Interpreting a Deep Reinforcement Learning Model of Sensory-Place Association

Juan Mendez *Andrea Pierré*

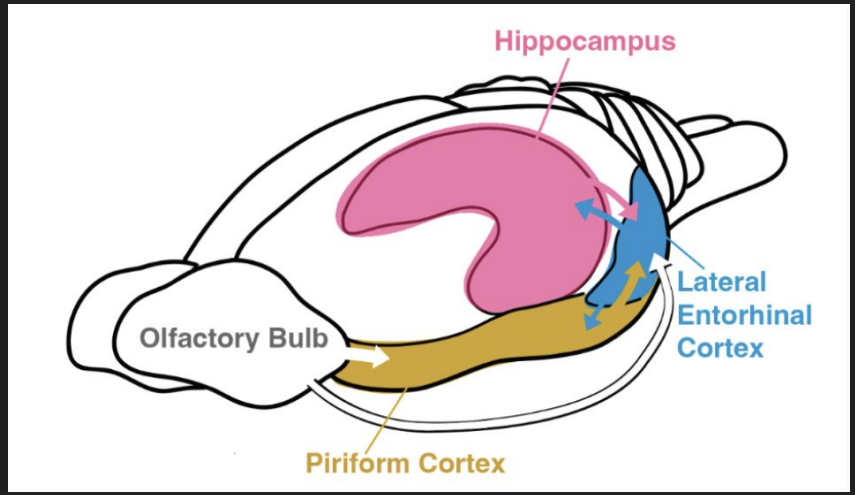
Jason Ritt *Alexander Fleischmann*

Contents

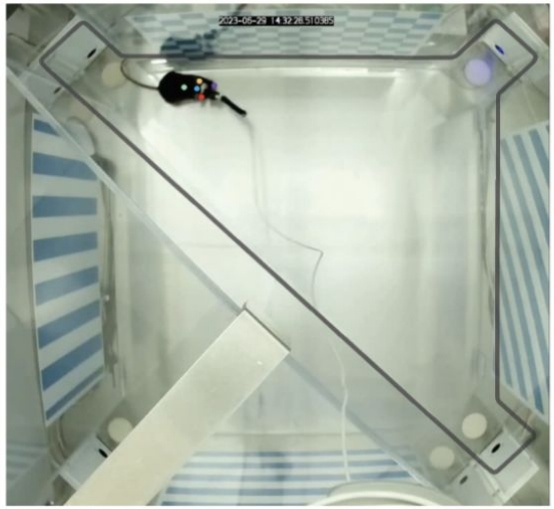
Introduction

Sensory-Place Association

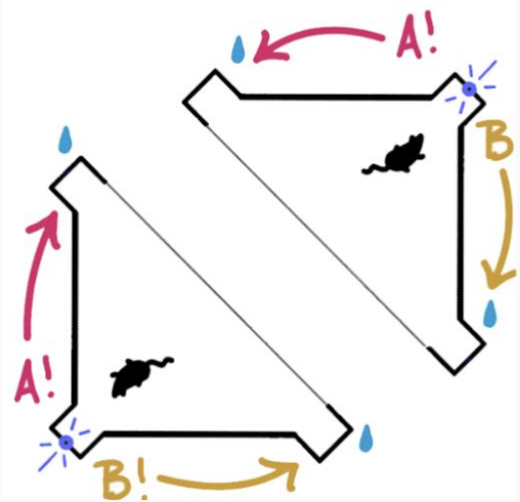
- The piriform cortex encodes olfactory information
- The hippocampus encodes spatial information
- The lateral entorhinal cortex encodes both olfactory and spatial information
- Associations between odor and space are thought to be supported by conjunctive neurons



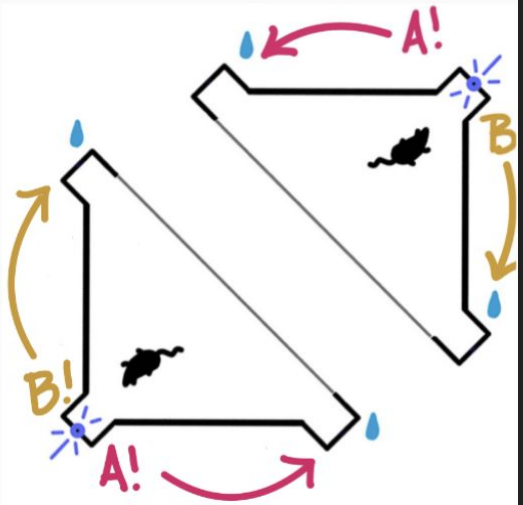
Learning Task for Odor-Place Association



East/West task



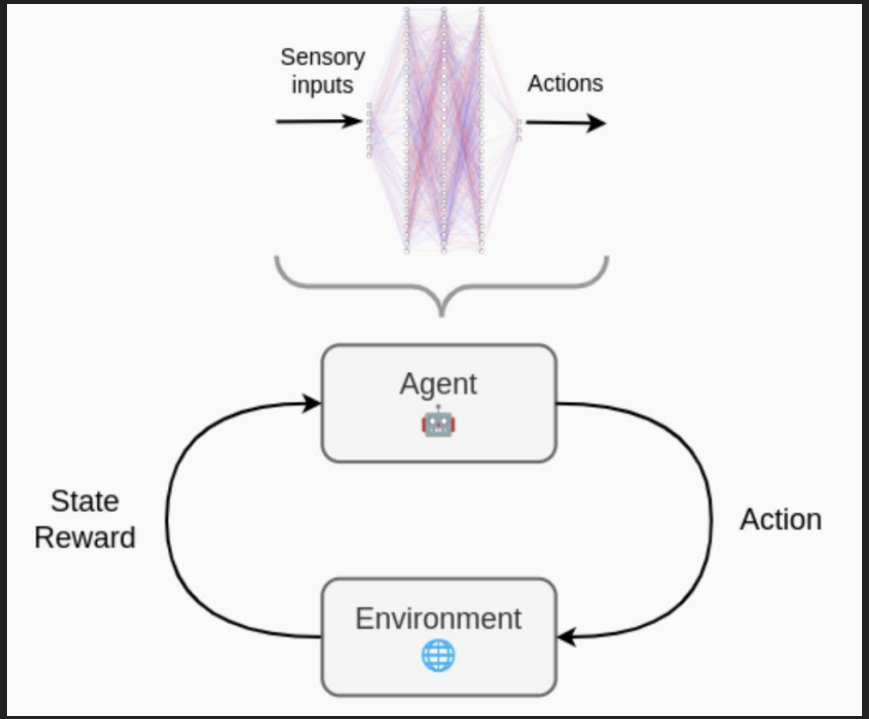
Left/Right task



Deep Reinforcement Learning (DRL)

- Reinforcement Learning: Machine learning where an agent learns to make decisions by interacting with an environment and maximizing reward

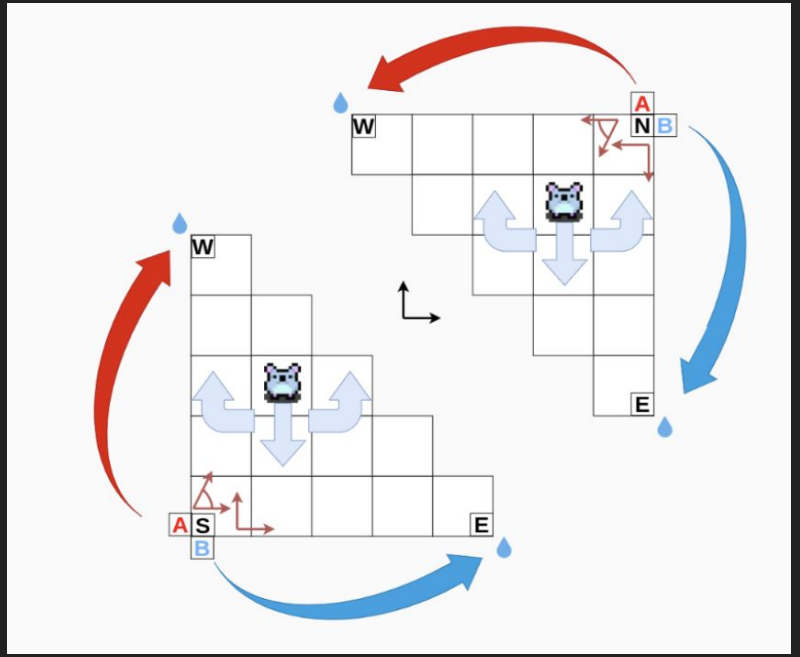
- Can be implemented with deep neural networks, which are used to learn what actions lead to higher reward



Motivation

- If we implement the mouse learning task in an analogous DRL task, can the digital agent effectively learn odor-place association?
  - Yes - Andrea's work

- *How* does the DRL agent learn the odor-place association task? Can we make any relations to the biological brain?



Motivation

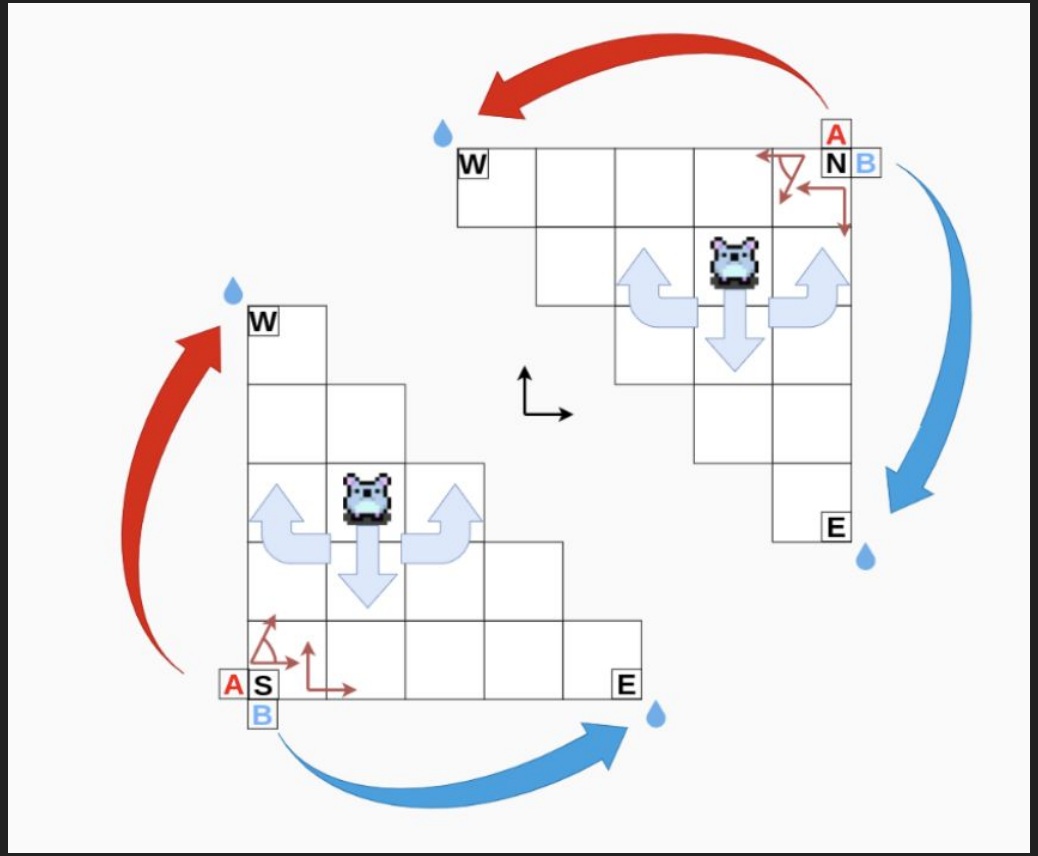
Can we find unintuitive ways in which the digital agent solves the task?

Can that inspire experimental questions for studying learning in the mouse brain?

