

Robust representations for olfactory-spatial association learning

Andrea Pierré

August 04, 2025

Outline

1. Project recap
2. Modeling & Simulation
3. Results
4. Conclusion

Outline

1. Project recap

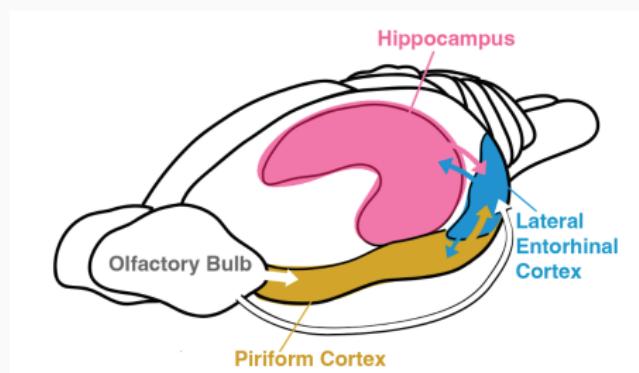
2. Modeling & Simulation

3. Results

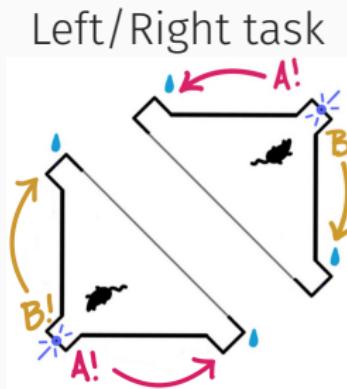
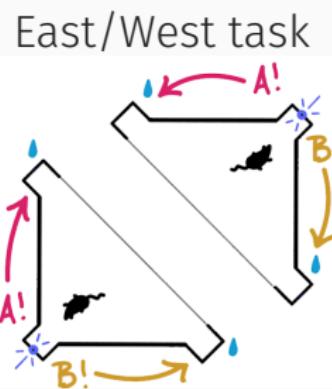
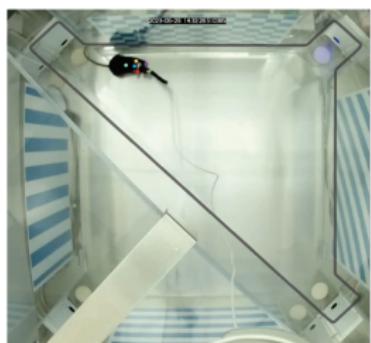
4. Conclusion

The LEC is key to sensory associations and spatial memory

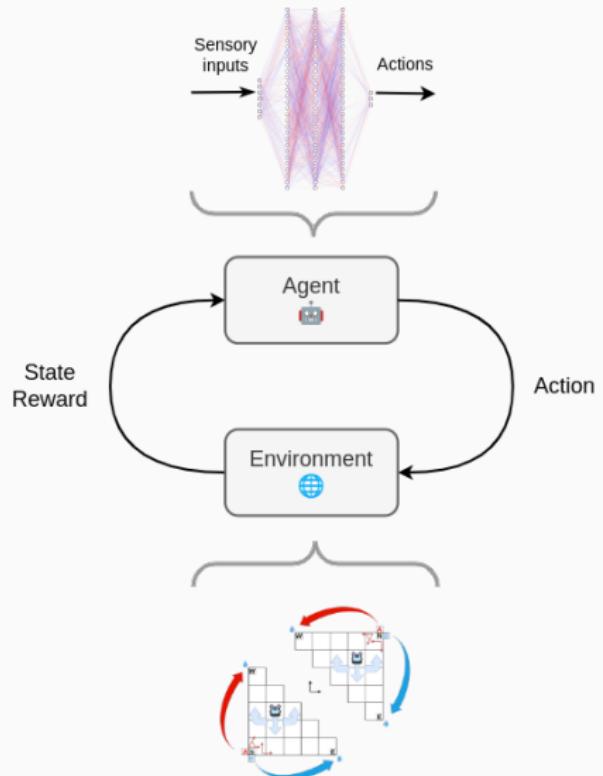
- **Piriform Cortex** encodes olfactory information
- **Hippocampus** encodes spatial information
- **Lateral Entorhinal Cortex (LEC)** encodes both olfactory & spatial information



