FAR.AI

Reverse-engineering a neural network that plans

a mesa-optimizer model organism

Mohammad Taufeeque
Aaron D. Tucker
Adam Gleave
Adrià Garriga-Alonso

Planning & Optimization

Plan
Sequence of actions to reach from a start state to a goal state (hopefully optimally)

Search / Optimization
algorithm that considers many plans & picks the best one according to an evaluation criteria

Grand Master Planning in Chess

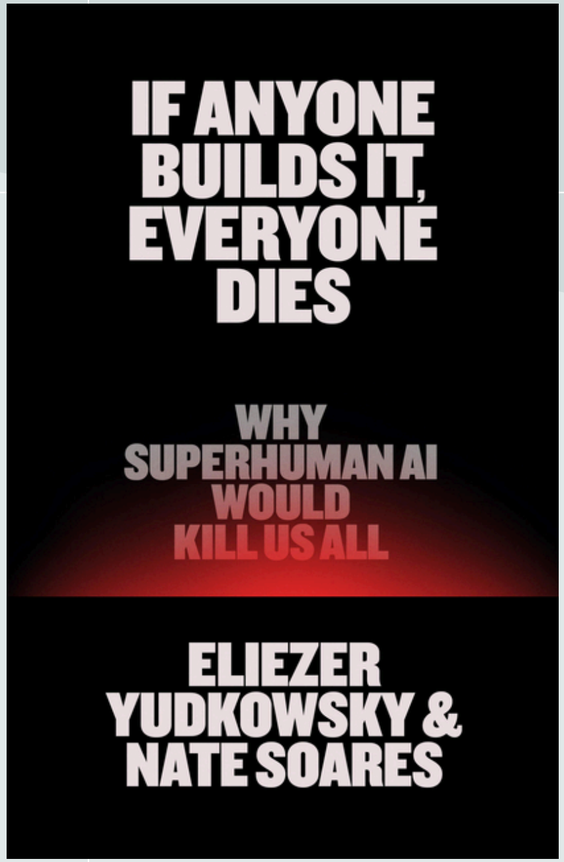
Source: https://www.youtube.com/watch?v=Eo4ETnTotc0&t=49s

Why care about planning in NN?

AI as optimizer threat model

- Classic threat model: AI as a ruthless optimizer towards a goal
  - **Evolution analogy**
  - “You don’t get what you train for”

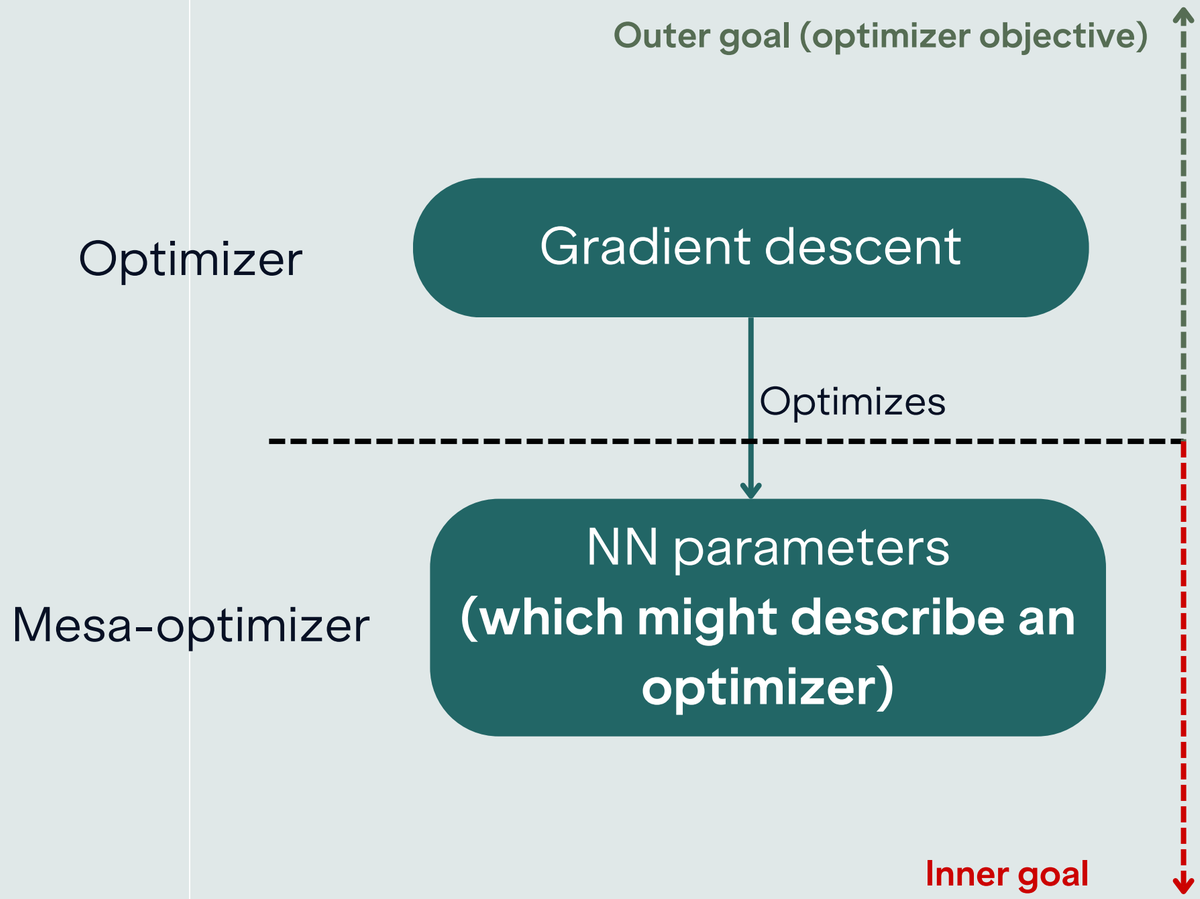
- In a machine learning context, **mesa-optimizer**
- **Theorized by Evan Hubinger et al. 2019**



It’s not using the term, but this book is about mesa-optimizers.

