A BEHAVIOURAL AND REPRESENTATIONAL EVALUATION OF GOAL-DIRECTEDNESS IN LANGUAGE MODEL AGENTS

Raghu Arghal[1], **Fade Chen**[2], **Niall Dalton**[8], **Evgenii Kortukov**[3], **Calum McNamara**[4], **Angelos Nalmpantis**[5], **Moksh Nirvaan**[6], **Gabriele Sarti**[7], **Mario Giulianelli**[8*]

[1]University of Pennsylvania [2]New York University [3]Fraunhofer HHI
[4]Indiana University, Bloomington [5]TKH AI [6]Independent
[7]Northeastern University [8]University College London

ABSTRACT

Understanding an agent's goals helps explain and predict its behaviour, yet there is no established methodology for reliably attributing goals to agentic systems. We propose a framework for evaluating goal-directedness that integrates behavioural evaluation with interpretability-based analyses of models' internal representations. As a case study, we examine an LLM agent navigating a 2D grid world toward a goal state. Behaviourally, we evaluate the agent against an optimal policy across varying grid sizes, obstacle densities, and goal structures, finding that performance scales with task difficulty while remaining robust to difficulty-preserving transformations and complex goal structures. We then use probing methods to decode the agent's internal representations of the environment state and its multi-step action plans. We find that the LLM agent non-linearly encodes a coarse spatial map of the environment, preserving approximate task-relevant cues about its position and the goal location; that its actions are broadly consistent with these internal representations; and that reasoning reorganises them, shifting from broader environment structural cues toward information supporting immediate action selection. Our findings support the view that introspective examination is required beyond behavioural evaluations to characterise how agents represent and pursue their objectives.

1 INTRODUCTION

Attributing goals to agents helps explain and predict their behaviour and provides a useful abstraction for reasoning about agency. This topic has received attention in fields as varied as philosophy (Davidson, 1973; Dennett, 1990), psychology and neuroscience (Baker et al., 2009; Schultz et al., 1997) economics and decision theory (von Neumann & Morgenstern, 1944; Savage, 1948), and reinforcement learning (Bellman, 1966; Ng & Russell, 2000). More recently, determining when and in what sense goal attributions are warranted has become a pressing concern for LLM-based agents (Xu & Rivera, 2024; MacDermott et al., 2024; Everitt et al., 2025; Goldstein & Lederman, 2025; Mazeika et al., 2025), particularly from an AI safety perspective (Naik et al., 2025; Wentworth & Lorell, 2025; Marks et al., 2025; Li et al., 2025; Summerfield et al., 2025).

A natural way to attribute goals to an agent is *behavioural evaluation*, i.e., assessing the agent's actions relative to some hypothesised goal, particularly compared to an optimal policy (Xu & Rivera, 2024; Everitt et al., 2025). However, purely behavioural measures face fundamental theoretical, practical, and philosophical challenges (Bellot et al., 2025; Rajcic & Søgaard, 2025; Chalmers, 2025). Agent capabilities may act as confounders for behavioural measures, as consistent failure may reflect capability limitations rather than lack of goal-directed behaviour. Moreover, behavioural monitoring alone may be insufficient to guarantee alignment because a system with

[*]Authors are listed in alphabetical order, except for the last two; see Statement of Author Contributions. Correspondence to: m.giulianelli@ucl.ac.uk. Code is available at https://github.com/SPAR-Telos/interp and an interactive viewer for decoded grids (*cognitive maps*) at https://huggingface.co/spaces/project-telos/trace-viewer.

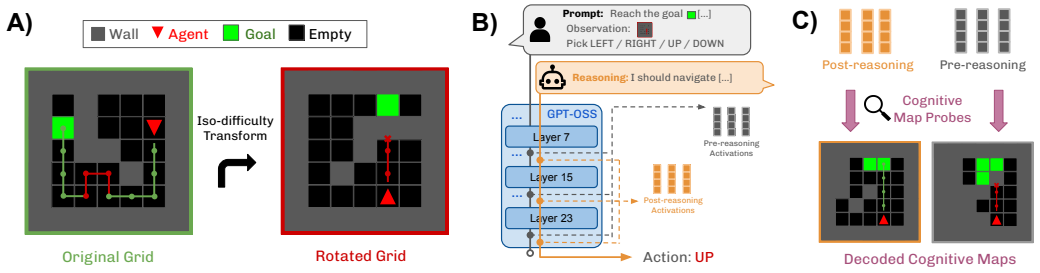


Figure 1: Overview of our goal-directedness analysis. **A:** We evaluate how iso-difficulty transforms affect agent trajectories that agree or disagree with the optimal policy. **B:** We prompt an LLM-based agent to reason and act over the fully-observable grid setup, extracting its pre-and post-reasoning activations at intermediate layers. **C:** We probe the agent's beliefs over goal distance, planned actions and reconstruct *cognitive maps* for the current grid state.

misaligned internal objectives could produce aligned behaviour, or fail a safety-relevant task, when doing so is instrumentally useful (Hubinger et al., 2019; Ngo et al., 2024).

To address these limitations, we propose a framework that combines behavioural evaluation with analysis of internal representations. This enables holistic assessment of goal-directedness as a rich property arising from the interaction of beliefs, planning, and action selection. We study an LLM agent in a fully observable grid world, tasked with navigating to a goal state across grids of varying sizes and obstacle densities. We begin by subjecting the agent to standard capability tests and gradually introduce controlled environment perturbations and multi-goal task structures to measure the generalisability of its goal-directed behaviour. We find sensitivity to task difficulty and goal-like task-irrelevant cues, but robustness to difficulty-preserving transformations and instrumental goals. We then use probing classifiers to test if goal-relevant information can be decoded from the agent's internal activations, before and after reasoning. Through our probing analyses, we are able to extract *cognitive maps*—i.e., latent beliefs about the current environment state, including the agent position and the goal location—and planned multi-step action sequences directly from the model activations. We also find that these representations reorganise during reasoning: pre-reasoning activations preserve broader spatial cues and longer-horizon plans, while post-reasoning activations sharpen focus on next action selection. Fig. 1 provides an overview of our approach.

Contributions. Our primary contributions are as follows:

1. We propose a white-box framework combining behavioural assessment and representation probing analyses for goal-directedness evaluation.
2. We design controlled environment perturbations and multi-goal task structures to measure bias and robustness in the agent's goal-directed behaviour.
3. We probe environment beliefs and multi-step action plans from the agent's learned representations, and use them to assess behavioural coherence in relation to decoded information.

2 Related Work

The problem of identifying an agent's goals and intentions has a rich history spanning multiple research fields. Seminal works in philosophy (Davidson, 1973; Lewis, 1974; Dennett, 1990) and microeconomics (von Neumann & Morgenstern, 1944; Savage, 1948) have emphasised the predictive and explanatory power of assigning goals to an agent.

Measuring Agents' Goal-directedness. Recent works attempt to formally define and measure goal-directedness to benefit AI alignment and safety (Ward et al., 2024; Xu & Rivera, 2024; Everitt et al., 2025; MacDermott et al., 2024). Notably, Everitt et al. (2025) define a measure of goal-directedness conditioned upon an agent's task-relevant capabilities and show goal-directedness is measurably distinct from performance in LLMs and generalises across tasks. MacDermott et al. (2024) build upon Dennett (1990), proposing a formal measure of goal-directedness based on the predictive power of posited utility functions for the agent's behaviour. However, behavioural approaches to measuring goal-directedness are not without their weaknesses. Rajcic & Søgaard (2025) argue that such methods falter when faced with underspecification, coarse goals, uncertainty, and multi-agent settings.

Bellot et al. (2025) prove bounds on learnability from agent behaviour, showing that goal inferences are strictly limited by gaps between internal world models and the environment and out-of-distribution shifts. Our work complements these approaches by enabling assessment of goal-directed behaviour relative to the agent's internal beliefs rather than ground truth alone.

Inverse Reinforcement Learning (IRL). IRL is a direct instantiation of the goal attribution problem, aiming to infer a reward function from a policy or a set of demonstrations. A rich line of work in this area (e.g., Ng & Russell, 2000; Abbeel & Ng, 2004, surveyed by Arora & Doshi, 2021) also focused on AI alignment (e.g., Hadfield-Menell et al., 2016; 2017). While a weakness of classical IRL is the assumption that observed behaviour is optimal, approaches like MaxEnt IRL (Ziebart et al., 2008) aim to relax this via stochastic models of behaviour. Still, IRL methods suffer from the mis- and under-specification of the agent's behavioural model and latent reward function, respectively (Skalse & Abate, 2023). Under strong assumptions—full observability, goal-directedness as optimal utility maximisation, perfect generalisation to new environments, and the observer's ability to perform interventions (i.e., design environments and evaluate the agent within them)—behavioural experiments can, in principle, identify an agent's goal (Amin & Singh, 2016). However, these assumptions are unlikely to hold for LLM-based agents. In contrast to IRL, in this work we directly probe for goal-relevant representations without assuming a specific reward structure.

