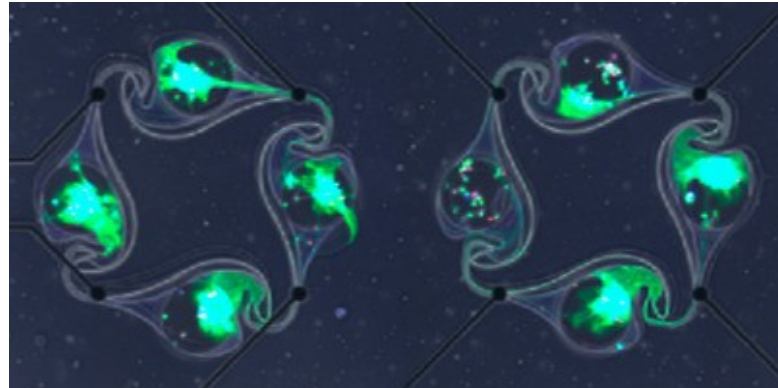


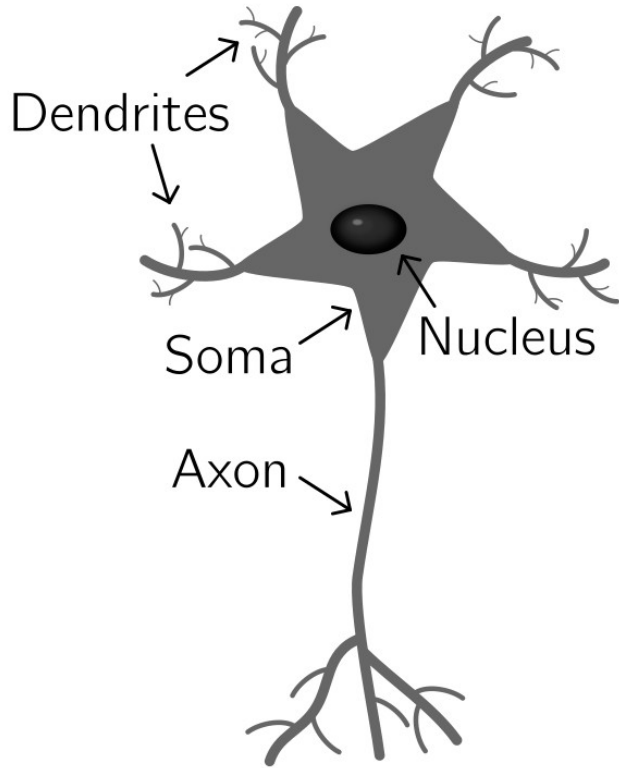
Reinforcement learning and biological networks

Stephan J. Ihle

Eric and Wendy Schmidt AI-Postdoctoral Fellowship
Biological Science Department
Physical Science Department
Data Science Institute University of Chicago