FAR.AI

Inner alignment problem



- Mesa-optimizer: an AI
  - which optimizes a goal
  - is itself the product of an optimizer (gradient descent)
- Since optimizer is learned, **its goal** might differ from **optimizer objective**
  - **Inner alignment**: ensure mesa-optimizer's objective is desirable

FAR.AI

Research objective:

Be able to reliably tell:

- whether a NN is an optimizer
- what its true goal is

Sokoban

FAR.AI

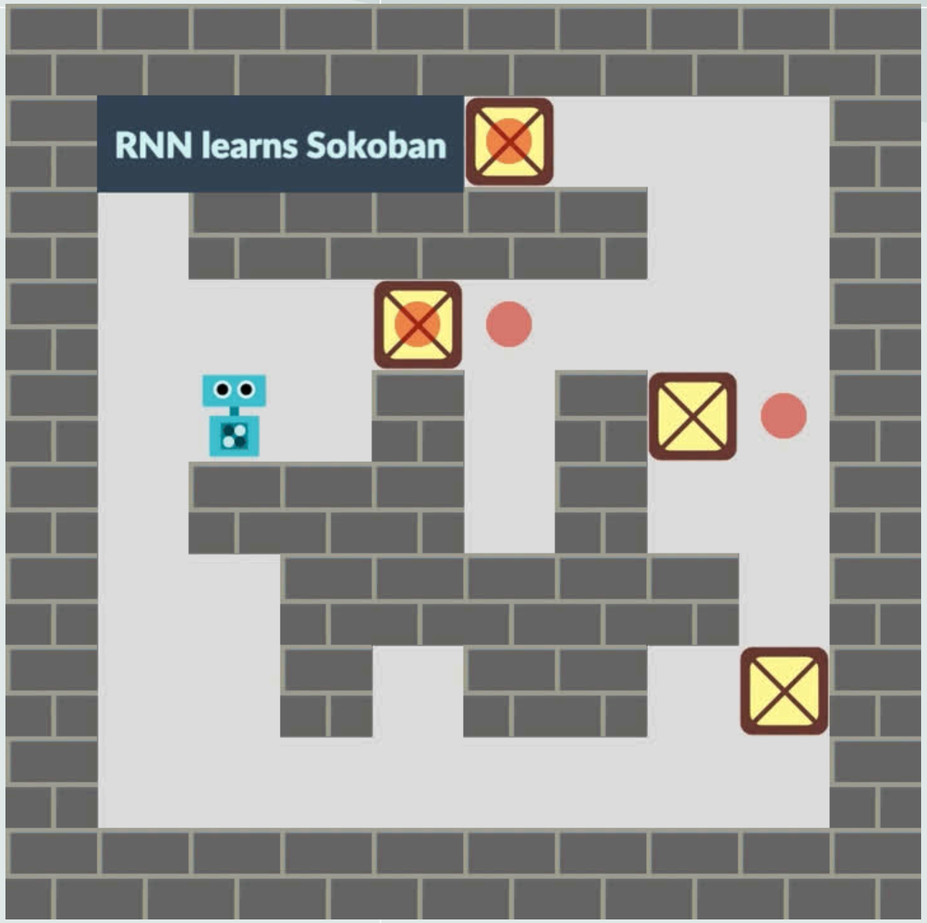
Sokoban

Sokoban is a game where the player has to put all the boxes on the targets

- Can only push boxes by moving
- Multiple boxes cannot be pushed simultaneously

Rewards:

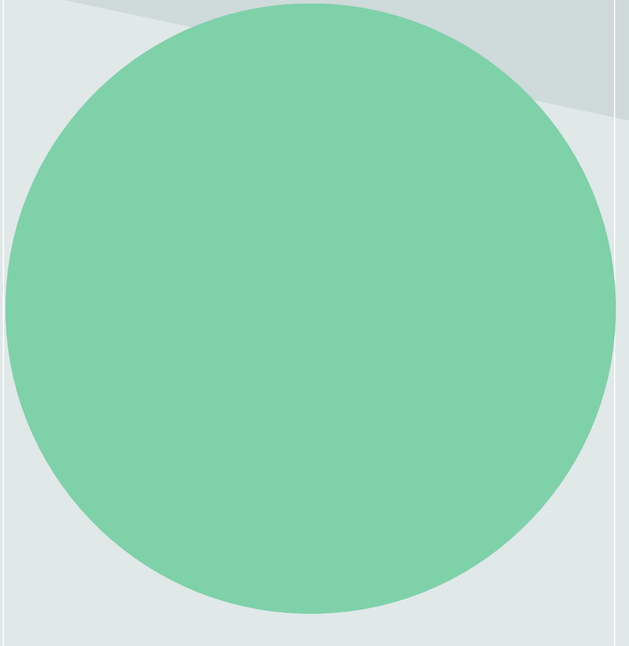
- +1: for putting a box onto any target
- -1: for removing boxes from target
- -0.1: for every step
- +10: for solving the level



FAR.AI

Why Sokoban?