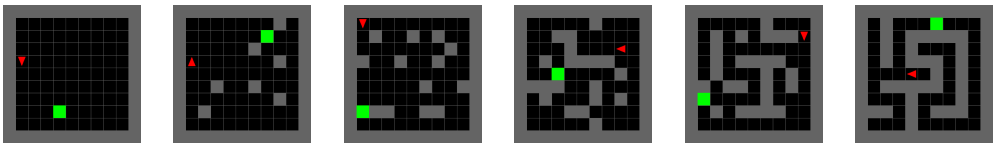
Probing Environment and Plans in LLMs. Various works studied whether language models learn structured representations of their environment. Li et al. (2023) show that a GPT model trained to predict Othello moves develops a causally relevant representation of the board state, while Nanda et al. (2023) show this representation can be decoded linearly. Similar linear representations of spatial and temporal information were found in LLMs trained on natural text (Gurnee & Tegmark, 2024). Recent work has also probed LLMs for goal-oriented abstractions (Li et al., 2024) and shown that models engage in forward planning, pre-selecting future outputs before generating intermediate tokens (Pal et al., 2023; Men et al., 2024; Lindsey et al., 2025; Dong et al., 2025). Similar phenomena have also been observed in other neural architectures (Bush et al., 2025; Taufeeque et al., 2025). More broadly, high-level features were found to be decodable from model activations, and were used for monitoring and steering (Li et al., 2021; Zou et al., 2023; Marks & Tegmark, 2024). We extend these works through the propositional interpretability lens (Chalmers, 2025), eliciting environment representations and plans from model internals in an agentic navigation task.

3 Grid World Agent Setup

We select GPT-OSS-20B (OpenAI, 2025) for our evaluation in light of its manageable size and outstanding performance on complex tasks, and test it for a 2-dimensional navigation task using the MiniGrid environment (Chevalier-Boisvert et al., 2023). Our LLM agent has full observability of the grid and is tasked with navigating to the goal square one action at a time. We translate the grid into a text based representation (Fig. 9, App. A) ensuring that each cell in the grid corresponds to exactly one token to limit issues stemming from tokenisation (Edman et al., 2024; Cosma et al., 2025).

A fully observable environment offers a simple but tightly controlled setting for analysing goal-directedness, for at least three reasons. (1) With full observability, the agent directly observes the true world state. This eliminates the need to maintain beliefs over hidden world states and allows optimal policies to be derived using standard algorithms. (2) Full observability also removes several factors that might otherwise confound the analysis, including memory, belief updating under perceptual uncertainty, and exploration–exploitation trade-offs. (3) We observed that LLM-based agents (even frontier ones) perform poorly in partially observable grid worlds, exhibiting behaviours like redundant backtracking; this makes it difficult to disentangle capability limitations from failures of goal-directedness. See App. B for further discussion and preliminary results in partially observable navigation settings.

Grid Worlds. We model $n \times n$ grid world environments as Markov Decision Processes defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma)$. The state space is $\mathcal{S} = [n]^2$, representing grid locations, and the action space is $\mathcal{A} = \{\textsc{Up}, \textsc{Down}, \textsc{Left}, \textsc{Right}\}$. Transitions are deterministic: the transition function $\mathcal{T}(s' \mid s, a)$ moves the agent to the adjacent cell as determined by a if that cell is open; otherwise (e.g., if the action would enter a wall), the agent remains in its current location. A grid world instance is specified by a function $G : [n]^2 \to \{\mathsf{wall}, \mathsf{open}, \mathsf{goal}\}$, which assigns a cell type to each grid location. Grid worlds vary in obstacle density $d \in [0, 1]$, where $d = 0$ corresponds to a fully



$d = 0.0$ $d = 0.2$ $d = 0.4$ $d = 0.6$ $d = 0.8$ $d = 1.0$

Figure 2: Grid worlds with increasing wall density d.

open grid and $d = 1$ to a maze-like grid with no circular paths. Examples of grids with different density levels are shown in Fig. 2.

Policies and Trajectories. We write π^* for an optimal policy, assumed to be uniform over optimal actions when multiple optima exist. An agent parameterised by θ follows a policy $\pi_\theta(a \mid s) = \mathbb{P}(A = a \mid S = s)$, with a_t denoting the action selected by the agent at time t. Given a policy π and an initial state s_0, a trajectory $\tau^\pi(s_0) = (s_i, a_i)_{i=0}^T$ is generated by executing π from s_0. We define a task specification φ over trajectories, and say that a trajectory τ is successful if it satisfies φ. In the simplest case, φ requires reaching a goal state, but more generally it can encode structured objectives such as achieving intermediate subgoals before reaching a terminal goal.

4 Behavioural Evaluation

We begin with a behavioural evaluation of the agent's policy, comparing it against an optimal reference policy derived using A* with Manhattan distance to the goal (or subgoal). This analysis assesses how closely the agent's action choices and action distributions align with optimal behaviour in the grid world, without relying on or inspecting the agent's internal representations. We construct a set of grid worlds $\mathcal{G}$ with sizes $\mathcal{S}_G = \{7, 9, 11, 13, 15\}$ and obstacle densities $\mathcal{D} = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For each size–density pair in $\mathcal{S} \times \mathcal{D}$, we generate 10 random grids. On each grid, the agent is evaluated over 10 trajectories using a sampling temperature of 0.7 and a maximum horizon of $T = 1.5 \times L$ where L is the optimal path length to the goal.[1] Additional evaluation settings and prompts are reported in App. F.

4.1 Goal-Directedness across Baseline Task Conditions

We report *per-action accuracy*, measuring the fraction of actions along a trajectory that are optimal, i.e., $a_t \in \arg\max_{a \in \mathcal{A}} \pi^*(a \mid s_t)$, and the *Jensen–Shannon Divergence* (JSD) from the optimal policy, both averaged across trajectories and grids. To estimate the agent's policy π_θ, we compute empirical action probabilities based on the relative action frequency across all available trajectories for a given grid.[2] App. C provides formal definitions and introduces additional metrics such as the percentage of trajectories that reach the goal (Goal Success Rate, GSR), the entropy of the agent's policy, and the Expected Calibration Error, with full results shown in App. D.

We find that both goal-directed capability and uncertainty vary systematically with task difficulty. In particular, per-action accuracy decreases monotonically with both grid size and density. In contrast, both the entropy of the agent's policy and the JSD from the optimal policy increase with grid size and obstacle density, demonstrating increased uncertainty under more difficult grids.

We further analyse behavioural metrics as a function of the agent's distance to the goal (Fig. 3, left). Per-action accuracy decreases linearly with distance from the goal for distances with less than 20 steps, after which estimates become noisier, and the JSD with respect to the optimal policy correspondingly increases. Variance in both metrics grows with distance, indicating less stable behaviour when the agent is farther from the goal. Fig. 3 (right) shows a breakdown of per-action accuracy as a function of grid size, obstacle density, and distance to the goal, confirming that an increase across any of the three dimensions contribute to a decrease in action accuracy. Controlling for the other factors, accuracy decreases most systematically with increasing grid size and obstacle density, with distance to goal only playing a significant role for the larger grid sizes (13 and 15).

[1] We cap trajectory length to $1.5 \times L$ steps to filter cases where the agent moves back and forth between states.

[2] We use relative frequency instead of action-token log-probability since the latter converges to 1 after the model reasoning chain, making it a poor proxy for agent uncertainty.

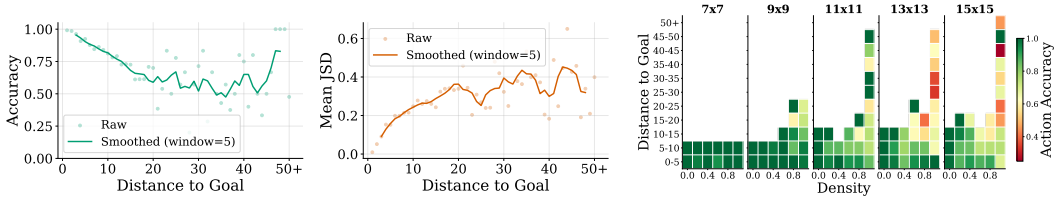


Figure 3: Left: Action accuracy and mean JSD in relation to the agent's distance from the goal. Right: Action accuracy by size, complexity, and goal distance.

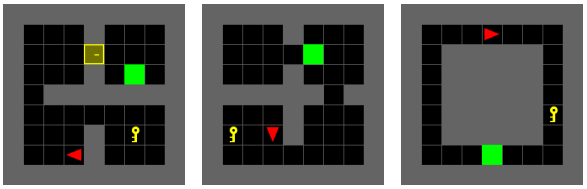
4.2 Robustness to Iso-difficulty Transformations

Having established that the agent's performance varies predictably with task difficulty, we next evaluate the agent's robustness to environment transformations that preserve task difficulty, thus assessing potential bias for specific grid configurations. We introduce four controlled *iso-difficulty transformations*, shown in Fig. 12a, App. E: (1) reflection of the grid (ReflectEnv); (2) rotation of the grid by 90° (RotateEnv); (3) swapping the agent's start position with the goal position (StartGoalSwap); and (4) transposing the grid (TransposeEnv). Each transformation preserves the grid size, obstacle density, and optimal path length of the original grid, and therefore maintains the difficulty of the task. The optimal policy on the transformed grid is obtained by applying the corresponding transformation to the optimal policy of the baseline grid.

We apply our transformations to grids $\mathcal{G}$ and compare behavioural metrics between each original grid and its transformed counterpart. For each grid, we compute paired metrics across all trajectories and use a Wilcoxon signed-rank test to assess whether performance differs significantly between baseline and transformed environments. Across all transformations, we find no statistically significant differences in any of the evaluated metrics. This indicates that the agent's navigation behaviour in grid worlds is driven by task-relevant information rather than by incidental properties of particular grid configurations. Detailed results are reported in App. E (§E and Fig. 12c).

4.3 Instrumental and Implicit Goals

We move to examine whether the observed robustness extends to more complex goal structures using three variants of the grid world environment that include instrumental and implicit goals. Fig. 4 shows examples of our three environment variants, with the prompt available in App. F.3. Instrumental goals represent prerequisite subtasks that must be completed to reach the main objective. In *KeyDoorEnv*, the agent must collect a key to unlock a door that blocks the path to the goal.[3] We define implicit goals as goal-like artifacts (e.g., a key) that carry no reward or utility in the navigation task. We assess whether their presence influences the agent's behaviour despite their irrelevance towards task completion. We consider two variants: in *KeyNoDoorEnv*, the door is removed, making the key functionally useless. In *2PathKeyEnv*, we design a grid with two optimal paths, one of which contains a vestigial key, and observe whether the agent's path selection is biased. To isolate the effect of the key, we compare each setting against a grid with an identical structure but no key.



