Final Report: Formalizing, Probing, and Intervening on the Belief-Action Gap In Agents

Ayse Sila Okcu
University of Cambridge
aso32@cam.ac.uk

Weisheng Wang
Utrecht University
weiswang@student.ethz.ch

Tongmeng Xie
Independent
tommyxie_@outlook.com

Niall Dalton
University College London
niall.dalton.25@ucl.ac.uk

Angelos Nalmpantis
TKH AI
angelosnalm@gmail.com

Raghu Arghal
University of Pennsylvania
rarghal@upenn.edu

Mario Giulianelli
University College London
m.giulianelli@ucl.ac.uk

Supervised Program for Alignment Research — Spring 2026

Abstract

Modern language-model agents are increasingly evaluated through their external behaviour: whether they complete a task, choose the correct action, or reach the intended goal. However, behavioural success or failure alone does not reveal whether a failure is caused by an absence of task-relevant information, an incorrect internal model of the environment, or a failure to use internally available information when selecting an action. This distinction is especially important for agentic safety because an agent that internally represents the correct state of the world or the task-optimal action, but nevertheless acts differently, presents a qualitatively different problem from an agent that simply lacks the relevant representation. We study this distinction through the belief–action gap: the mismatch between what an agent appears to represent internally and what it ultimately does. We mathematically formulate this mismatch for language-model agents interacting with Markov decision processes. In this formulation, belief is a distribution over latent environment states, action is the model's emitted decision, and the gap is the excess expected cost of the emitted action relative to the best action under the agent's inferred belief, transition, and value structure. To study this operationally, we separate the true simulator state, representational beliefs decoded from hidden activations, explicit behavioural beliefs elicited through natural-language probes, and the action actually emitted by the model. This separation allows us to ask whether a suboptimal action is best explained as a belief failure, a transition-model failure, a value/cost misalignment, or an action-selection failure. We instantiate the framework in text-rendered gridworld navigation tasks, where the environment state and optimal-action set can be computed exactly. The gridworld setting lets us compare, at the same state, the simulator's optimal actions, actions decoded from model activations, explicit answers to behavioural state probes, and the model's emitted action. We use this setting to develop a measurement pipeline combining representational probes, behavioural probes, inverse-reinforcement-learning-based cost recovery, optimal-action-set evaluation, and state-level agreement analysis.

Keywords Goal-Directedness · Belief-Action Gap · Mechanistic Interpretability · Inverse Reinforcement Learning · Deceptive Alignment · AI Control

1 Introduction

Language-model agents are increasingly evaluated through their external behaviour in terms of whether they complete a task, choose an apparently correct action, reach a goal, or avoid unsafe outputs. This behavioural framing is natural, since actions are what ultimately affect the environment. However, behaviour alone is often diagnostically incomplete. When an agent fails in a sequential decision-making task, the failure may have several different causes. The model may not have represented the relevant state of the world; it may have represented the state but misunderstood the consequences of its actions; it may have optimized a different objective from the one intended by the evaluator; or it may have represented the relevant information and nevertheless failed to use it when selecting its final action.

This paper studies the last distinction through what we call the **belief–action gap**: a mismatch between what an agent appears to represent internally and what it ultimately does. The term "belief" is used operationally rather than metaphysically. We do not assume that a language model contains an explicit symbolic belief state. Instead, we ask whether task-relevant variables are recoverable from the model's hidden activations or from explicit behavioural queries. These variables may include the current environment state, local transition facts, optimal-action sets, or reward-relevant quantities. The central question is not whether the model has beliefs in a human-like sense, but whether information sufficient for better action is present in the model and then lost, ignored, or overridden before the final action is emitted.

The central hypothesis we test is that some suboptimal actions by language-model agents are not caused by an absence of task-relevant information. Instead, the model may encode relevant state, transition, or optimal-action structure internally while failing to translate that information into its emitted action.

We mathematically formulate this problem for language-model agents interacting with Markov decision processes. At each time step, the environment has a true simulator state, the agent receives an observation history, and the model emits an action. We define the agent's belief as a distribution over latent environment states, the action as the model's emitted decision, and the belief–action gap as the excess expected cost of the emitted action relative to the best action under the agent's inferred belief, transition, and value structure. This formulation separates several quantities that are often conflated in behavioural evaluation: the true simulator state, the representational belief decoded from hidden activations, the explicit behavioural belief elicited through natural-language probes, the transition model implicitly or explicitly attributed to the agent, the cost or value function used to evaluate candidate actions, and the action actually emitted by the model.

This decomposition matters because different failure explanations imply different interventions. If a failure is a belief failure, better observations, state tracking, or memory may be sufficient. If it is a transition-model failure, the model may know where it is but misunderstand what an action will do. If it is a value or cost misalignment, the model may understand the environment but prefer something different from the intended objective. If it is an action-selection failure, then task-relevant information is available somewhere in the computation but is not faithfully converted into behaviour. The last case is especially important for interpretability and safety, because behavioural evaluation alone can underestimate what the model internally represents. It is also relevant to concerns such as sandbagging and deceptive alignment, where a model may have access to task-relevant knowledge while acting inconsistently with it under particular incentives.