- Sokoban is still used as a benchmark for planning algorithms & RL
- The trained RNN shows planning-like capabilty
- The network is small which makes it easier to do interpretability research



Sokoban RNN
1.28 M

GPT-2 small
124 M

FAR.AI

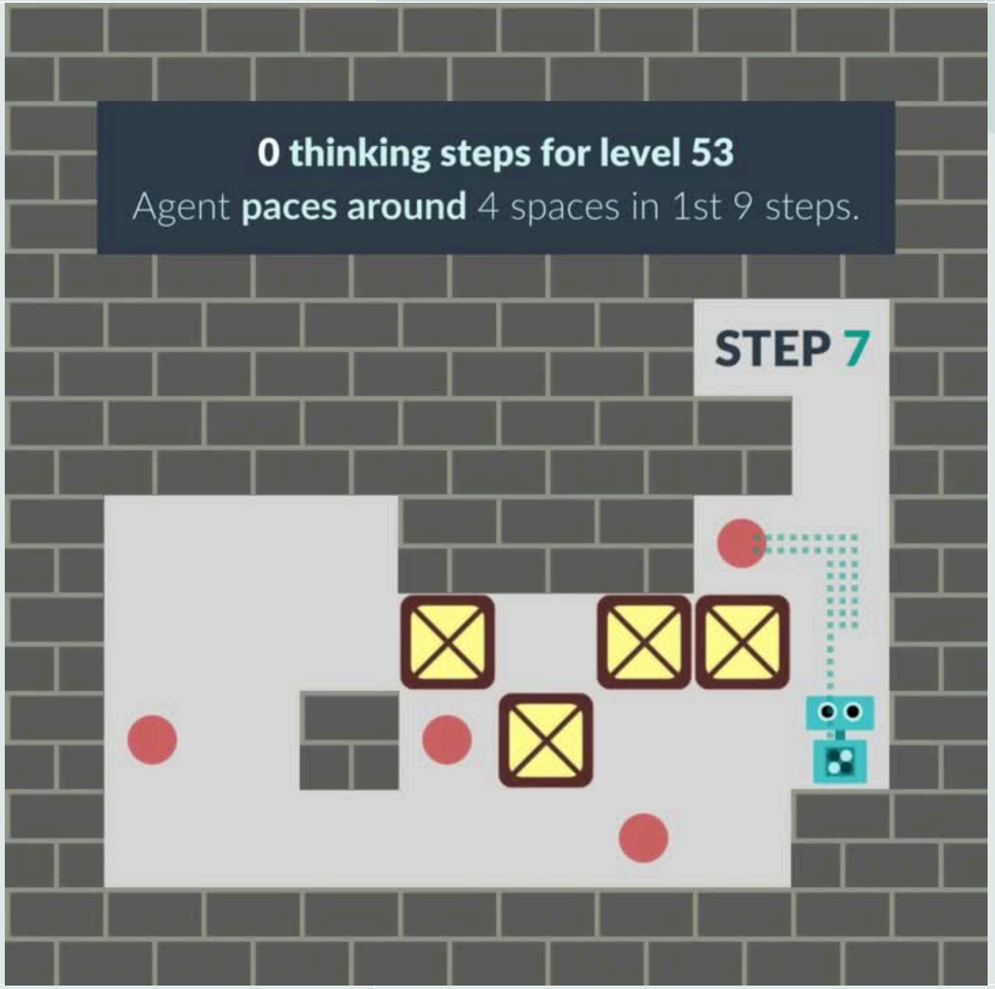
Behavioral evidence of planning

Agent thinks before solving the level

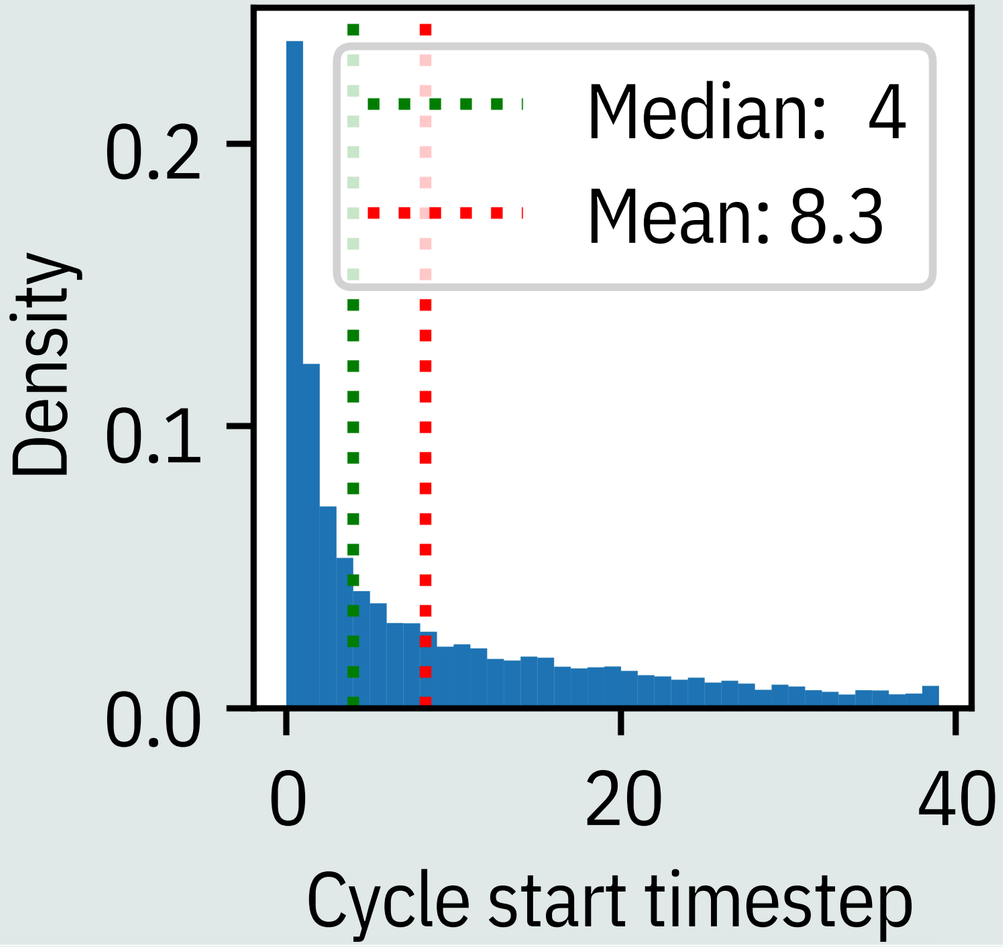
Agent paces around in cycles before starting to move the box

waiting for the eureka moment when it thinks it can solve the level

The agent learned to pace during RL training to get more time (& compute) to solve a level