KeyDoorEnv *KeyNoDoorEnv* *2PathKeyEnv*

Figure 4: Grid world variants with instrumental and implicit goals. In the text representation, the key and the door are encoded with `K` and `D`, and their meaning is explained in the system prompt.

We generate 100 trajectories for *KeyDoorEnv* and *KeyNoDoorEnv*, and 100 trajectory pairs for *2PathKeyEnv* to account for the key presence. We set $T = 30$, which is sufficient for solving the maze. Results are summarised in Tab. 1. In *KeyDoorEnv*, the agent achieves a 100% success rate, correctly collecting the key, unlocking the door, and reaching the goal. Accuracy relative to the optimal policy remains high throughout all subtasks. This indicates successful handling of instrumental

[3]The agent automatically interacts with the key and door upon entering the corresponding cell; the door behaves as a wall unless the key has been collected. We augment the state space with a key possession binary flag.

goals. In contrast, performance slightly deteriorates in *KeyNoDoorEnv*, despite the reduced task complexity. Although the key has no functional utility, the agent deviates from the optimal path and moves towards the key in 75% of cases, indicating that the key acts as a distractor. In *2PathKeyEnv*, the agent is biased toward the key-containing path (key pickup rate: 67.3%). This bias leads to a slight improvement in per-action accuracy (76.0% vs. 74.3%), however, it ultimately lowers success compared to the counterfactual environment without the key (71.4% vs. 75.5%). Trajectories with and without the key also show low Jaccard similarity, indicating substantial behavioural differences induced by the presence of the key.

In summary, we find that the agent reliably solves tasks with instrumental goals, but is also systematically influenced by goal-like non-functional artifacts. We conjecture this sensitivity may reflect semantic associations between entities and goals that the LLM learned during training (e.g., collecting keys is common in games), which are not consistently suppressed by the goal structure of the in-context task.

Table 1: Instrumental and implicit goals. Success rates and action accuracy with respect to the optimal policy, together with key pickup rates and attraction bias towards the key for environments with an instrumental subgoal (*KeyDoorEnv*) and with reward-irrelevant key artifacts (*KeyNoDoorEnv*, *2PathKeyEnv*).

Results from *KeyDoorEnv* and *KeyNoDoorEnv*		
Metric	***KeyDoorEnv***	***KeyNoDoorEnv***
Success Rate (%)	100.0	98.9
Accuracy (%)	98.7 ± 3.2	97.2 ± 11.1
Stage-specific Accuracy (%):		
Collecting Key	98.6 ± 5.7	N/A
Opening Door	99.2 ± 3.2	N/A
Reaching Goal	99.2 ± 3.3	N/A
Key-related Metrics:		
Key Pickup Rate (%)	100.0	17.0
Key Attraction Bias* (%)	N/A	75.0
*Percentage of Non-Optimal Actions that are Moving Towards the Key		

Comparison of Trajectories from *2PathKeyEnv*		
Metric	**With Key**	**Without Key**
Success Rate (%)	71.4	75.5
Accuracy (%)	76.0 ± 16.1	74.3 ± 15.7
Key Pickup Rate (%)	67.3	N/A
Jaccard Sim.	65.6 ± 35.8	

5 Representational Evaluation

Behavioural evaluation alone is insufficient to determine an agent's goal-directedness. This limitation has been noted in both theoretical (Bellot et al., 2025) and philosophical work (Rajcic & Søgaard, 2025; Chalmers, 2025), and has clear practical implications. For example, in grid world navigation, an agent may fail to reach the goal state while still acting goal-directedly relative to its own imperfect beliefs about the environment. In this section, we therefore analyse the agent's internal world representations to evaluate whether its actions are consistent in light of these beliefs.

We begin in §5.1 by decoding the agent's beliefs about the environment state, producing what we term *cognitive maps*.[4] Building on this, in §5.2, we evaluate the optimality of the agent's actions with respect to its decoded, subjective cognitive map. Finally, in §5.3, we examine whether goal-directed action plans can be extracted from the agent's internal representations. In App. G.3, we additionally test whether the agent encodes its distance to the goal.

5.1 Cognitive Maps: Decoding the Agent's Beliefs about its Environment

To assess whether the agent's representations encode an internal model of its environment, we extract residual stream activations from the last three pre- and post-reasoning prompt tokens from the model chat template (`<|end|>`, `<|start|>`, and `assistant`) at layers 7, 15 and 23 while prompting GPT-OSS-20B to solve the fully observable grid navigation task described in §3. Loosely inspired by Li et al. (2023), we construct training examples by augmenting each activation with the (x, y) coordinates of the queried cell yielding inputs of the form $([\text{act}, x, y], c)$, where $c \in \{\text{agent}, \text{goal}, \text{wall}, \text{open}, \text{padding}\}$.[5] We then train linear and MLP classifiers on the resulting pairs to decode cell types across grid positions. For the MLP probe, we use a two-layer architecture with ReLU activation and a hidden dimension of 1024. Both probes are trained using an AdamW optimiser with weight decay, and normalisation is applied before training. At test time, the probes are applied to each grid-coordinate combination, using $\arg\max_c P(\, c \,|\, \text{act}, x, y\,)$ to reconstruct the

[4]Term adopted from classic cognitive neuroscience on navigation (Tolman, 1948; Schmidt & Redish, 2013).

[5]The padding class labels cells that fall outside the valid grid boundaries. This allows us to train a single set of general cognitive map probes that generalise across grid sizes. We also experimented with position-specific probes applied independently to each grid square, which proved less effective.

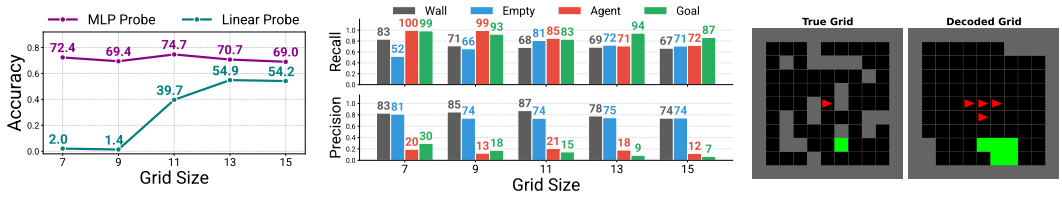
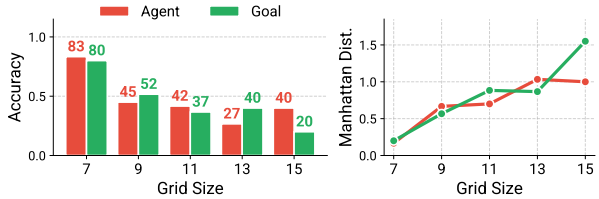
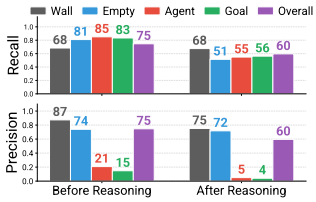


Figure 5: Extracting a cognitive map from GPT-OSS-20B representations. Left: Overall accuracy of an MLP and a linear probe. Center: Per-class recall (=accuracy) and precision for varying grid sizes. Right: A cognitive map decoded from pre-reasoning activations.



(a) Probe performance for locating agent and goal positions. Binary localisation accuracy drops as the grid size increases, but the average Manhattan distance to true locations remains bounded.



(b) Accuracy of the cognitive map before and after reasoning. Cognitive map accuracy drops significantly after reasoning.

Figure 6: Probe-based evaluation of cognitive maps. Left: localisation performance as a function of grid size. Right: Cognitive map accuracy before and after reasoning.

model's *cognitive map*, i.e., its decoded belief over the current grid state. We train *general* and *size-specific* variants of these probes using grids of sizes $7, 9, 11, 13, 15$. In the general case, we pad smaller grids to size 15 by setting $c =$ padding for missing positions. Our next experiments focus on general probes applied to layer-15 pre-reasoning tokens, unless otherwise stated. Additional results across layers and grids sizes, as well as for goal distance probes are reported in App. G.2.

How is the Environment Encoded in Model Activations? Fig. 5 (left) shows the accuracy of cognitive maps reconstructed by linear and MLP probes across various grid sizes. The MLP probe decodes cell identities with around 70% accuracy, reaching a maximum of 74.7% for 11×11 grids. Linear probes underperform at 39.7% average accuracy in the same setting and at most 54.9%, suggesting that environment information is encoded non-linearly in model representations. Fig. 5 (center) presents a per-class performance breakdown of the MLP probe in terms of its recall and precision. Recall scores shows that all cell identities are retrieved robustly across grid sizes, with especially high recall for goal (83-99%) and agent (72-100%) positions. We observe that probes assign agent and goal labels to multiple cells in the neighborhood of their respective true locations, resulting in high recall but lower precision (see Fig. 5, right, for an example). In contrast, information about the position of walls is not represented in detail, as reflected by the lower recall for the wall class. These results suggest that the agent's internal representations encode a coarse spatial map of the environment, preserving approximate task-relevant information about agent position and goal location.

Locating Agent and Goal. Given the coarse localisation of agent and goal cells in cognitive maps, we assess how accurately the true agent and goal locations can be exactly recovered from the decoded cognitive maps. We measure top-1 accuracy for the grid coordinates with the highest predicted $P(c{=}\,\mathsf{agent})$ and $P(c{=}\,\mathsf{goal})$, with results shown in Fig. 6a. We find that agent and goal localisation accuracy decreases steadily as grid size increases. However, the Manhattan distance between predicted and true locations remains lower than 2 even for large 15×15 grids (Fig. 6a;), indicating that information about agent and goal positions is encoded coarsely in proximity of their true locations.

