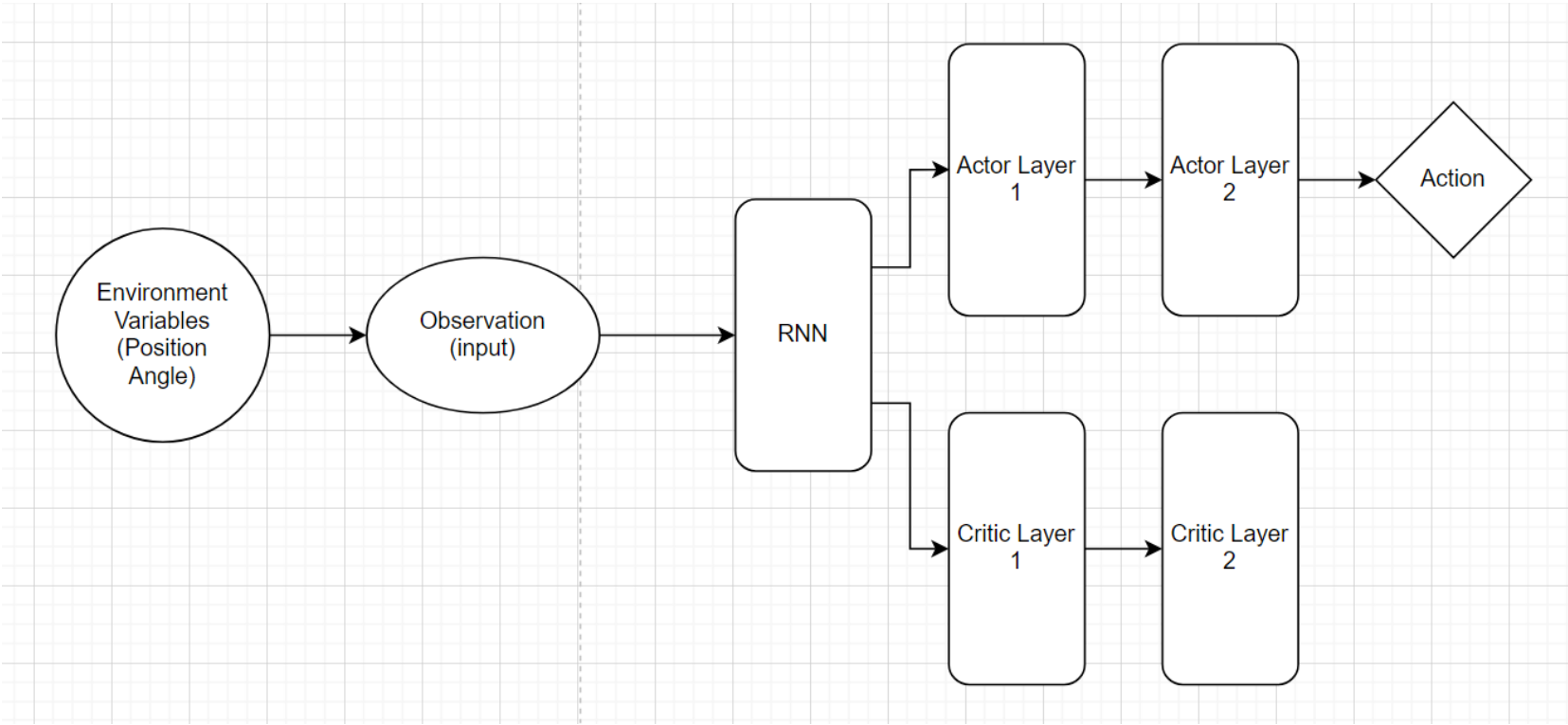