Cycles occur at the start of a level



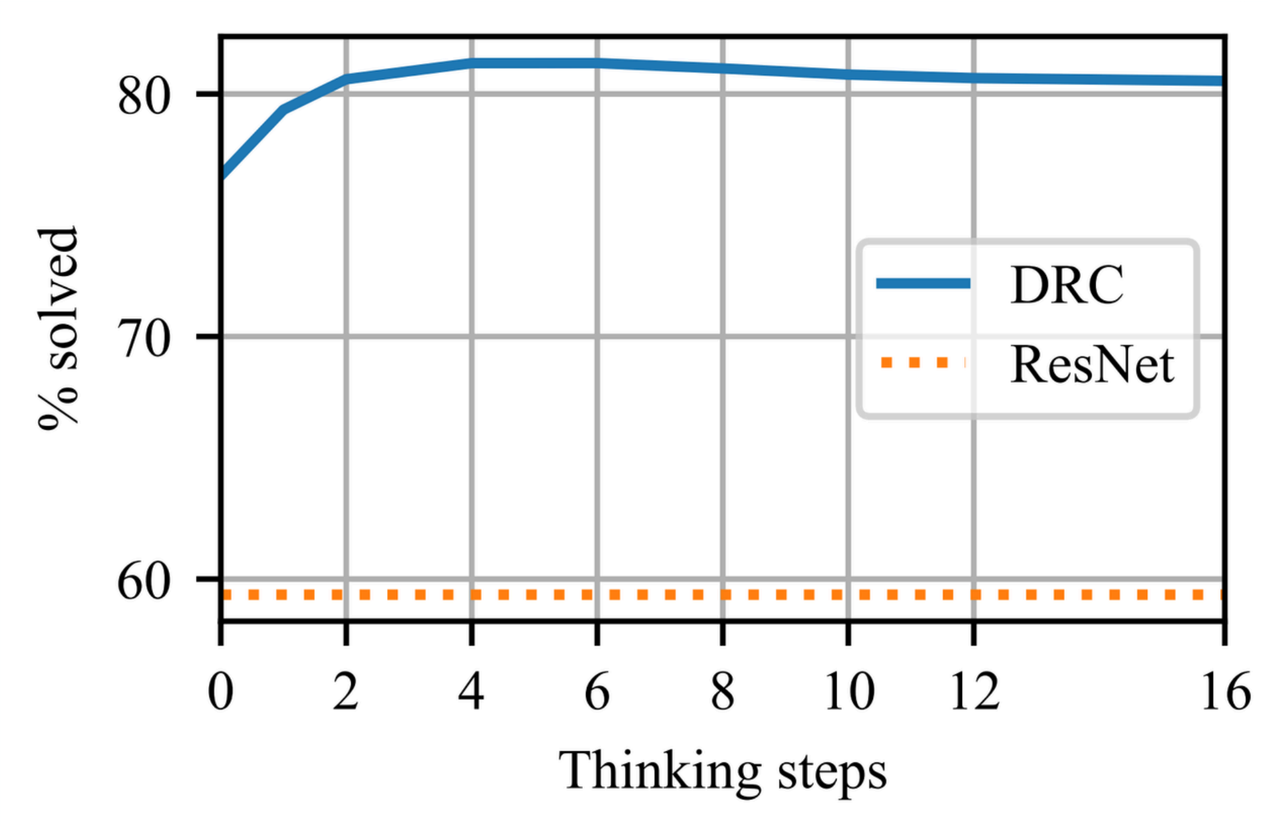
FAR.AI

Artificial Thinking Steps

During evaluation, give the starting position multiple times to the agent before making its first move

This allows the network to utilize more compute during inference to play the level