Half triangle task for olfactory-spatial association learning

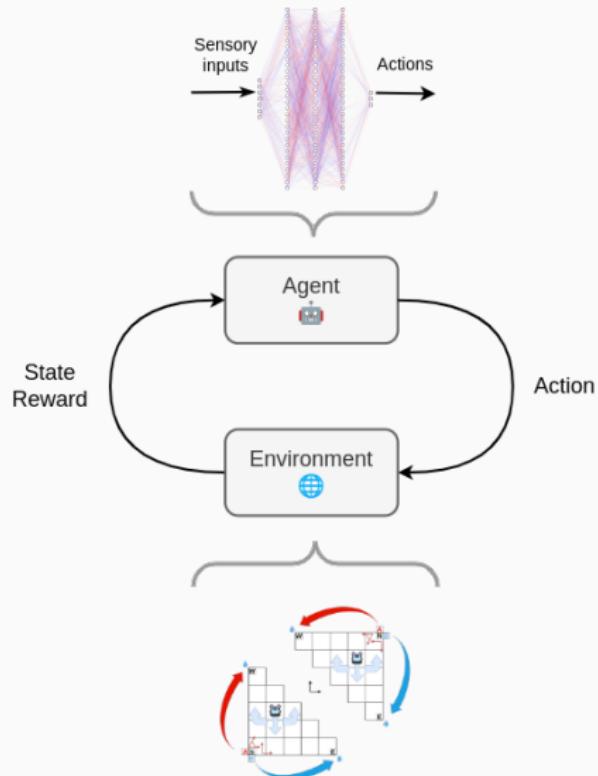


Deep Reinforcement Learning model



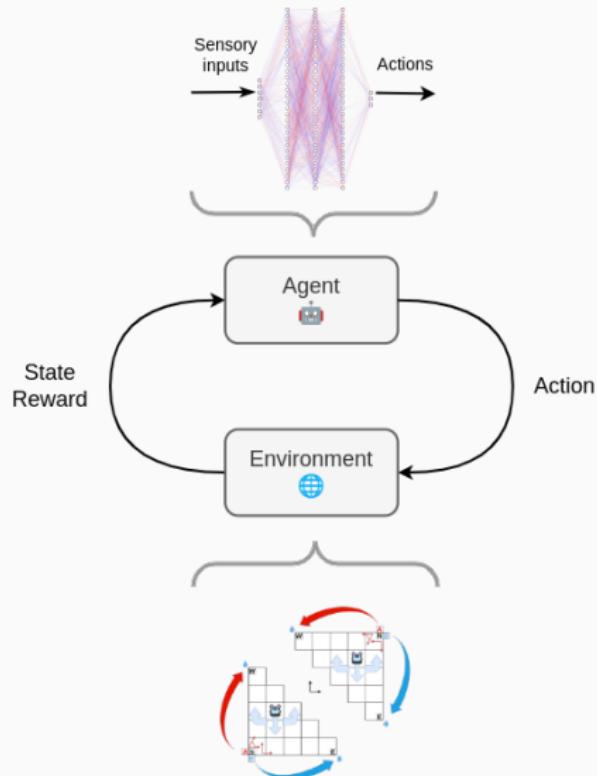
- Agent learns a policy (sequences of actions) that maximize rewards
- RL → no need to have ground truth data (as in supervised learning)

Deep Reinforcement Learning model



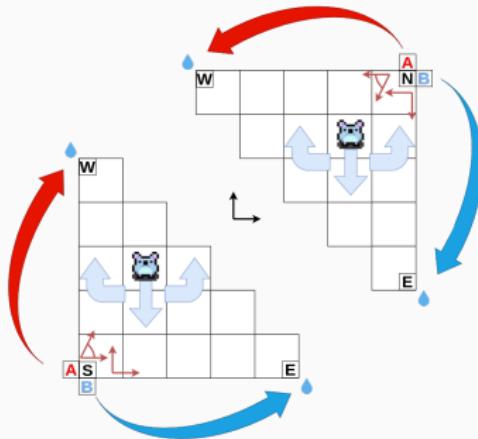
- Agent learns a policy (sequences of actions) that maximize rewards
- RL → no need to have ground truth data (as in supervised learning)

Deep Reinforcement Learning model



- Agent learns a policy (sequences of actions) that maximize rewards
- RL → no need to have ground truth data (as in supervised learning)

The environment



- 3 actions: $\leftarrow \uparrow \rightarrow$
- **Duplicated** coordinates inputs:
 - Cartesian coordinates from north & south port
 - Polar coordinates from north & south port

Questions & Hypothesis

Questions

- What **function** does the network learn?
- How the constraints of the task affect learning & the representations learned?
- How do the representations learned compare between the *in vivo* and the *in silico* neurons?