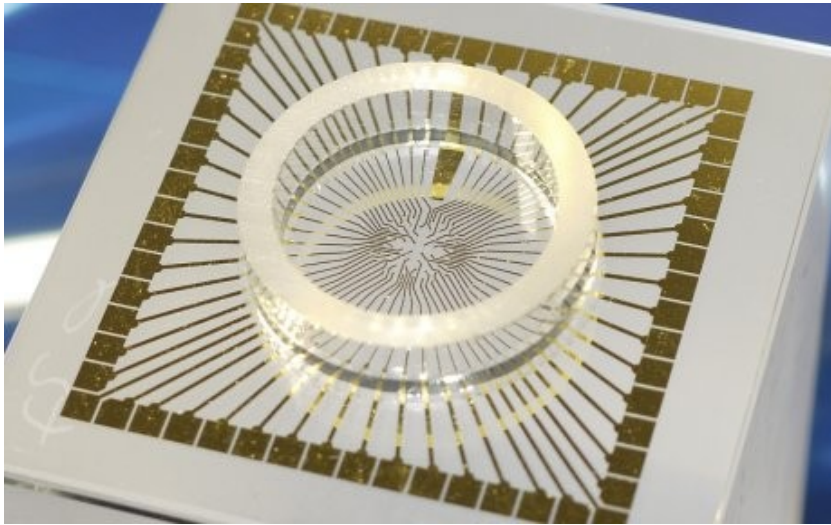


How neurons work



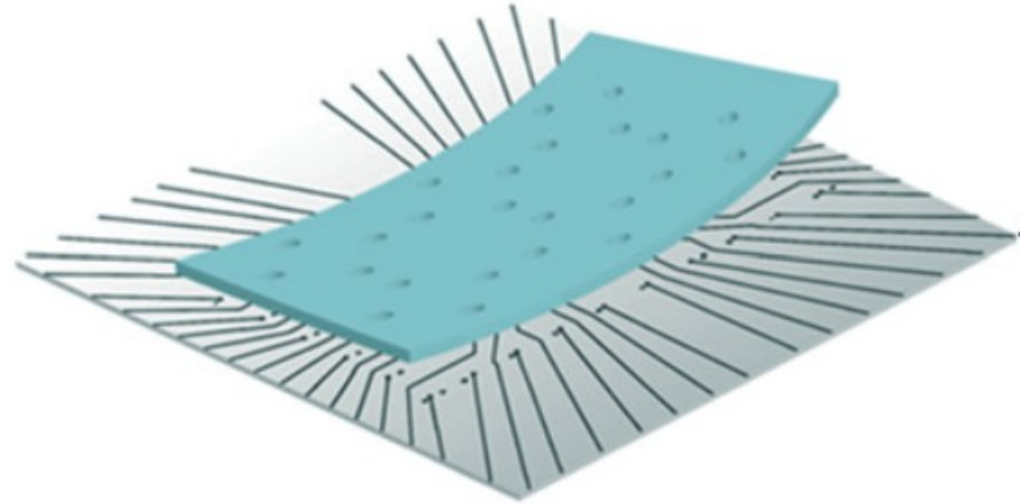
Culturing neurons in a dish

Microelectrode array for neuronal recording



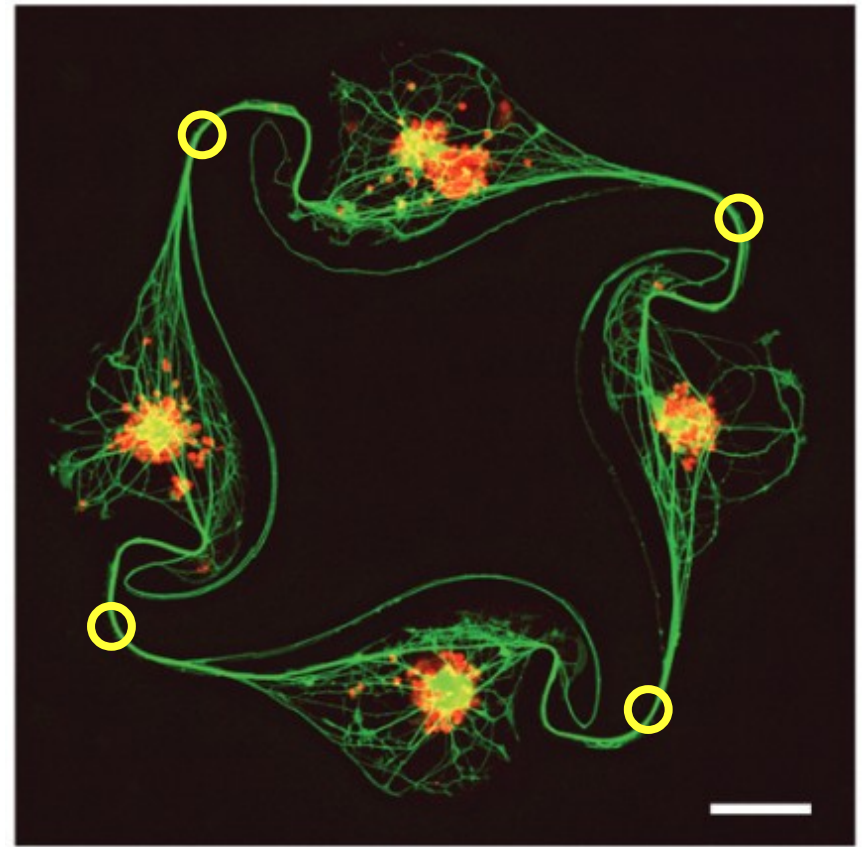