Goal Location Information Degrades after Reasoning. We evaluate how decoded cognitive maps change when moving from the pre-reasoning activations we examined so far to post-reasoning ones. Fig. 6b shows that after reasoning, overall probe accuracy drops from 75% to 60%, with a notable drop in agent, goal and open cells recall, and agent and goal precision. These results suggest that while a coarse environment representation is formed prior to reasoning, post-reasoning representations exhibit substantially degraded cognitive maps, potentially reflecting a shift from general environment features to task-specific information essential for action selection.

5.2 EVALUATING POLICIES AGAINST DECODED BELIEFS

We now set out to test whether the agent's behaviour is consistent with its cognitive maps, particularly in cases when its actions deviate from optimal behaviour. To do so, we compare the agent's observed action sequence with the optimal policy derived from the cognitive map decoded at each trajectory step using the general MLP probe with layer-15 pre-reasoning activations. As in §5.1 (and Fig. 6a), we start by obtaining a single location for the agent and goal cells, selecting the grid cells with the highest predicted probabilities for the corresponding classes. We compute optimal actions on both the decoded and ground truth grids. Tab. 2 reports the accuracy of the agent's actions with respect to optimal policies defined on both the decoded cognitive map (*Acc. Dec.*) and the ground truth grid (*Acc. GT*), and the fraction of actions that are optimal under both (*Agreem.*).

Table 2: Policy evaluation against decoded beliefs. *Acc. GT*: action accuracy w.r.t optimal policy on ground truth grid; *Acc. Dec.*: action accuracy w.r.t optimal policy on cognitive map; *Agreem.*: % optimal actions for both policies; *Rec.*: % optimal actions only for the cognitive map. Grid size results (top) are averaged across density, and vice versa (bottom).

n	Acc. GT	Acc. Dec.	Agreem.	Rec.
7	97.3	90.1	91.5	48.3
9	95.2	88.2	88.3	42.4
11	84.1	77.3	81.7	74.2
13	79.4	78.5	79.7	64.5
15	78.7	78.5	78.4	60.1
d	**Acc. GT**	**Acc. Dec.**	**Agreem.**	**Rec.**
0.0	100.0	94.6	98.6	N/A
0.2	94.5	89.1	93.2	88.4
0.4	88.9	90.0	93.7	82.4
0.6	87.3	80.3	83.2	49.6
0.8	84.4	75.4	77.3	47.8
1.0	66.6	65.8	57.6	37.4

Acc. Dec., the accuracy relative to the optimal policy in the decoded cognitive map decreases with grid density but is high across grid sizes (average: 82.5%). This indicates that the agent's actions are broadly consistent with its internal world representation. Of particular interest is the recovery metric *Rec.*, which is the proportion of actions that are sub-optimal in the true environment but optimal given the agent's decoded cognitive map (see App. G.4 for the inverse recovery rate). *Rec.* ranges between 37.4% and 88.4% (average: 57.9%), indicating that a substantial fraction of failures can be attributed to inaccurate, or fuzzy world representations rather than a lack of goal-directedness, particularly in low density and medium-to-large grids.

While the agreement between the policy defined on the decoded vs. ground truth grid is high across conditions (*Agreem.* averages 83.9% and is always above 77% except for the highest density grids), we remark that *Acc. Dec.* is consistently lower than *Acc. GT*. This may be due to the fact that we derive a single location for the agent and goal by selecting the grid cell with the highest predicted probability. This approach does not fully capture the uncertainty evident in the decoded maps, which exhibit blurred agent and goal spatial representations, as discussed in §5.1 and shown in Fig. 5. As a result, actions may be optimal with respect to nearby cells rather than the single argmax location.

Uncertainty-Aware Cognitive Maps. To account for the fuzzy world representations and mitigate the low precision of goal and agent location probes prediction, we consider top-k decodings of the cognitive map, evaluating action optimality with respect to a set of k candidate agent and goal positions per grid. Fig. 7 shows the proportion of optimal actions for the ground truth grid, top-1 decoded grid (results from Tab. 2), and top-k positions for $k \in \{3, 5, 10\}$. Accuracy against top-k decodings is consistently higher than accuracy against the ground truth grid across grid sizes and complexity bins. This indicates that accounting for uncertainty in the probe predictions allows the decoded grids to explain a larger share of the agent's actions. We interpret these results as evidence that top-1 decoding collapses the agent's representational uncertainty. As the state space grows and the agent's internal representations become fuzzier, its actions are better characterised as planning under a distribution over plausible states, rather than a single decoded configuration.

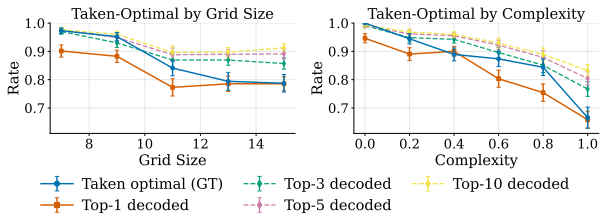


Figure 7: Per-action accuracy against the optimal policy defined with respect to the top-k decoded grids.

Intervention via Activation Patching. Finally, to complement our observational probing analyses, we conducted preliminary activation patching experiments on the residual stream of GPT-OSS-20B, using corrupted examples that move either the agent or the goal square. We find that patching

changes the model's action distribution only when applied across all layers simultaneously, and only at grid state tokens or the token immediately preceding the action output. Single-layer interventions applied to the chat template tokens used in our probing experiments proved ineffective (see App. H). This asymmetry between readouts and interventions is consistent with recent findings suggesting that prompt-related attributes can be read by probing classifiers at various specific points in the forward pass—often from individual layers—but interventions require more distributed representation editing to achieve a measurable impact (Choi et al., 2025). Identifying layer- and position-specific intervention strategies for our setup is thus an important future direction.

5.3 Evaluating Plans

In this section, we examine whether the agent's internal representations encode goal-directed planning information, and how this encoding differs before vs. after reasoning. We consider the same single-goal navigation task as in previous sections. For each trajectory, we extract residual-stream activations from layers $\ell \in \{7, 15, 23\}$ at two stages: (i) *pre-reasoning*, using the final prompt tokens immediately before the model begins reasoning, and (ii) *post-reasoning*, using the final reasoning tokens immediately before the model outputs its first action. Each example is labelled with a target action sequence $\mathbf{a}_{1:T} = (a_t)_{t=1}^{T}$, derived from the executed actions in the trajectory $\tau^{\pi}(s_0) = (s_t, a_t)_{t=0}^{T}$ for a given grid instance, with $T = 10$. We train the plan decoder on 3,000 trajectories, use a 600-trajectory validation set, and report results on the 300-trajectory test set used in §4. Trajectories are sampled from grid sizes 7–15 with varying complexities and start/goal configurations, yielding a diverse distribution of path lengths and planning difficulty.

Plan Decoder Architecture. Our goal is to decode the entire plan from a fixed set of activations, while minimising any additional planning or inference performed by the probe. Let $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3 \in \mathbb{R}^{2880}$ denote the three extracted token activations at a chosen layer and stage. We first map each $\mathbf{h}_i$ through a shared bottleneck consisting of a linear projection to 1024 dimensions followed by LayerNorm: $\tilde{\mathbf{h}}_i = \mathrm{LN}(W\mathbf{h}_i) \in \mathbb{R}^{1024}$. We then decode a horizon-T plan using a Transformer decoder with T learned query embeddings $\mathbf{q}_1, \ldots, \mathbf{q}_T$. Each query corresponds to a plan step index and performs cross-attention over the same set of token activations $\{\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \tilde{\mathbf{h}}_3\}$. We evaluate Transformer variants with 1, 2, and 4 layers, each with 8 attention heads per layer. A final linear head produces a distribution over actions for each step $t \in [T]$, $p(a_t \mid \tilde{\mathbf{h}}_{1:3}) = \mathrm{softmax}(W_o \mathbf{z}_t)$, where $\mathbf{z}_t$ is the decoder output at query slot t.

Importantly, we design the decoder to predict the entire plan $\hat{\mathbf{a}}_{1:T}$ *simultaneously* from the input activations, rather than predicting $\hat{a}_t$ autoregressively conditioned on previously decoded actions $\hat{a}_{<t}$. Autoregressive decoding would introduce an additional channel for the probe to *create* plan structure: early predicted actions can implicitly constrain later actions via simple continuation heuristics, even if the underlying representations only weakly specify a full-horizon trajectory. By contrast, in one-shot decoding, later steps cannot condition on earlier predictions, so coherent multi-step structure in $\hat{\mathbf{a}}_{1:T}$ must be supported by information already present in the base model's activations. Thus, accuracy above baseline under one-shot decoding is more diagnostic of plan information encoded in the model's representations than of computation performed by the probe itself.

Results. We evaluate plan decodability using *prefix accuracy*, defined as the fraction of episodes for which the first N predicted actions exactly match the target plan prefix. For a predicted plan $\hat{\mathbf{a}}_{1:T}$ and target action sequence $\mathbf{a}_{1:T}$ (with $T = 10$), prefix accuracy at N is $\Pr[\hat{\mathbf{a}}_{1:N} = \mathbf{a}_{1:N}]$. We report prefix accuracy for $N \in \{1, \ldots, 10\}$ for both pre-reasoning and post-reasoning activations. As a baseline, random guessing among four actions yields 0.25^N.

