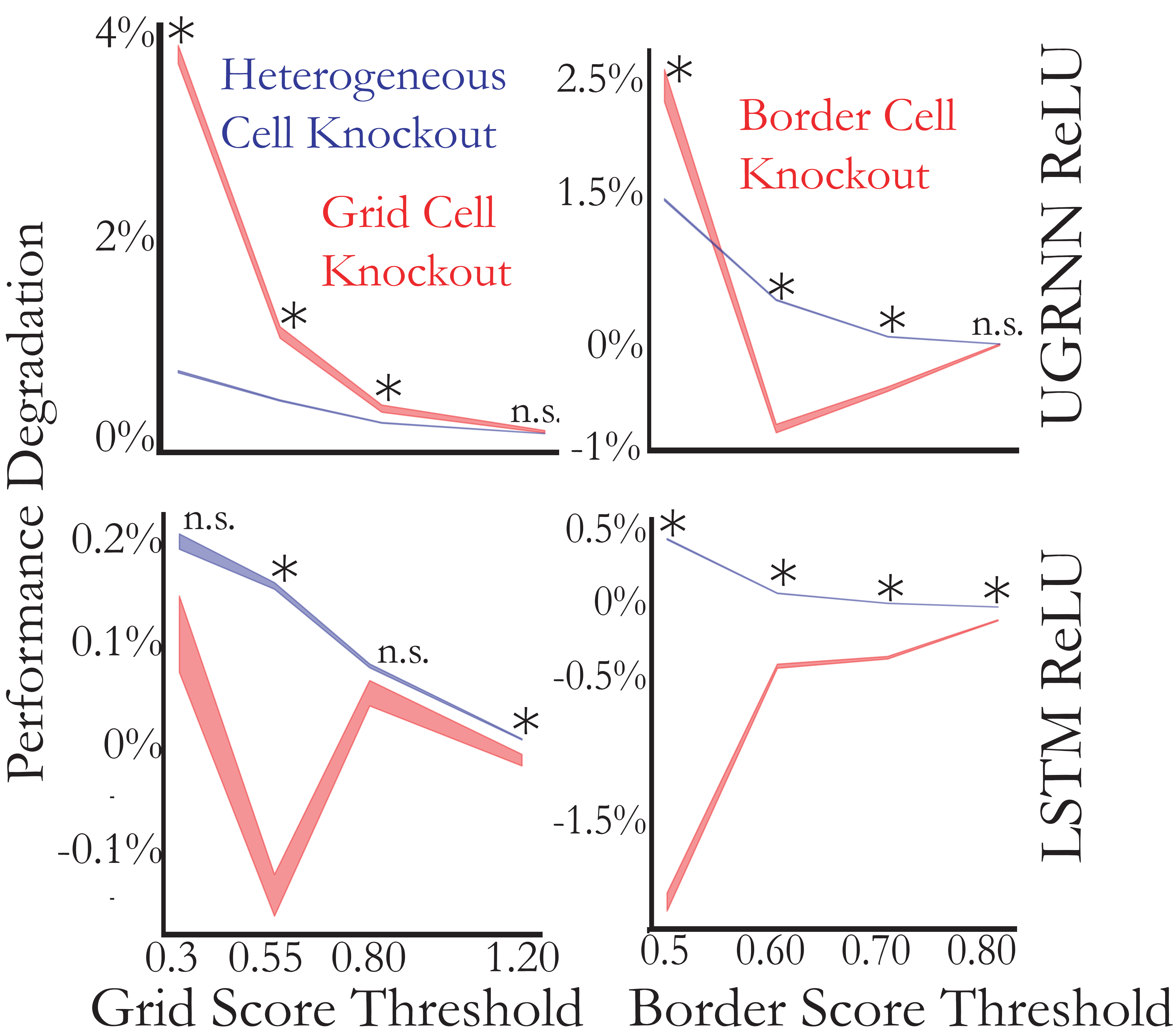
Neural network models are differentially better at heterogeneous cells than NMF



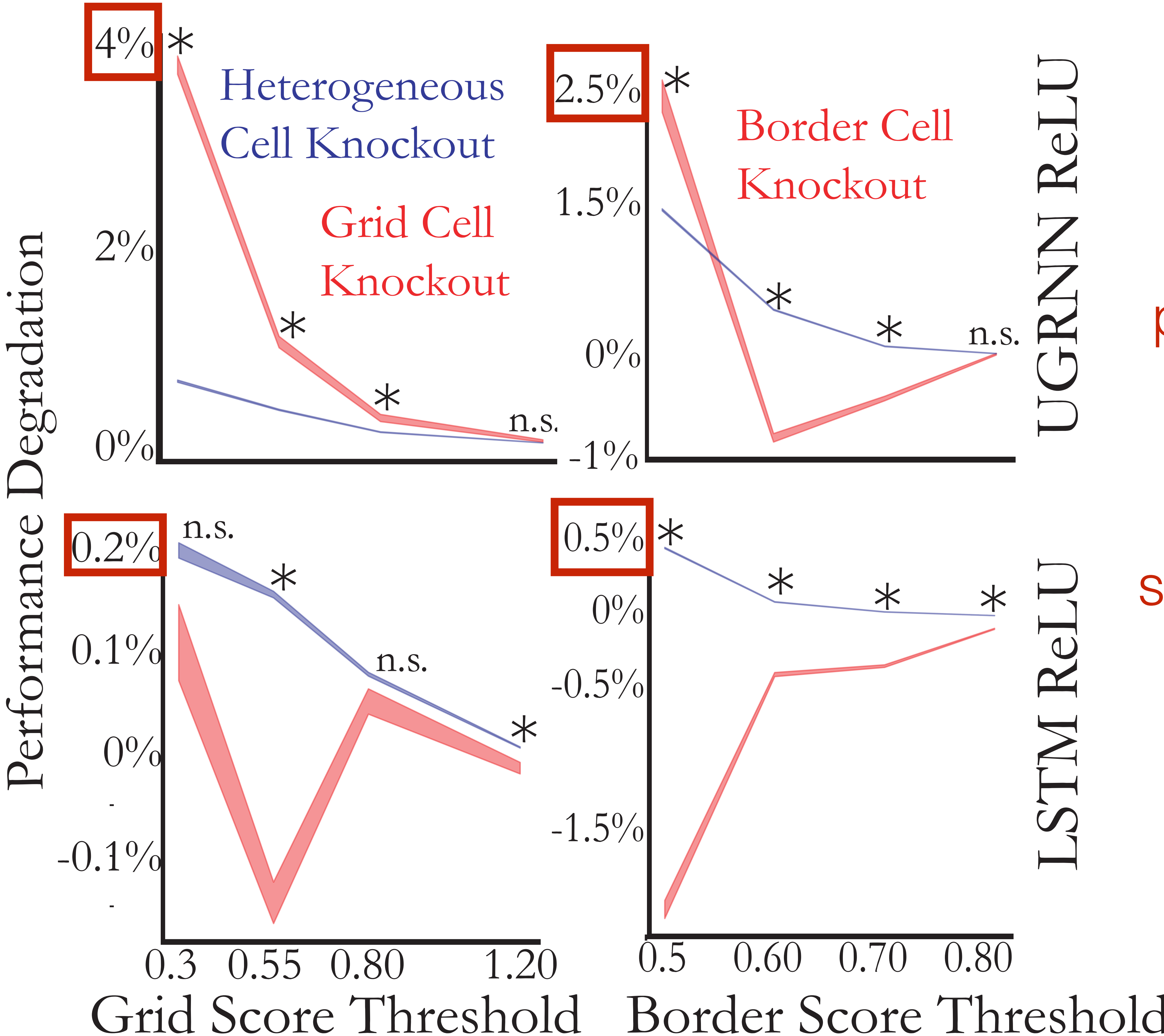
Knockout experiments

Given that we have a model that exhibits close similarity to MEC, we can use it to generate predictions for experiments that are very difficult to do