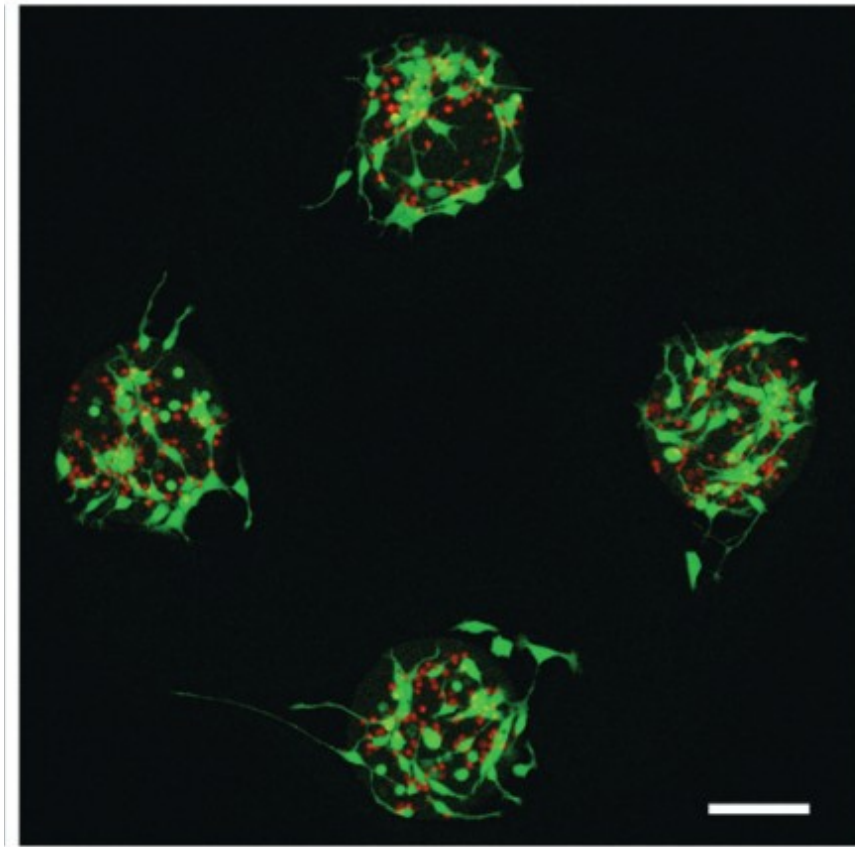
multichannelsystems.com

Microstructure to constrain growth

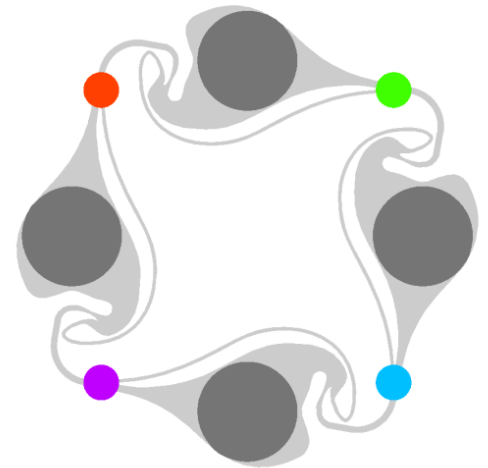
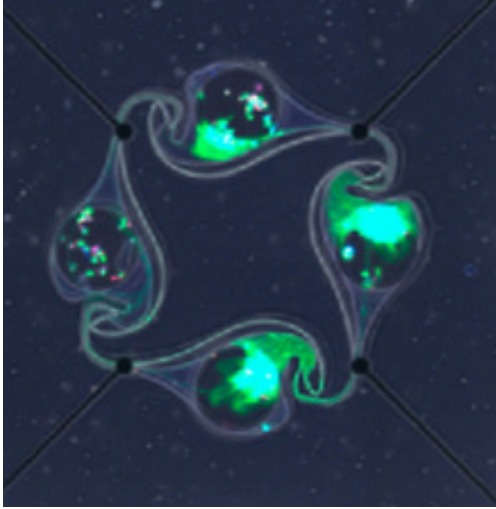