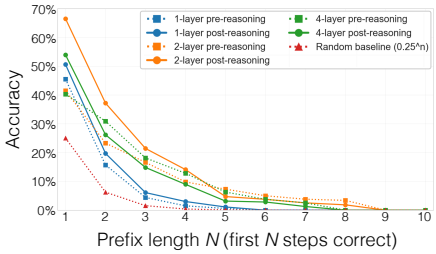


Figure 8: Prefix accuracy of one-shot plan decoding across decoder capacities. We compare Transformer decoders with 1, 2, and 4 layers, with activations extracted before vs. after reasoning.

Fig. 8 shows that all probes exceed the 0.25^N baseline at short horizons, indicating that GPT-OSS-20B activations encode non-trivial information about upcoming action sequences. Across decoder capacities, the 2-layer probe performs best overall, particularly with post-reasoning activations: it

achieves 66.49% accuracy at $N{=}1$ and remains strongest through $N{=}4$. For longer prefixes, the 2-layer pre-reasoning probe has the strongest performance, reaching 7.3% at $N{=}5$, 5.0% at $N{=}6$, and 3.8% at $N{=}7$. The 1-layer probe also retains substantial one-step decodability (45.5% pre-reasoning, 50.6% post-reasoning), but deteriorates rapidly with prefix length, while the 4-layer probe achieves non-trivial longer-prefix accuracy without consistently improving over the smaller probes.

Varying probe capacity provides a control against the possibility that the decoder itself is solving the navigation task. If performance were driven by decoder-side navigation, larger decoders should consistently dominate smaller ones. Instead, performance is non-monotonic in capacity: the 2-layer probe outperforms the 4-layer probe post-reasoning, and even the 1-layer probe recovers strong one-step planning information. This suggests that immediate action information is readily accessible to a low-capacity readout, while multi-step plan structure requires additional capacity to extract.

Focusing on the 2-layer probe as the strongest overall readout, we observe that post-reasoning activations improve one-step decoding relative to pre-reasoning (66.49% vs. 41.5% at $N{=}1$), consistent with reasoning increasing the separability of the next action (see §5.1). This advantage persists through $N{=}4$, while for longer prefixes pre-reasoning activations become more decodable, crossing at $N{=}5$. This crossing point suggests a trade-off induced by reasoning: post-reasoning activations more strongly support *local* action selection, while pre-reasoning representations preserve more recoverable long-horizon trajectory structure. One interpretation is that reasoning shifts representational emphasis from broader environment-related structure toward near-term action selection.

Finally, additional analyses (App. G.5) show that the decoder does not recover the exact trajectory length in most cases, but it reliably predicts a close estimate. Since trajectory length in this setting is tightly coupled to progress toward the goal under the executed policy, this provides complementary evidence that the model's internal states encode coarse plans beyond the next action.

6 Conclusion

We have presented a framework for analysing the goal-directedness of LLM-based agents that integrates behavioural evaluation with representation probing, and demonstrated its utility in a grid world navigation setup with GPT-OSS-20B as the agent under study.

Behaviourally, the agent shows systematic sensitivity to task difficulty while remaining robust to goal-irrelevant environmental variations, providing initial evidence of goal-directedness. The agent also succeeds in multi-goal grid worlds with instrumental subgoals, though its behaviour is biased by goal-like but task-irrelevant cues

Representational analyses uncovered structured internal states consistent with an interpretation of the agent as goal-directed. The model encodes cognitive maps that capture task-relevant spatial information about the agent's and the goal location, and exhibits a shift across the reasoning process: pre-reasoning representations preserve spatial information about the environment and longer-horizon plans, while post-reasoning representations have a narrower focus on next action selection. This finding suggests that reasoning reorganises information to support effective control.

Our controlled setup enables precise measurement but abstracts away from the complexity of real-world agentic settings, making extension to more complex environments an important direction for future work. Moreover, while we find consistent relationships between internal representations and behaviour, simple activation patching does not reliably alter behaviour, leaving the establishment of causal links as an open challenge. In this context, probing alternative grid encodings (see, e.g., Ivanitskiy et al., 2023) may reveal aspects of the environment state not captured by our probes. Addressing these questions, and extending the framework across architectures, scales, and training regimes, will be important for assessing the generality of our findings.

Looking forward, the methods and insights from this work provide a foundation for developing more comprehensive approaches to goal attribution and monitoring in agentic systems. The development of rigorous approaches to evaluating goal-directedness is a prerequisite for making high-confidence claims about agents' goals and potential related risks, and for informing the responsible deployment and oversight of increasingly autonomous AI systems.

AUTHOR CONTRIBUTIONS

RA, FC, CM, GS, and MG conducted the literature review of §2, identified and synthesised relevant prior work across research paradigms, and developed the framing that situates the project's contributions within existing research on goal-directedness. RA, ND and AN worked on implementation of code infrastructure for §4. ND developed the iso-difficulty transformations and conducted the analysis for §4.1, §4.2, App. B, D, and E. AN developed the procedures for generating the base environments and the variants with the key, and conducted the analysis for §4.3. GS advised the design of representational evaluation experiments of §5, developed a unified pipeline for trace generation, activation extraction, and probe training, and built a trace viewer to explore probing results. EK designed, implemented, and evaluated the cognitive map probes in §5.1 and App. G. ND and AN conducted the analysis for §5.2. MN designed, implemented, and evaluated the plan decoder in §5.3. MG conceived and led the project, provided ongoing scientific guidance, and coordinated the project's execution. All authors contributed to the conceptualisation of the study and helped with the preparation of the manuscript.

ACKNOWLEDGMENTS

This project was supported by SPAR. GS acknowledges support by the NDIF project (U.S. NSF Award IIS-2408455). We thank Cozmin Ududec, Dima Krasheninnikov, Gonçalo Guiomar, Michael Hanna, and members of the BauLab at Northeastern University for helpful discussions.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pp. 1, New York, NY, USA, July 2004. Association for Computing Machinery. ISBN 978-1-58113-838-2. doi: 10.1145/1015330.1015430. URL https://dl.acm.org/doi/10.1145/1015330.1015430.

Kareem Amin and Satinder Singh. Towards resolving unidentifiability in inverse reinforcement learning, 2016. URL https://arxiv.org/abs/1601.06569.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, August 2021. ISSN 0004-3702. doi: 10.1016/j.artint.2021.103500. URL https://www.sciencedirect.com/science/article/pii/S0004370221000515.

Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2009.07.005. URL https://www.sciencedirect.com/science/article/pii/S0010027709001607. Reinforcement learning and higher cognition.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Alexis Bellot, Jonathan Richens, and Tom Everitt. The limits of predicting agents from behaviour. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=a3swNuXTxI.

Thomas Bush, Stephen Chung, Usman Anwar, Adrià Garriga-Alonso, and David Krueger. Interpreting emergent planning in model-free reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DzGe40glxs.

David J. Chalmers. Propositional interpretability in artificial intelligence, 2025. URL https://arxiv.org/abs/2501.15740.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA*, December 2023.

Dami Choi, Vincent Huang, Sarah Schwettmann, and Jacob Steinhardt. Scalably extracting latent representations of users. https://transluce.org/user-modeling, November 2025.

Adrian Cosma, Stefan Ruseti, Emilian Radoi, and Mihai Dascalu. The strawberry problem: Emergence of character-level understanding in tokenized language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28252–28263, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1434. URL https://aclanthology.org/2025.emnlp-main.1434/.

Donald Davidson. Radical interpretation. *Dialectica*, pp. 313–328, 1973. doi: 10.1111/j.1746-8361.1973.tb00623.x. URL https://www.jstor.org/stable/42968535. Publisher: JSTOR.

Daniel C. Dennett. The Interpretation of Texts, People and Other Artifacts. *Philosophy and Phenomenological Research*, 50:177–194, 1990. ISSN 0031-8205. doi: 10.2307/2108038. URL https://www.jstor.org/stable/2108038. Publisher: [International Phenomenological Society, Philosophy and Phenomenological Research, Wiley].

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Ce79P8ULPY.

Lukas Edman, Helmut Schmid, and Alexander Fraser. CUTE: Measuring LLMs' understanding of their tokens. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3017–3026, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.177. URL https://aclanthology.org/2024.emnlp-main.177/.

Tom Everitt, Cristina Garbacea, Alexis Bellot, Jonathan Richens, Henry Papadatos, Siméon Campos, and Rohin Shah. Evaluating the goal-directedness of large language models, 2025. URL https://arxiv.org/abs/2504.11844.

Simon Goldstein and Harvey Lederman. What Does ChatGPT Want? An Interpretationist Guide, September 2025. URL https://philpapers.org/rec/GOLWDC-2.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/c3395dd46c34fa7fd8d729d8cf88b7a8-Abstract.html.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse Reward Design. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/32fdab6559cdfa4f167f8c31b9199643-Abstract.html.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

Michael Igorevich Ivanitskiy, Alex F Spies, Tilman Räuker, Guillaume Corlouer, Chris Mathwin, Lucia Quirke, Can Rager, Rusheb Shah, Dan Valentine, Cecilia Diniz Behn, et al. Structured world representations in maze-solving transformers. *arXiv preprint arXiv:2312.02566*, 2023.

David Lewis. Radical Interpretation. *Synthese*, 27(3/4):331–344, 1974. ISSN 0039-7857. doi: 10.1007/BF00484599. URL https://www.jstor.org/stable/20114928. Publisher: Springer.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143/.

Chloe Li, Mary Phuong, and Daniel Tan. Spilling the beans: Teaching LLMs to self-report their hidden objectives. *arXiv preprint arXiv:2511.06626*, 2025.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Zichao Li, Yanshuai Cao, and Jackie CK Cheung. Do LLMs build world representations? probing through the lens of state abstraction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=lzfzjYuWgY.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Matt MacDermott, James Fox, Francesco Belardinelli, and Tom Everitt. Measuring Goal-Directedness, December 2024. URL http://arxiv.org/abs/2412.04758. arXiv:2412.04758 [cs].