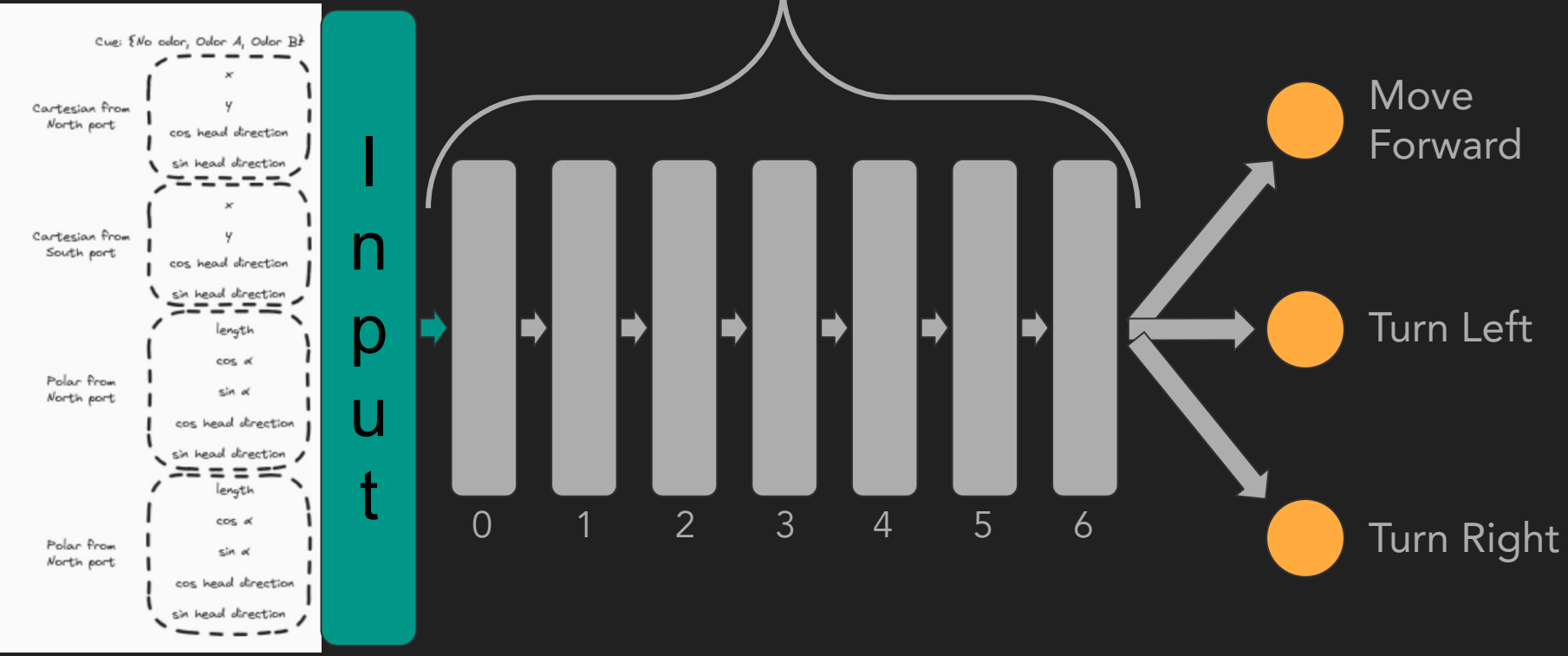
Does the digital agent show preference for learning allocentric vs. egocentric representations?

What are systematic ways to gain interpretability into deep reinforcement learning models?

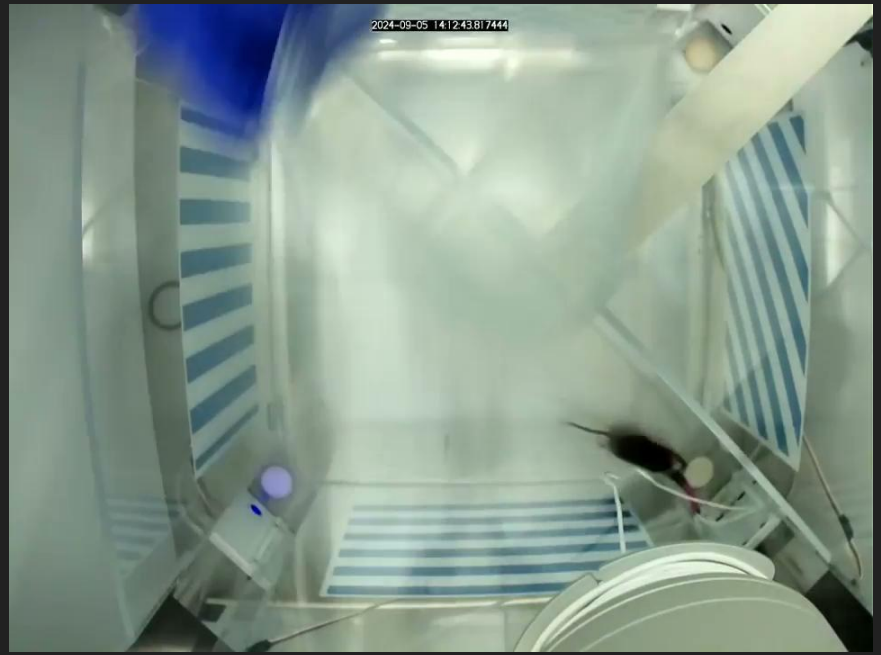
The Environment



The Model


Hidden Layers - 512 nodes each
Cue: {No odor, Odor A, Odor B}
Cartesian from North port
x
y
cos head direction
sin head direction
Cartesian from South port
x
y
cos head direction
sin head direction
Polar from North port
length
cos α
sin α
cos head direction
sin head direction
Polar from North port
length
cos α
sin α
cos head direction
sin head direction
Input
0
1
2
3
4
5
6
Move Forward
Turn Left
Turn Right

Model Video



Model Video - Expert

Questions & Hypotheses

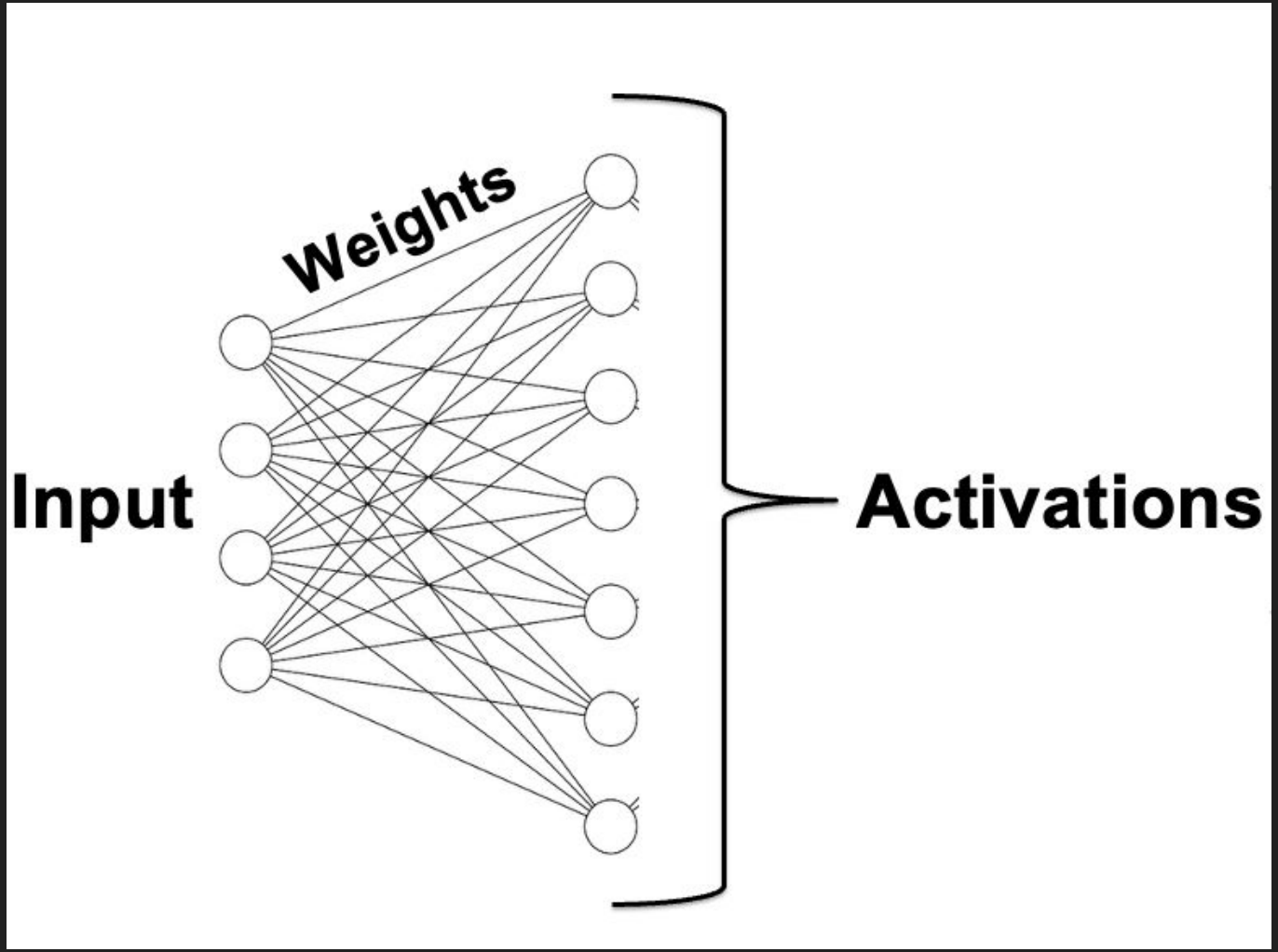
Q: What internal representations is the network learning to solve the task?

H: The network supports *conjunctive activations*; that is, nodes that activate only at a specific grid position with a specific odor

Q: What coordinate system does the network prefer when solving EastWest vs. LeftRight?

H: Cartesian (allocentric) coordinates are preferred for EastWest, while Polar (egocentric) coordinates are preferred for LeftRight