Forró *et al.*, Biosensors and Bioelectronics, 2018

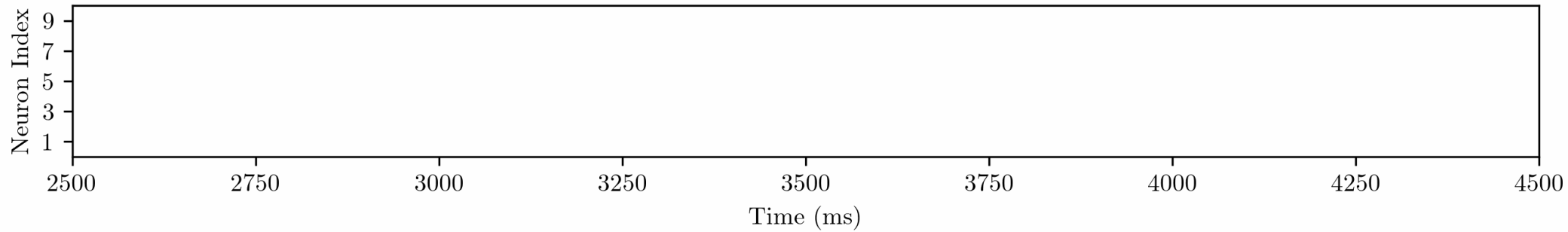
Culturing neurons in a dish