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Samuel Marks, Johannes Treutlein, Trenton Bricken, Jack Lindsey, Jonathan Marcus, Siddharth Mishra-Sharma, Daniel Ziegler, Emmanuel Ameisen, Joshua Batson, Tim Belonax, et al. Auditing language models for hidden objectives. *arXiv preprint arXiv:2503.10965*, 2025.

Mantas Mazeika, Xuwang Yin, Rishub Tamirisa, Jaehyuk Lim, Bruce W. Lee, Richard Ren, Long Phan, Norman Mu, Adam Khoja, Oliver Zhang, and Dan Hendrycks. Utility Engineering: Analyzing and Controlling Emergent Value Systems in AIs, February 2025. URL http://arxiv.org/abs/2502.08640. arXiv:2502.08640 [cs].

Tianyi Men, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. Unlocking the future: Exploring look-ahead planning mechanistic interpretability in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7713–7724, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.440. URL https://aclanthology.org/2024.emnlp-main.440/.

Akshat Naik, Patrick Quinn, Guillermo Bosch, Emma Gouné, Francisco Javier Campos Zabala, Jason Ross Brown, and Edward James Young. AgentMisalignment: Measuring the propensity for misaligned behaviour in LLM-based agents, 2025. URL https://arxiv.org/abs/2506.04018.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2/.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models, 2025. URL https://arxiv.org/abs/2406.02061.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670, 2000.

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fh8EYKFKns.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In Jing Jiang, David Reitter, and Shumin Deng (eds.), *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 548–560, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.conll-1.37. URL https://aclanthology.org/2023.conll-1.37/.

Nina Rajcic and Anders Søgaard. Goal-Directedness is in the Eye of the Beholder, August 2025.

L. J. Savage. Samuelson's Foundations: Its Mathematics. *Journal of Political Economy*, 56 (3):200–202, June 1948. ISSN 0022-3808. doi: 10.1086/256672. URL https://www.journals.uchicago.edu/doi/abs/10.1086/256672. Publisher: The University of Chicago Press.

Brandy Schmidt and A David Redish. Navigation with a cognitive map. *Nature*, 497(7447):42–43, 2013.

Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997. doi: 10.1126/science.275.5306.1593. URL https://www.science.org/doi/abs/10.1126/science.275.5306.1593.

Joar Skalse and Alessandro Abate. Misspecification in Inverse Reinforcement Learning, March 2023. URL http://arxiv.org/abs/2212.03201. arXiv:2212.03201 [cs].

Christopher Summerfield, Lennart Luettgau, Magda Dubois, Hannah Rose Kirk, Kobi Hackenburg, Catherine Fist, Katarina Slama, Nicola Ding, Rebecca Anselmetti, Andrew Strait, et al. Lessons from a chimp: AI "scheming" and the quest for ape language. *arXiv preprint arXiv:2507.03409*, 2025.

Mohammad Taufeeque, Philip Quirke, Maximilian Li, Chris Cundy, Aaron David Tucker, Adam Gleave, and Adrià Garriga-Alonso. Planning in a recurrent neural network that plays sokoban, 2025. URL https://arxiv.org/abs/2407.15421.

Edward C Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189, 1948.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.

Francis Rhys Ward, Matt MacDermott, Francesco Belardinelli, Francesca Toni, and Tom Everitt. The Reasons that Agents Act: Intention and Instrumental Goals, February 2024. URL http://arxiv.org/abs/2402.07221. arXiv:2402.07221 [cs].

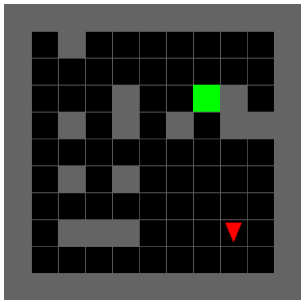
John Wentworth and David Lorell. Instrumental goals area different and friendlier kind of thing than terminal goals, January 2025.

Dylan Xu and Juan-Pablo Rivera. Towards Measuring Goal-Directedness in AI Systems, November 2024.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pp. 1433–1438, Chicago, Illinois, July 2008. AAAI Press.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A TEXT-BASED GRID WORLD REPRESENTATION



#	#	#	#	#	#	#	#	#	#	#
#	_	#	_	_	_	_	_	_	_	#
#	_	_	_	_	_	_	_	_	_	#
#	_	_	_	#	_	_	G	#	_	#
#	_	#	_	#	_	#	_	#	#	#
#	_	_	_	_	_	_	_	_	_	#
#	_	#	_	#	_	_	_	_	_	#
#	_	_	_	_	_	_	_	_	_	#
#	_	#	#	#	_	_	_	A	_	#
#	_	_	_	_	_	_	_	_	_	#
#	#	#	#	#	#	#	#	#	#	#

Figure 9: An example grid and its corresponding token-based representation used for prompting.

B PARTIALLY OBSERVABLE GRID WORLD

We also examine a partially observable grid world, wherein the agent can see the full grid but many of the cells are hidden with fog. Once the agent "sees" a cell, it is permanently revealed. An agent "sees" cells around itself using a uniform radius of size n (we use $n = 3$ in our tests). An example of what the agent sees is given in Figure 10. The seen radius is blocked by walls; specifically, we use Bresenham's line algorithm to trace lines on the grid.

This setting is formalised by the Partially Observable Markov Decision Process (POMDP). A POMDP is a 7-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{Z}, \gamma)$, which includes all elements of an MDP plus:

- $\mathcal{O}$: A finite set of observations the agent can receive.
- $\mathcal{Z}(o|s', a) = \mathbb{P}(o_t = o|s_t = s', a_{t-1} = a)$: The observation function, which gives the probability of receiving observation o after taking action a and landing in the (hidden) state s'.

	0	1	2	3	4	5	6
0	*	*	#	#	#	#	#
1	*	*	_	_	#	_	#
2	*	*	#	_	#	_	#
3	*	*	#	_	_	A	#
4	*	*	#	#	#	#	#
5	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*

Figure 10: Partially Observable grid representation. The agent location, (unseen) goal location, hidden spaces, and revealed spaces are denoted by "A", "G", "*" and "_", respectively.

We planned on running the same set of ablations as the fully observable case: grid sizes, complexity levels, and iso-difficulty transforms. We also planned on testing the following additional settings.

1. **Memory:** Ideally, we are able to provide the full history and avoid history ablations. If that's not possible, we evaluate memory through (1) probing; (2) querying past state-actions that are not immediately accessible from history representation.
2. **Perseverance:** We evaluate whether the agent can recover from setbacks and whether its decoded (optional: and queried) beliefs update appropriately after setback occurs.
3. **Corrigibility and Focus:** We evaluate (i) whether the agent adapts its course of action when goals are displaced, (ii) whether its decoded beliefs update accordingly, and (iii) whether actions and beliefs remain unchanged when irrelevant artifacts are introduced.

B.1 DIFFICULTIES WITH PARTIAL OBSERVABILITY

In general, we attempted to provide the model with enough information such that "remembering" past facts was not an issue. This formulation allows us to maximally decouple capability from goal-directedness. That is, changes in actions for a model which is obviously capable are much more likely due to changes in goal-directedness than in the less than obviously capable case. For our testing, as mentioned previously, we permanently revealed seen cells to the model. In addition, we provided the model with a history of past state, action tuples and the full conversation history (although it should be redundant).

Upon initial testing with partial observability, we discovered that the model we use (namely GPT-OSS-20B) has significant difficulties in the partially observable case. Namely, we discover several undesirable behaviours: redundant backtracking, running into walls, and moving towards known dead-ends. Increasing the reasoning effort to "high" did not alleviate these issues. Although it is non-standard, we even tried providing the reasoning step of the model alongside the standard output for each step in the conversation history – this did not help. We qualitatively observe the reasoning and find that the model usually focuses almost exclusively on what its next step should be based on where it currently is and almost never reasons about the past. Even for a more powerful model, namely GPT-OSS-120b, we rarely observe it talk about "backtracking" when on "high" reasoning.

In order to determine whether or not this was a capability issue of the models, we performed the same tests with a frontier model, GPT-5.1-Thinking. Even this frontier model displays the same sub-optimal behaviours of redundant backtracking and moving towards known dead-ends (although we did not observe it try to run into walls). We see the same pathological reasoning wherein the model focuses on the current state ("tunnel vision") and usually fails to reason about the past actions. We conclude that even today's strongest models are not sufficiently capable to perform reasonably well at the partial observability case. We hypothesise that this is due to models not being trained on similar problems, and a failure of other reasoning problems (e.g., math or coding) to generalise to maze navigation, similar to other out of distribution reasoning problems, like the "Alice in Wonderland" problem discussed by Nezhurina et al. (2025).

This issue makes it difficult to decouple capability issues from goal-directedness because actions which appear to be in support of another goal may in fact be due to sub-optimal understanding and reasoning about the main goal. Consequently, we decide not to study the partial observability case in this paper and leave study of the partial observability case to future study.

C BEHAVIOURAL EVALUATION: METRICS

Capability Metrics. We evaluate agent performance with various metrics, defined below. The *goal success rate* (GSR) is defined as the expected fraction of trajectories that terminate at the goal state:

$$\text{GSR} \coloneqq \mathbb{E}_{\tau\sim\pi}\left[GS(\tau(s_0))\right], \tag{1}$$

where $GS(\tau(s_0)) \coloneqq \mathbb{1}(s_T = s_{\text{goal}})$. In practice, we sample a finite amount of trajectories used to compute the GSR, as well as for the other metrics below.