Hypothesis

- The network will use the **most efficient** coordinate information (Cartesian vs. polar) based on the task (left/right vs. east/west)
- The structure of the network's weights will reflect this prioritization of information
- Some neurons will encode a joint representation of {odor & space} (i.e., conjunctive cells)

Questions & Hypothesis

Questions

- What **function** does the network learn?
- How the constraints of the task affect learning & the representations learned?
- How do the representations learned compare between the *in vivo* and the *in silico* neurons?

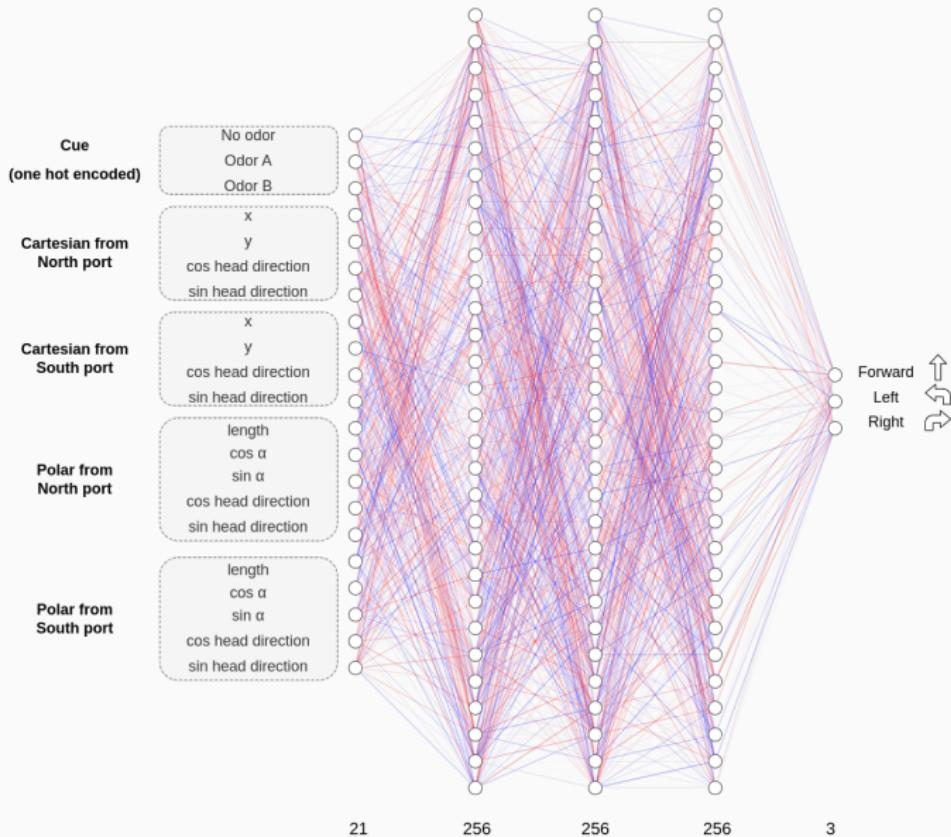
Hypothesis

- The network will use the **most efficient coordinate information** (Cartesian vs. polar) based on the task (left/right vs. east/west)
- The structure of the network's weights will reflect this prioritization of information
- Some neurons will encode a **joint representation of {odor & space}** (i.e., conjunctive cells)

Outline

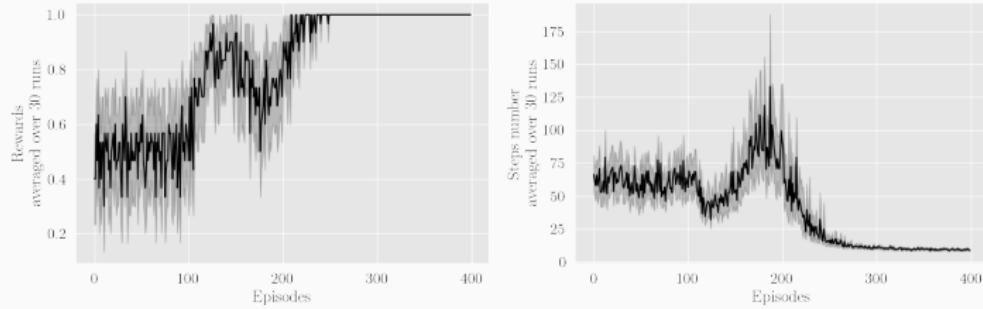
1. Project recap
2. Modeling & Simulation
3. Results
4. Conclusion