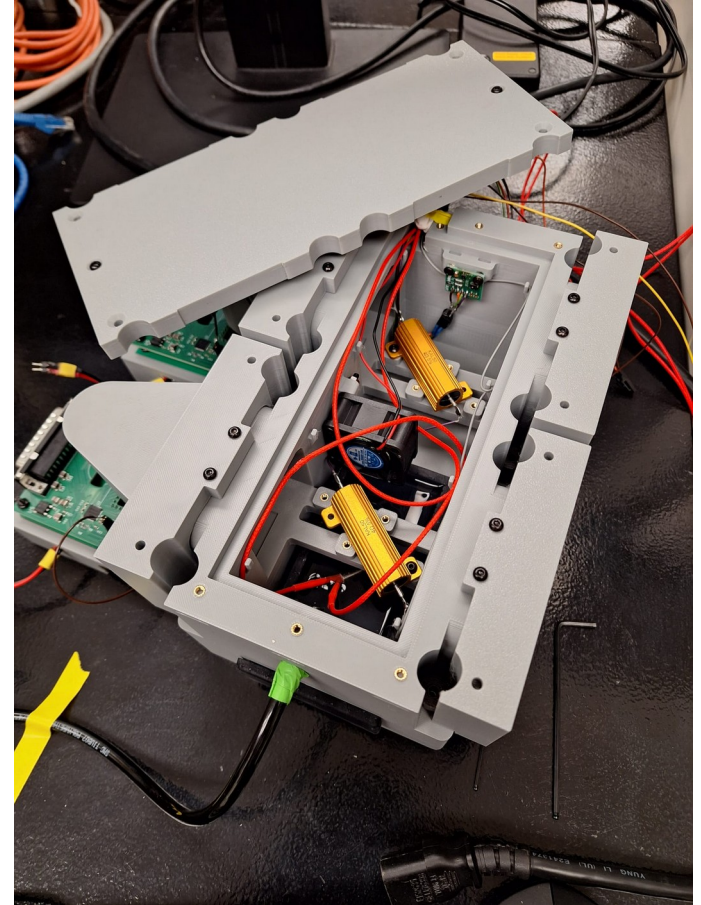
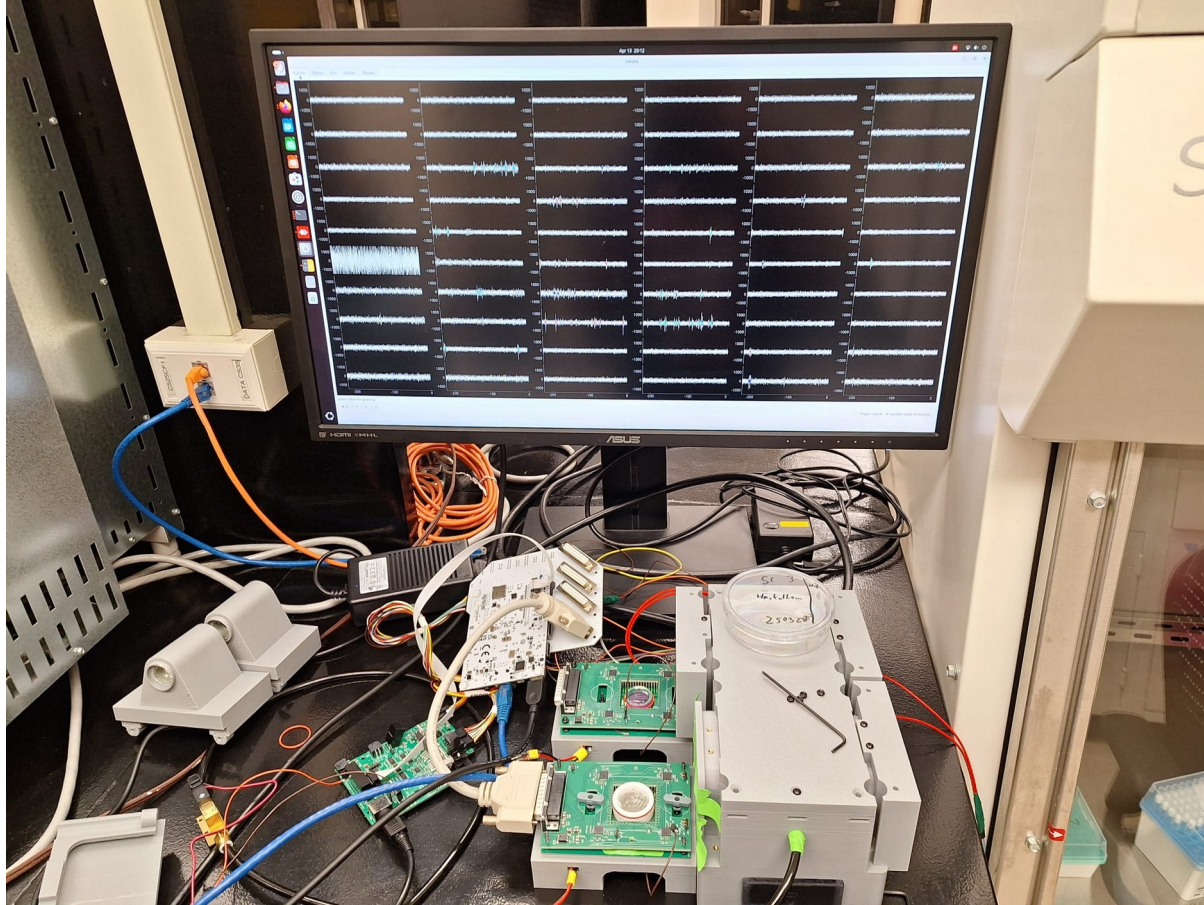
Measuring neuronal activity