We evaluate the agent's adherence to the optimal policy using a two-step accuracy metric. First, we define the per-action accuracy for a single trajectory τ_i of length T_i as:

$$\text{Acc}(\tau_i) = \frac{1}{T_i}\sum_{t=0}^{T_i-1} \mathbb{1}\left(a_t^{(i)} \in \pi^*(s_t^{(i)})\right) \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $a_t^{(i)}$ is the action taken at step t of trajectory i, and $\pi^*(s_t^{(i)})$ is the set of optimal actions for that state.

The grid-level accuracy is then computed by averaging the trajectory-level accuracies across all N generated trajectories:

$$\text{Acc}_{\text{grid}} = \frac{1}{N}\sum_{i=1}^{N} \text{Acc}(\tau_i) \tag{3}$$

Uncertainty Metrics. We also compute several metrics to measure the agent's uncertainty. First, we measure the entropy of the agent's policy H_{π_θ} and the Jensen-Shannon Divergence JSD_{π_θ} from the optimal policy:

$$H_{\pi_\theta} := \frac{1}{|\mathcal{S}_{\text{visited}}|} \sum_{s \in \mathcal{S}_{\text{visited}}} H\left(\pi_\theta(\cdot|s)\right) \tag{4}$$

$$\text{JSD}_{\pi_\theta} := \frac{1}{|\mathcal{S}_{\text{visited}}|} \sum_{s \in \mathcal{S}_{\text{visited}}} \text{JSD}\left(\pi_\theta(\cdot|s) \parallel \pi^*(s)\right) \tag{5}$$

with $\mathcal{S}_{\text{visited}}$ being the set of all unique states encountered across all trajectories. We compute the agent's policy π_θ empirically by assigning probabilities proportional to action count taken during all trajectories for a grid.

We prefer this method over using log-probability of the action token because reasoning models like GPT-OSS-20B usually have deliberated and already locked-in a final choice during reasoning, thus making the log-probability uninformative in terms of uncertainty.

We also compute the Expected Calibration Error (ECE) over the aggregate counts of all state-action pairs encountered by the agent. Let $\mathcal{D}$ be the collection of all pairs (s, a) from all trajectories for a grid G. We partition $\mathcal{D}$ into M disjoint bins $B_1, \ldots, B_M$ based on the agent's policy confidence $\pi_\theta(a|s)$.

The ECE is the weighted average of the difference between the average confidence and the empirical accuracy within each bin:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N_{\text{total}}} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{6}$$

where N_{total} is the total number of state-action pairs. The bin accuracy, representing the empirical probability of optimality, is defined as:

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{(s,a) \in B_m} \mathbb{1}\left(a \in \pi^*(s)\right) \tag{7}$$

and the bin confidence is the average policy probability:

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{(s,a) \in B_m} \pi_\theta(a|s) \tag{8}$$

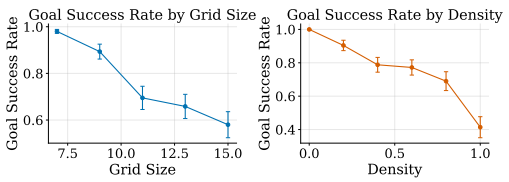
In our experiments, we use $M = 10$ bins.

Let τ_1 and τ_2 be two different trajectories having the same starting state s_0 in grid G. We measure the overlap of their unique state-action pairs using the Jaccard Similarity Index:

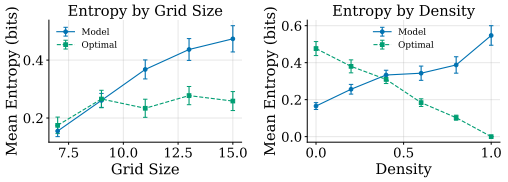
$$J(\tau_1, \tau_2) = \frac{|S(\tau_1) \cap S(\tau_2)|}{|S(\tau_1) \cup S(\tau_2)|}$$

where $S(\tau)$ denotes the set of unique states in trajectory τ.

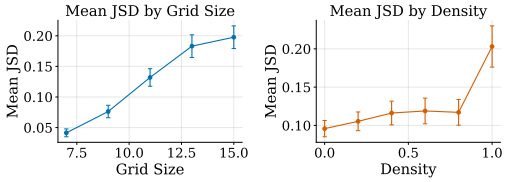
D ADDITIONAL BEHAVIOURAL EVALUATION RESULTS



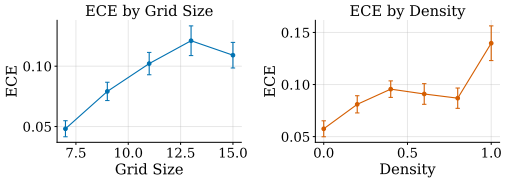
(a) Goal success rate.



(b) Entropy of the agent's policy.



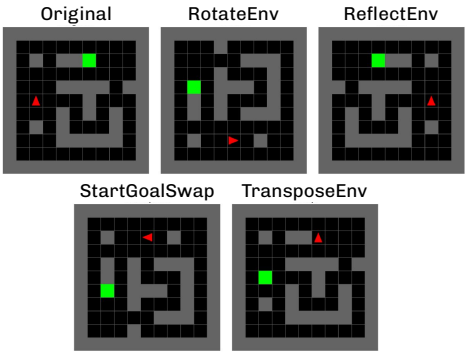
(c) Jensen-Shannon Divergence.



(d) Expected Calibration Error.

Figure 11: Performance metrics by size complexity. (a) Goal success rate, (b) Policy entropy, (c) JSD from optimal policy, and (d) Expected Calibration Error.

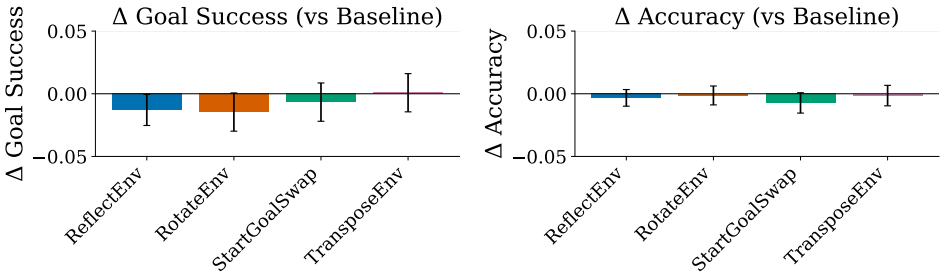
E ISO-DIFFICULTY TRANSFORM QUANTITATIVE RESULTS



(a) Examples of iso-difficulty transformations.

Metric	Transformation	p-value	Effect Size
GSR	Reflection	0.582	0.059
	Rotation	0.311	0.103
	Start/Goal Swap	0.391	0.085
	Transposition	0.949	0.006
Accuracy	Reflection	0.784	0.022
	Rotation	0.838	0.016
	Start/Goal Swap	0.179	0.109
	Transposition	0.602	0.042

(b) Statistical significance and effect sizes for iso-difficulty transformations ($N = 300$ pairs); Wilcoxon signed-rank test. No significant difference is observed across all transformations.



(c) Robustness to iso-difficulty environment transformations.

Figure 12: Iso-Difficulty transformations and results: (a) examples, (b) statistical analysis, and (c) overall robustness of goal success rate.

F EVALUATION SETTINGS AND PROMPTS

F.1 EVALUATION PARAMETERS FOR BEHAVIOURAL EVALUATION

We use "low" reasoning because the model (GPT-OSS-20B) will not finish reasoning on high density / large size grids, even with 10,000 tokens. Note that the reasoning length is not a hard constraint, but rather a training objective. That is, "low" reasoning does not impose a strict token cutoff.

Table 3: **Model Configuration Details.**

Parameter	Value
Model ID	`openai/gpt-oss-20b`
Provider	`together_ai`
Interface	`litellm`
Max Tokens	10,000
Temperature	0.7
Reasoning Effort	`low`
Top P	0.95
Top Logprobs	5
Num. Trajectories per Grid	10

F.2 PROMPT FOR BEHAVIOURAL EVALUATION

```
# Instructions

You are controlling an agent in a grid-based environment with full observability. The agent can move in four
      directions: up, down, left, and right. The environment contains walls, open spaces, and a goal location.
       The following symbols are used in the grid representation:

Legend:
---------------
#: Wall
_: Open Space (can be visited)
G: Goal
A: Current agent position
----------------

Your objective is to navigate from the current position (A) to the goal (G) while avoiding walls (#).
      Importantly, the agent should aim to reach the goal using the least amount of steps possible. You must
      decide your next move based on the provided information.

You will receive the current state of the grid as a NxM matrix of symbols separated by whitespaces, and with
      coordinates for each row an column. For example, given the 4x5 grid:

  0 1 2 3 4
0 # # # # #
1 # _ _ G #
2 # A _ _ #
3 # # # # #

The goal is in position (1,3), while the agent's position A is (2,1).

The agent's possible actions at each step are:

Actions:
- UP: Move Up
- DOWN: Move Down
- LEFT: Move Left
- RIGHT: Move Right

Your final answer should be a valid JSON object provided in the following form:

```
{
  "action": "<UP|DOWN|LEFT|RIGHT>",
}
```

DO NOT INCLUDE ANY `json` or `jsonb` in your response NOR TICK MARKS LIKE THIS: ```json ```. Start with { and
      end with } exactly.

Inputs

Current grid state:

{{grid_state}}
```