Thinking improves performance



Thinking steps let the network solve a higher number of levels

The agent learned to pace during RL training to get extra compute

Extra compute results in better performance

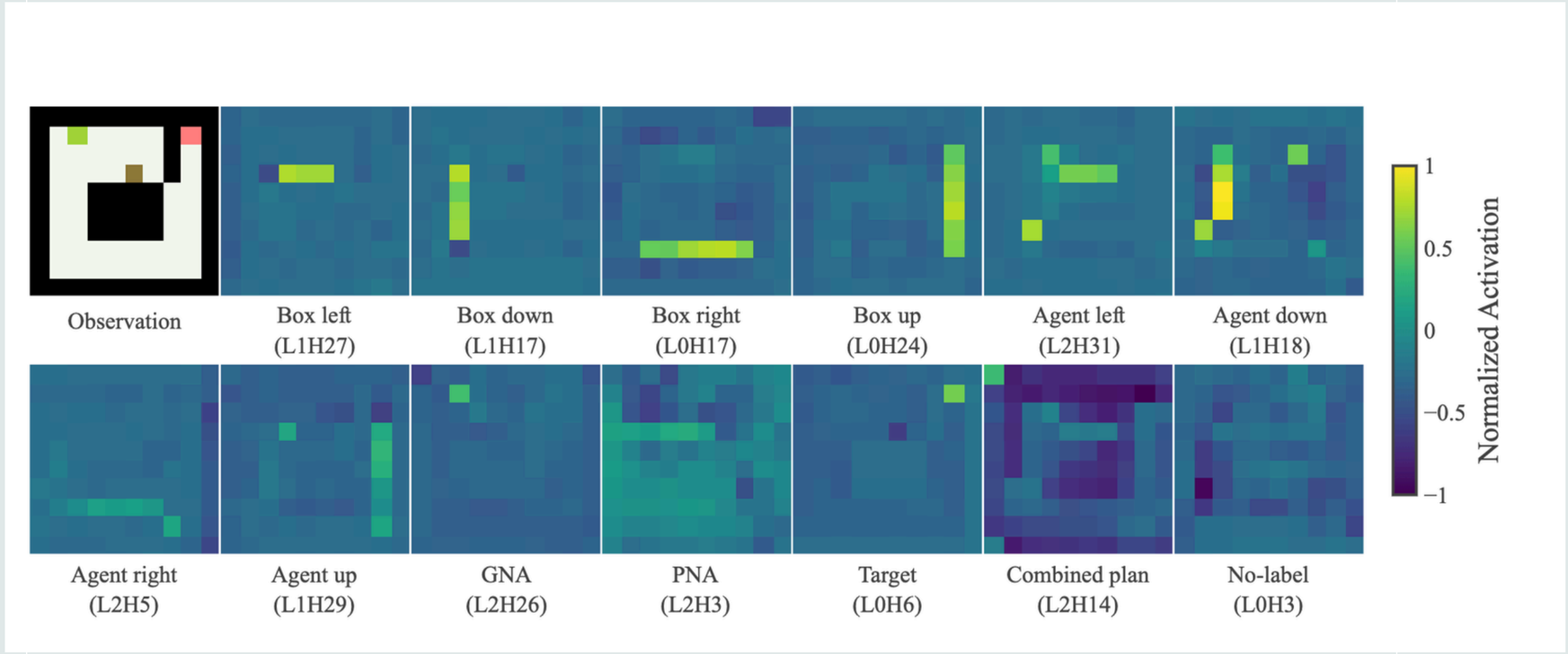
Internal representation

Identified channel groups

Table 1: Channel groups, their definitions and counts for each direction (up, down, left, right).

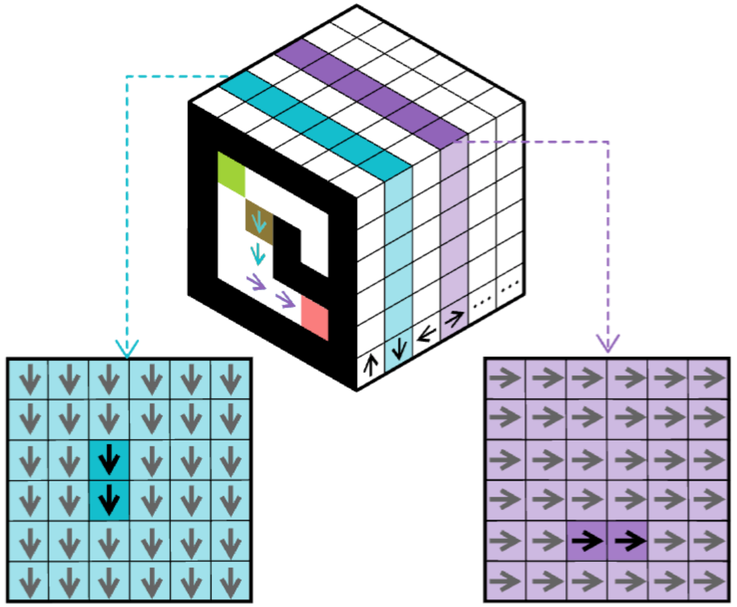
Group	Definitions	Channels
Box-movement	Path of box (short- and long-term)	20 (3, 6, 5, 6)
Agent-movement	Path of agent (short- and long-term)	10 (3, 2, 1, 4)
Grid Next Action (GNA)	Immediate next action, represented at agent square	4 (1, 1, 1, 1)
Pooled Next Action (PNA)	Pools GNA to represent next action in all squares	4 (1, 1, 1, 1)
Entity	Target, agent, or box locations	8
Combined path	Aggregate 2+ directions from movement channels	29
No label	Difficult to interpret channels	21

Identified channel groups

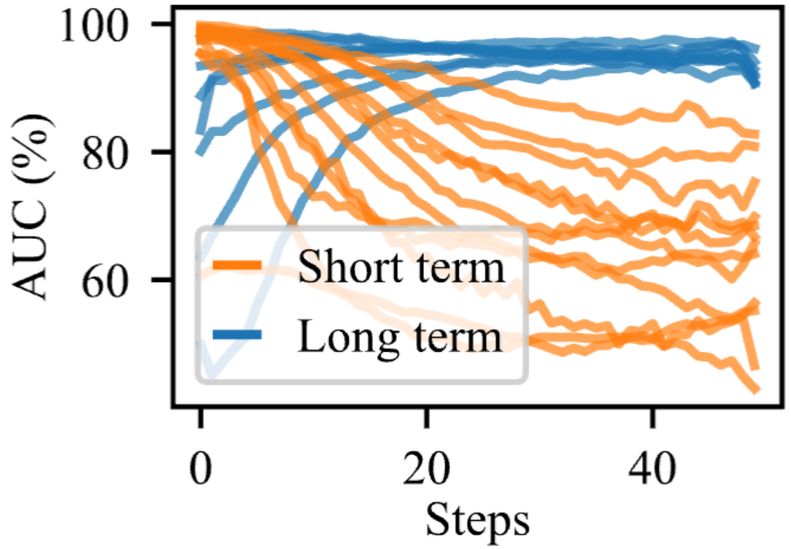


FAR.AI

State-action representation



Representing plan “go down then right”