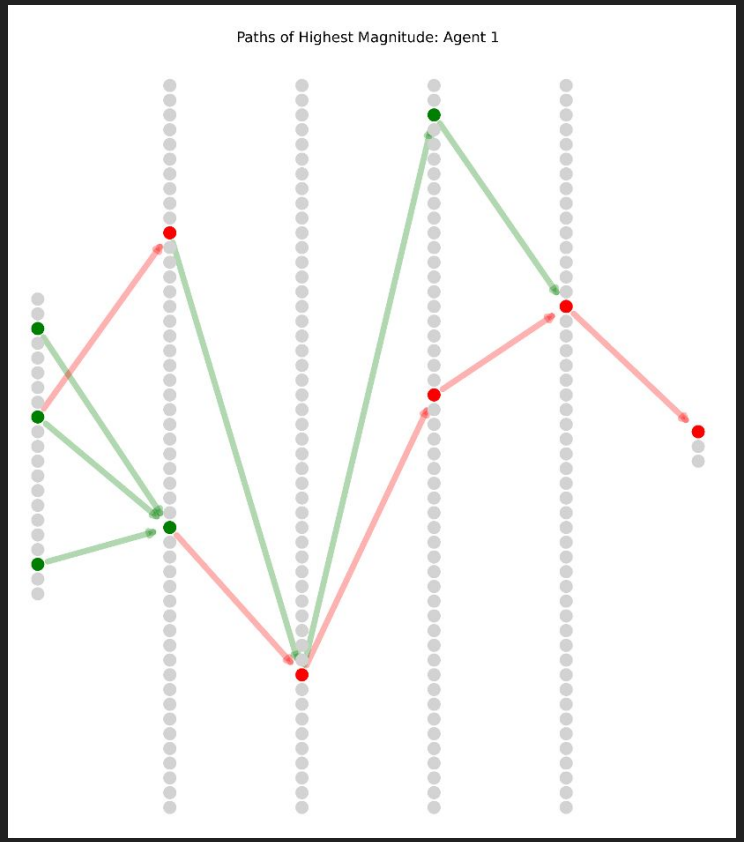
Results


Weights
Input
Activations

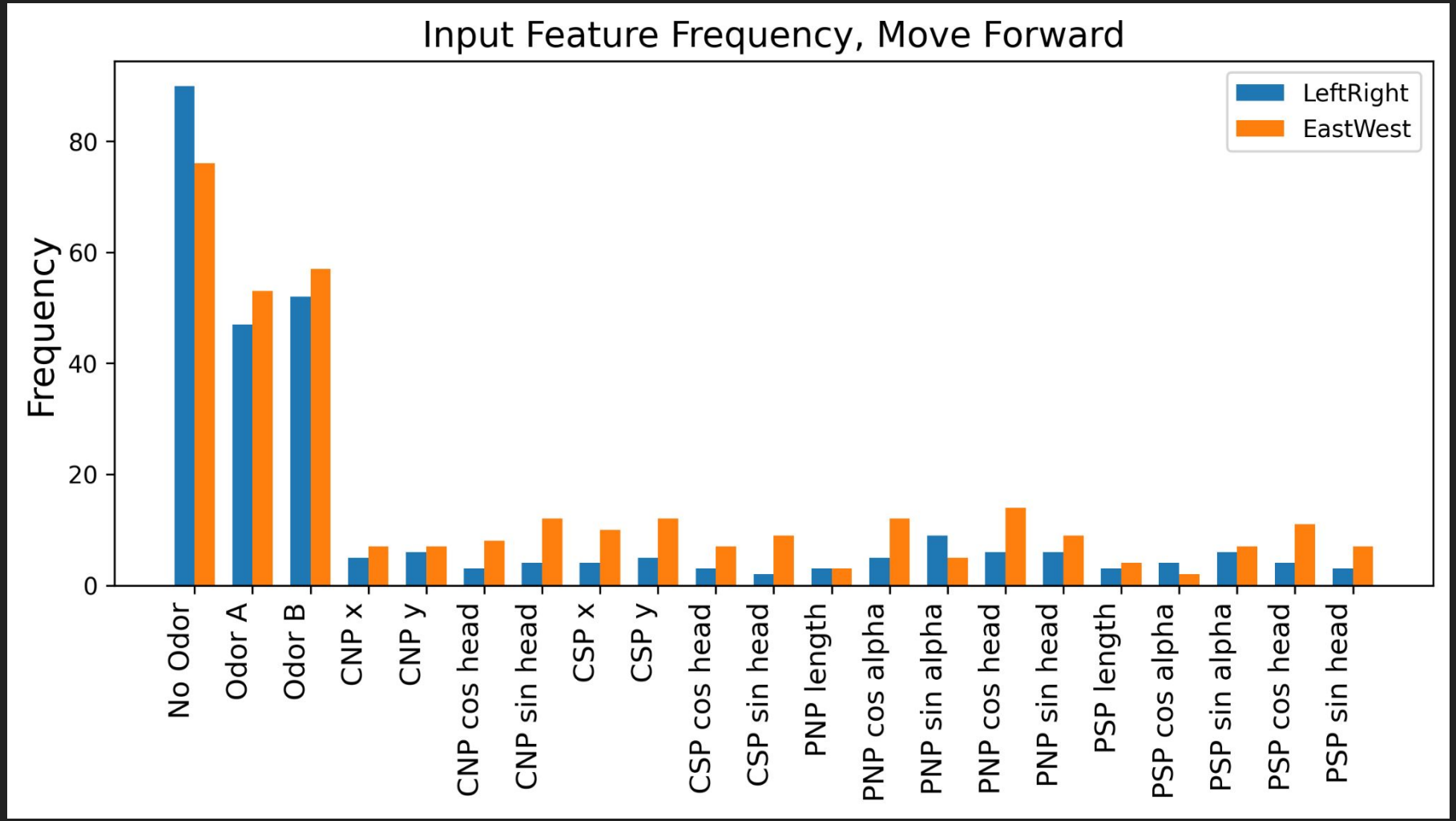
Model Weights

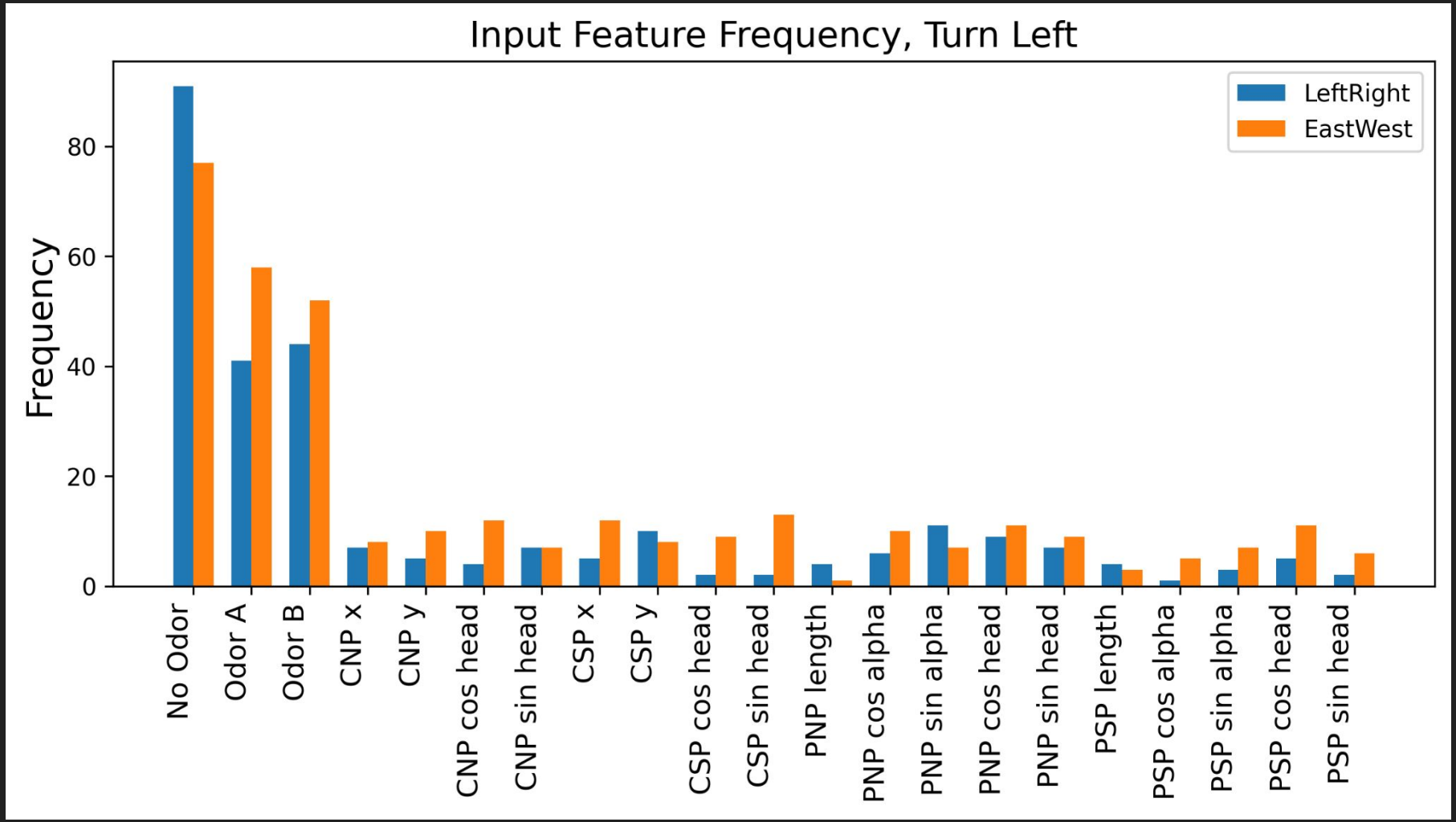
Strongest Weight Paths

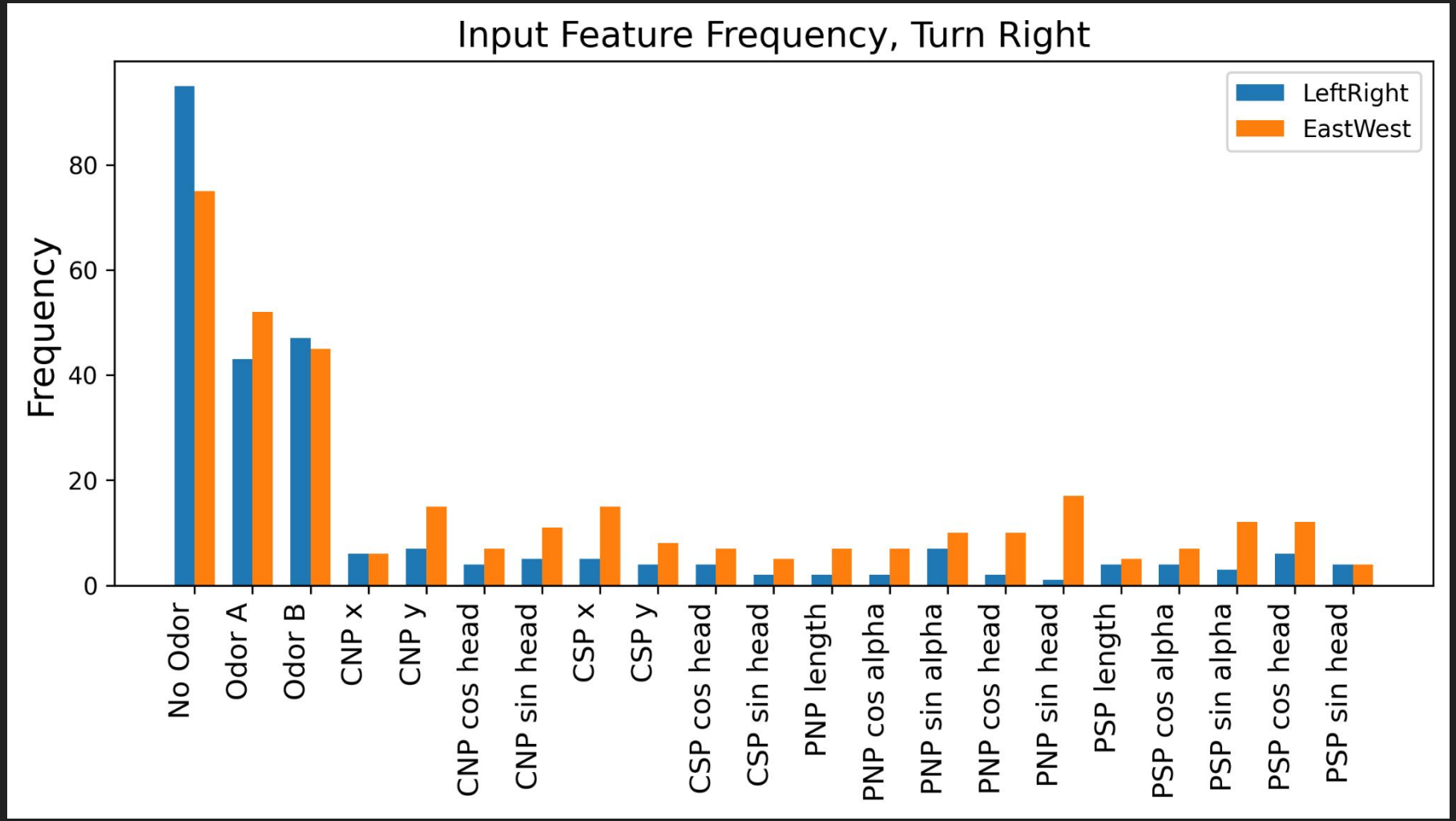
Strongest Weight Paths



- Every node has a vector of weights
  - Weights contribute to activations in the next layer
- We can look at *paths of high magnitude* – that is, paths of nodes with strong weight
- These paths can help tell us which nodes strongly modulate certain activations in the next layer


Input Feature Frequency, Move Forward
LeftRight
EastWest
Frequency
0
20
40
60
80
No Odor
Odor A
Odor B
CNP x
CNP y
CNP cos head
CNP sin head
CSP x
CSP y
CSP cos head
CSP sin head
PNP length
PNP cos alpha
PNP sin alpha
PNP cos head
PNP sin head
PSP length
PSP cos alpha
PSP sin alpha
PSP cos head
PSP sin head


Input Feature Frequency, Turn Left
LeftRight
EastWest
Frequency
0
20
40
60
80
No Odor
Odor A
Odor B
CNP x
CNP y
CNP cos head
CNP sin head
CSP x
CSP y
CSP cos head
CSP sin head
PNP length
PNP cos alpha
PNP sin alpha
PNP cos head
PNP sin head
PSP length
PSP cos alpha
PSP sin alpha
PSP cos head
PSP sin head