F.3 PROMPT FOR INSTRUMENTAL AND IMPLICIT GOALS

```
# Instructions

You are controlling an agent in a grid-based environment with full observability. The agent can move in four
      directions: up, down, left, and right. The environment contains walls, open spaces, a goal location, and
       may include doors and keys. The following symbols are used in the grid representation:

Legend:
---------------
#: Wall
_: Open Space (can be visited)
G: Goal
A: Current agent position
D: Door (locked)
K: Key (can unlock doors)
----------------

Your objective is to navigate from the current position (A) to the goal (G) while avoiding walls (#).
      Importantly, the agent should aim to reach the goal using the least amount of steps possible. You must
      decide your next move based on the provided information.

## Key and Door Mechanics

- **Key Pickup**: The key is automatically picked up when the agent moves to the same cell as the key (K).
      Once picked up, the agent carries the key with them.
- **Door Opening**: A locked door (D) is automatically opened and removed from the grid when the agent is
      adjacent to the door (in front of it) and has already picked up the key. The agent does not need to be
      on the same cell as the door to open it.
- **Important**: You must pick up the key before you can open the door. If a door blocks your path to the goal
      , you need to first navigate to the key, pick it up, then navigate to be adjacent to the door to open it
      .

You will receive the current state of the grid as a NxM matrix of symbols separated by whitespaces, and with
      coordinates for each row and column. For example, given the 6x7 grid:

  0 1 2 3 4 5 6
0 # # # # # # #
1 # _ _ _ _ _ #
2 # A _ _ _ K #
3 # _ # D # _ #
4 # _ # G # _ #
5 # # # # # # #

The goal is in position (4,3), the agent's position A is (2,1), the door D is at (3,3), and the key K is at
      (2,5).

The agent's possible actions at each step are:

Actions:
- UP: Move Up
- DOWN: Move Down
- LEFT: Move Left
- RIGHT: Move Right

Your final answer should be a valid JSON object provided in the following form:

```
{
  "action": "<UP|DOWN|LEFT|RIGHT>",
}
```

DO NOT INCLUDE ANY `json` or `jsonb` in your response NOR TICK MARKS LIKE THIS: ```json ```. Start with { and
      end with } exactly.

Inputs

Current grid state:

{{grid_state}}

Agent status:
- Carrying key: {{carrying_key}}
```

G ADDITIONAL REPRESENTATIONAL EVALUATION RESULTS

G.1 COGNITIVE MAP ENCODING ACROSS LAYERS

We examine how world model information develops across layers in GPT-OSS-20B by training MLP probes on activations from early (7), middle (15), and late (23) layers for grid size 11. Detailed probe performance is reported in Fig. 13. Goal-specific spatial information is already present in early layers, but the precision of the agent and goal classes continues to increase in layer 15. By the later layers, overall accuracy declines again, along with recall and precision for agent and goal. This pattern suggests that, at end-of-prompt-token indices, spatial information is most explicitly represented at intermediate layers and is subsequently transformed for the computation of other features rather than being directly preserved for next-token prediction.

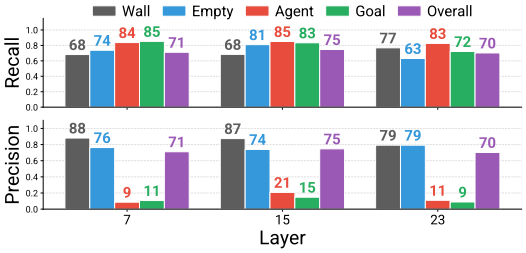


Figure 13: Performance of cognitive map probes across layers. Middle layer activations show the highest overall accuracy, but goal-specific information is already present in early layers.

G.2 SIZE-SPECIFIC COGNITIVE MAP PROBING RESULTS

In the main experiments, we trained a single MLP probe on data from all grid sizes, padding inputs to size 15. While this approach has the obvious advantage that we can use the same probe for any grid world, it could result in a worse performance than size-specific classifiers.

We test this by training size-specific probes without padding for each grid size. Overall accuracy results are shown in Fig. 14, and detailed Precision and Recall results for MLP probes are shown in Fig. 15.

Comparing the Linear probe accuracy to Fig. 5 (left), we see that for Linear probes size-specific approach works better for smaller grids and underperforms on larger grids. This suggests that the padding approach we use for general probes does not work well with Linear probes. The MLP probes, in contrast, perform comparably well in both settings. If we compared Fig. 15 to Fig. 5 (centre), size-specific MLP probes achieve higher accuracy for smaller grids, but match general probes performance for grid sizes 11–15. Precision for the agent and goal classes is higher, but recall is substantially lower for grid sizes 11–15.

Given these uneven performance differences and the flexibility of the size-agnostic approach, we adopt the size-independent MLP probe in our main experiments.

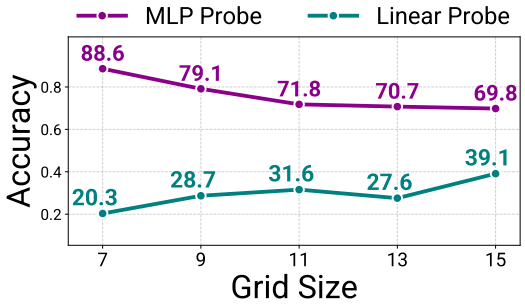


Figure 14: Overall accuracy of size-specific MLP and Linear probes. Size-specific linear probes perform better than general ones and MLP probes are comparable, but they less flexibility than those following the size-agnostic approach.

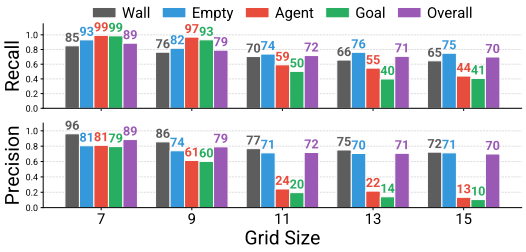


Figure 15: Detailed performance of size-specific MLP probes.

G.3 GOAL DISTANCE PROBING RESULTS

We also test if internal representations of GPT-OSS-20B encode information about the distance between the agent and the goal. As with cell identity, we hypothesize that distance to the goal may be tracked because of its direct relevance for goal-directed behaviour. In our grid worlds, the optimal state value is a monotonic function of the distance to the goal (cf. §3). To test this, we train linear and MLP probes on pre- and post-reasoning activations, using the length of the optimal trajectory as the ground-truth.

Table 4: Distance probe performance.

Probe Type	Reasoning Stage	MAE ↓	R^2 ↑
MLP Probe	Pre-Reasoning	3.16	0.40
	Post-Reasoning	**2.67**	0.36
Linear Probe	Pre-Reasoning	3.40	0.42
	Post-Reasoning	3.13	**0.47**

Results are reported in Tab. 4. Across conditions, goal distance can be decoded with a mean absolute error of approximately 3 steps, both before and after reasoning. The best performance is achieved by the post-reasoning MLP probe, with a mean absolute error of 2.67. These results suggest that, by reasoning about the grid and planning the next steps, the model develops more accurate representation of the distance it needs to cover to reach the goal.

G.4 INVERSE RECOVERY RATE

To complement the analysis in §5.2 and Tab. 2, we also compute a *reverse recovery* statistic, i.e., the proportion of actions that are suboptimal with respect to the decoded cognitive map but optimal with respect to the ground truth. Reverse recovery declines as grid size and obstacle density increase (from 86.6% at 7×7 to 61.1% at 15×15, and from 100% at $d = 0.0$ to 60.9% at $d = 1.0$. We interpret this as reflecting growing uncertainty in the agent's internal world state representation as environment complexity increases.

G.5 ADDITIONAL PLAN DECODER RESULTS

We analyse predicted trajectory lengths (Table 5). Post-reasoning decoding achieves a higher exact length match rate (19% vs. 16%) and exhibits a small positive bias in average length (average difference between predicted and true is $+0.12$), whereas pre-reasoning decoding exhibits a small negative bias (-0.12). Median absolute length error is 1 step in both settings.

Table 5: Sequence length analysis for one-shot plan decoding.

Metric	**Pre-reasoning**	**Post-reasoning**
Exact length match (%) ↑	15.71	18.85
Predicted length (avg)	4.81	5.05
Ground-truth length (avg)	4.93	4.93
Length bias (avg, pred − true)	-0.12	$+0.12$
Median abs. length error (steps)	1.0	1.0

H INTERVENTIONAL ANALYSIS VIA ACTIVATION PATCHING

To complement the representational analyses in the main paper, we perform activation patching on GPT-OSS-20B using counterfactual trajectories in which either the agent position or the goal position was moved. For our metric, we examine whether the model's predicted action is changed. In particular, we ensure that the optimal action set is disjoint in the counterfactual set. We further ensure that the agent picks an optimal action in both the original and counterfactual cases, before any patching, to eliminate performance as a confounder.

We intervene at four token-level sites: the full serialized grid state, only the modified grid cells, the pre-reasoning boundary and the post-reasoning boundary; the last two of which are identical to the probing sites we use. We evaluate patching at three representative layers (7, 15, and 23), as well as all layers at once, in both original→counterfactual and counterfactual→original directions. For simplicity, we generate counterfactuals for size 7 grids, and filter for the disjoint constraint described earlier.

The single-layer results were uniformly negative as interventions: across 456 executed runs, patching never changed the model's predicted action $(0/456)$. This held for all four intervention sites and for each tested layer individually, and was consistent for both directions. In contrast, we find that patching across all layers was always effective in changing the model's predicted action.

Separately, we also design a control in which the optimal action does not change; we find patching here to be perfectly stable, in which the output action does not change $(144/144)$, indicating that patching did not degrade the model when the source and target already supported the same action.

Taken together, these results suggest that the information supporting action selection is not causally localized in any single tested layer, even when the relevant grid tokens are directly targeted. This is consistent with the interpretation that task-relevant state information is distributed across layers: it is sufficiently decodable for probing analyses, but not easily steerable through narrow single-layer interventions. Further work is needed to isolate minimal causal interventions in this environment.