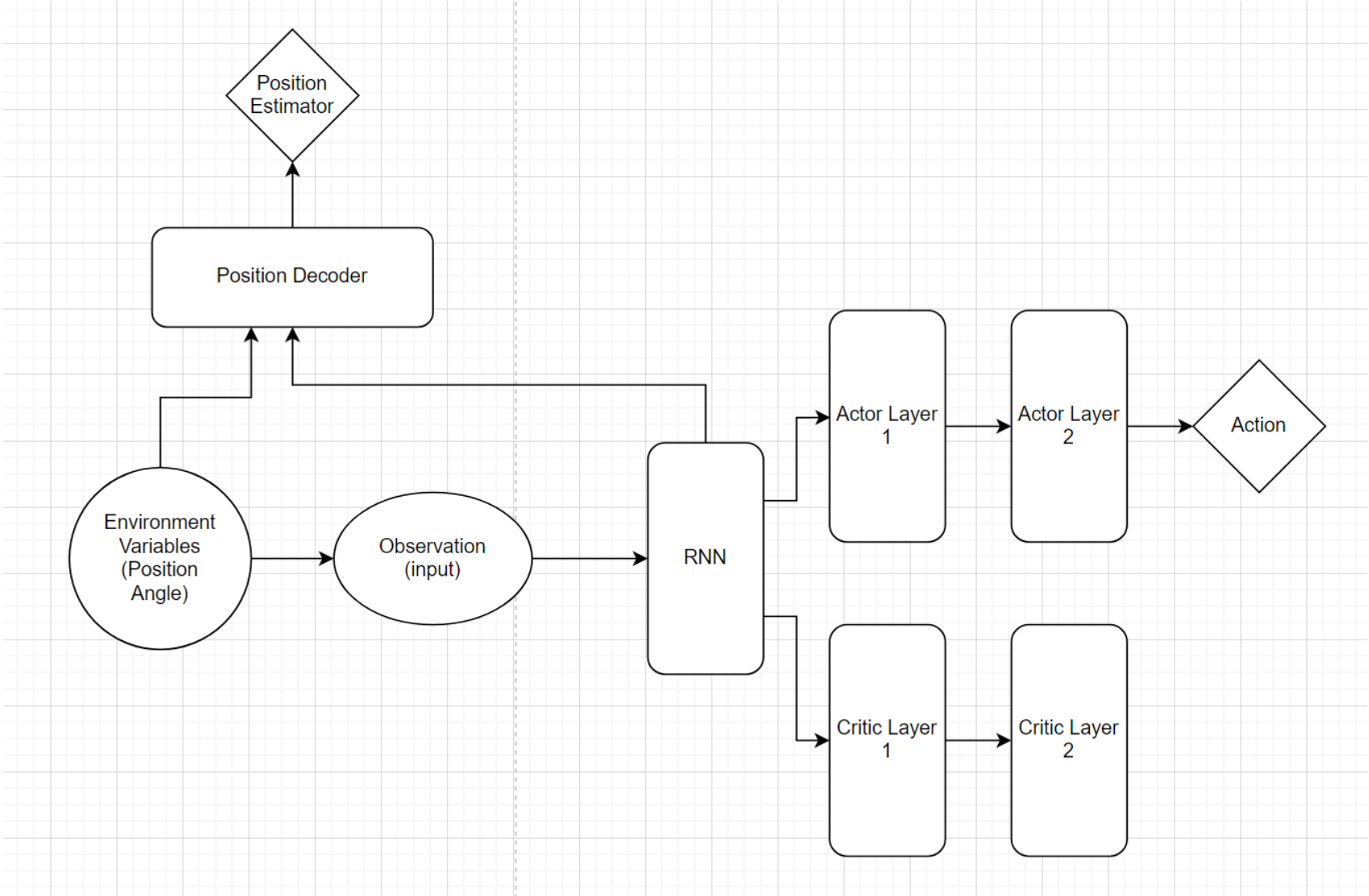
Usual Agent Diagram