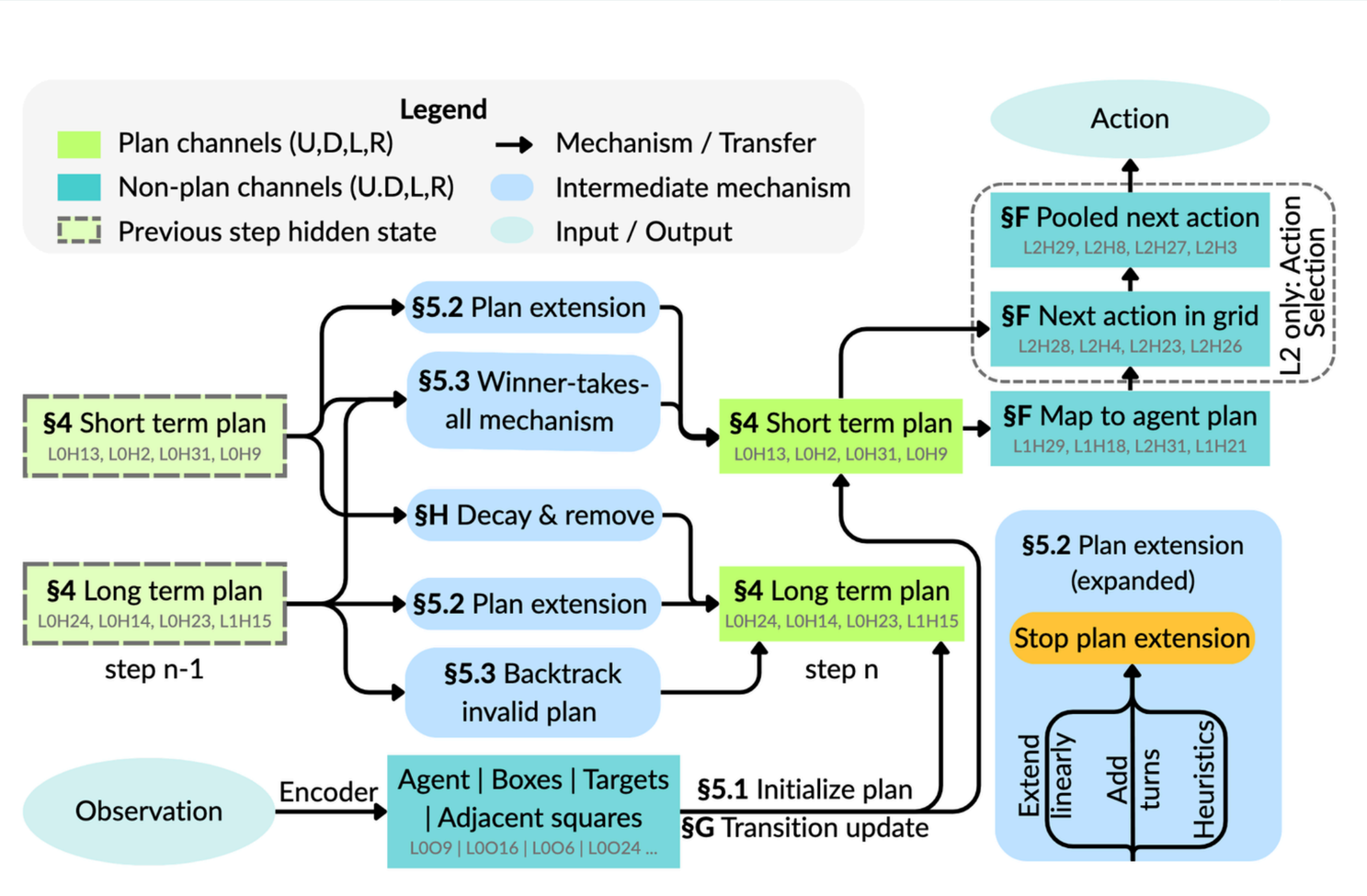
Short-term channels predict actions very well in short-term, long-term ones don’t result contradictives

FAR.AI

Evidence for path channels: causality

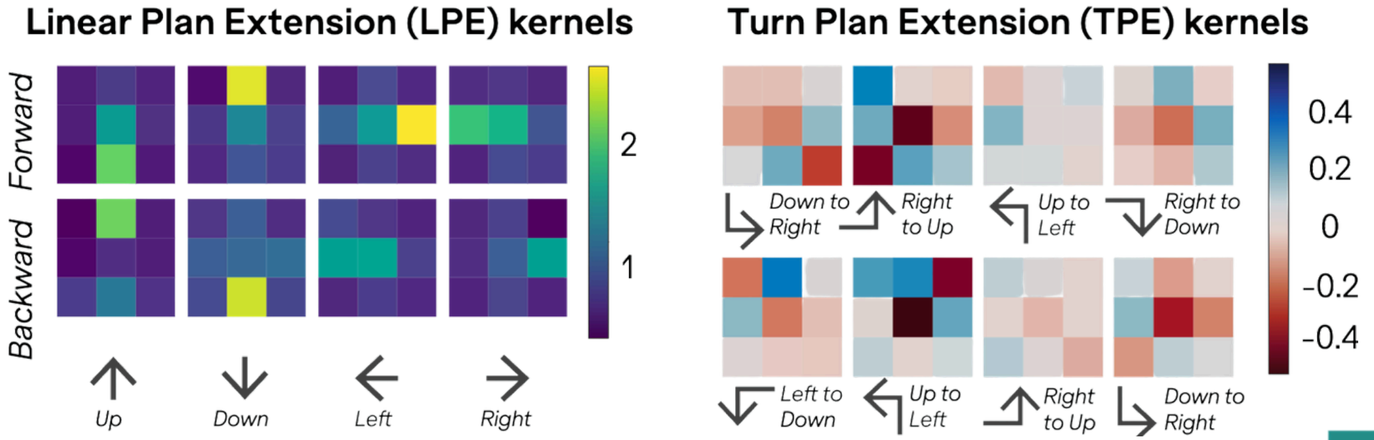
Group	Score (%)
Pooled Next Action (PNA)	99.7 ± 0.2
Grid Next Action (GNA)	98.9 ± 0.4
Box- and agent-movement	88.1 ± 1.9
Box-movement	86.3 ± 2.1
Agent-movement	53.2 ± 2.1
Probe: box movement	82.5 ± 2.5
Probe: agent movement	20.7 ± 0.7