Knockout experiments

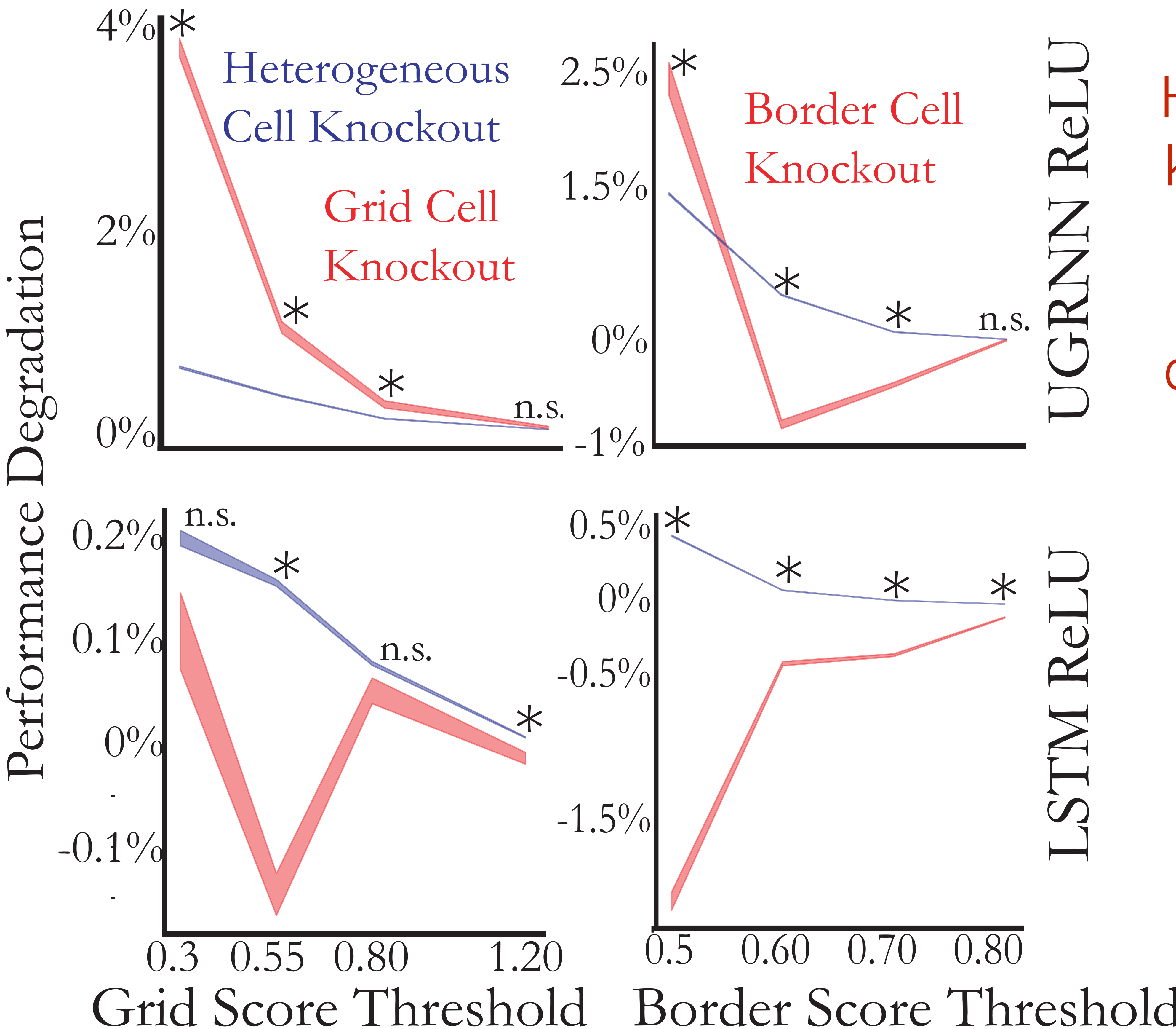


Networks are robust to knockouts



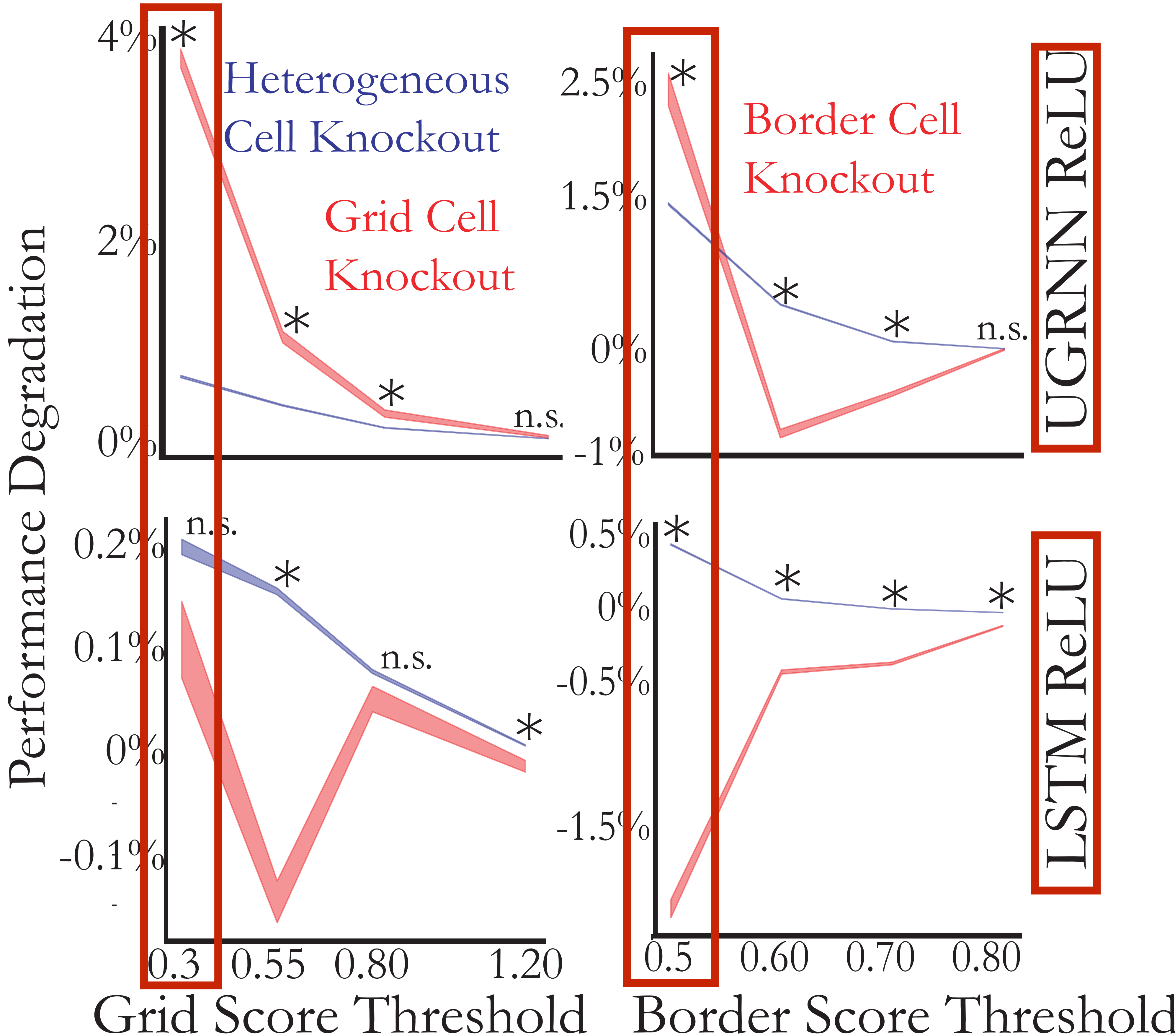
Network performance is robust to knockouts on the order of several hundred units

Heterogeneous cells are relevant to navigation



Heterogeneous knockout gives similar performance degradation as cell type specific knockout, especially as threshold increases

Differences in gating architecture



At the lowest threshold of cell type specificity, different gating architectures give somewhat different predictions, which may be useful to gather evidence for in future experiments

Accounting for rewards

Why do we want the same model to account for rewards?

Accounting for rewards

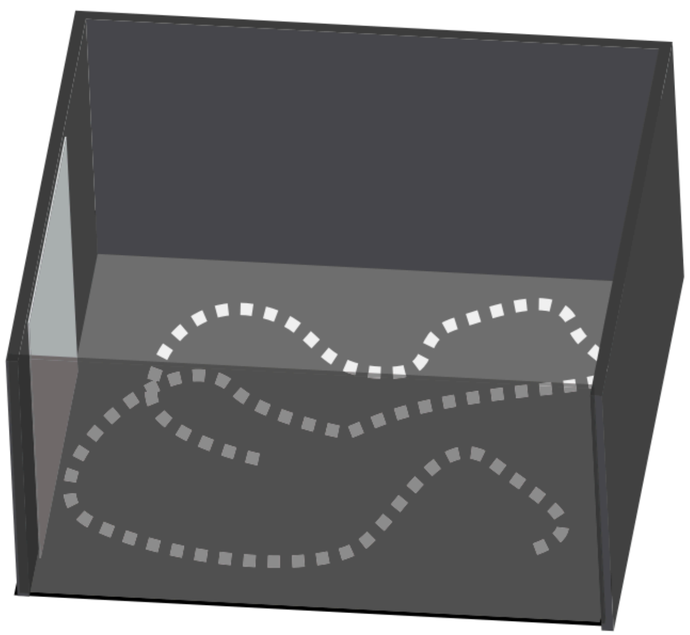
Why do we want the same model to account for rewards?

Because we think that non-spatial rewards are nonetheless part of the same underlying framework.

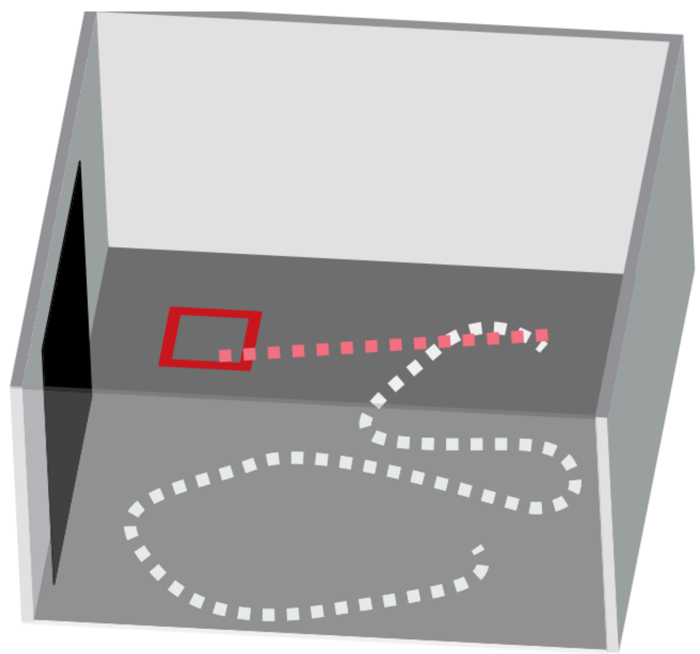
Remapping in the presence of reward

Remembered reward locations restructure entorhinal spatial maps

William N. Butler*, Kiah Hardcastle*, Lisa M. Giocomo†



free foraging (ENV1)

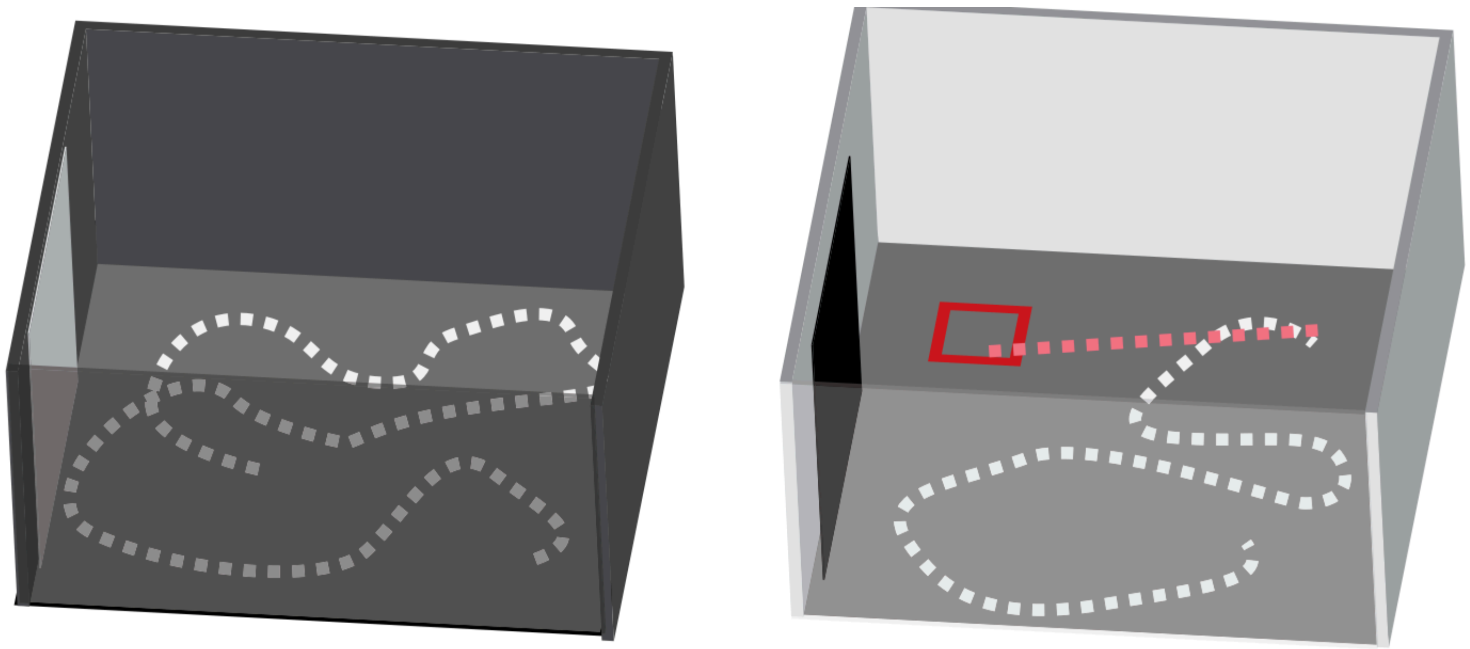


spatial task (ENV2)