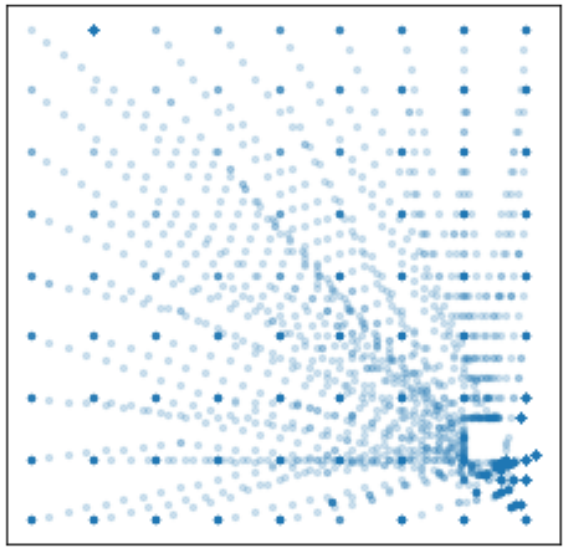


Activations and Position used to train Decoder

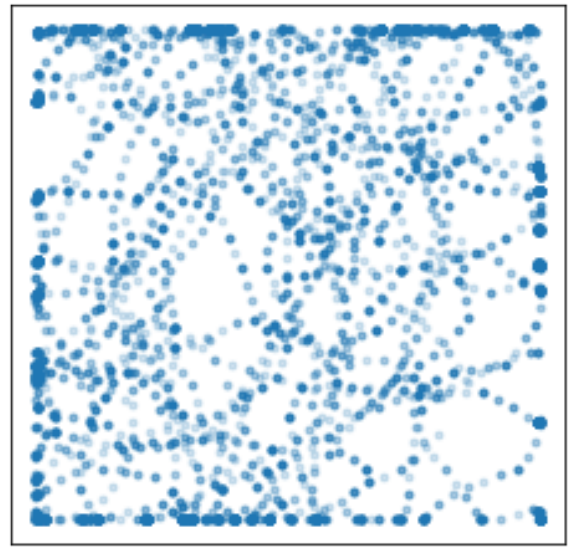


Generate some episodes, record position and activaations of NN during episode

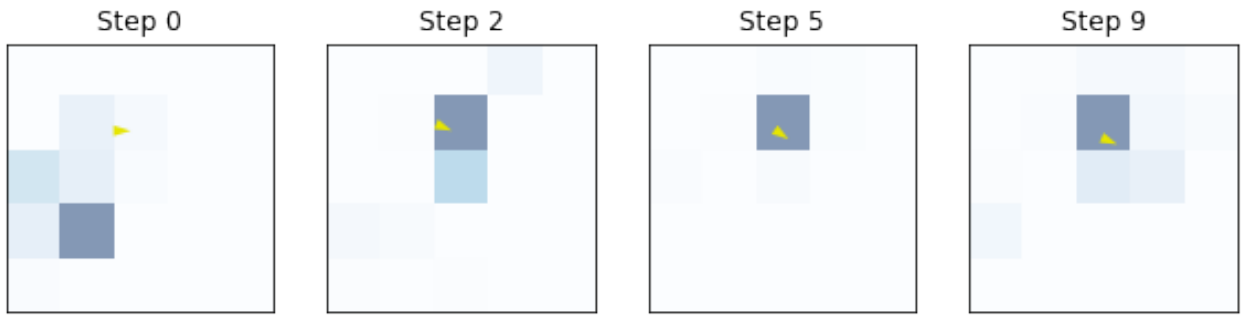
Example of trajectories from fixed starting positions