Identified algorithm

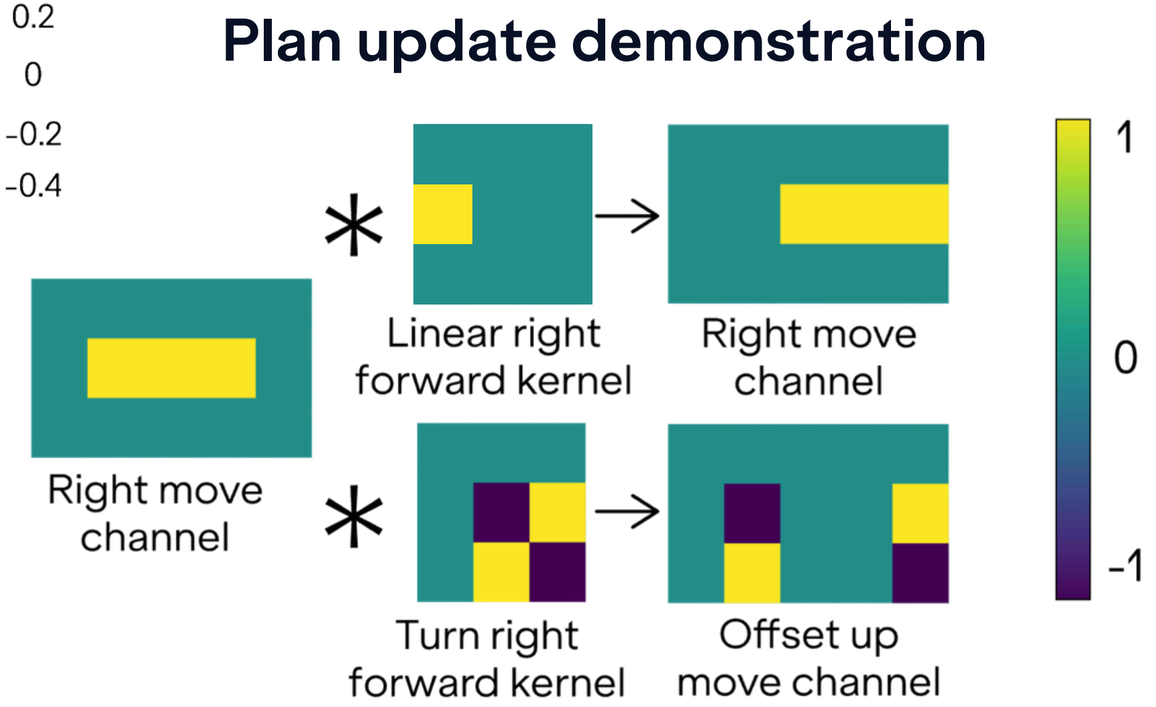


FAR.AI

Transition (world) model

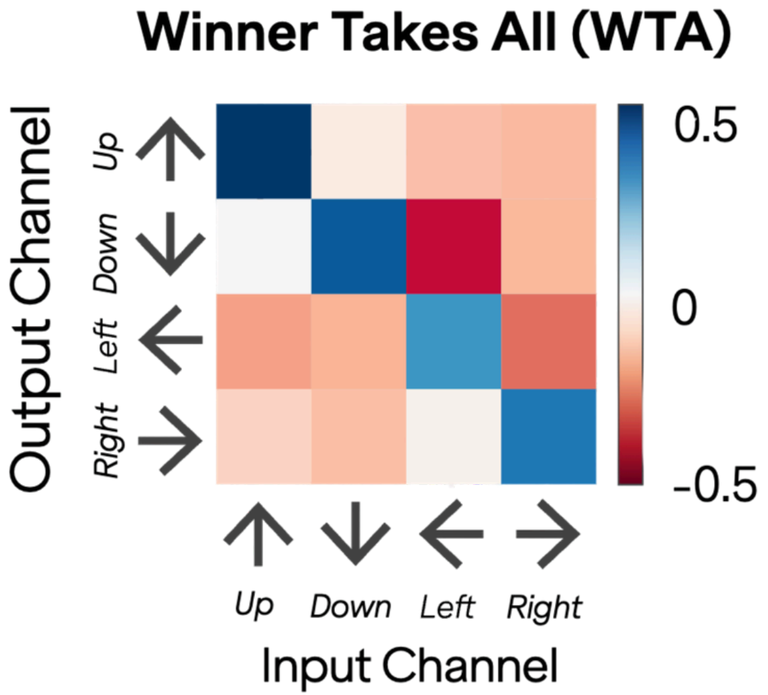


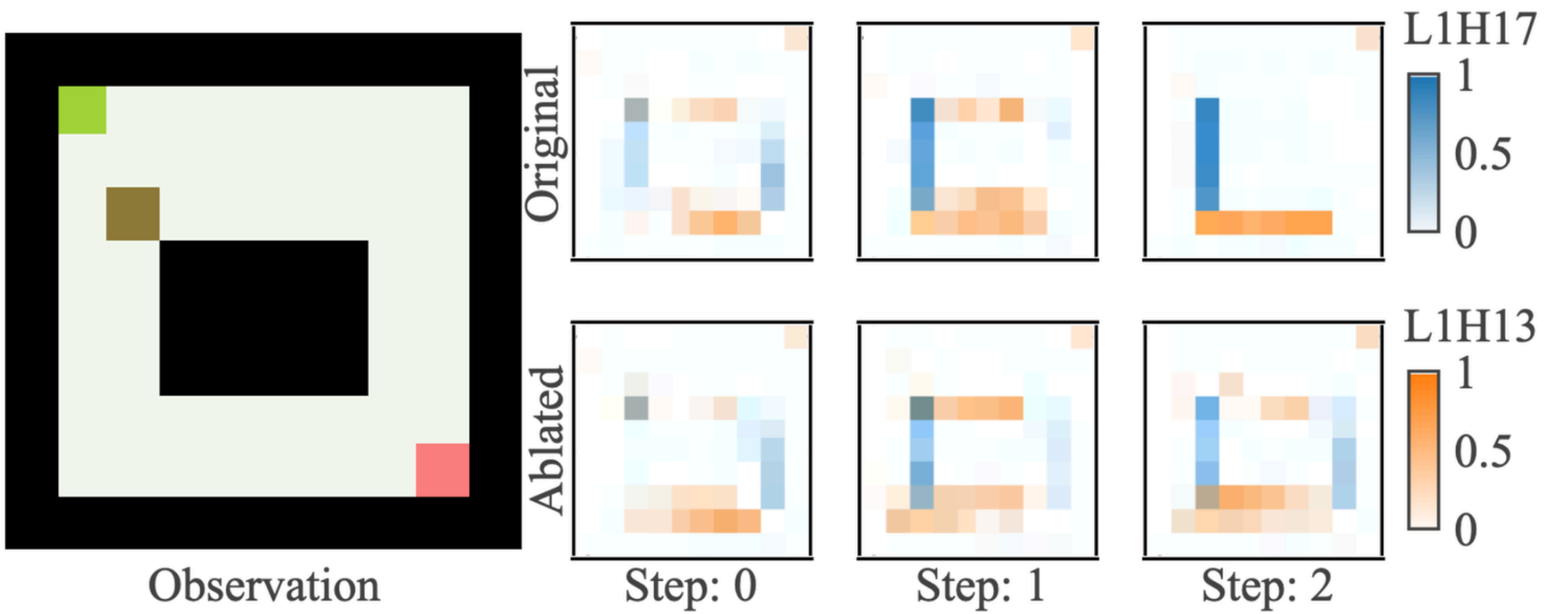
Kernels encoding how to update plan: first linearly along each direction and then adding turns near walls or squares of other orthogonal directions