Remapping in the presence of reward

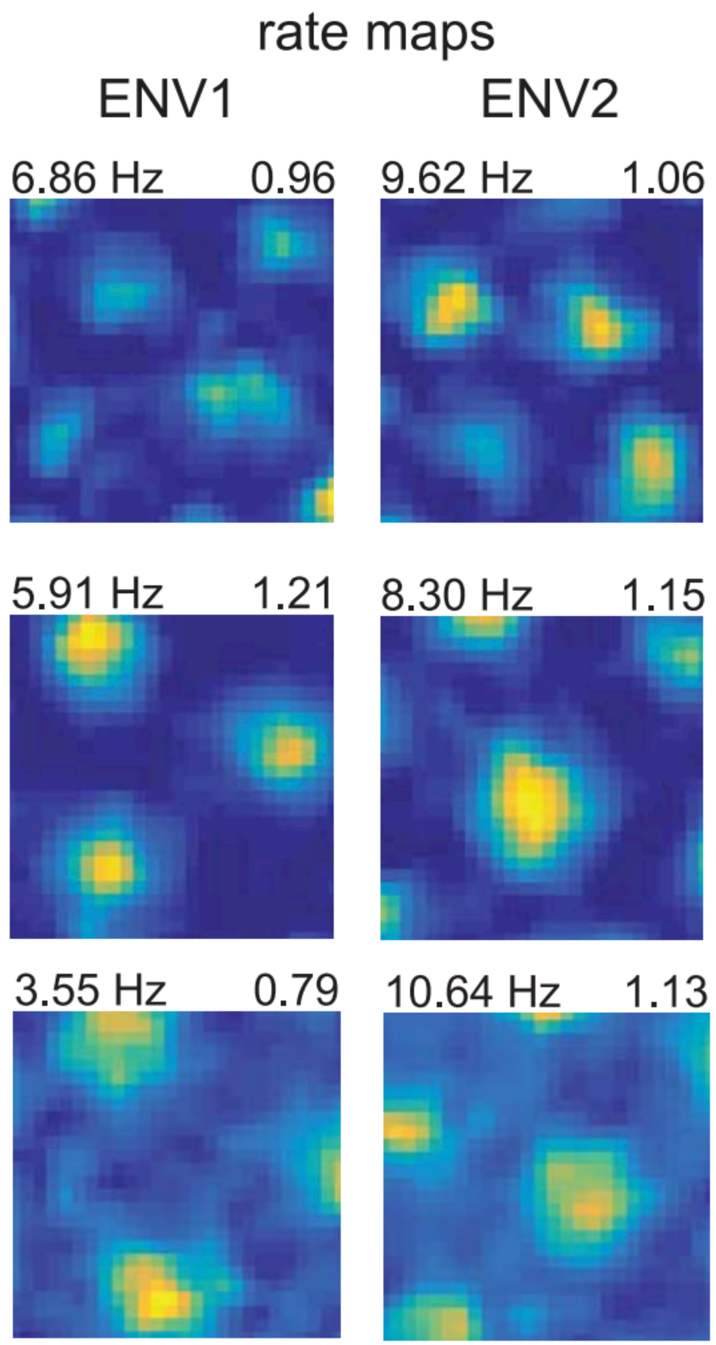
Remembered reward locations restructure entorhinal spatial maps

William N. Butler*, Kiah Hardcastle*, Lisa M. Giocomo†

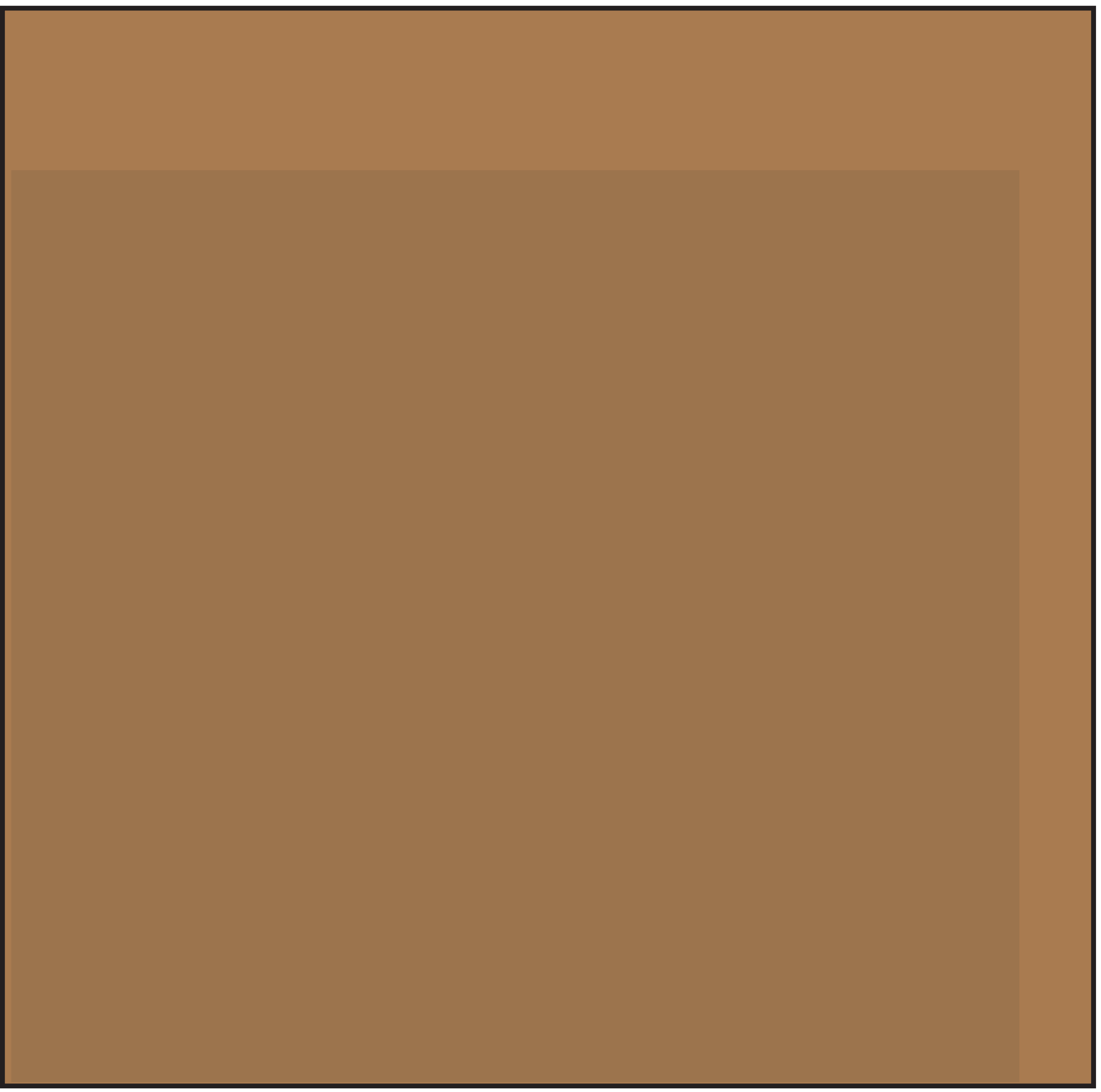


free foraging (ENV1)

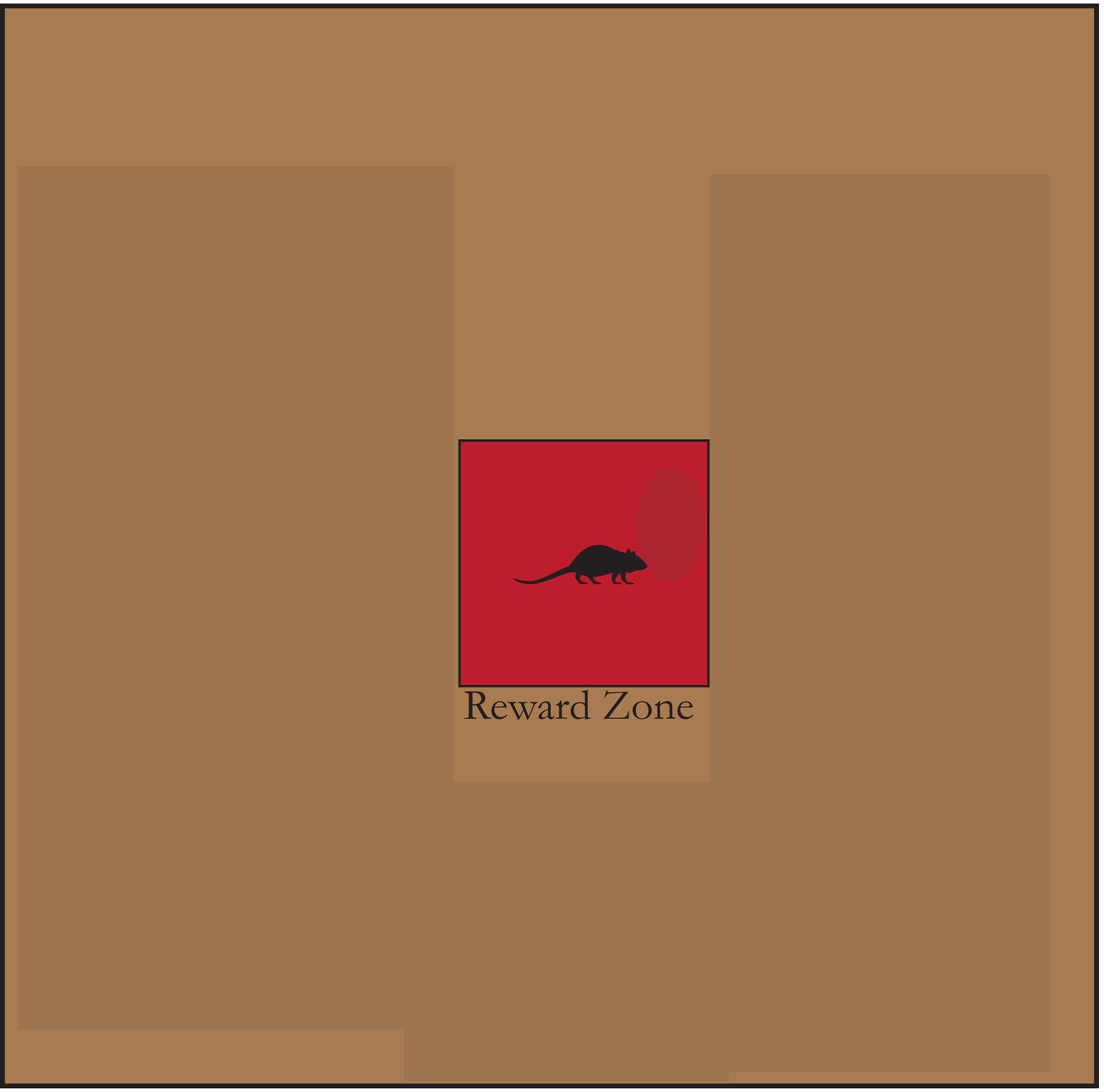
spatial task (ENV2)