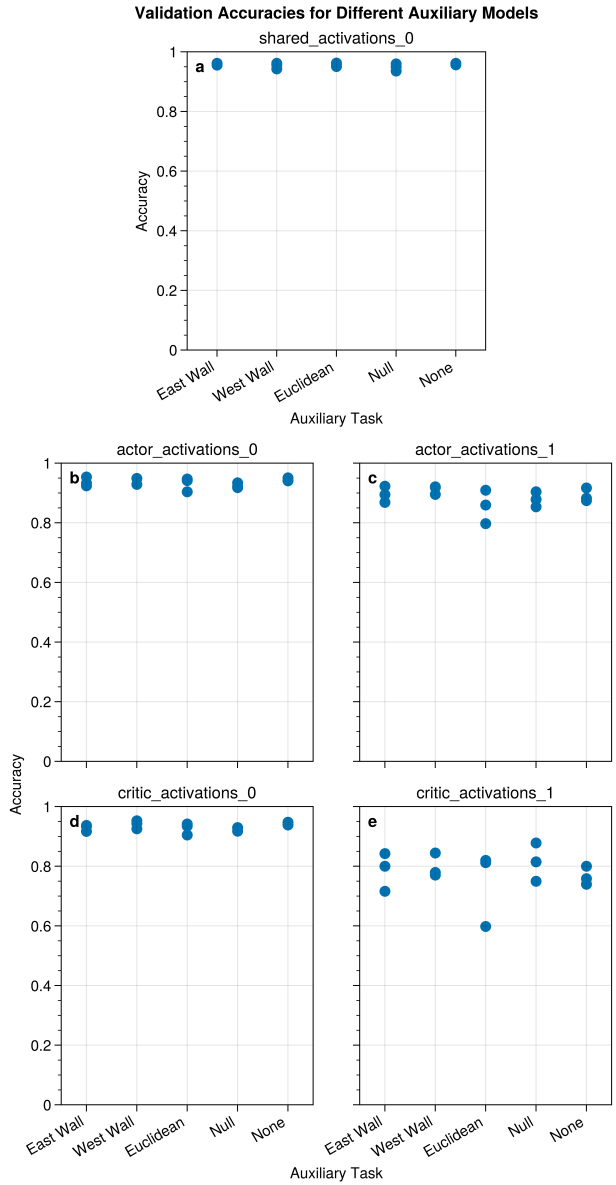
Example of 50 random action trajectories



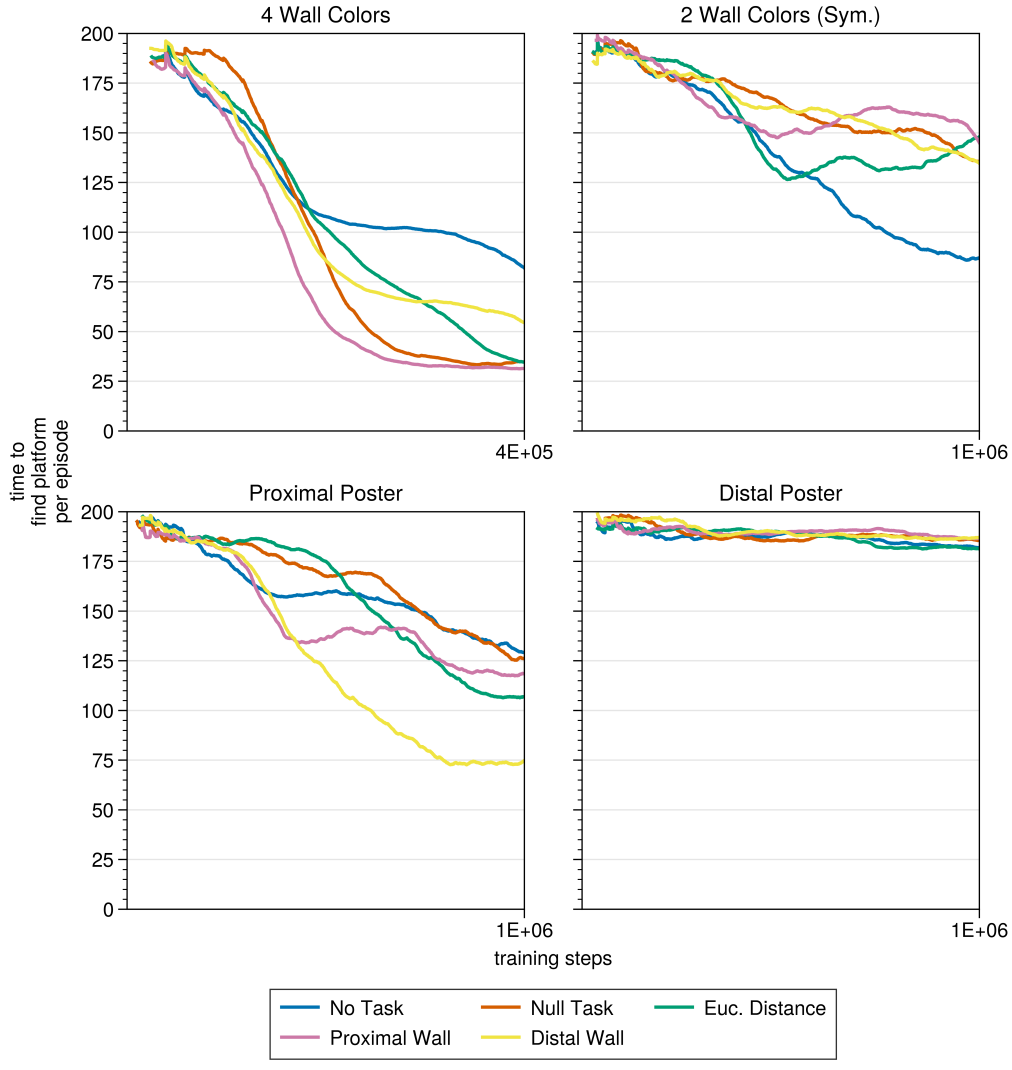
A resulting linear machine learning model can accurately predict position from activations alone (position predicted as which 5x5 grid the agent is currently in)