Input Feature Frequency, Turn Right
LeftRight
EastWest
Frequency
0
20
40
60
80
No Odor
Odor A
Odor B
CNP x
CNP y
CNP cos head
CNP sin head
CSP x
CSP y
CSP cos head
CSP sin head
PNP length
PNP cos alpha
PNP sin alpha
PNP cos head
PNP sin head
PSP length
PSP cos alpha
PSP sin alpha
PSP cos head
PSP sin head

Observations and Limitations

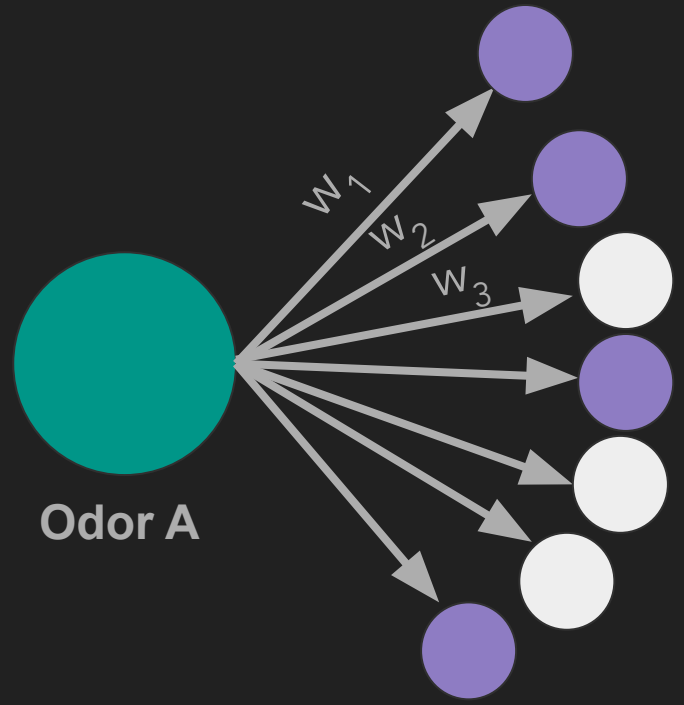
- Odor Cue is most frequently included in strongest weight paths throughout the network
- Odor may be the most salient input feature carried through from the input layer
    - Specifically No Odor

- Odor is one-hot encoded; could that play a factor?
- Networks may aim to generalize information across many nodes; looking at individual paths significantly reduces information

Hierarchical Clustering

- Every input node has 512 weights projected out to the first hidden layer, corresponding to the 512 nodes in that layer

- We can run a clustering algorithm on nodes in the first hidden layer, telling us clusters of nodes that receive similar weight information from inputs

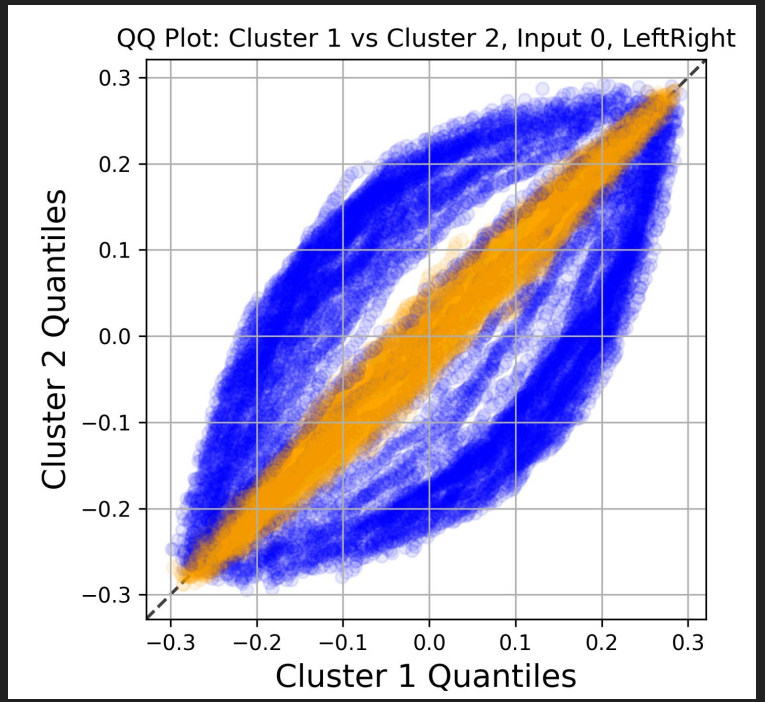


No Odor

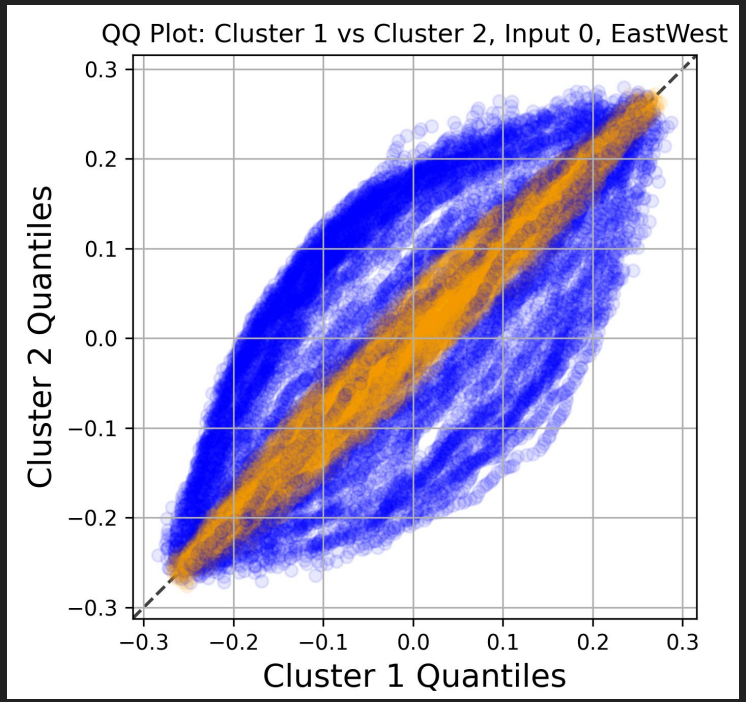
Observed Distribution

Null Distribution

LeftRight



EastWest

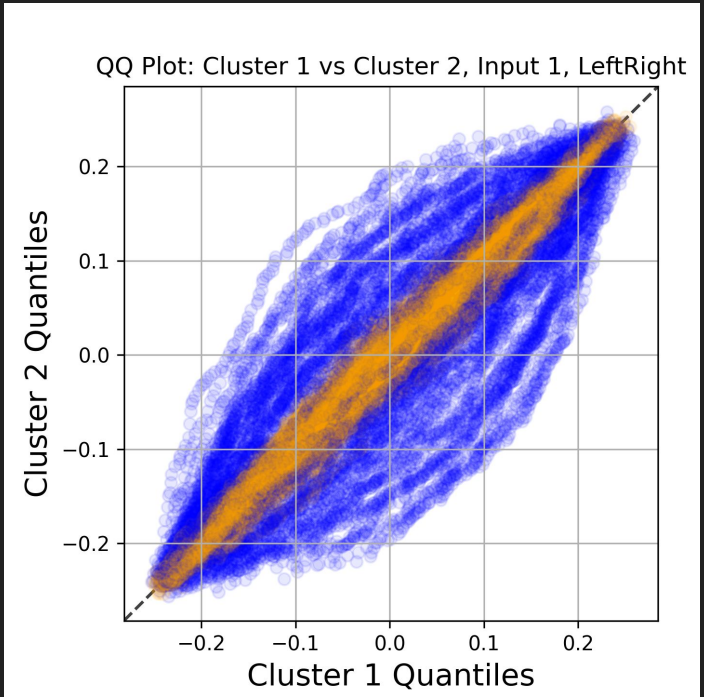


Odor A

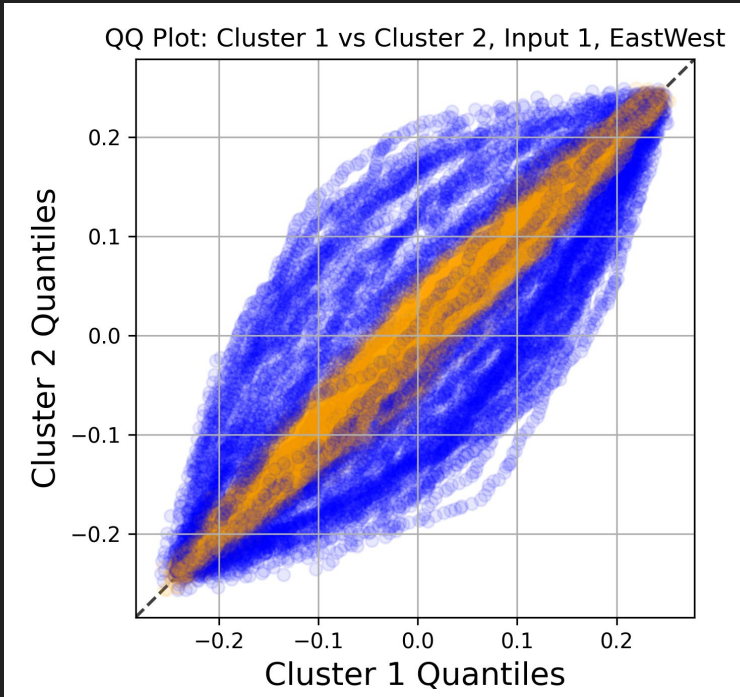
Observed Distribution

Null Distribution

LeftRight



EastWest

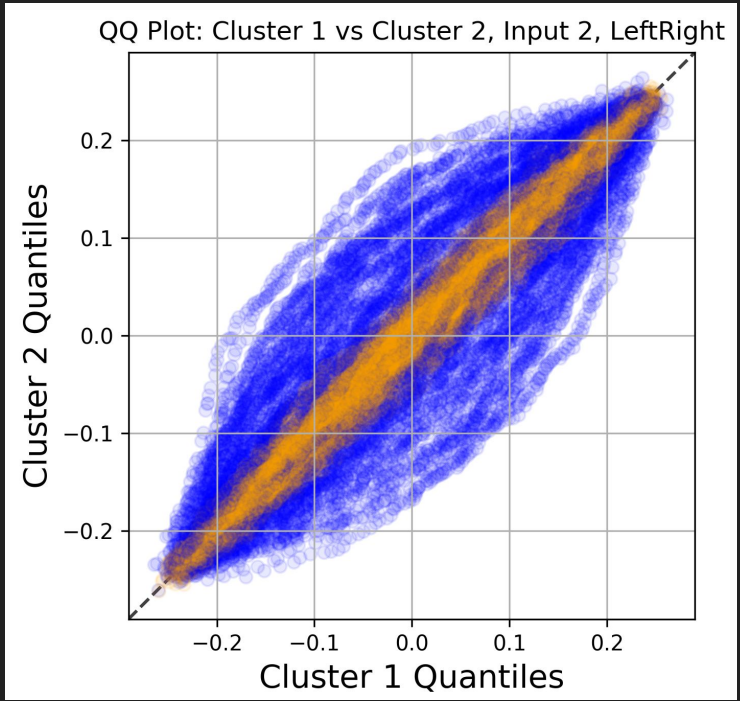


Odor B

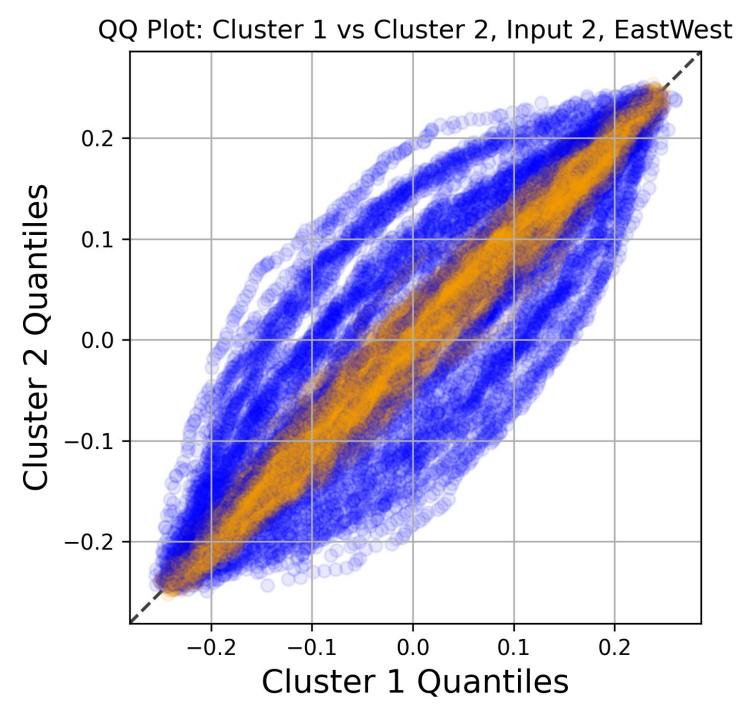
Observed Distribution

Null Distribution

LeftRight



EastWest



Observations and Limitations

- Using this clustering algorithm, the 512 nodes in the first hidden layer has one cluster that receives significantly different weight information from No Odor than the other cluster
  - Less significant for Odor A and B, and even less so for the other inputs
- The weights to the first hidden layer modulate activations according to the No Odor cue