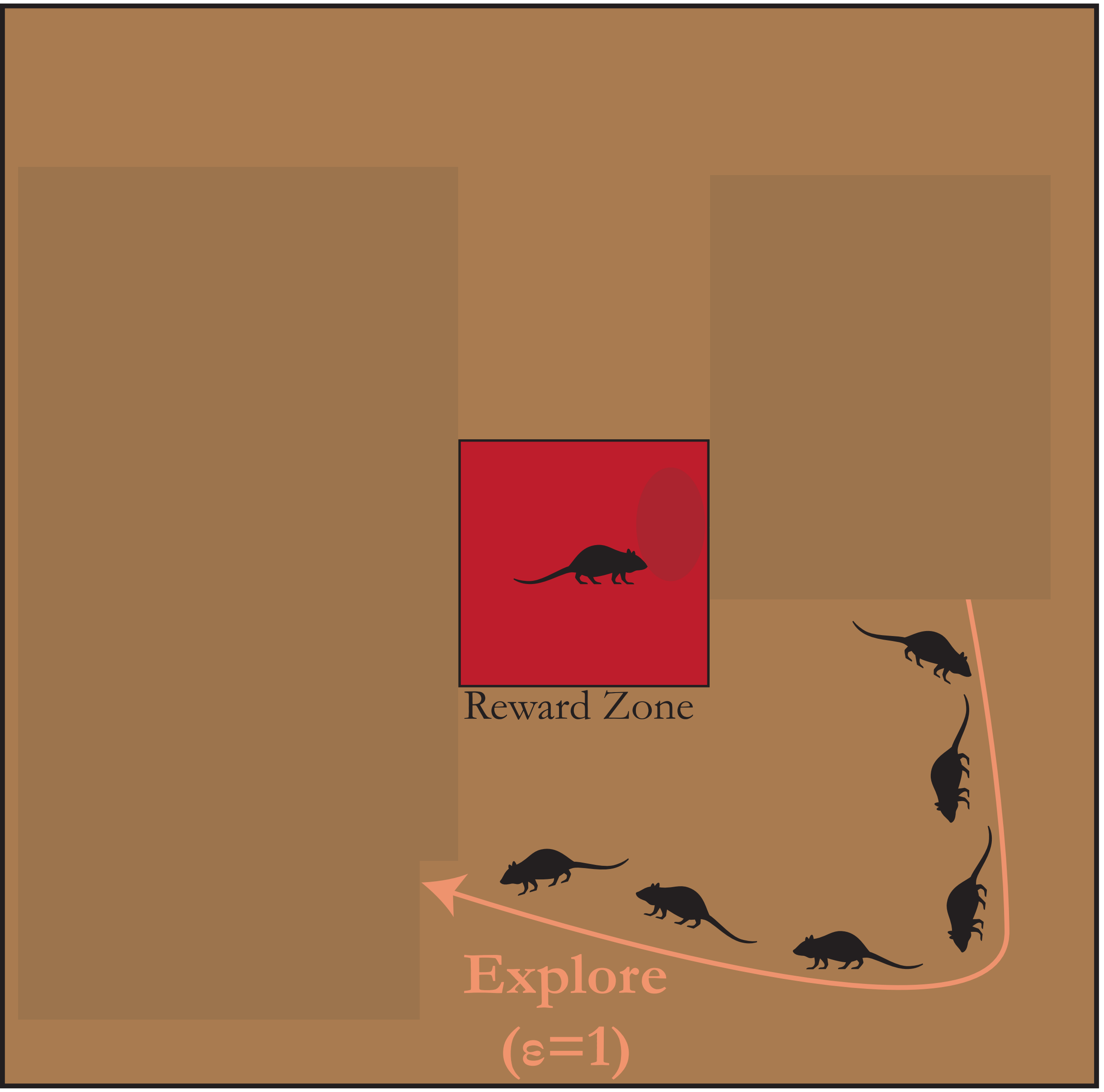
Modeling rewards



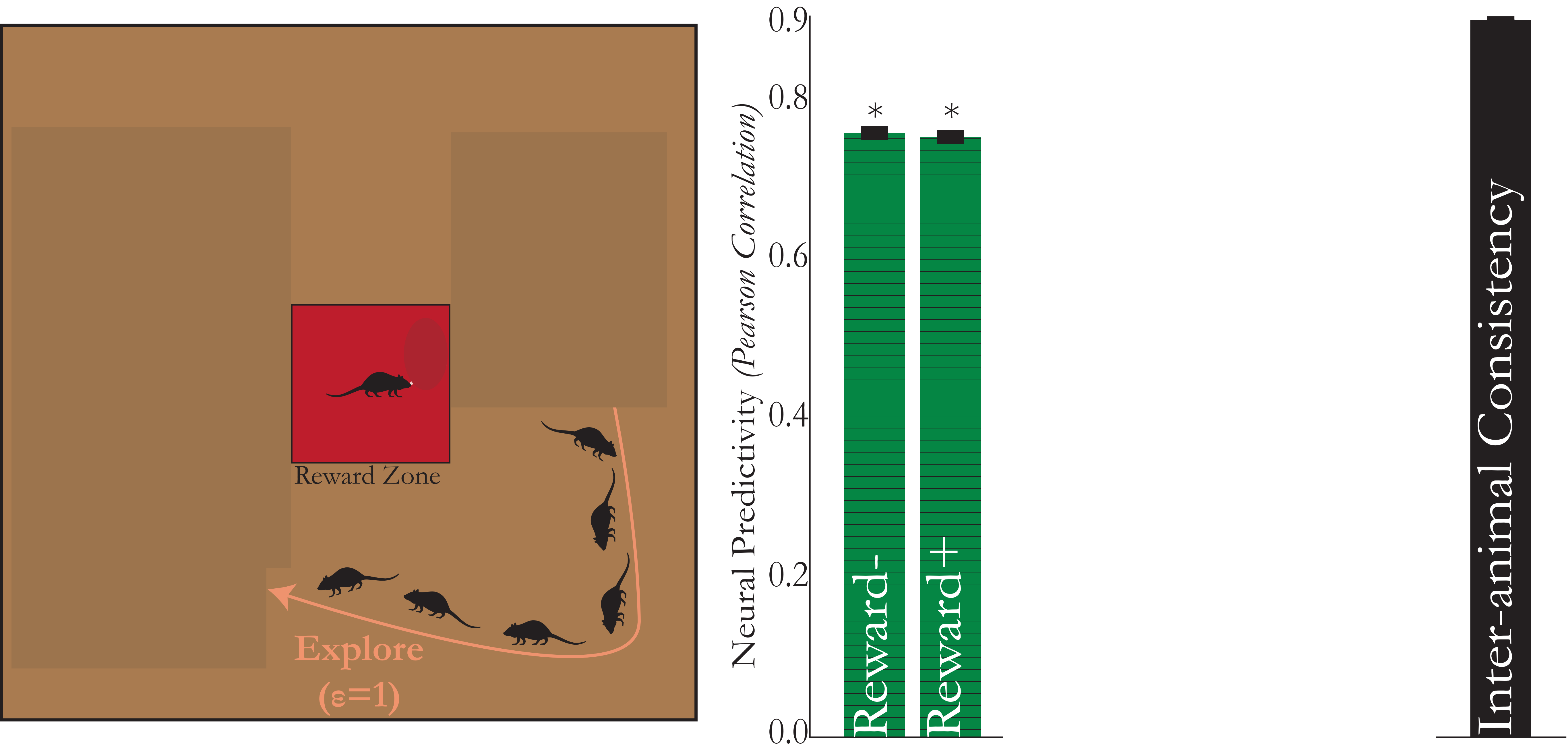
Modeling rewards



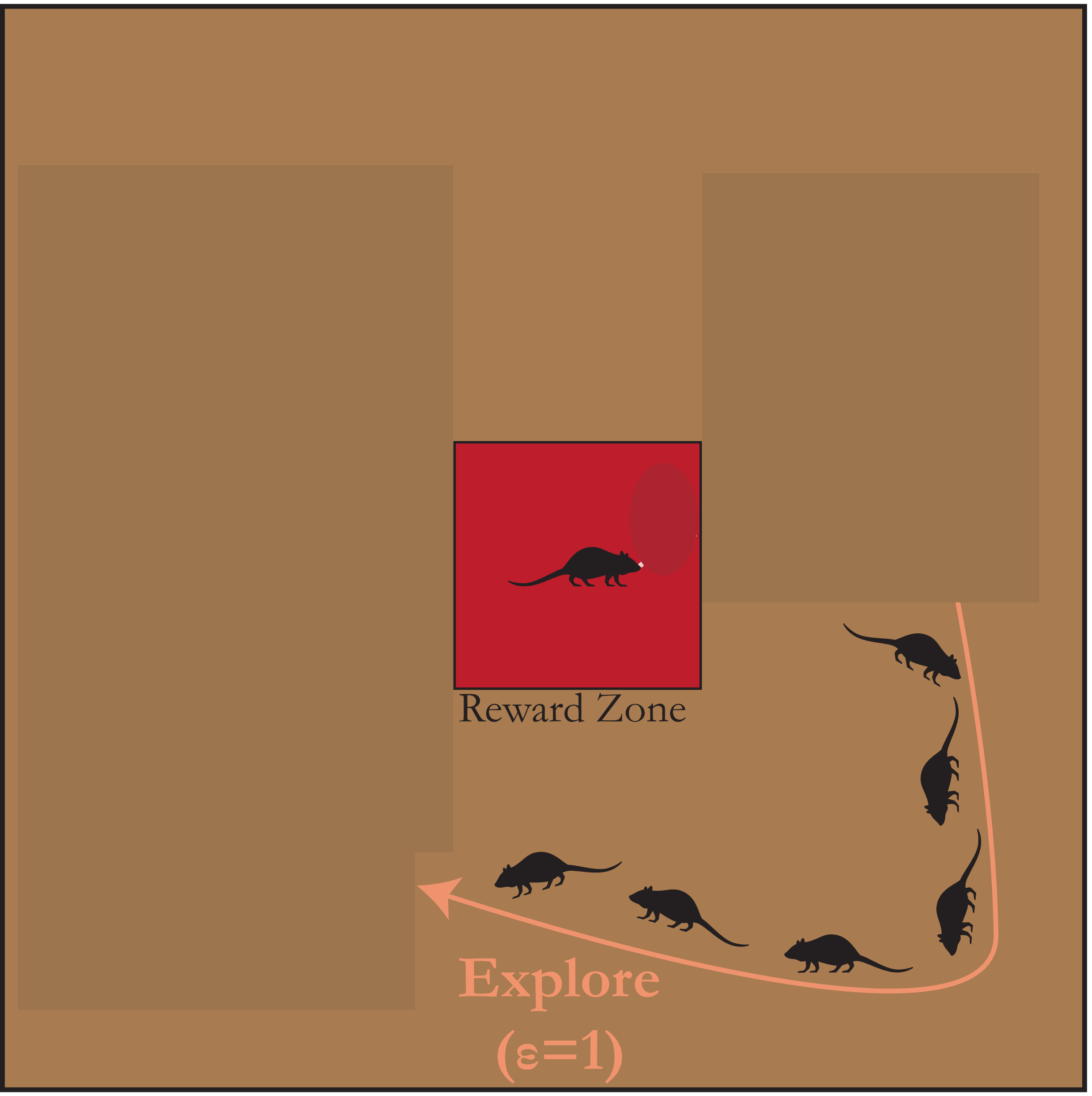
Modeling rewards - What we have done previously

