Structured Representations in Maze-Solving Transformers

Michael I. Ivanitskiy[1], Alex F. Spies[23], Tilman Räuker

Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn[1], Katsumi Inoue[3], Samy Wu Fung[1]

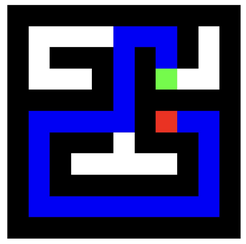
[1]Colorado School of Mines [2]Imperial College London [3]National Institute of Informatics

Why Study Maze-solving Models?

- **Small models** are easier to analyze and can help us to develop techniques for investigating large models
- **Spatial structure**: visually intuitive to interpret
- **Internal search:** solving mazes is a good test to see if models can plan, which has implications for AI safety

Our Setup

- Train transformers to solve mazes, or follow paths
- Inputs are tokenized mazed, including the shortest path
- Setup is the same as a normal next-token prediction task



Example of a maze. The model must output the shortest path from the start to the goal

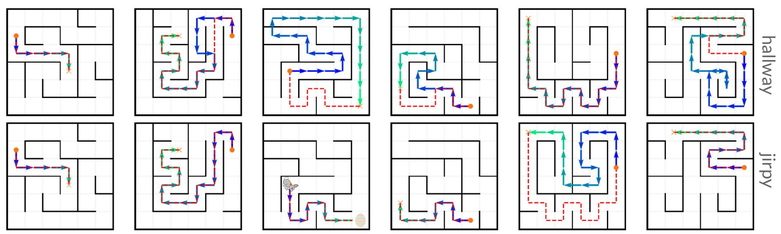
```
<ADJLIST_START> (0,0) <-> (1,0) ; (2,0) <-> (3,0) ; (4,1) <-> (4,0) ;
(2,0) <-> (2,1) ; (1,0) <-> (1,1) ; (3,4) <-> (2,4) ; (4,2) <-> (4,3) ;
[...] (3,1) <-> (3,2) ; (1,3) <-> (1,4) ; <ADJLIST_END>
<ORIGIN_START> (1,3) <ORIGIN_END> <TARGET_START> (2,3) <TARGET_END>
<PATH_START> (1,3) (0,3) (0,2) (1,2) [...] (2,3) <PATH_END>
```

A truncated example of the input provided to a model at train time (the path is omitted during inference)

The Models

We focus our analyses on two performant toy models:

- **"jirpy"** - trained to solve mazes, high data variety
  - 9.6M params, 12 layers, 16 heads
- **"hallway"**- trained to follow "forkless" paths
  - 1.2M params, 6 layers, 4 heads

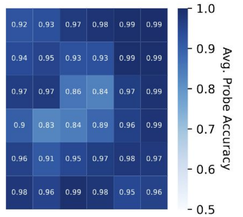


Trained models were consistently able to solve the tasks. The best models had >95% accuracy on the validation set

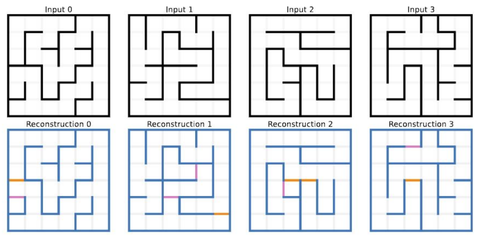
Models Learn Structured Representations

Linear "World Models" are Learned

In our jirpy model we can faithfully decode the full maze from a single token's* residual stream state at Layer 2!



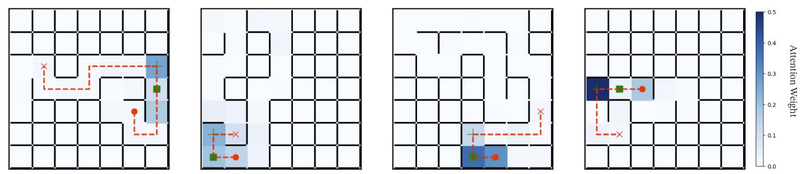
Average accuracy of the 4-wall probes. *Using <PATH_START> residuals from jirpy at layer 2



Some examples of reconstructed mazes (not cherry-picked). Here orange → missed and pink→ hallucinated

Adjacency Heads Encode Valid Neighbors

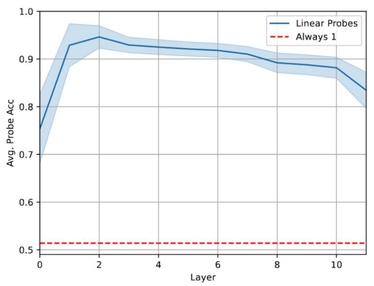
Looking at per-head logit contributions we identify heads in the hallway model which capture valid neighbor structure



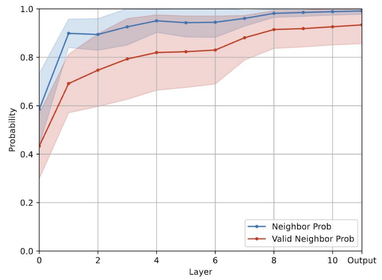
Logit Contributions of the "Adjacency Head" on various mazes. Here the dashed line from the red circle to the red cross indicates the true path, and the current position is the green cube.

Models Capture Neighbor Structure Early

- Linear probing works best at early layers
- TunedLens provides further evidence that by layer 2 models already encode the maze's topology



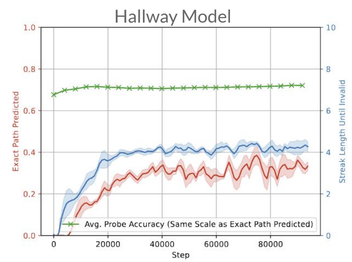
Accuracy of the linear world model probes peaks at layer 2. This held across similar models.

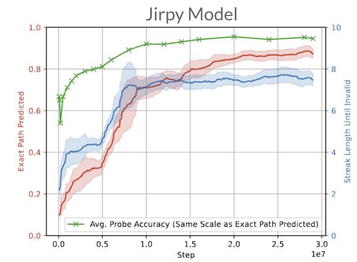


We use TunedLens to decode the residual stream, finding that models already encode neighbor and valid-neighbor info in early layers

When do World Models Form?

- Training to follow paths (hallway) rather than solve mazes (jirpy) doesn't lead to World Models (WMs)
- Formation of World Models coincides with our models learning to solve the mazes





World Model accuracy (green) against model accuracy throughout training,

Takeaways

- Linear WMs form at early layers
- Specialized Attention heads respect maze topology
- WMs don't form in transformers which only need to follow paths (rather than solve mazes)
- During training, WMs become more accurate when Transformers get better at solving mazes.

Next Steps

- Do the structures we find play a causal role? If so, what does the circuit they are part of look like?
- Can we intervene on the model in order to make it better at satisfying constraints we like?
- Can we retarget the model to act sensibly with respect to different goals?

Author Notes

- The code and models required to reproduce these experiments is available on our github (linked above) as well as the libraries we created for Maze Datasets and Maze Transformers
- Not all transformers form linear world models (but the best ones do)
- We have since trained models which leverage relative position encodings and generalize much more strongly to longer and shorter mazes than those seen during training.