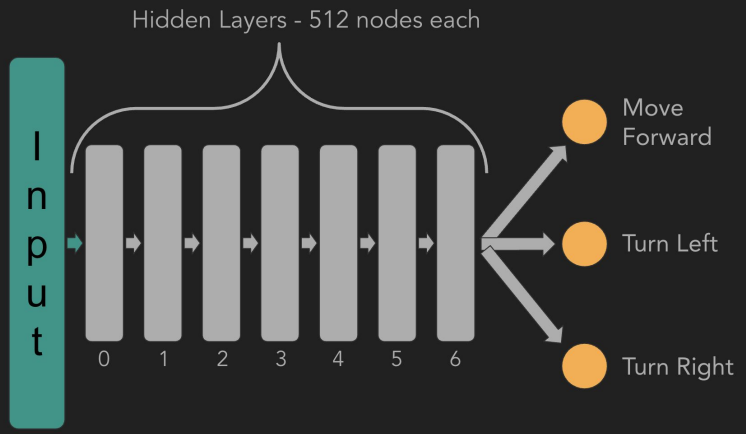
- How much can we trust typical clustering algorithms in this high-dimensional space?
- How much interpretation can we gain from *one* layer? How can we incorporate this with multiple?

Model Activations

PCA Plots

PCA

- We can extract the activations of a particular layer in our network as a vector
- These vectors are high-dimensional – 512 dimensions for 512 nodes
- In order to visualize activations, we can use dimensionality reduction techniques such as PCA, and observe any clustering



LeftRight Expert

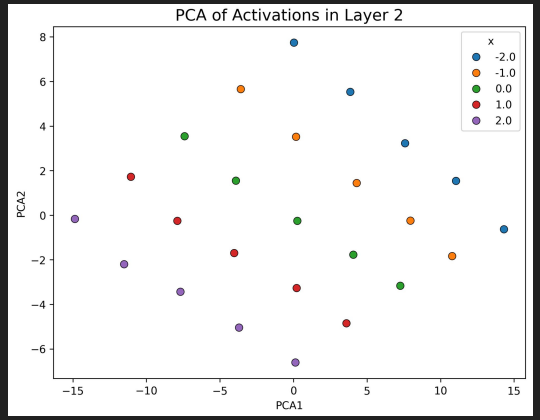
Filtering States

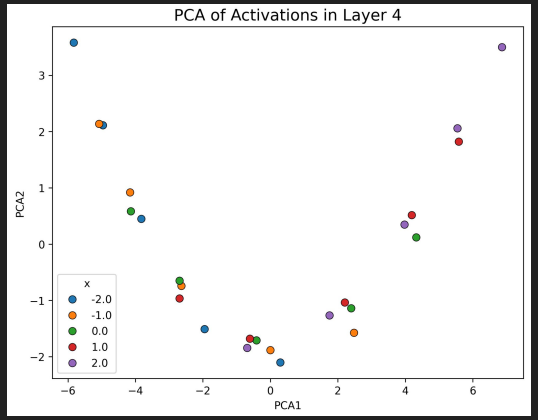
- Instead of recording activations for every possible state, we can filter down states to consider less information

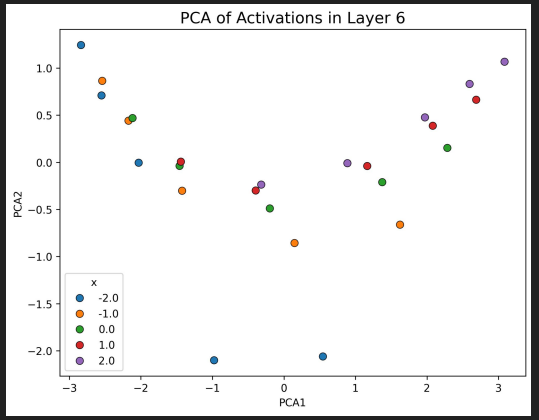
- For instance, what if we consider only states where the agent has smelled Odor A, and is facing North?

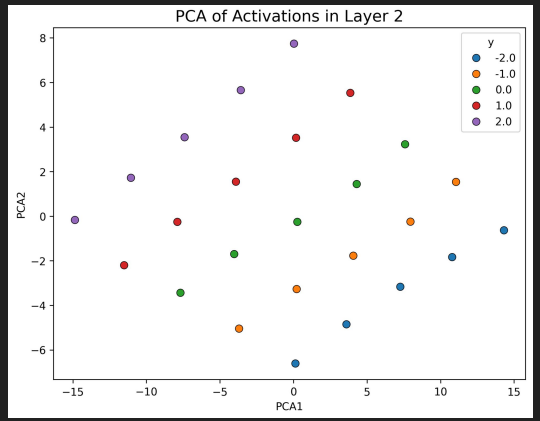
- 25 states; can visualize effectively with PCA

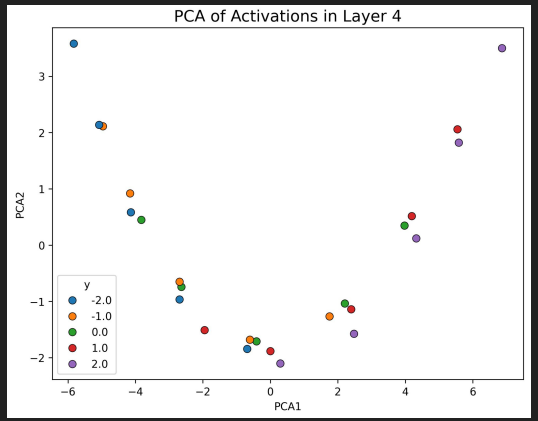
Upper and lower triangle states activate differently

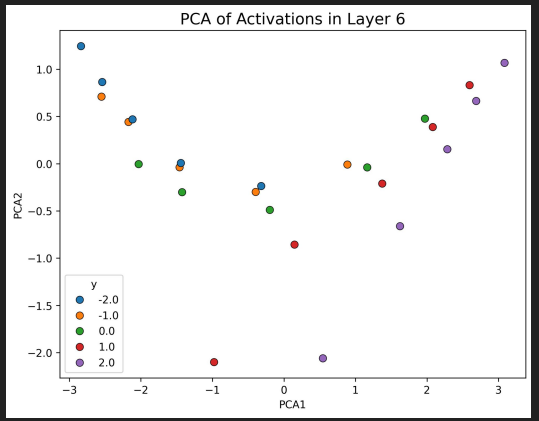




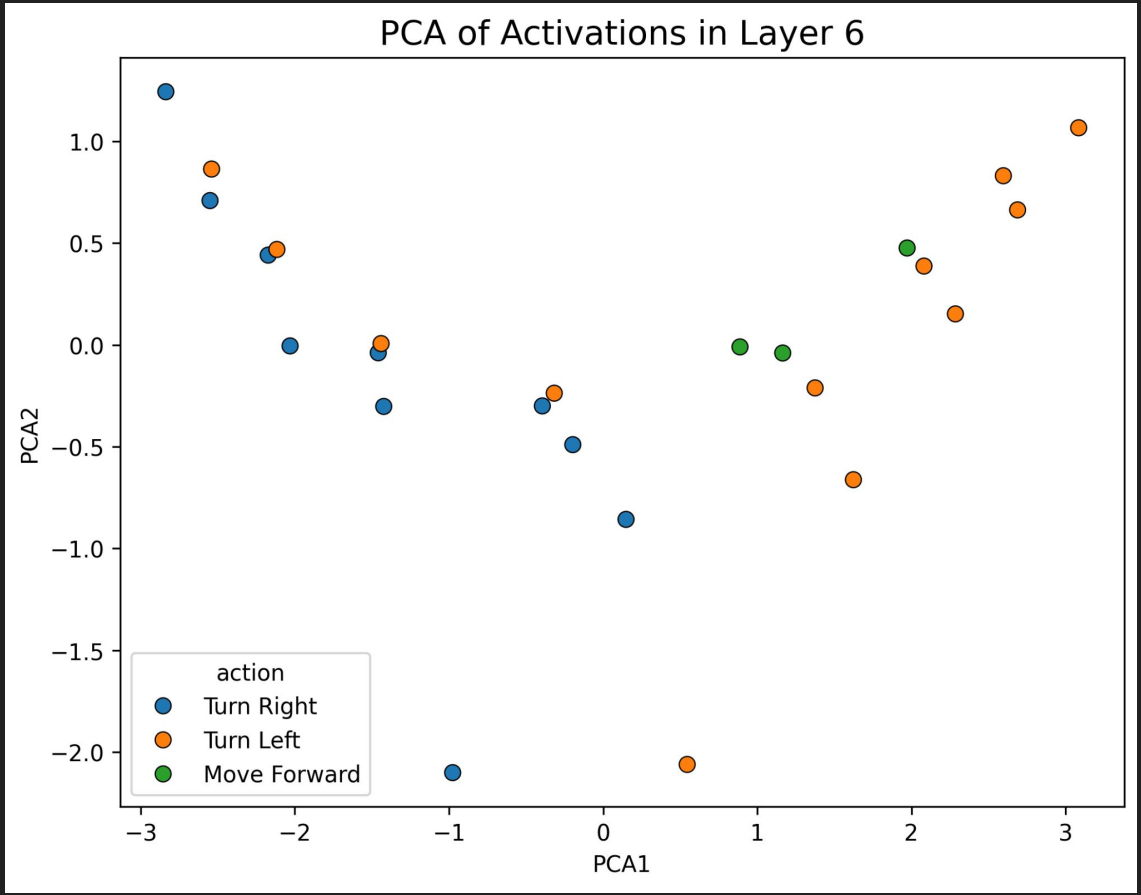




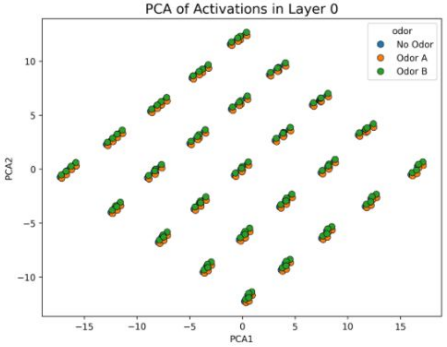


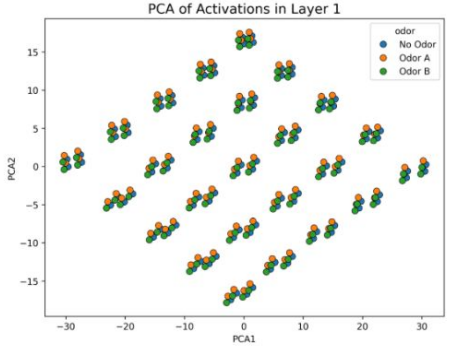


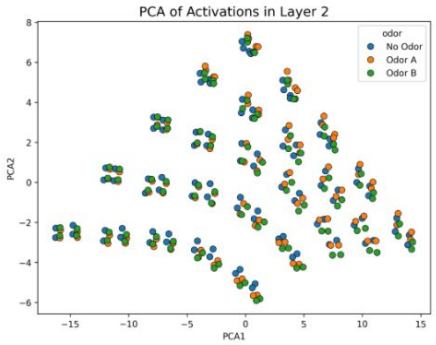
Differing upper and lower triangle activations reflect behavior differences