State space & network architecture

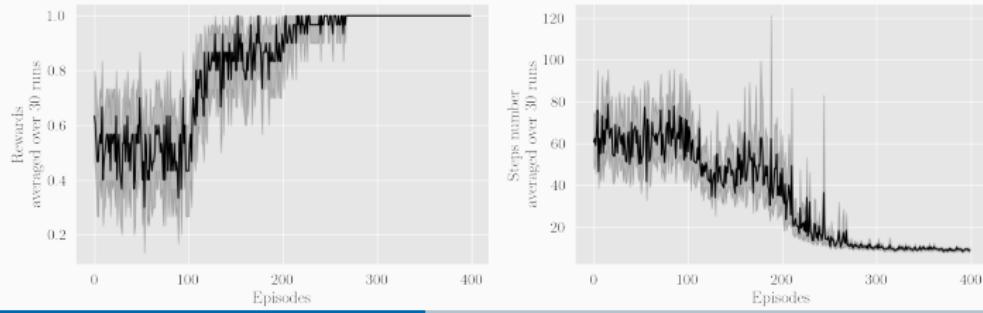


Training

East/West

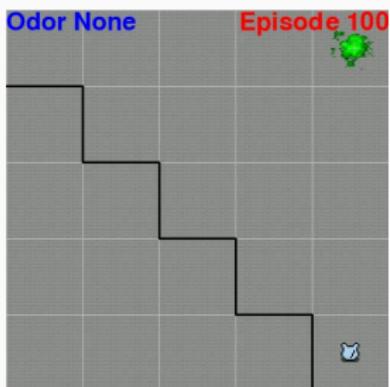


Left/Right

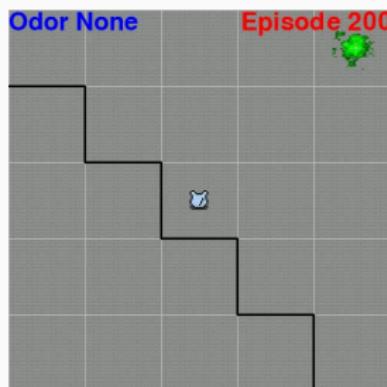


Agent behavior

Naive agent



Intermediate agent



Trained agent