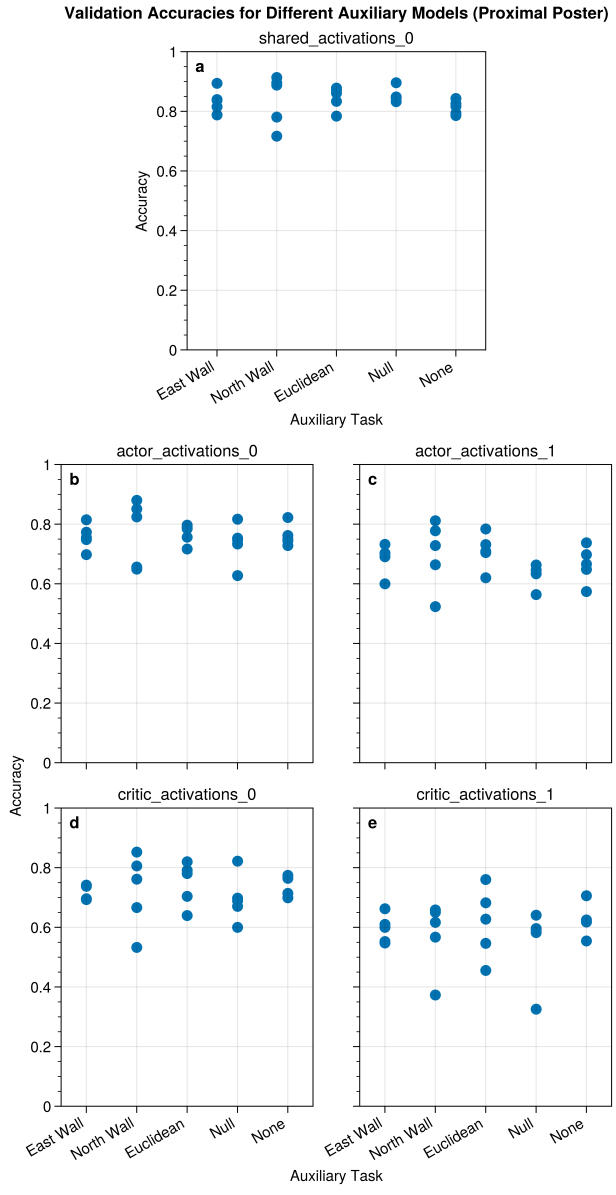
Position decoders can consistently be trained (Basic 4 wall color task)