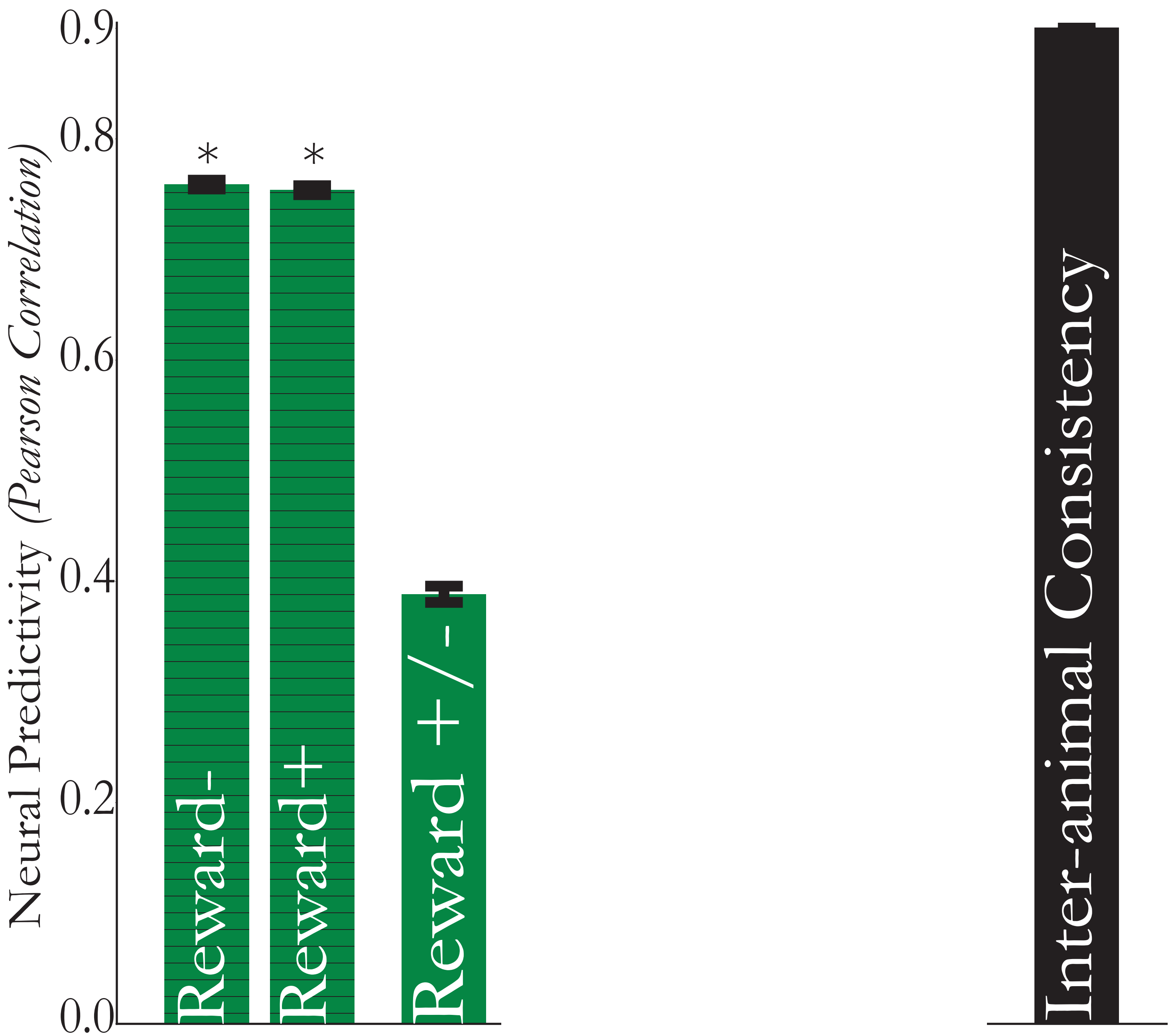


Exploration only model captures each condition *separately*



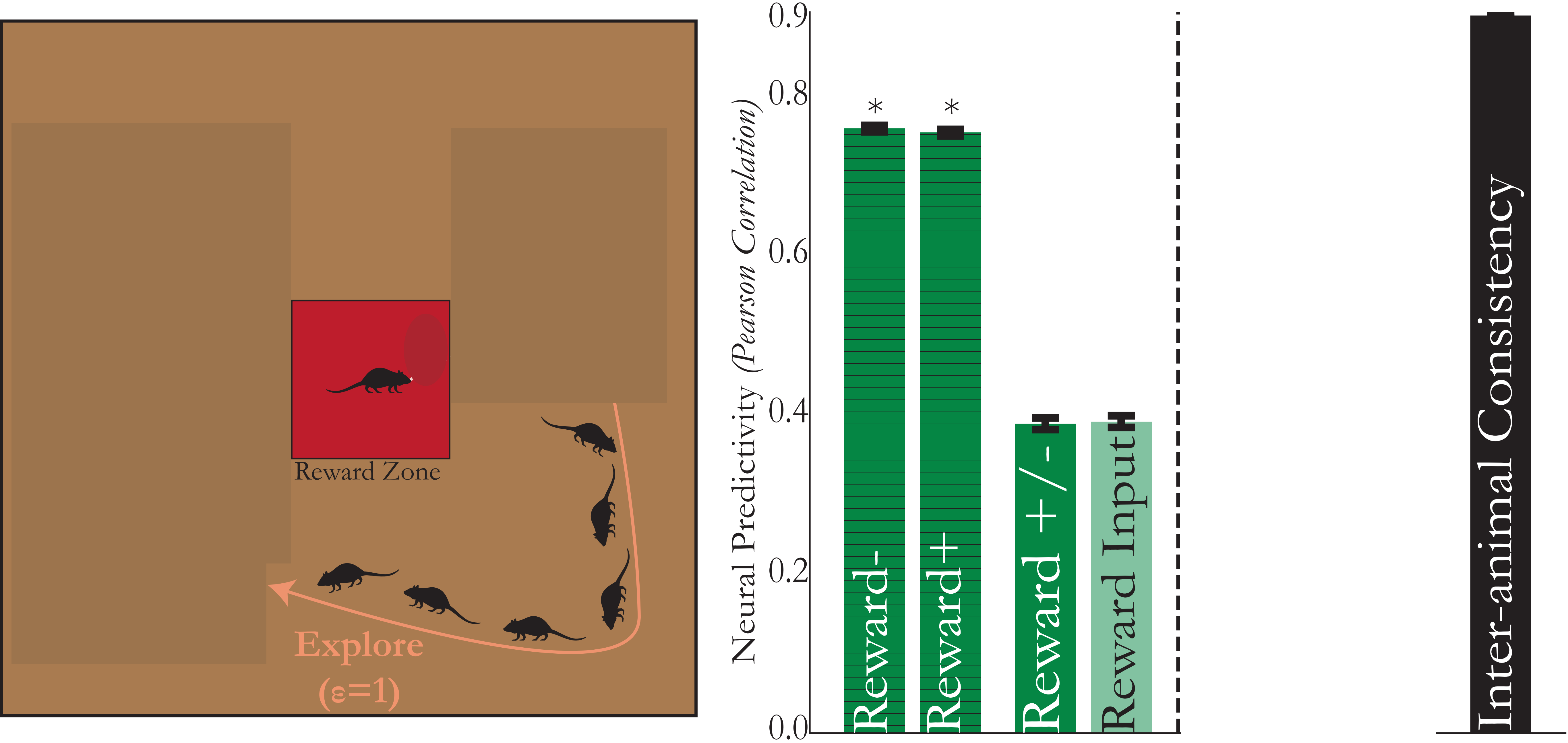
Exploration only model fails to capture remapping





Failure of pure exploration!