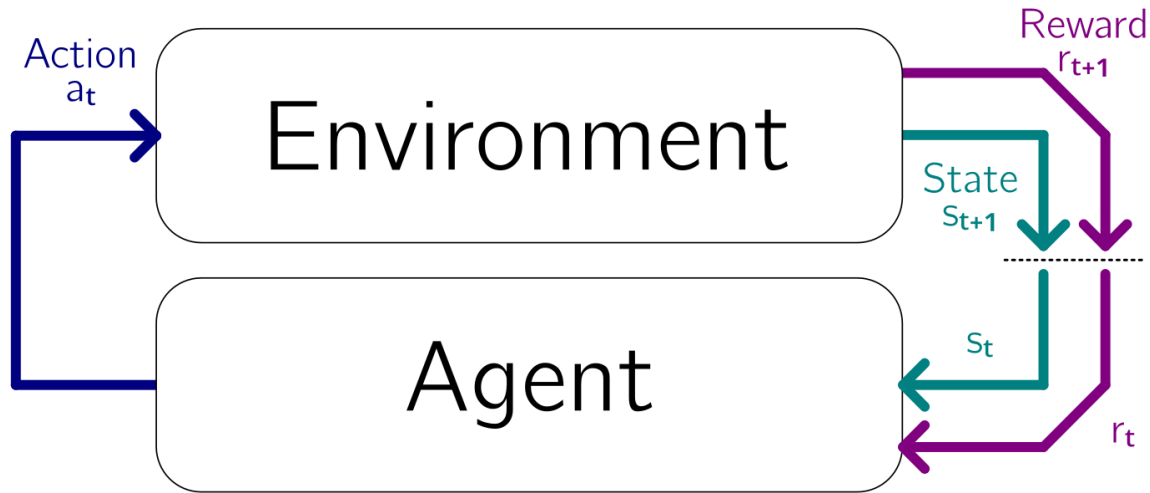
Ihle et al. 2022



Recording setup + incubator



Reinforcement learning

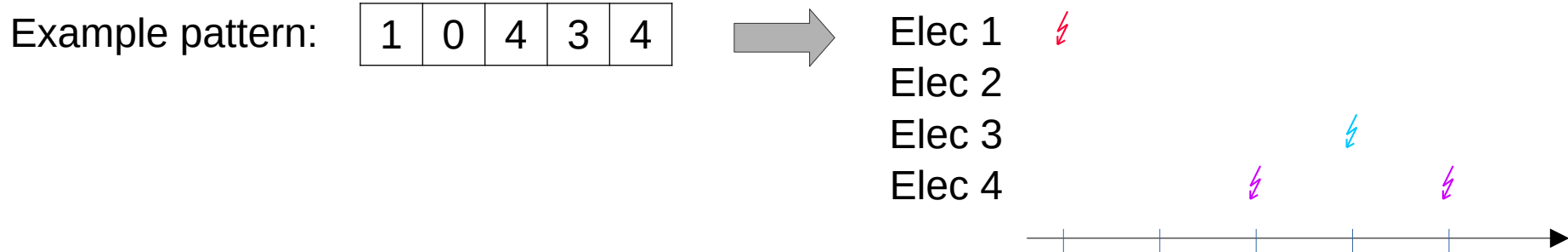
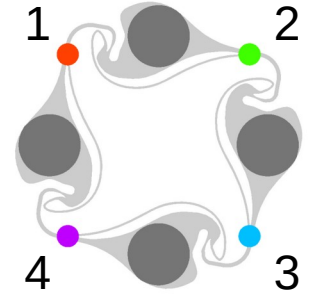


The network is stimulated every 250 ms

What actions should you use?

You have up to 5 slots per action.

Each slot can have value 0 (no stimulus), or numbers 1,2,3,4



Advice: Especially at the beginning, do not use all 5 slots, but only the last 1,2, or 3

What states should you use?

There are different state functions.

Reward functions can be static (easier) or dynamic (harder). Dynamic functions need to be trained on the data but contain in general more useful information.

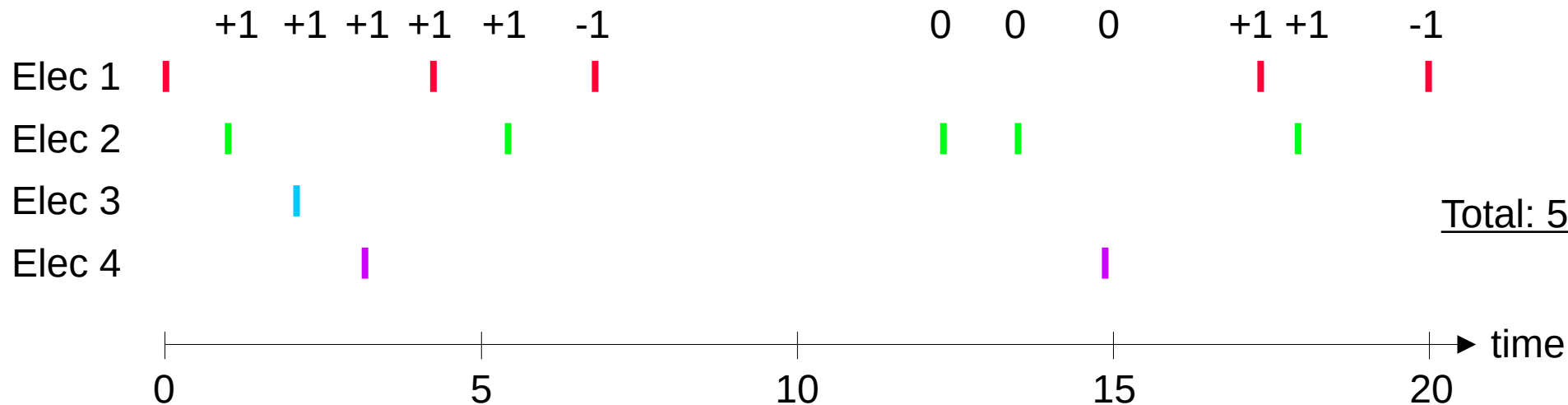
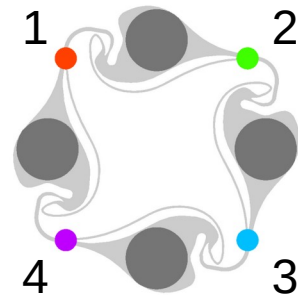
There exists 2 jupyter notebook examples with these two types of functions.