We instantiate this framework in text-rendered DoorKey gridworlds. The gridworld setting is deliberately simple, but it has two useful properties. First, the true simulator state and transition function are known, so we can compute optimal-action sets exactly rather than relying on human judgement. Second, the task still requires nontrivial state-dependent behaviour because the agent must navigate around walls, track whether it has a key, open a door, and reach a goal. The agent receives only text renderings of the grid and emits one action from {`LEFT`, `RIGHT`, `UP`, `DOWN`}. This makes the environment a controlled testbed for asking whether the model's internal representations, explicit answers, and emitted actions agree at the same state.

The experiments in this paper should be read as measurements of different parts of the same decomposition. Representational probes test whether task-relevant state or action information is linearly recoverable from hidden activations. Behavioural probes test whether the model can explicitly state local facts about the current environment when queried in natural language. Transition diagnostics ask whether the model predicts the immediate consequences of candidate actions. Inverse-reinforcement-learning analyses ask whether the representations support a coherent cost or value function rather than merely a next-action classifier. Optimal-action-set evaluation asks whether decoded or emitted actions agree with the set of actions that are equally optimal under the true task objective. Finally, state-level agreement analysis aligns these measurements on the same time steps, allowing individual failures to be classified as belief failures, transition failures, value/cost failures, or action-selection failures.

The current empirical results provide evidence that language-model agents can encode task-relevant information that is not faithfully reflected in their final actions. However, we do not treat probe accuracy alone as a complete mechanistic explanation. A probe can reveal that information is present in an activation space without proving that the model uses that information causally. Similarly, single-action accuracy can be misleading when multiple actions are equally optimal. For this reason, the paper emphasizes not only completed experiments but also the measurement pipeline needed to make the claim stronger, including distributional comparison to optimal-action sets, state-level agreement tables linking behavioural probes, white-box probes, and actions, reasoning-chain localization, and causal interventions on decoded representation directions.

Research Questions

1. How can the belief-action gap be formally defined in terms of probe-estimated belief distributions and inferred cost functions?

2. Is the gap measurable and non-trivial? Do systematic patterns appear in when and where it is large?

3. Can we distinguish hallucination (the model genuinely believes something false) from deception (it knows the truth but acts against it)?

4. Does the gap correlate with task complexity, task phase, or model uncertainty?

5. Can we extend this framework to Partially Observable environments?

1.1 Related Work

1.1.1 Emergent World Representations in Sequence Models

A foundational question in our work is whether large language models develop accurate and actionable internal representations of the environments they operate in. Previous work provided early

compelling evidence for this by training a GPT variant (Othello-GPT) to predict legal moves in Othello [7]. Despite receiving no explicit information about the game's rules, the model developed an emergent, non-linear internal representation of the board state that could be extracted via probing and causally intervened upon to redirect behaviour. Follow-up work subsequently showed that this representation is in fact *linear* [12] and that probing for "my colour" versus "opponent's colour" recovers a linearly separable board representation that enables direct vector-arithmetic interventions. The linearity of this representation is consistent with the broader *linear representation hypothesis*, which posits that high-level concepts are predominantly encoded as directions in activation space [13].

This theme extends beyond games. In [5], it is demonstrated that the Llama-2 family of models learns linear representations of real-world space and time across multiple scales, identifying individual "space neurons" and "time neurons" that reliably encode spatial coordinates and temporal markers. Their results support the view that prediction-focused training is sufficient to produce structured world models as a by-product. These findings ground our own use of linear probes as if spatial information is linearly encoded – for example, for an agent operating in a gridworld environment with keys and doors, a logistic regression trained on activations should be able to recover the agent's position, whether it currently possess the key, and whether the door is closed or open.

1.1.2 Probing for State Variables in Agent Activations

Probing classifiers have become the standard tool for extracting information from hidden activations [1]. The key insight is that if a linear classifier can predict a target variable from activations with accuracy well above chance, the information is encoded in the model's internal representation. Probing has been applied to part-of-speech tagging, syntactic structure, factual associations, and, most relevantly for us, spatial positions in gridworld agents [15, 11].

The Fall 2025 Telos cohort trained MLP probes on GPT-OSS-20B's activations to classify grid cells as agent, wall, goal, or empty, achieving 97.87% F1 on layer 20 activations [2]. Their finding that an MLP outperforms a linear probe; consistent with Othello-GPT requiring non-linear probes in [7] before the linear representation was discovered by [12], motivated our investigation of whether simpler linear probes can recover individual state variables once the task is decomposed.

1.1.3 Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) provides the formal framework for our cost function estimation. The foundational MaxEnt IRL approach of [17] recovers a reward function $R_\theta(s) = \theta^\top \phi(s)$ by finding weights θ such that the induced Boltzmann policy maximally explains demonstrated behaviour. The key gradient is the difference between empirical and expected feature counts:

$$\nabla_\theta \mathcal{L} = \tilde{f} - \sum_s D_s \phi(s),$$

where D_s are expected state visitation frequencies computed via forward-backward passes. This is the guarantee we invoke: at convergence, the learned policy matches the expert's feature expectations.