FAR.AI

Value function (plan selection)



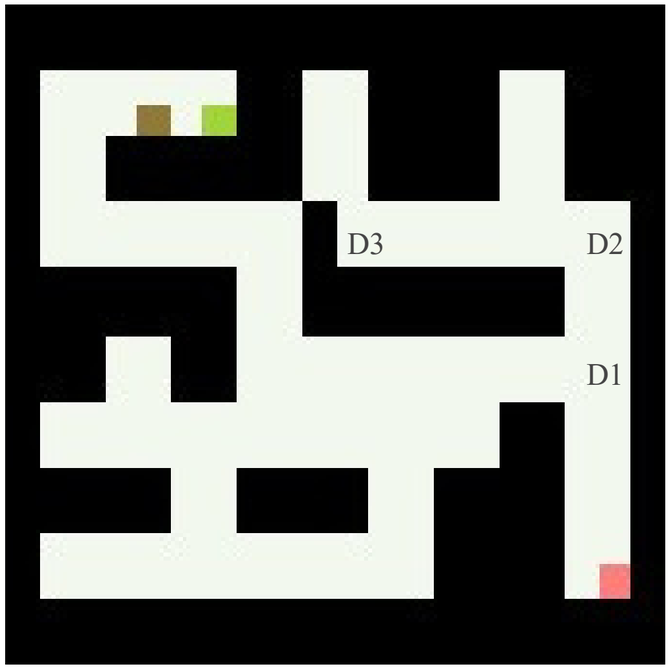


Different channel directions inhibit each other which results in one direction “winning”

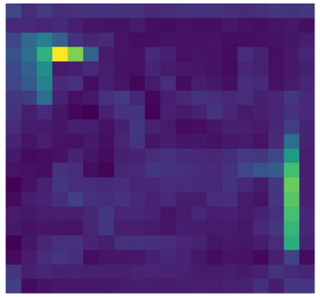
Down direction “wins” at the box square to select plan to go down first. Upon ablating kernels b/w directions, network is not able to select the plan

FAR.AI

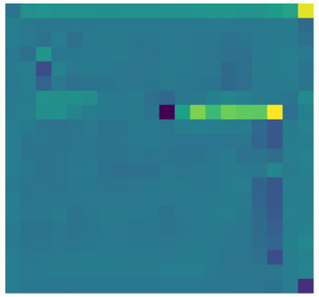
Backtracking Mechanism



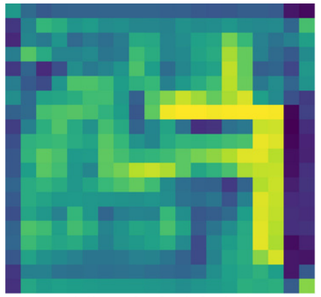
(a) Observation



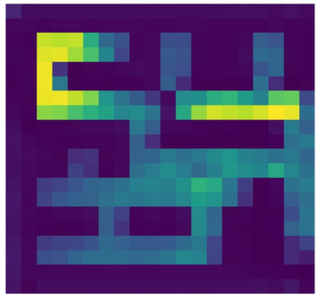
(b) All box channels
step 1 tick 0



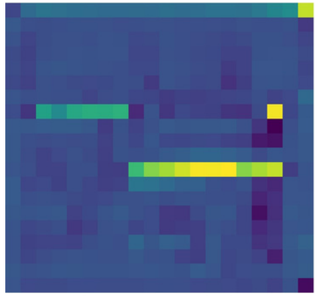
(c) L2H9 (right)
step 5 tick 1



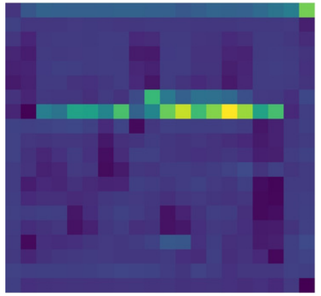
(d) L2I9 (right)
step 5 tick 1



(e) L2O9 (right)
step 5 tick 1



(f) L2H9 (right)
step 9 tick 1



(g) L2H9 (right)
Abl. step 9 tick 1

Path channel activations for a plan from D1->D2->D3, backtracking from a wall by propagating negative activations backward. Ablating negative activations makes the agent ignore the wall.

FAR.AI

Conclusion

- The network shows planning-like capability: more compute => better performance
- It learned this capability through RL training
- We can identify the plans, transition model, and sort of a value function
  - Concrete example of a mesa-optimizer
- Interpretability can identify fairly complicated algorithms
  - Test case for future interpretability automation?

Thank you!

Correspondence:

- taufeeque@far.ai
- adriagarriga@gmail.com

FAR.AI