Outline

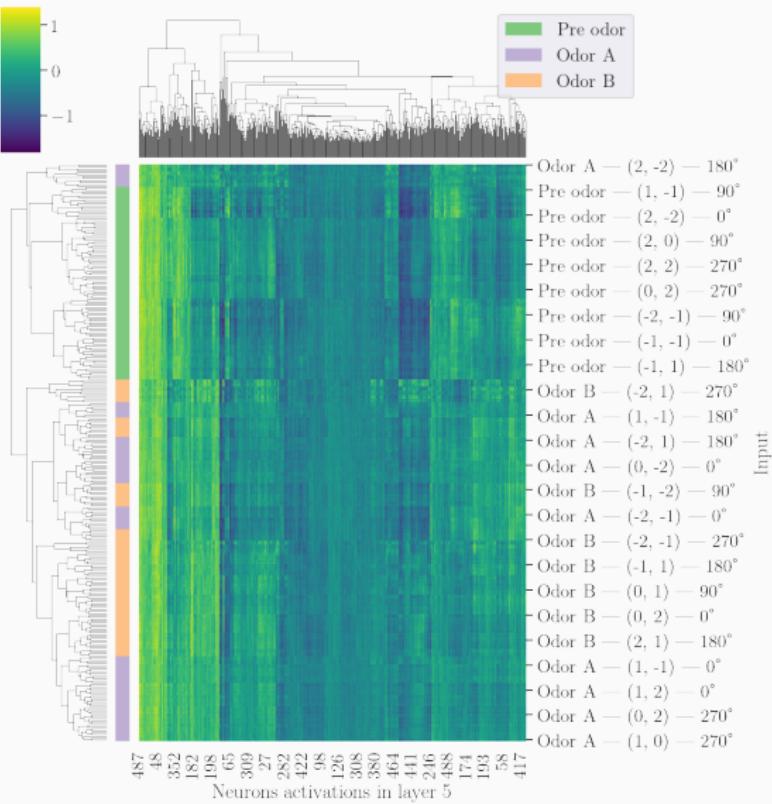
1. Project recap

2. Modeling & Simulation

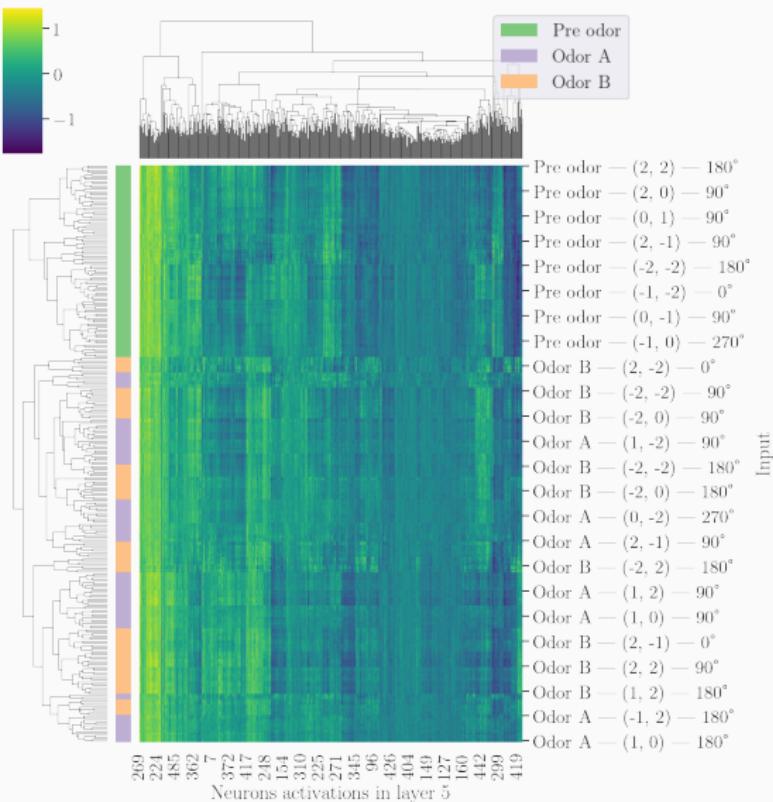
3. Results

4. Conclusion

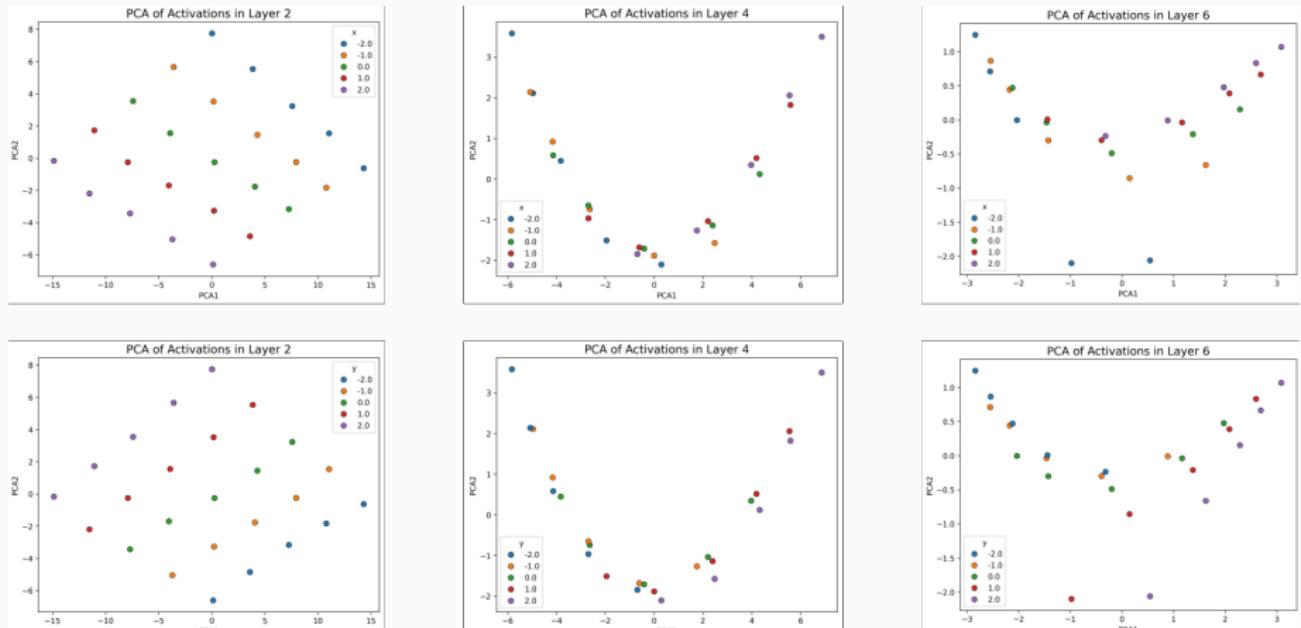
Pre-odor activations cluster together – East/West