Attention: Your state representation may not be complete
Your state is not necessarily Markovian

Bonus task: Write your own state function (you will likely perform much better)
Can you figure out an approach to use the spiking activity directly as a state?

What reward function should you use?

Linear reward: +1 if 2 spikes are clockwise (and $\Delta t < 5$ ms)
-1 if 2 spikes are counter-clockwise (and $\Delta t < 5$ ms)



Bonus task: Write your own reward function (you will likely perform much better)

Your final score:

Send me your code **by 3pm on 04/28**: ihles@uchicago.edu

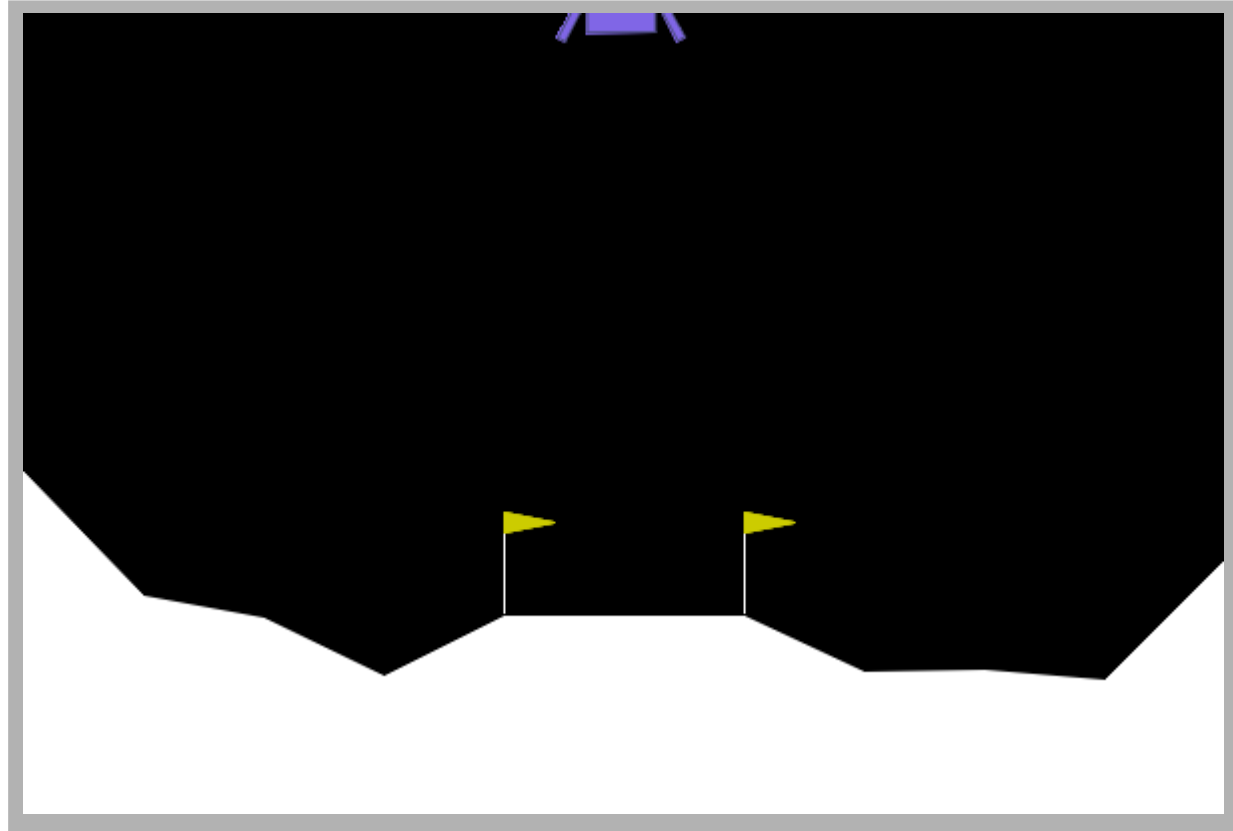
We will run your code directly on our setup and networks. It will run for exactly 2 hours on a network (28800 actions). The final score will be your average reward over the last 30 min of this time period (7200 actions).