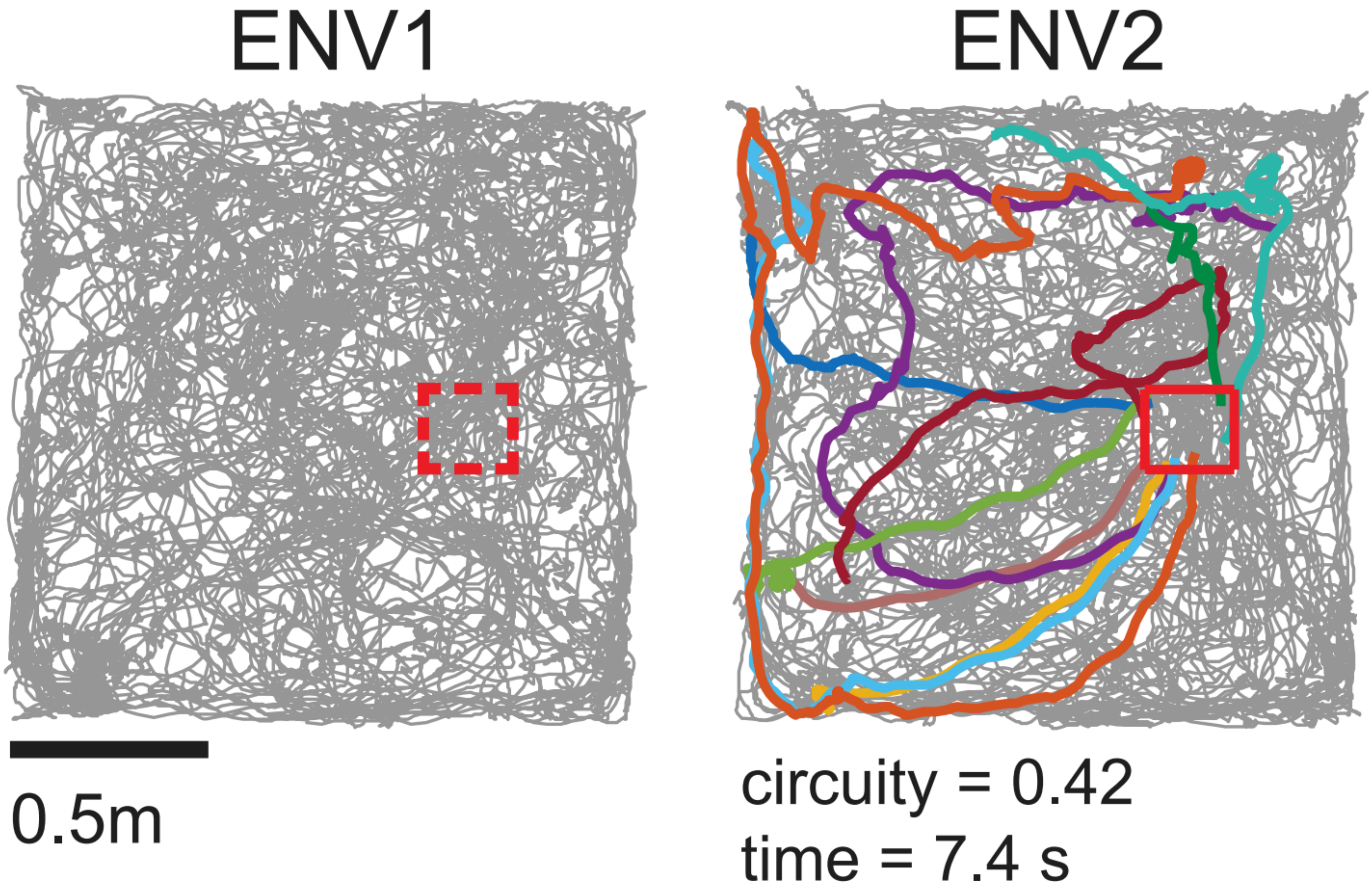
Reward must be extrinsically modeled



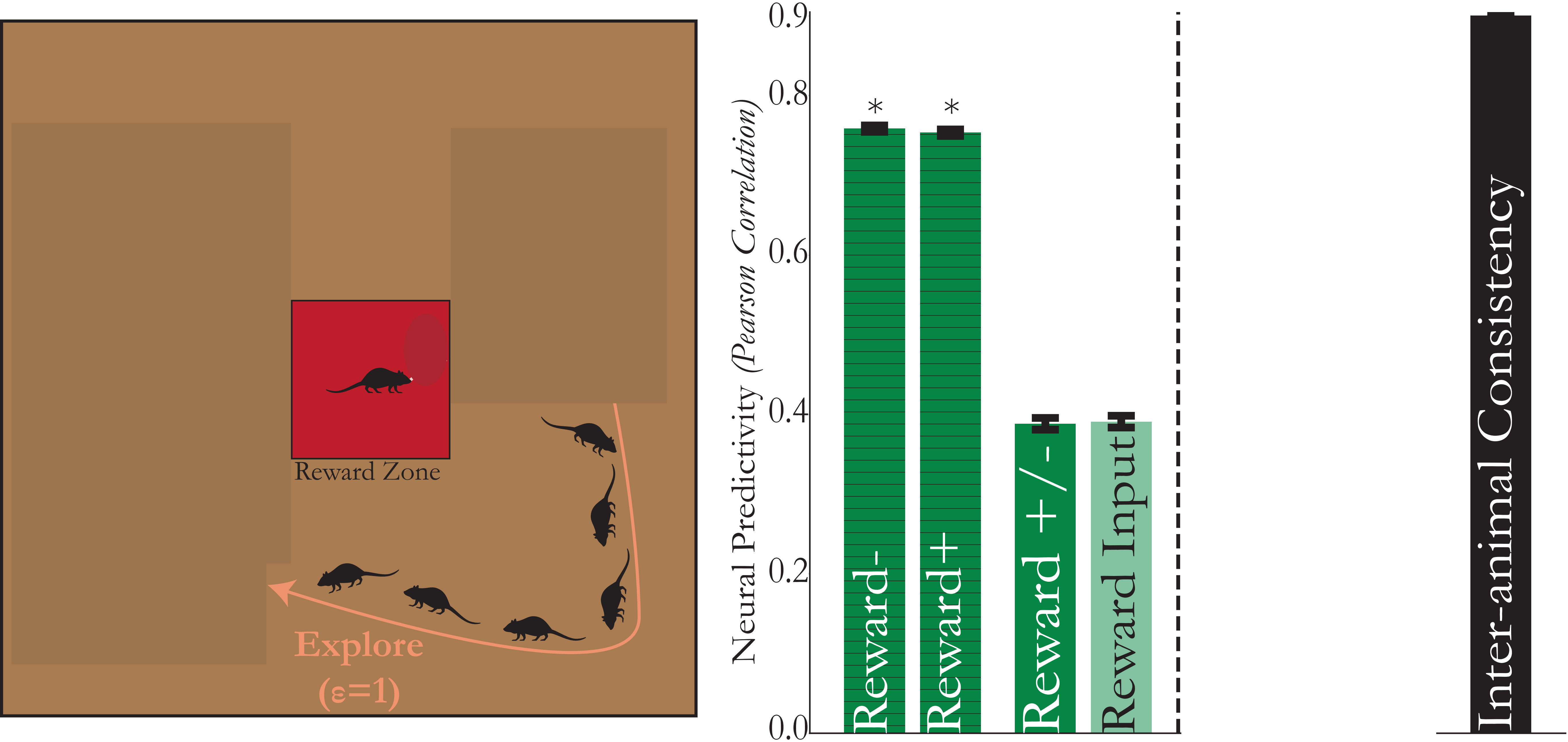
Simply augmenting inputs does not help either

Inspiration from animal behavior — rapid, direct paths

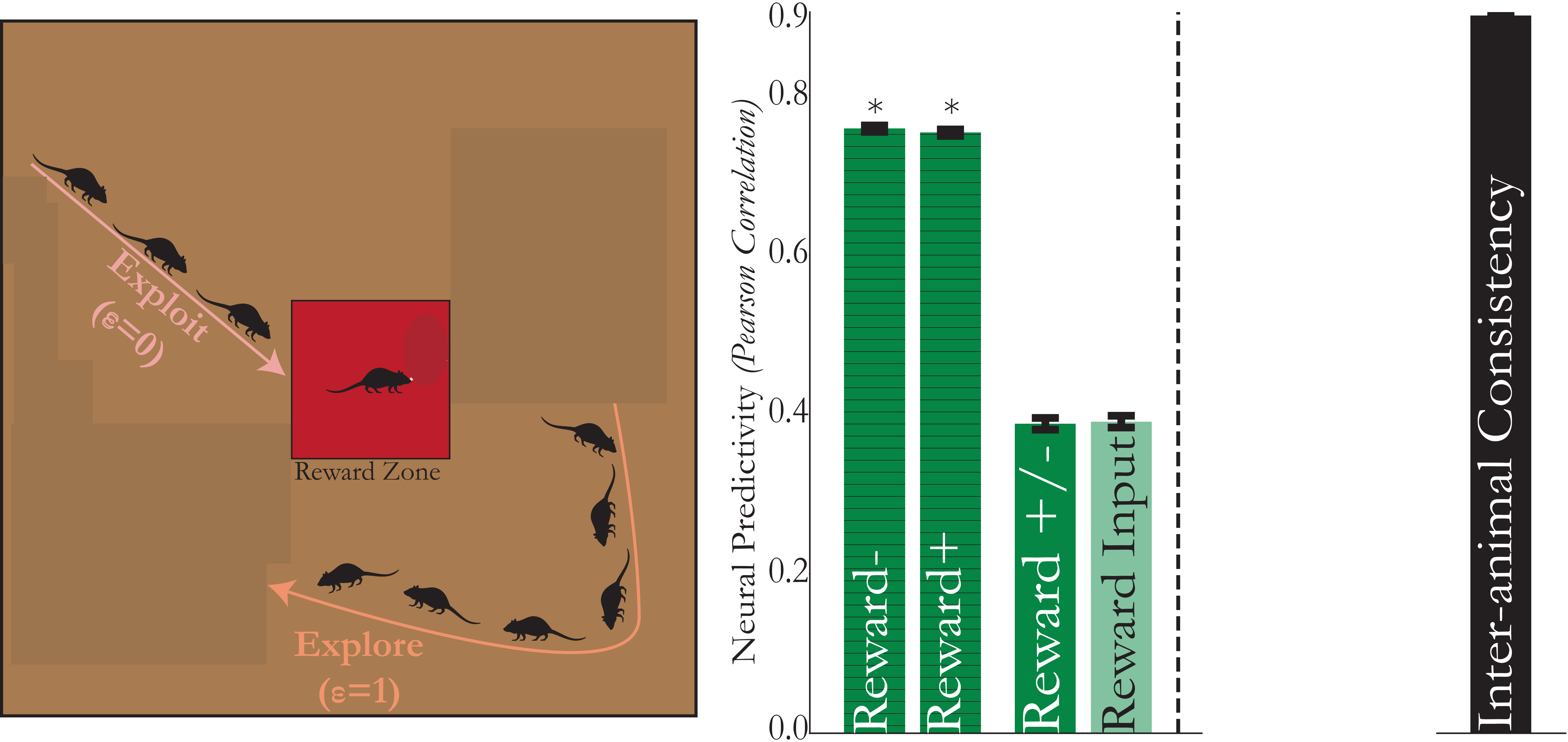
Animals tend to take rapid, direct paths to reward zone