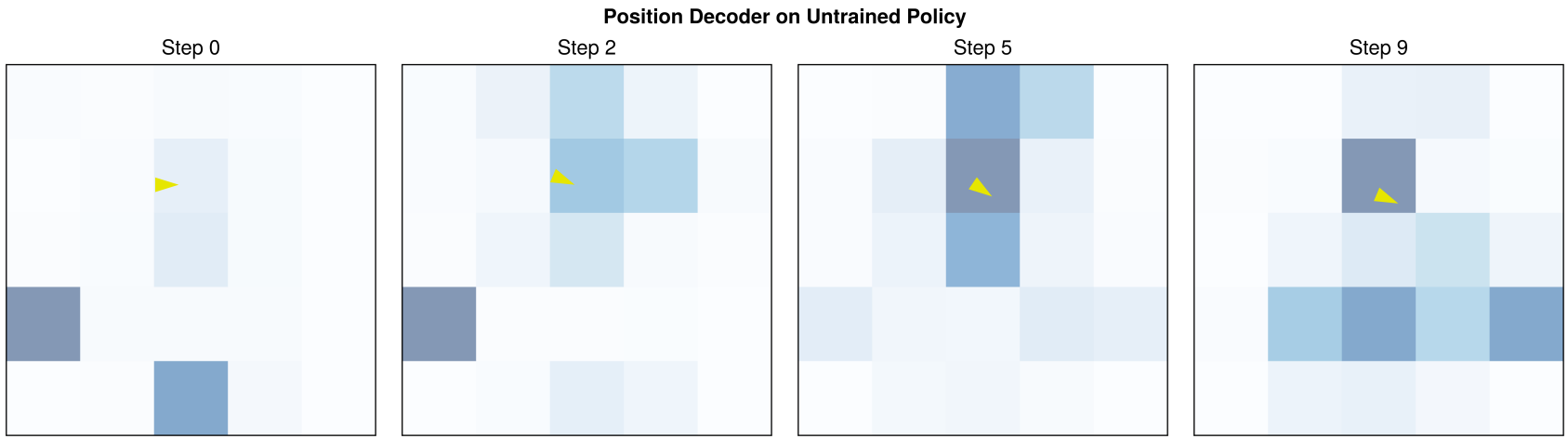
Accuracy seems to drop off for deeper layers


4 Wall Colors
2 Wall Colors (Sym.)
Proximal Poster
Distal Poster
time to find platform per episode
200
175
150
125
100
75
50
25
0
4E+05
1E+06
training steps
No Task
Null Task
Euc. Distance
Proximal Wall
Distal Wall

Position decoders less accurate for proximal poster environment



Accuracy does not show the whole picture. A position decoder can still achieve about 0.4 accuracy given activations from a totally untrained agent.