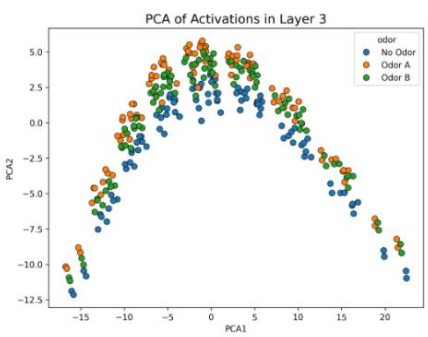


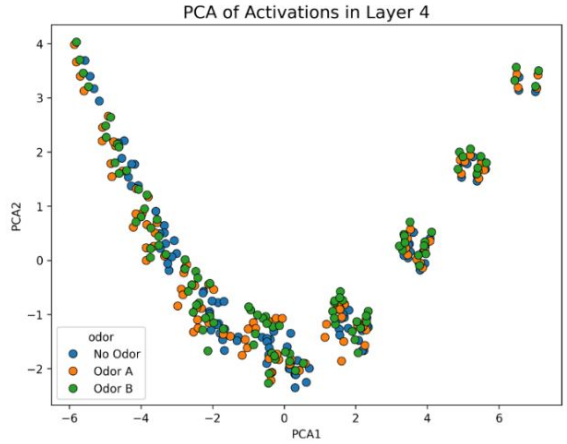
Odors Differentiate U/L Triangle, but Odor A and B Similar

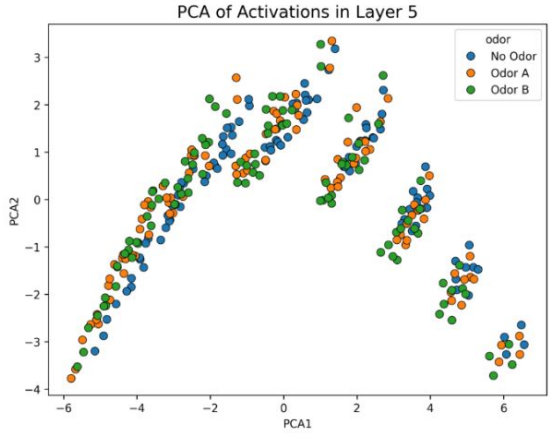


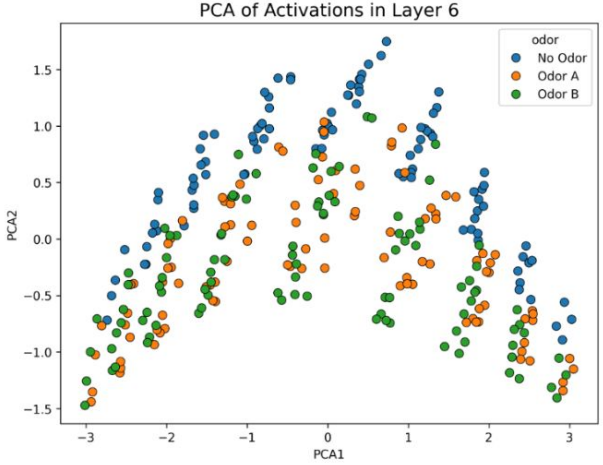












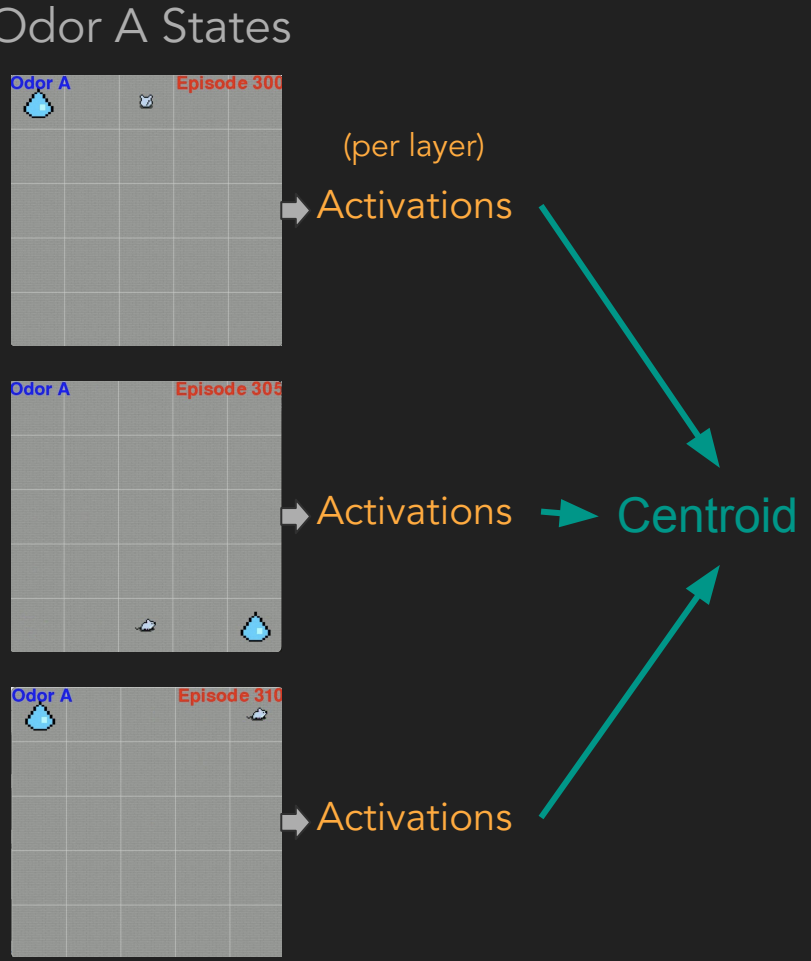
Observations and Limitations

- The model effectively learns to represent the upper and lower triangle differently in activation space, reflecting the different reward location
- Each odor maintains an encoding of the upper and lower triangle, and within each encoding there is a spatial map
    - But Odor A and B exhibit similar activation patterns; No Odor is more distinguished

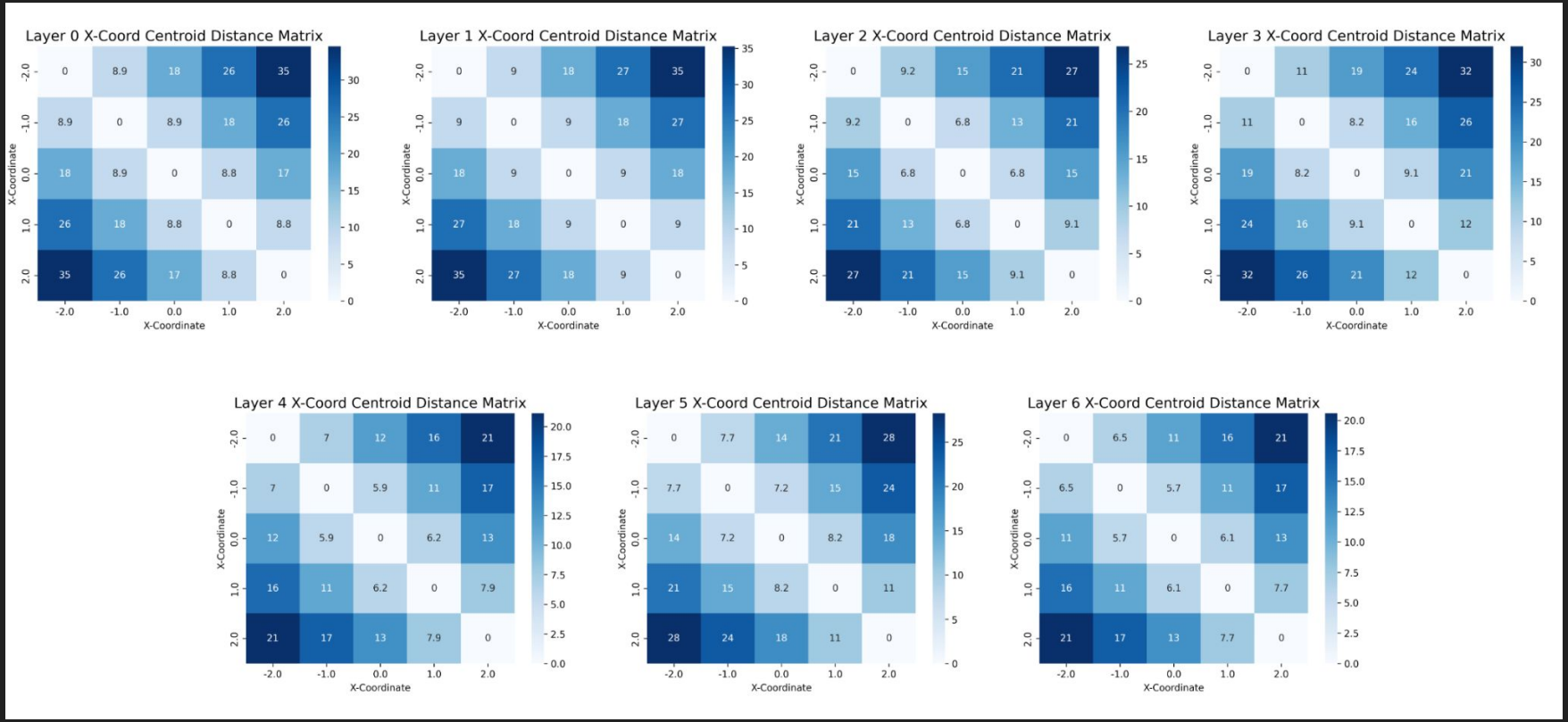
- Must be wary of interpreting visualizations of reduced-dimensional data
    - Makes sense when tied to behavior

Centroids

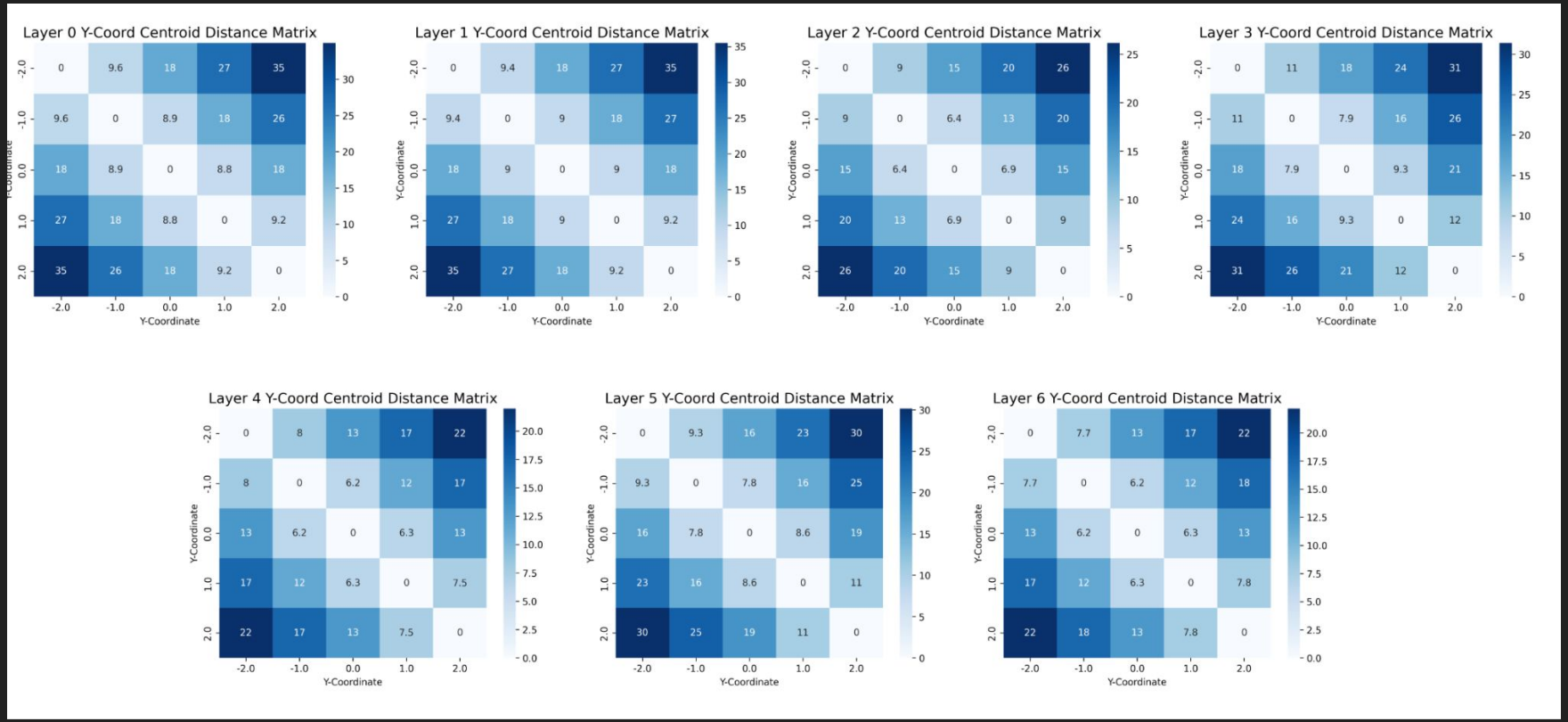
- Centroids are the <u>average activation</u> for a particular category
- For instance, an Odor A centroid represents the average activation of Odor A states
- We can compute differences between centroids to see how categorical representations differ in activation space