Pre-odor activations cluster together – Left/ Right

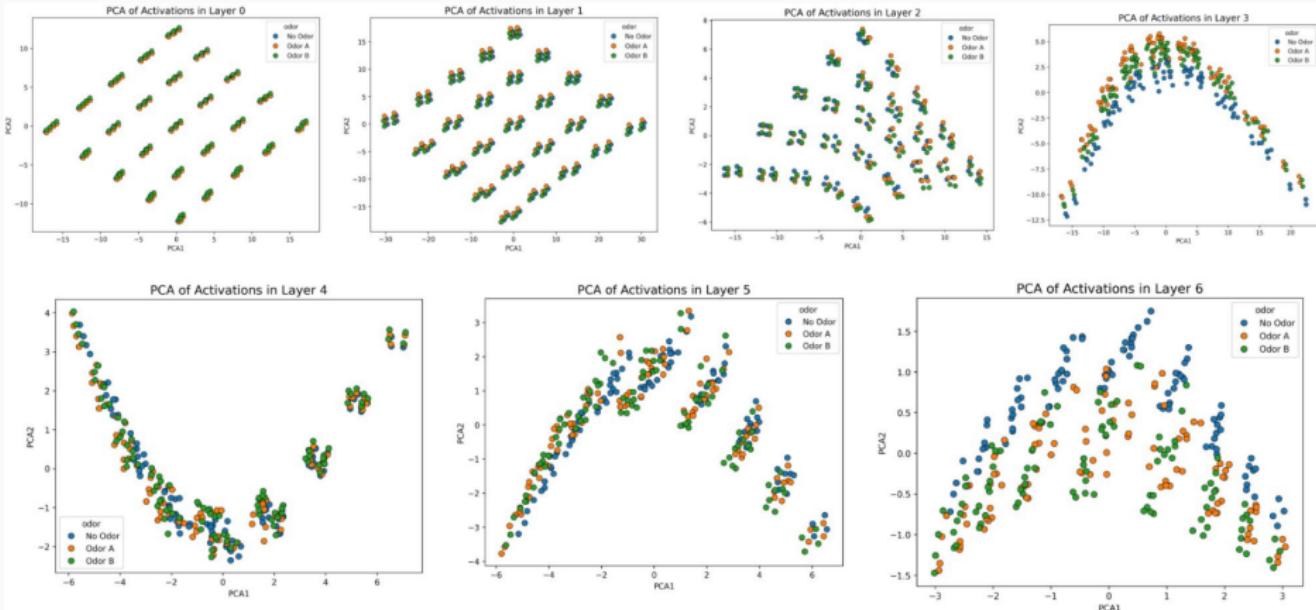


PCA on activations by layer on $\{x, y\}$ coordinates – Left/Right



- From grid shape in early layer to half moon cluster in late layers
- Seems to encode the lower/upper triangle

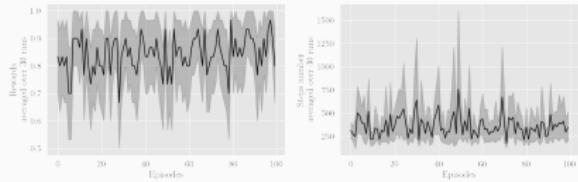
PCA on activations by layer on odor cue – Left/Right