For the reward function, we are using: “linear_reward()”

You will be scored based on your performance, your approach, and the presentation.

Important consideration: Use RL gymnasium

<https://gymnasium.farama.org/index.html>



Important consideration: Use RL gymnasium

```
import gymnasium as gym

# Initialise the environment
env = gym.make("LunarLander-v3", render_mode="human")

# Reset the environment to generate the first observation
observation, info = env.reset(seed=42)
for _ in range(1000):
    # this is where you would insert your policy
    action = env.action_space.sample()

    # step (transition) through the environment with the action
    # receiving the next observation, reward and if the episode has terminated or truncated
    observation, reward, terminated, truncated, info = env.step(action)

    # If the episode has ended then we can reset to start a new episode
    if terminated or truncated:
        observation, info = env.reset()

env.close()
```

First, learn reinforcement learning using the RL gymnasium

Look into:

- Multi-armed bandit
- Deep Q-network

Then, you will find example scripts for the environments used in the project in Code/Examples

Important consideration: We are data limited

At the end, you will have **28800 stimulation events** (actions) in total (of which your performance is judged on the last 7200).

Action space size: $5^5 = 3125$ Action space is huge for RL

Rule of thumb: Use small networks!
Number of weights in network should be $\sqrt{\text{samples}}$ $\rightarrow 170$

Use experience replay: https://deeplizard.com/learn/video/Bcuj2fTH4_4
<https://arxiv.org/abs/2007.06700>

Important consideration: We are time critical



- Stimulation:** Requires 10 ms
- Recording:** We look at 20 ms of neuronal activity
- Processing:** Spike detection / clustering takes 20 ms
- Receiving data:** From laboratory computer to RCC (50 ms)
- Prediction:** **You have roughly 100 ms to decide on your next action**
- Sending data:** From RCC to laboratory computer (50 ms)

Use 1 network to decide on your next action and (at least) one network for training!

Why/how to use multiple networks: https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html

Important consideration: Each group has 4 networks

Not all 4 networks are always real networks, some (or all) are simulated

Each network is unique (both simulated and real)

Network assignment changes every 2 hours: Important, **Check yourself, when that happens** (see for that “stim_id” in “info” from the environment)

You need to set your group id and password in auth.py to access these networks (see 5_StaticRealExample.ipynb)

Important consideration: Information in info!

```
[10]: state, reward, terminated, truncated, info = env.step(action)
```

Apply action to environment

```
[11]: state, reward
```

```
[11]: (array([-1.          ,  0.36726528, -0.11623551,  0.          ]), 2)
```

state reward

```
[12]: info
```

Spike times since stimulus [ms]

```
[12]: {'spikes': array([2.09947933, 5.47157396, 5.80934131, 6.43806268]),  
      'elecs': array([0, 3, 0, 1]),  
      'action': array([1, 0]),  
      'missed_cyc': 0,  
      'stim_id': 102,  
      'simulated': True,  
      'comment': 'none'}
```

spike location (electrode ID)

applied action (may be different then applied)

tells you, if you were on time (should be 0)

from 0 to 28799 (timing of the stimulus)

If TRUE: simulated network. If False: real network

Tells you, if something went wrong

Happy Hacking!

Questions?