X-Coordinates Maintain Separate Activations



Y-Coordinates Maintain Separate Activations



Observations and Limitations

- A fairly consistent spatial grid is maintained as the network processes input
  - This makes intuitive sense early in the network (different grid positions have different input values), but it's interesting it's maintained until late in the network
- Activations reflect an understanding of position

- We are taking averages across states with very different pieces of information; different Odors, different head directions
  - What information are we losing or being biased towards?

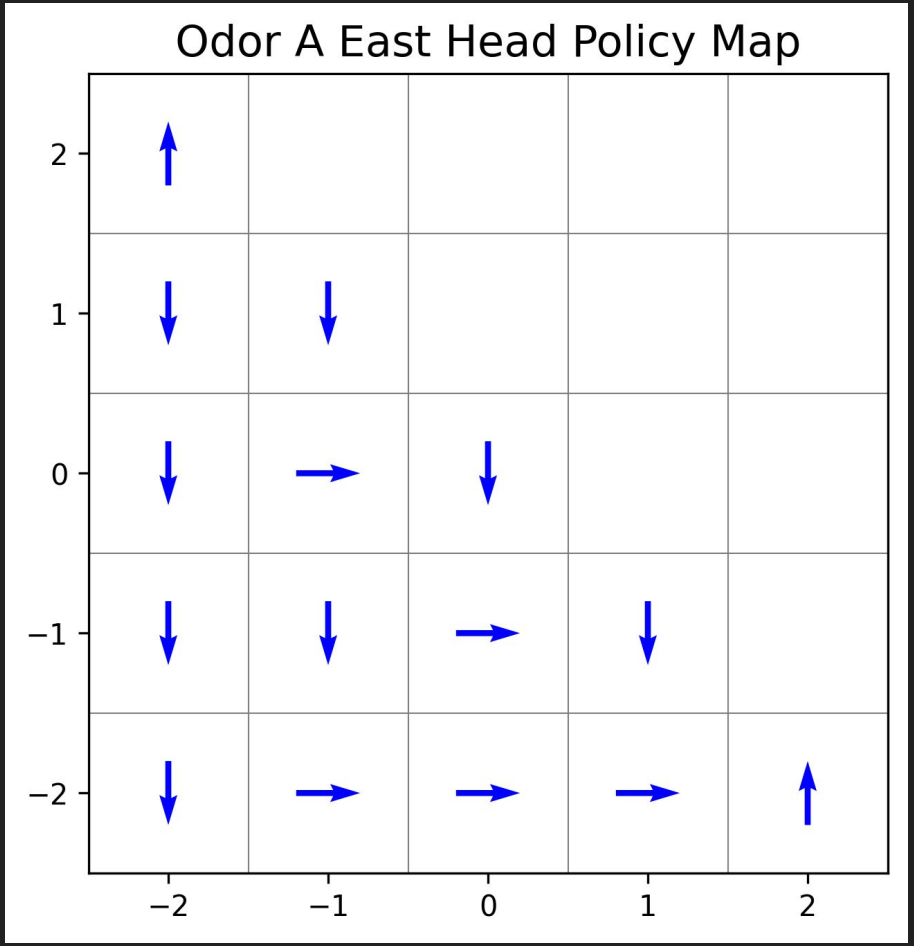
Model Behavior

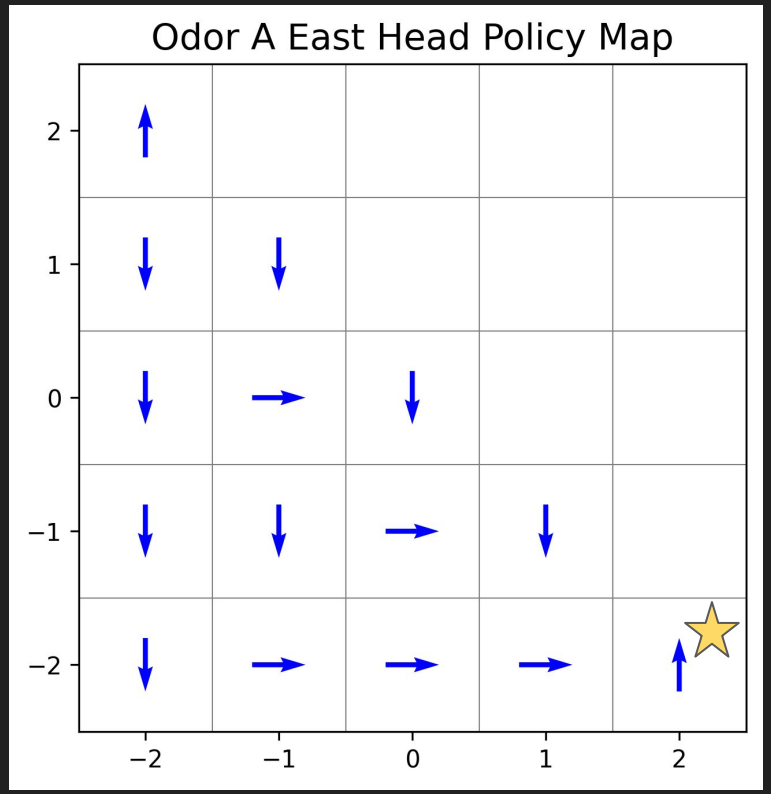
SHAP Values

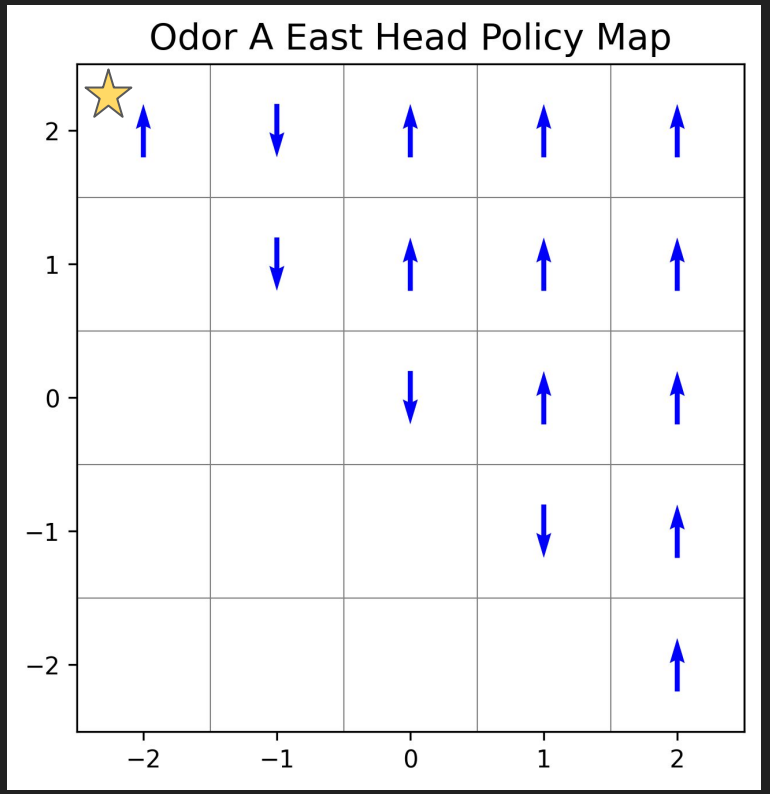
LeftRight Expert

Policy Maps

- Policy maps help us see the choices an agent makes in a given state




Odor A East Head Policy Map
2
1
0
−1
−2
−2
−1
0
1
2


Odor A East Head Policy Map
2
1
0
−1
−2
−2
−1
0
1
2

Observations and Limitations

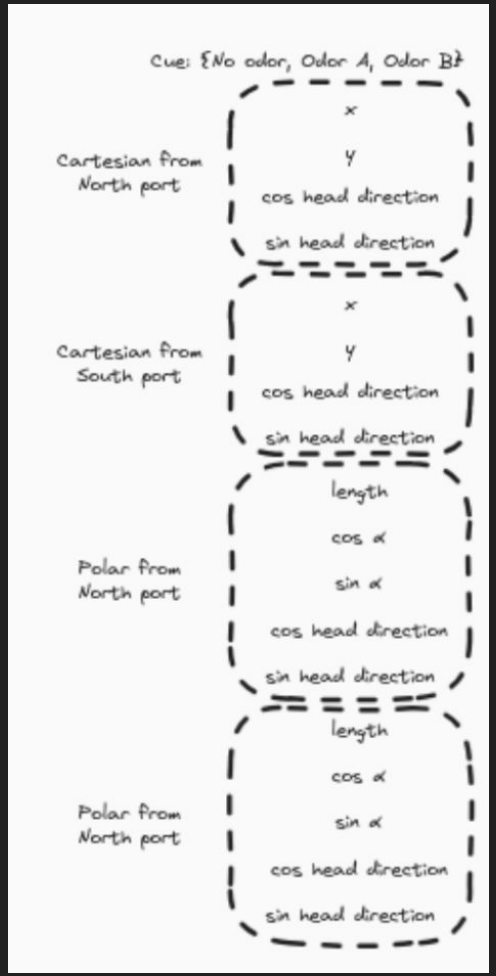
- Different behavior in upper and lower triangle – good!
- Agent seems to make some unintuitive choices – does it create pre-established paths that always lead to reward?
    - We *don't* penalize for efficiency

- Hard to see overall behavior trends, have to limit to specific states

SHAP Values

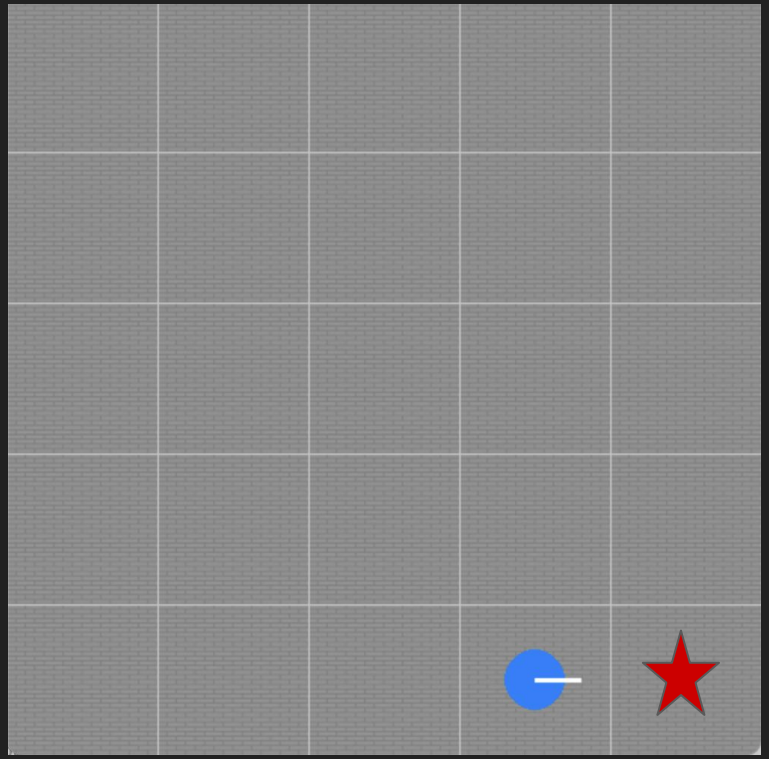
- SHAP values assign *feature importance*

- That is, for a given state, what input features are most important for the prediction of the action to take?