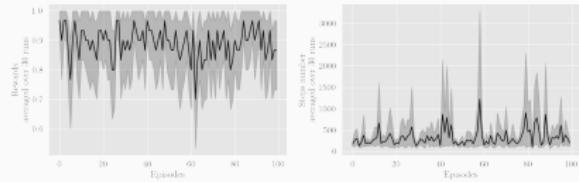
- From grid shape in early layer to half moon cluster in late layers
- Seems that odor cues cluster by tile/grid

Cartesian inputs unchanged – polar inputs perturbed

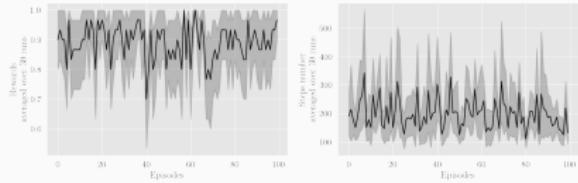
East/West
Silencing inputs



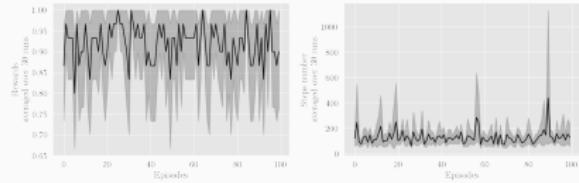
Left/Right
Silencing inputs



Randomizing inputs

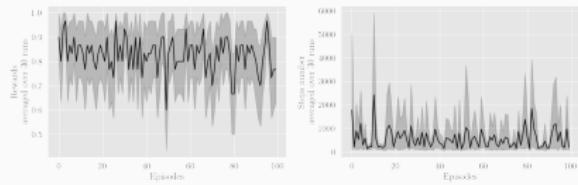


Randomizing inputs

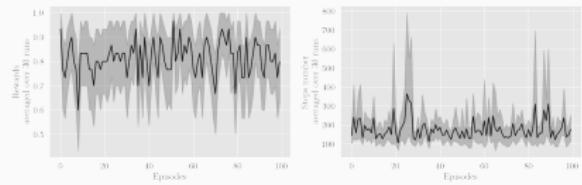


Polar inputs unchanged – Cartesian inputs perturbed

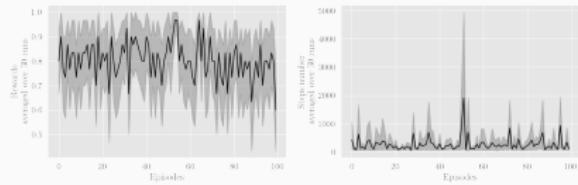
East/West
Silencing inputs



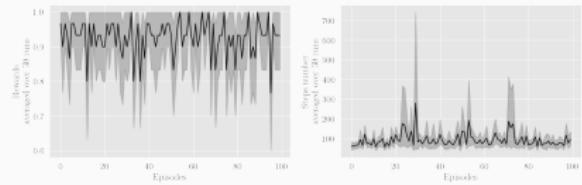
Left/Right
Silencing inputs



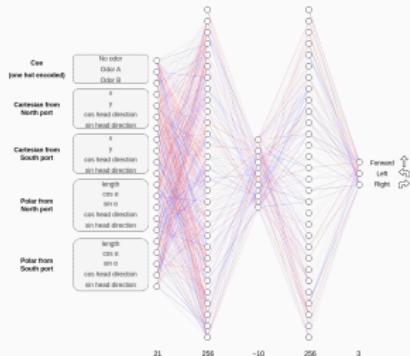
Randomizing inputs



Randomizing inputs

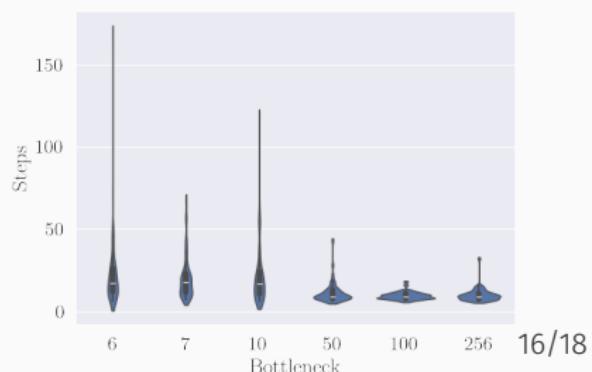
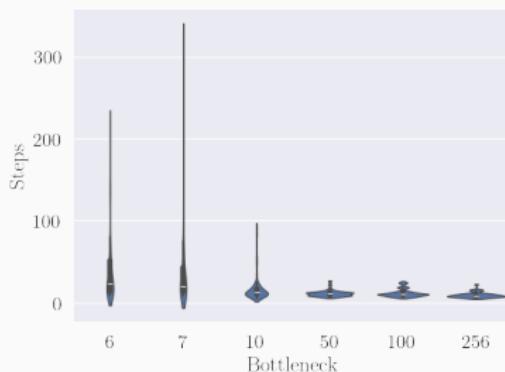