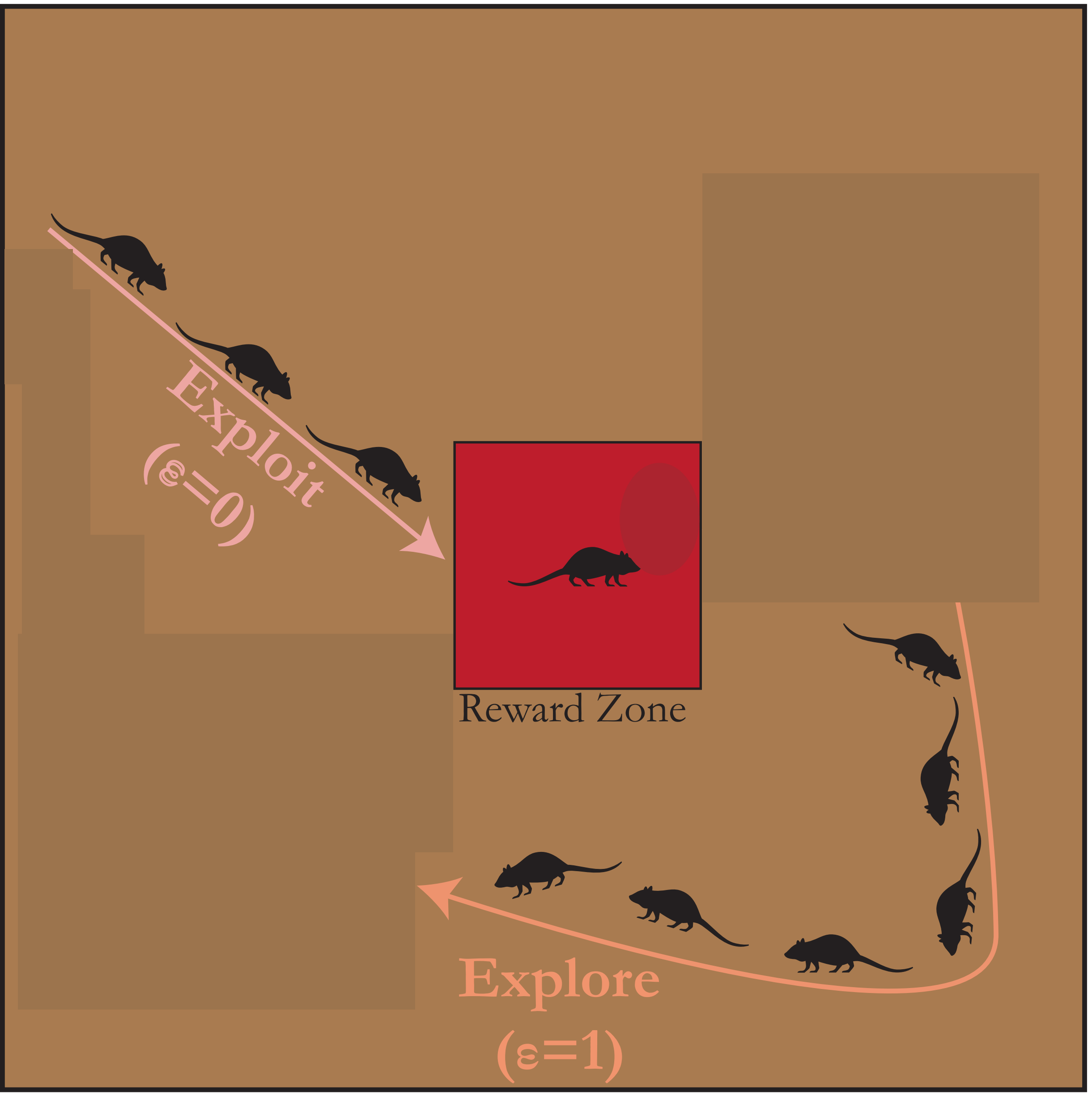
Reward must be extrinsically modeled

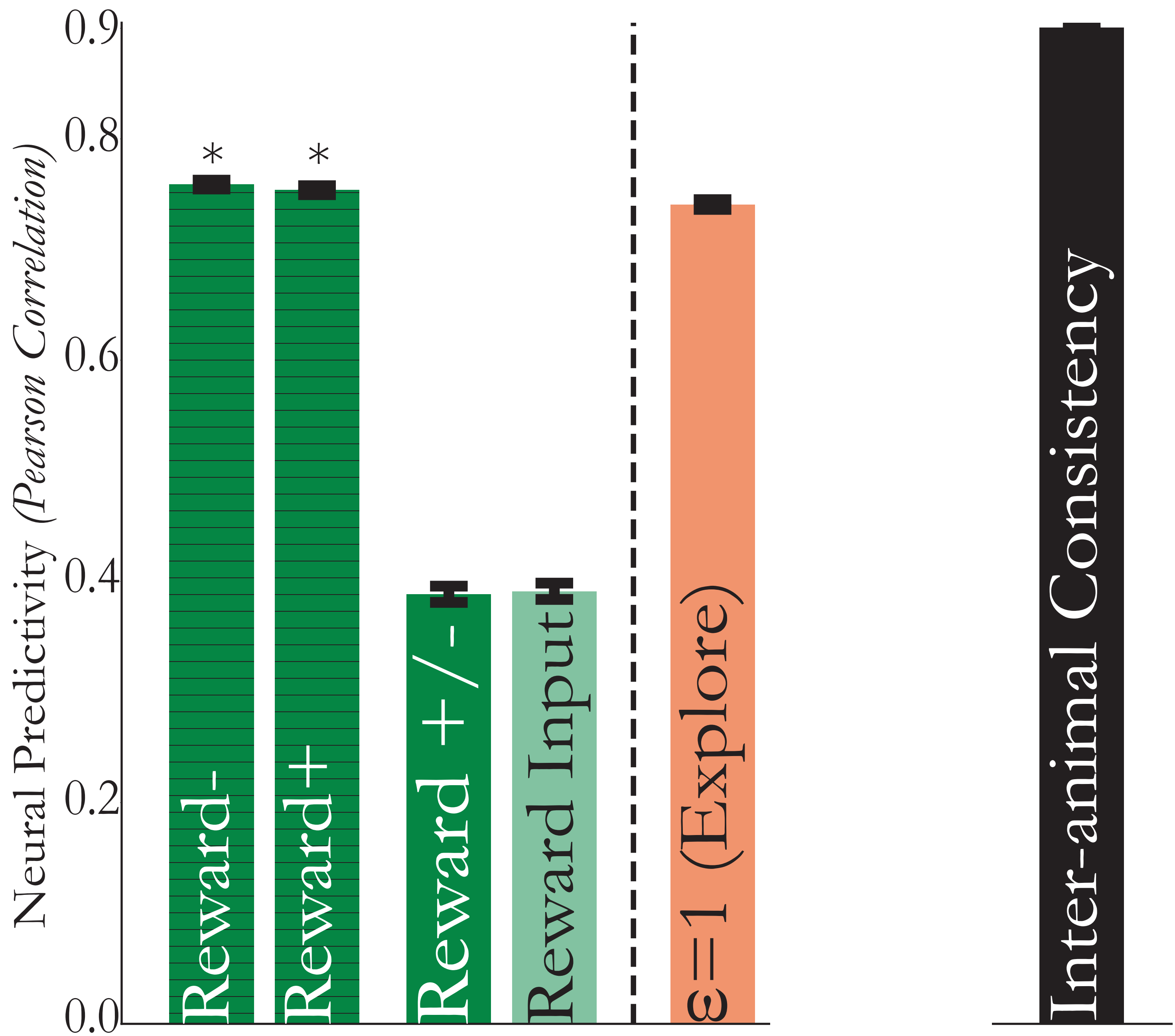


Modeling rewards as biased path integration

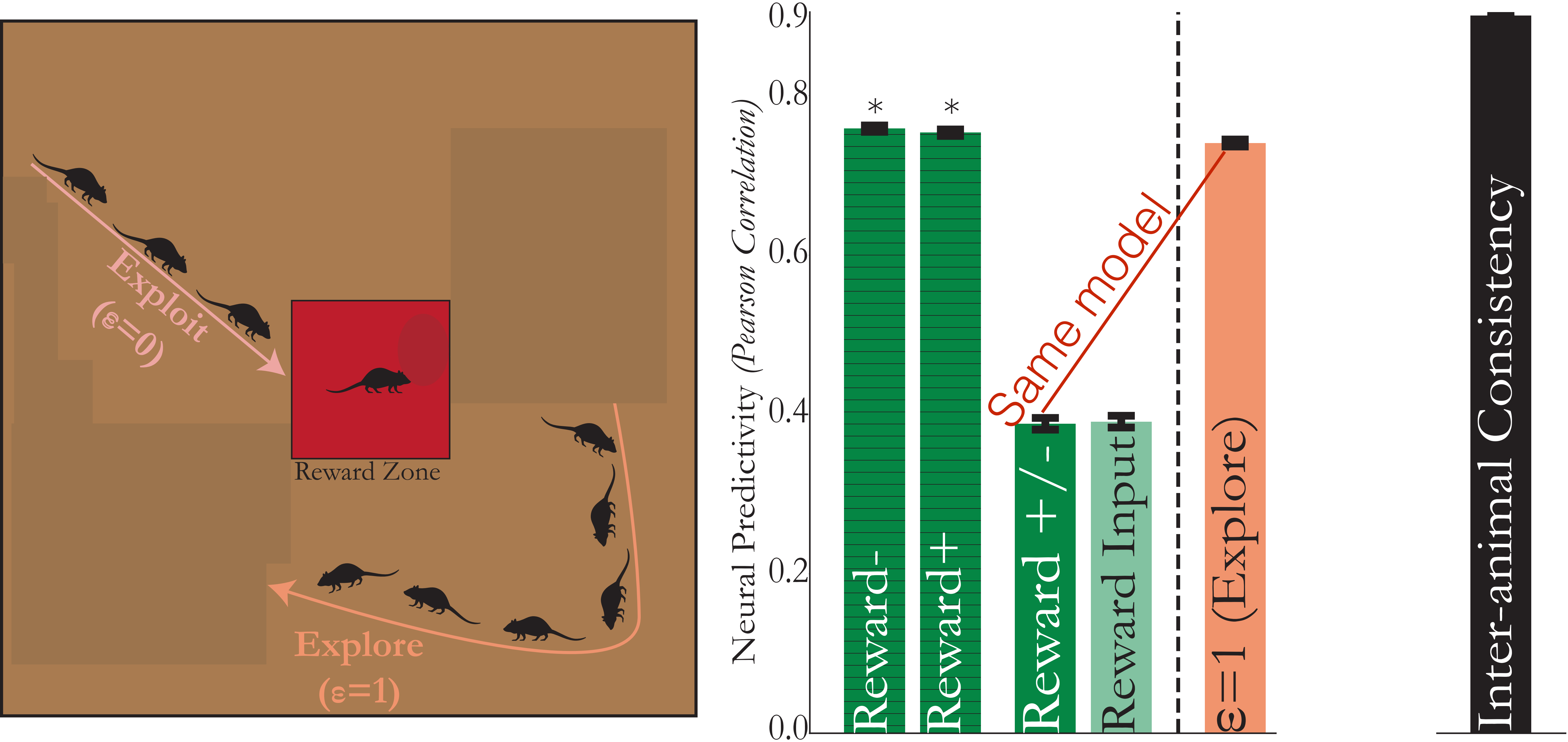


Modeling rewards as biased path integration



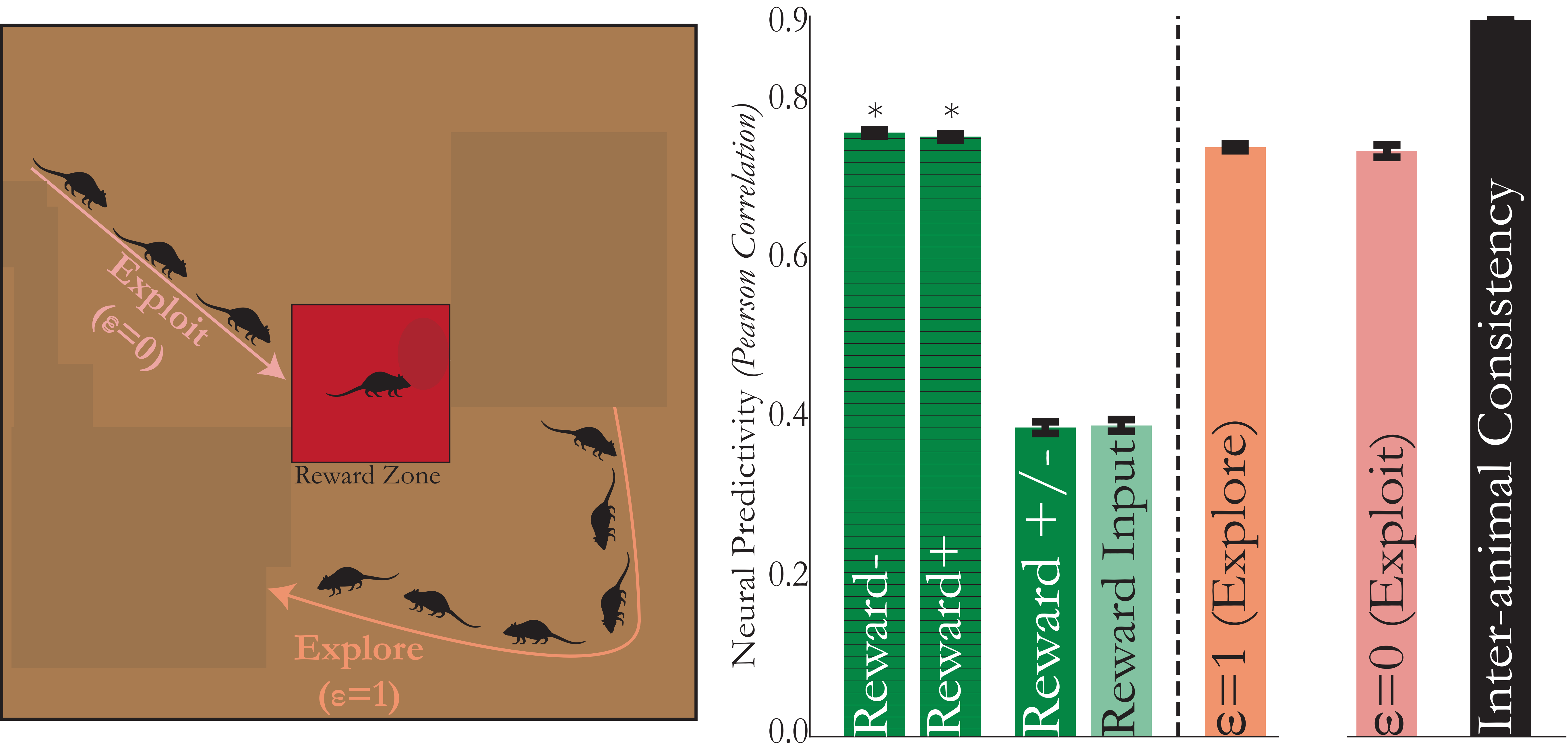


Modeling rewards as biased path integration

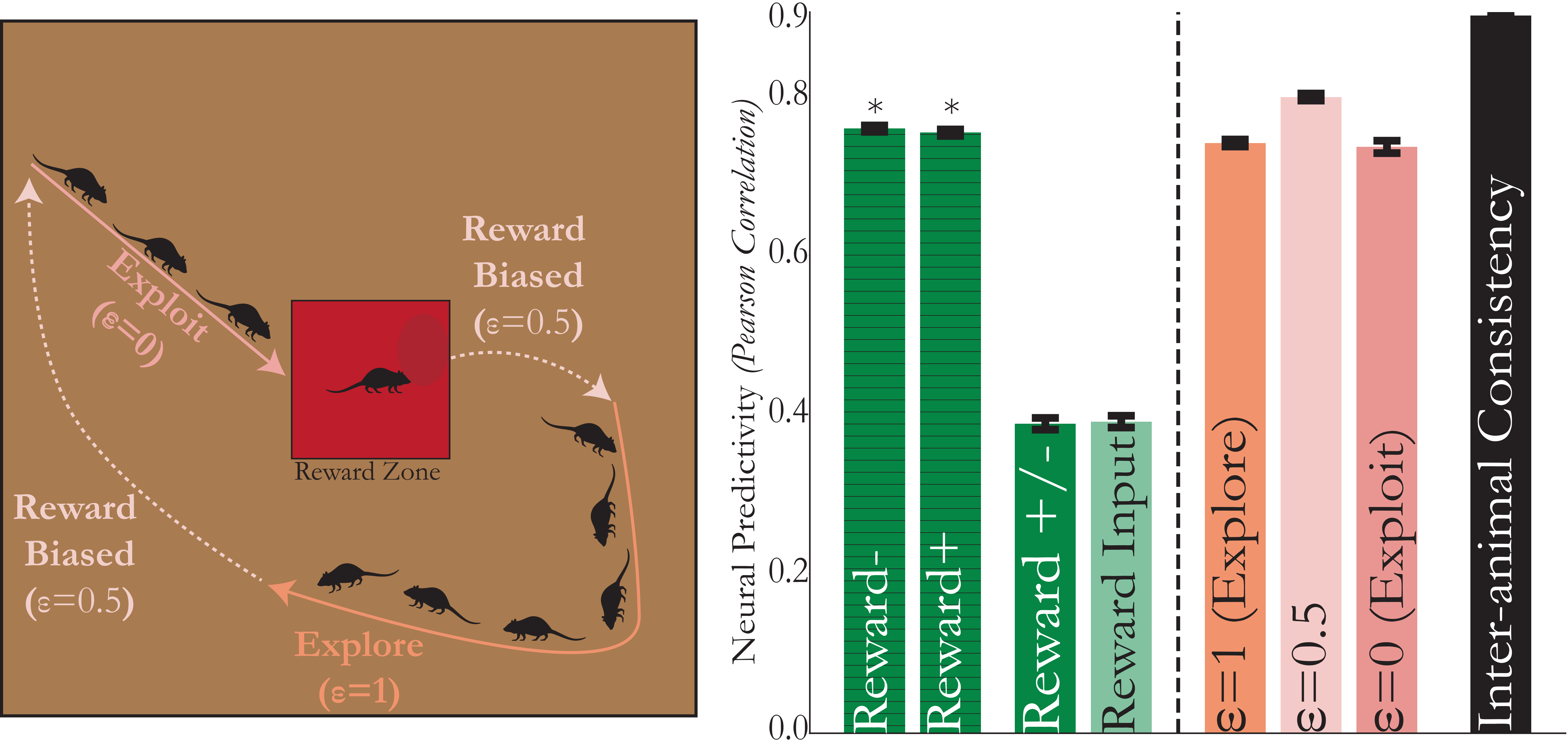


Reward remapping strongly input driven!

Pure exploitation isn't any better



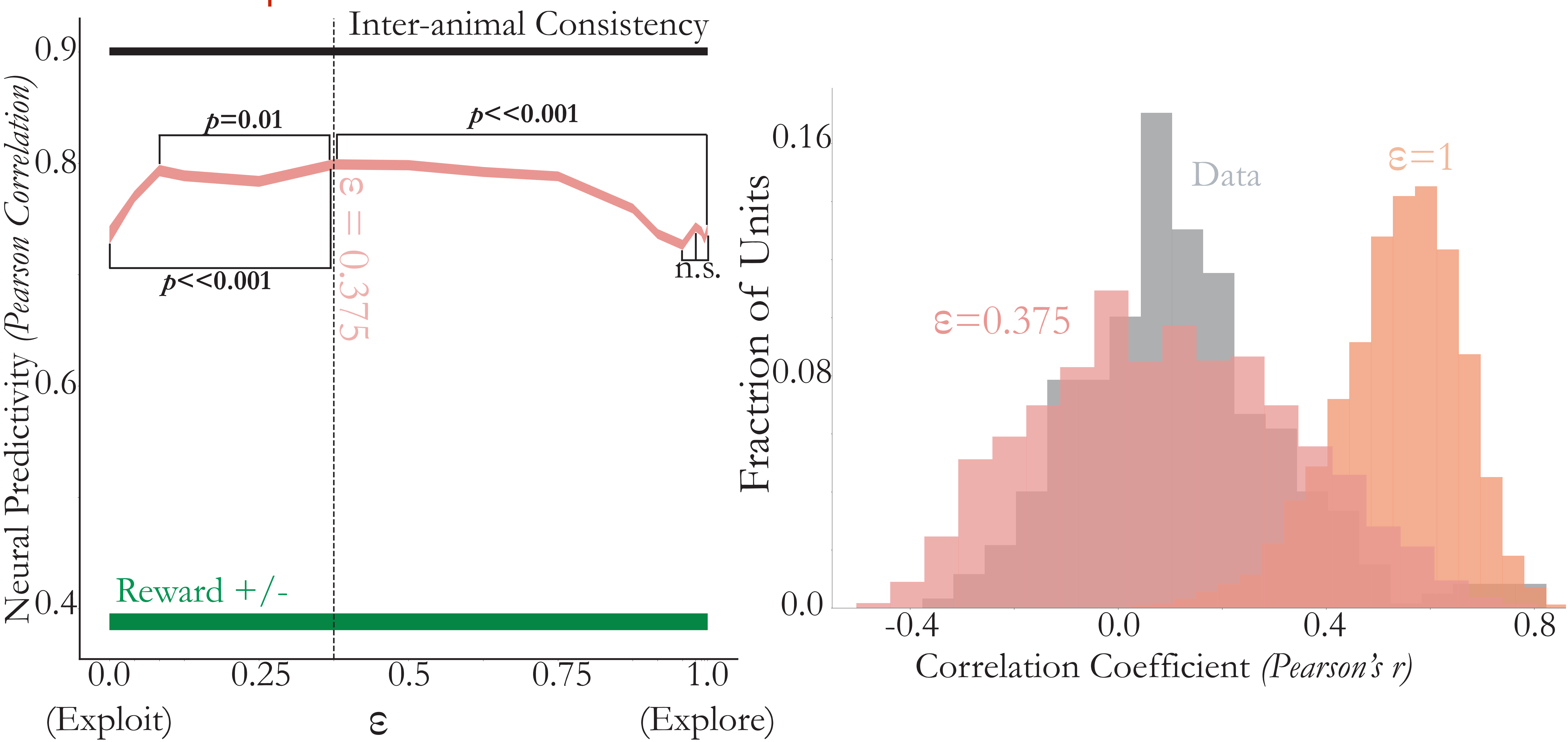
Reward-biased path integration captures remapping of responses in the presence of reward



Reward-biased path integrator best captures remapping

Reward-biased path integration captures remapping of responses in the presence of reward

Slight bias to exploitation preferred



Reward-biased path integrator best captures remapping

Main Conclusions

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

3. *Task Differentiation:* Navigational task training loss gives you higher correlation than NMF loss, especially for the non-grid like units. Intermediate Place Cell representation is important.

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

3. <u>*Task Differentiation:*</u> Navigational task training loss gives you higher correlation than NMF loss, especially for the non-grid like units. Intermediate Place Cell representation is important.

4. <u>*Circuit Differentiation:*</u> UGRNN ReLU gives the best match overall, and approaches the inter-animal consistency even when trained in a different environment.

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

3. *Task Differentiation:* Navigational task training loss gives you higher correlation than NMF loss, especially for the non-grid like units. Intermediate Place Cell representation is important.

4. *Circuit Differentiation:* UGRNN ReLU gives the best match overall, and approaches the inter-animal consistency even when trained in a different environment.

5. Non-spatial **rewards** can be accounted for in the *same* path integration framework (and are very input-driven).