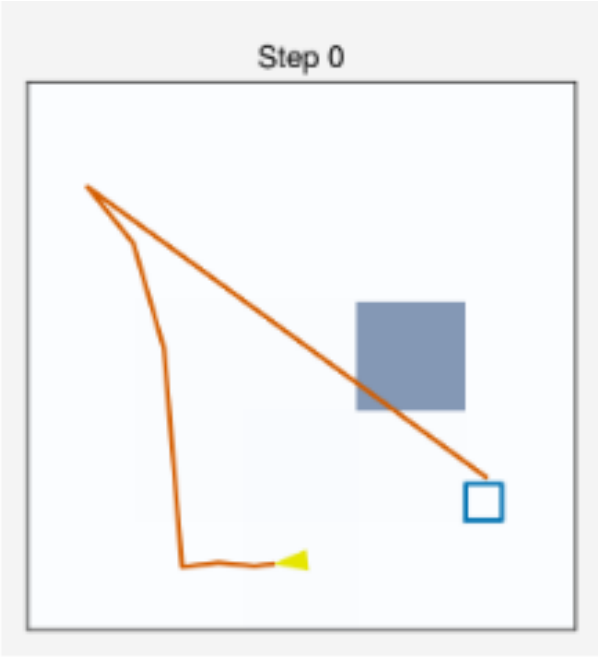


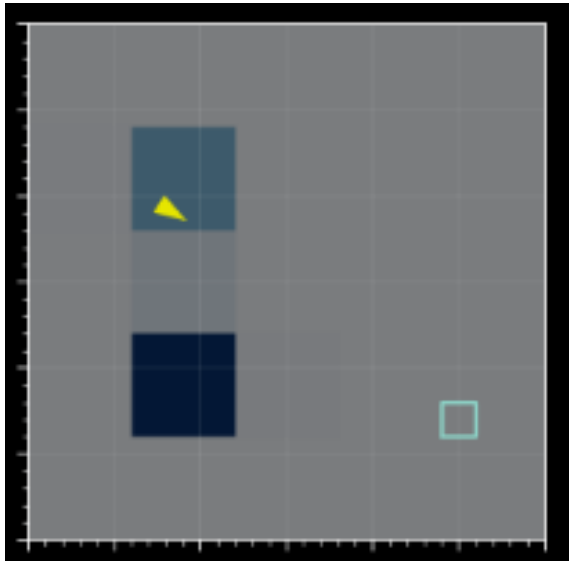
Questions

Does "having good representations" correlate with performance?
Can we quantify what good representations are?
What are deeper layers representing?

Exploration of Behavior and Representations

Proximal Poster environment leads to interesting policy and position decoder

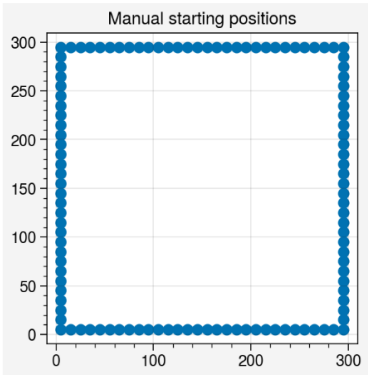


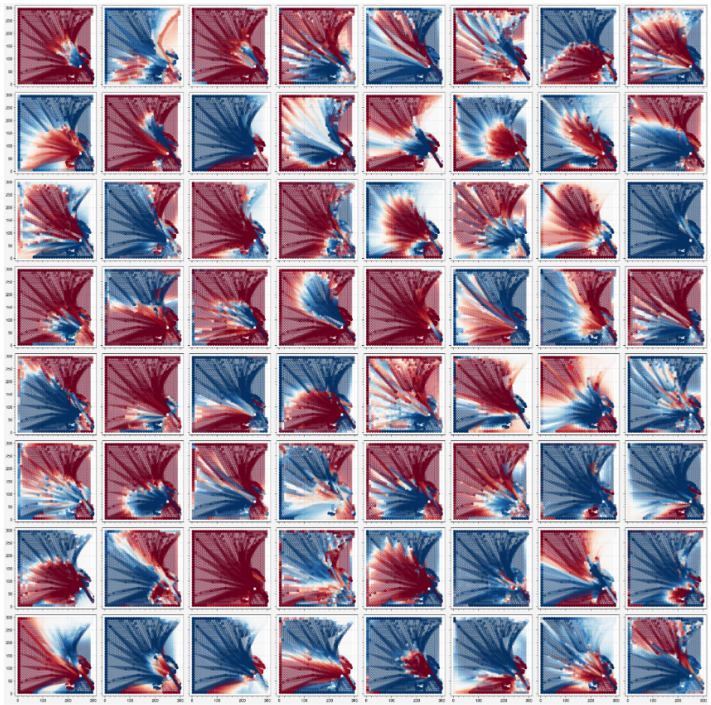


Left: a sample trajectory of proximal poster environment
Right: a sample step where the position decoder guesses the agent to be in symmetrical position