Interesting State

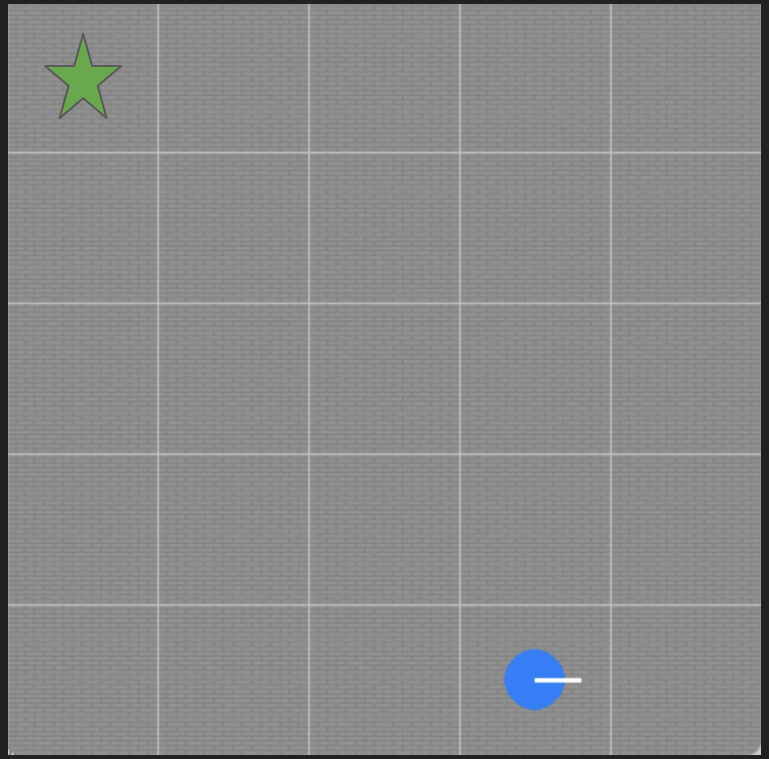
Left/Right



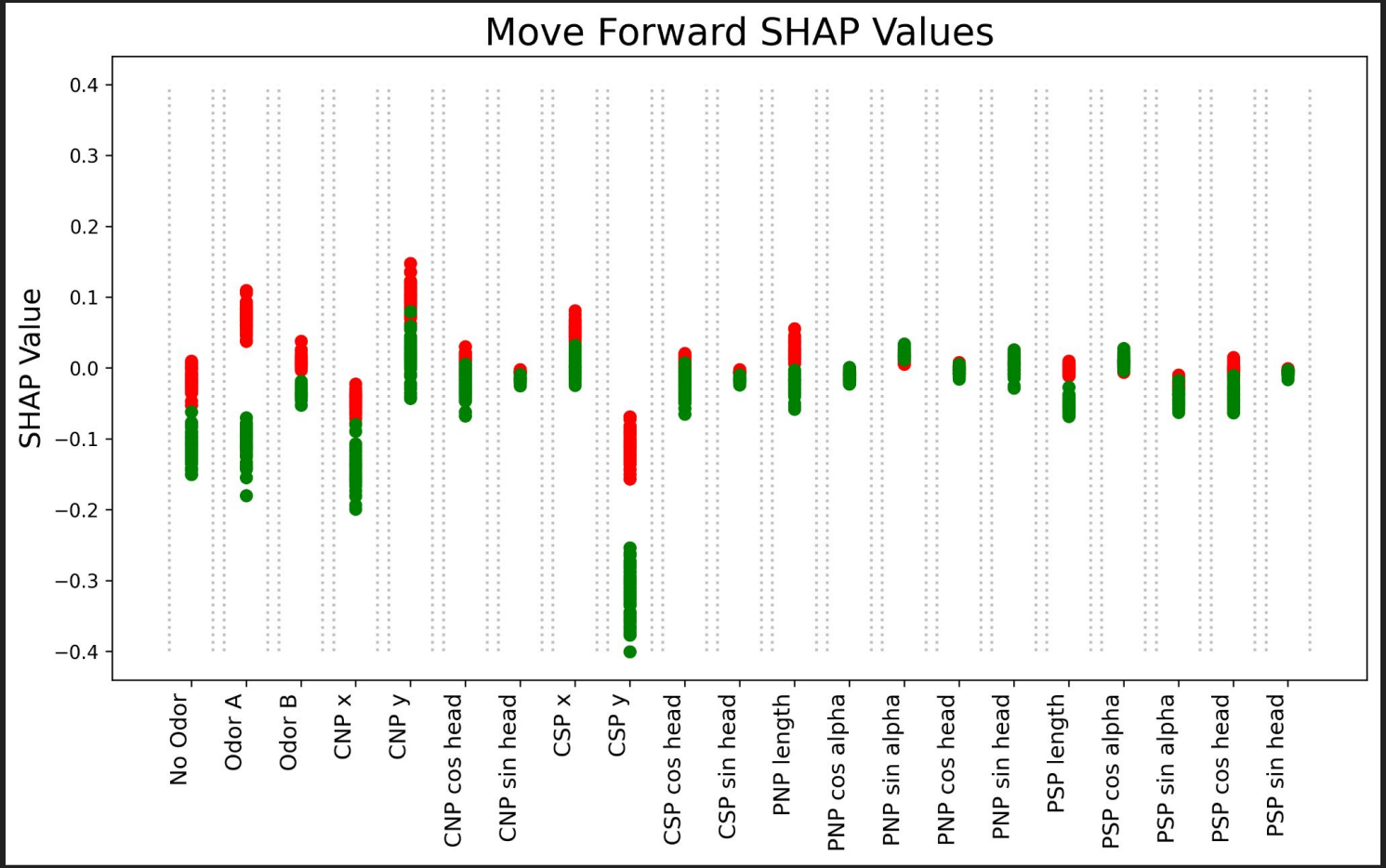
Odor A

vs.

East/West



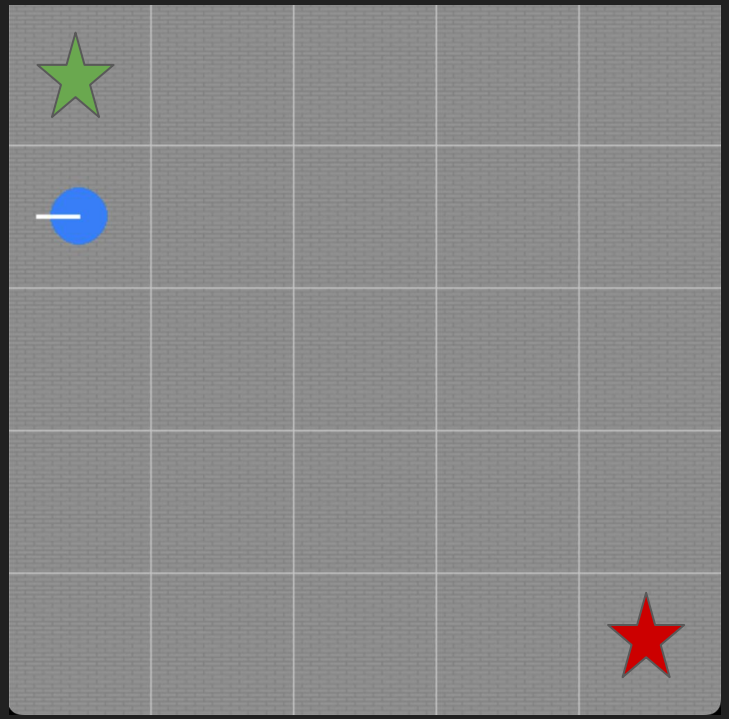
Odor and Cartesian Coordinates have Most Importance for E/W

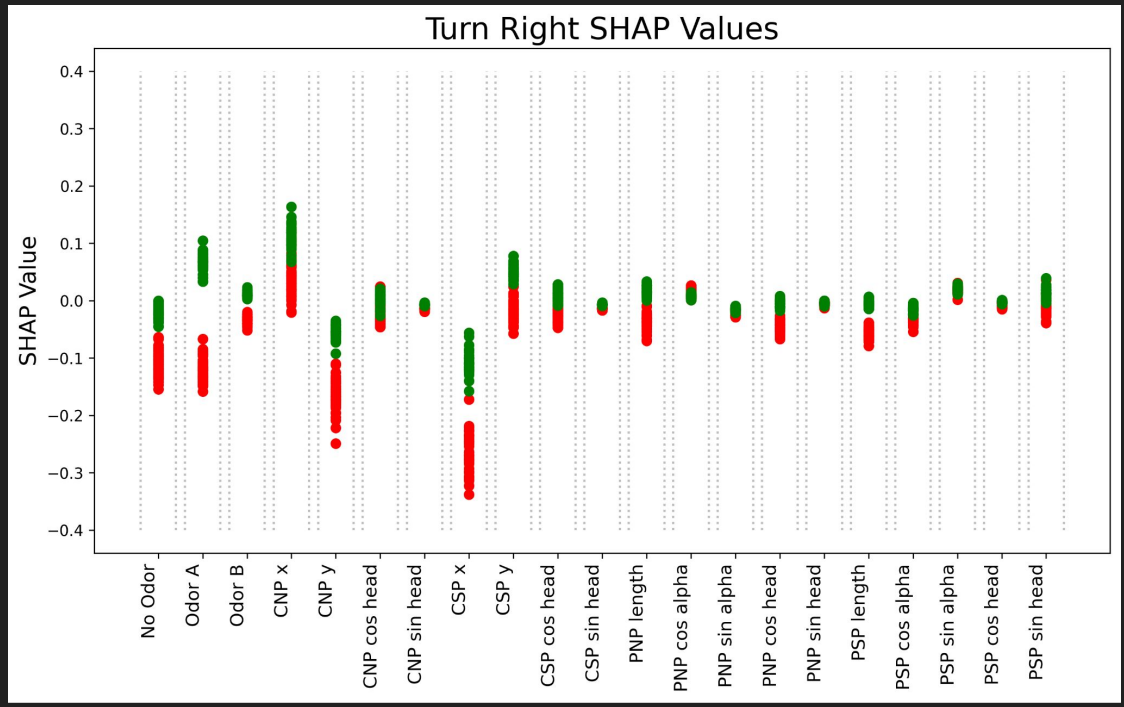


Red: LeftRight

Green: EastWest

Odor and Cartesian Coordinates have Most Importance for L/R





Red: LeftRight Green: EastWest

Observations and Limitations

- The agent seems to place highest importance on Odor and Cartesian coordinates when making decisions
- This is true for both LeftRight and EastWest agents – against our hypothesis?

- Have to look at a lot of other states to gain a fuller picture

Conclusions

What can we say?

- This has been a preliminary overview into the kinds of things the agent is learning

- Odor and Cartesian coordinates are input features that are salient in the network *(weights, SHAP values)*, across both LeftRight and EastWest agents

- The activation space for LeftRight agents is dominated by a difference between the Upper and Lower Triangle, reflecting behavior differences
- Every Odor identity maintains a spatial map of the Upper and Lower Triangle, but Odor A and B activate similarily

Future Directions

Where to Go From Here

- LOTS of potential directions…
  - Still a lack of streamlined technique for interpretability in DRL

- Andrea has been experimenting with autoencoders; the agent can still learn the task with a bottleneck of ~10 nodes
  - Investigating representations in this simpler network – could lead to easier analysis

- In this vein, conduct perturbation + compression experiments…

- Organizing a comprehensive analysis of the techniques shown here

Thanks for listening :)