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

3. *Task Differentiation:* Navigational task training loss gives you higher correlation than NMF loss, especially for the non-grid like units. Intermediate Place Cell representation is important.

4. *Circuit Differentiation:* UGRNN ReLU gives the best match overall, and approaches the inter-animal consistency even when trained in a different environment.

5. Non-spatial **rewards** can be accounted for in the *same* path integration framework (and are very input-driven).

Overall Conclusion: A process of biological performance optimization directly shaped the neural mechanisms in MEC as a whole *(normative explanation for grid & non-grid cells alike)*.

Main Conclusions

1. **Heterogeneous cells are reliable:** Animals can explain each other quite well, but under a suitably chosen transform class (ridge regression)

Modeling conclusions (under transform class):

2. Classic theoretical model does not quantifiably explain all of the data: NMF, (dimensionality reduction on simulated place cells) is very far from the inter-animal consistency.

3. *Task Differentiation:* Navigational task training loss gives you higher correlation than NMF loss, especially for the non-grid like units. Intermediate Place Cell representation is important.

4. *Circuit Differentiation:* UGRNN ReLU gives the best match overall, and approaches the inter-animal consistency even when trained in a different environment.

5. Non-spatial **rewards** can be accounted for in the *same* path integration framework (and are very input-driven).

***NeurIPS 2021* Paper:** https://www.biorxiv.org/content/10.1101/2021.10.30.466617
Pretrained Models & Neural Fitting Pipeline: https://github.com/neuroailab/mec

Acknowledgments


Alexander Attinger


Malcolm G. Campbell


Kiah Hardcastle


Isabel I.C. Low


Caitlin S. Mallory


Gabriel C. Mel


Ben Sorscher


Alex H. Williams


Surya Ganguli


Lisa M. Giocomo


Daniel L.K. Yamins