Coloring Neuron Activations

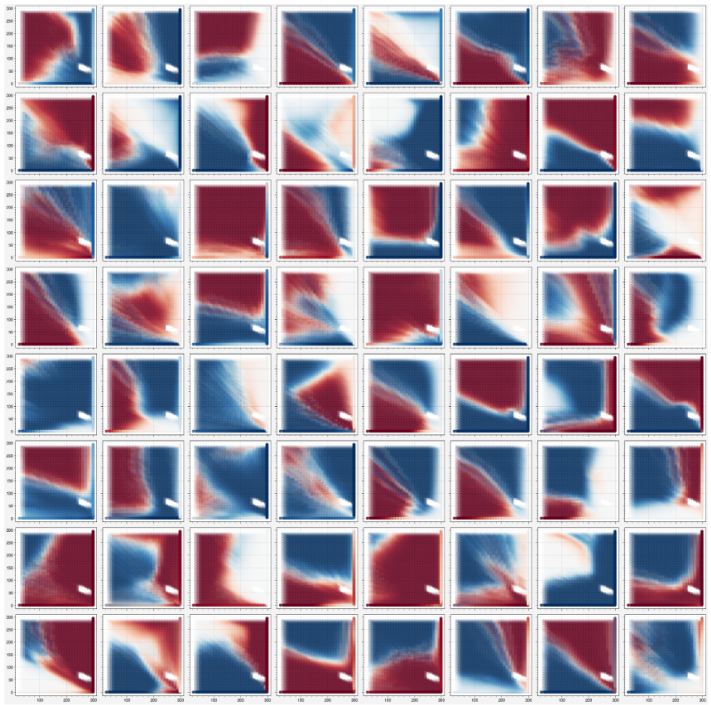
(4 Wall Color env) Starting from outer edge, track position and activations for first layer



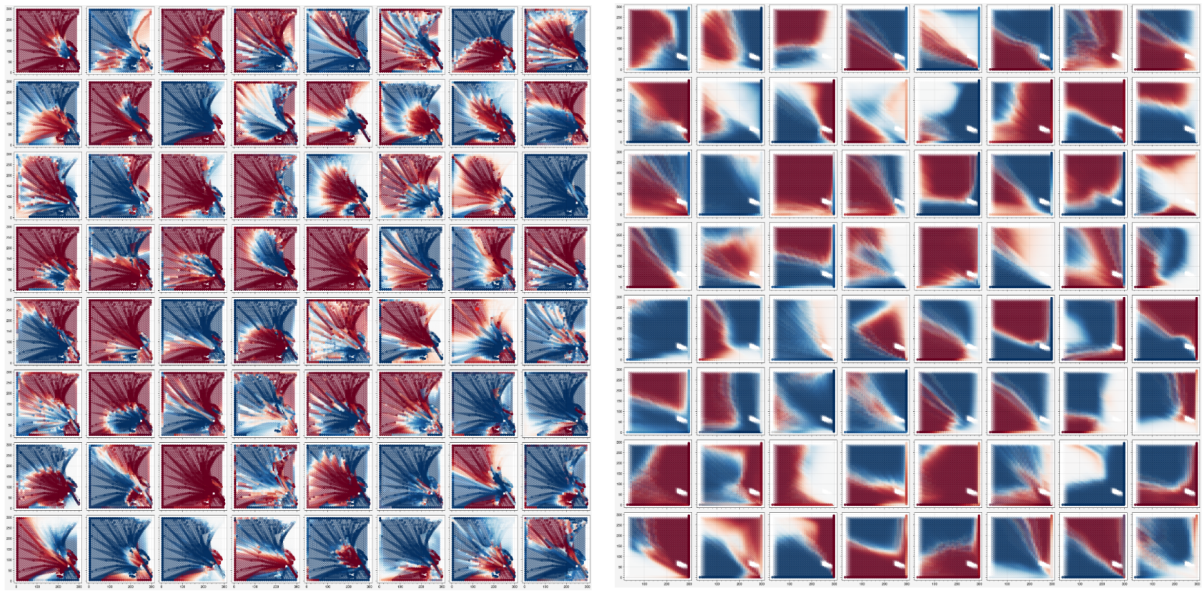


Coloring Neuron Activations

Same env, but fixed actions of moving toward bottom right



For comparison



UMAP reducing all activations