Bottleneck network → performance degrades below ~10 neurons



East/West

Left/Right



Outline

1. Project recap

2. Modeling & Simulation

3. Results

4. Conclusion

Partial conclusions so far

- The pre-odor activations cluster together, but no other clear pattern seems to emerge
- Dimensionality reduction analysis → the network seems to mainly encode the position of the agent on the grid
- With this task setup, it seems both types of coordinates information are required to solve the task → is this a negative result that invalidates our original hypothesis or is this due to overfitting?
 - From earlier experiments, we know the agent can perfectly solve the task without needing duplicated inputs..

Partial conclusions so far

- The pre-odor activations cluster together, but no other clear pattern seems to emerge
- Dimensionality reduction analysis → the network seems to mainly encode the position of the agent on the grid
- With this task setup, it seems both types of coordinates information are required to solve the task → is this a negative result that invalidates our original hypothesis or is this due to overfitting?
 - From earlier experiments, we know the agent can perfectly solve the task without needing duplicated inputs..

Partial conclusions so far

- The pre-odor activations cluster together, but no other clear pattern seems to emerge
- Dimensionality reduction analysis → the network seems to mainly encode the position of the agent on the grid
- With this task setup, it seems both types of coordinates information are required to solve the task → is this a negative result that invalidates our original hypothesis or is this due to overfitting?
 - From earlier experiments, we know the agent can perfectly solve the task without needing duplicated inputs...

Partial conclusions so far

- The pre-odor activations cluster together, but no other clear pattern seems to emerge
- Dimensionality reduction analysis → the network seems to mainly encode the position of the agent on the grid
- With this task setup, it seems both types of coordinates information are required to solve the task → is this a negative result that invalidates our original hypothesis or is this due to overfitting?
 - From earlier experiments, we know the agent can perfectly solve the task without needing duplicated inputs...

Next steps

- What happens in between tiles? Did the network learn to interpolate?
- Add dropout during training to make the network more robust to perturbations and avoid overfitting?
- Identify the potential functions of the neurons in the bottleneck layer? What type of information do they encode? Any sign of potential conjunctive cells?
- End of this project → still looking for well defined results/principles/rules for a potential MVP (Minimum Viable Publication)

Next steps

- What happens in between tiles? Did the network learn to interpolate?
- Add dropout during training to make the network more robust to perturbations and avoid overfitting?
- Identify the potential functions of the neurons in the bottleneck layer? What type of information do they encode? Any sign of potential conjunctive cells?
- End of this project → still looking for well defined results/principles/rules for a potential MVP (Minimum Viable Publication)

Next steps

- What happens in between tiles? Did the network learn to interpolate?
- Add dropout during training to make the network more robust to perturbations and avoid overfitting?
- Identify the potential functions of the neurons in the bottleneck layer? What type of information do they encode? Any sign of potential conjunctive cells?
- End of this project → still looking for well defined results/principles/rules for a potential MVP (Minimum Viable Publication)

Next steps

- What happens in between tiles? Did the network learn to interpolate?
- Add dropout during training to make the network more robust to perturbations and avoid overfitting?
- Identify the potential functions of the neurons in the bottleneck layer? What type of information do they encode? Any sign of potential conjunctive cells?
- End of this project → still looking for well defined results/principles/rules for a potential MVP (Minimum Viable Publication)

Questions ?

Pytorch network

```
DQN(  
    (mlp): Sequential(  
        (0): Linear(in_features=21, out_features=512, bias=True)  
        (1): Linear(in_features=512, out_features=512, bias=True)  
        (2): ReLU()  
        (3): Linear(in_features=512, out_features=512, bias=True)  
        (4): ReLU()  
        (5): Linear(in_features=512, out_features=512, bias=True)  
        (6): ReLU()  
        (7): Linear(in_features=512, out_features=3, bias=True)  
    )  
)
```