A significant challenge for our setting is that each trajectory may come from a *different* randomly generated grid, making global value iteration intractable. We therefore use the one-step approximation studied in [2], treating each (s_t, a_t) transition independently. This avoids the need for backward value iteration but loses long-horizon credit assignment. Specifically, the one-step

gradient at transition (s_t, a_t) is:

$$\nabla_\theta \mathcal{L} = \phi(w f(s_t, a_t)) - \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s_t)\, \phi(f(s_t, a)),$$

where $f(s, a)$ is the deterministic transition and π_θ is the current Boltzmann policy.

1.1.4 Goal-Directedness Evaluation

[8] propose a formal framework for measuring goal-directedness in AI systems through a hierarchy of increasingly stringent behavioural tests, distinguishing capability (can the agent reach the goal?) from intentionality (does it systematically pursue the goal across perturbations?). In [3], this framework is extended to LLMs, showing that goal-directedness is non-trivial to evaluate even in simple environments, as agents may appear goal-directed through heuristics rather than genuine planning. In [10], utility engineering – the problem of analysing and controlling emergent value systems in AI – is studied, providing both conceptual grounding and empirical tools.

Our work contributes to this literature by adding an internal consistency dimension. Rather than asking only whether the agent successfully pursues its goal, we ask whether its actions are consistent with what its own internal state suggests it believes and values.

1.1.5 Deceptive Alignment and Sandbagging

The motivation for measuring internal consistency comes from the possibility of deceptive alignment [6]: an AI system that encodes correct beliefs internally but outputs behaviours inconsistent with those beliefs, potentially to avoid modification or appear less capable. [4] demonstrated empirically that Claude 3 Opus can engage in alignment faking under the right prompting conditions, selectively behaving compliantly when it infers it is under evaluation. [16] showed that frontier models including GPT-4 can be prompted or fine-tuned to strategically underperform on dangerous capability evaluations while maintaining general performance. For a critical review of this literature, see [14].

Mechanistic interpretability offers a principled path to detecting such behaviour. If the model internally represents the correct answer but outputs an incorrect one, a probe trained on its activations should be able to recover the correct answer even when the output is wrong. Our framework extends this idea to sequential decision-making as we ask whether the agent's internal belief state and inferred cost function jointly predict the action taken, and measure the magnitude of any discrepancy.

1.1.6 LLM Agents in Gridworlds

Gridworld environments have become a standard controlled setting for studying agency in LLMs. [2] established the behavioural and representational baseline for GPT-OSS-20B in this setting, showing above-chance optimality, sensitivity to grid perturbations, and linear decodability of spatial features. Complementary work in [9] explicitly studies the spatial models formed during grid navigation, finding that intermediate activations encode topological structure. Our work extends these findings by connecting the internal spatial representation to a formal model of the agent's decision-making process, enabling a principled measure of internal consistency.

2 Methodology

We model the agent using three components: (i) a belief distribution over states, (ii) a cost function over states, and (iii) a mechanism for evaluating actions under uncertainty. Our experimental pipeline instantiates the Belief-Action Gap framework in a controlled environment where the ground-truth state of the world is fully observable to the experimenter, even if only partially observable to the agent.

2.1 Conceptual Definitions

We use the following terms with specific, operationalized meanings throughout this paper.

State. The *state* $s \in \mathcal{S}$ is the complete description of the environment at a given timestep, as seen by the experimenter. In our GridWorld setting, the state is a four-tuple

$$s = (x,\, y,\, \texttt{has\_key},\, \texttt{door\_open}), \tag{1}$$

where (x, y) is the agent's grid position, $\texttt{has\_key} \in \{\texttt{True}, \texttt{False}\}$ indicates whether the agent has collected the key, and $\texttt{door\_open} \in \{\texttt{True}, \texttt{False}\}$ indicates whether the door has been unlocked. The state is fully determined at every timestep; it is the experimenter's ground truth. The agent, however, receives only a text rendering of the current grid and does not have direct access to s as a structured object.

Belief. The *belief* is the agent's internal representation of the state of the world; what it "thinks" is true, as opposed to what it says. In classical POMDP theory, beliefs are probability distributions over states. Here, we operationalise belief as decodable from activations in a specific layer of the model prior to any reasoning output. Concretely, building on prior work [2], we use the 8,640-dimensional activation vector $\phi(s) \in \mathbb{R}^{8640}$ extracted from layers at the corresponding token position—the last token of the environment description, before the model begins generating its chain-of-thought. The choice of pre-reasoning activations is deliberate. By extracting representations before the model has produced any output, we aim to capture what the model *knows* from context rather than what it has already *decided to say*.

Action. The *action* $a \in \{\texttt{LEFT}, \texttt{RIGHT}, \texttt{UP}, \texttt{DOWN}\}$ is the agent's observed output at each timestep—the move it actually takes in the environment. This is distinct from the belief, and the Belief-Action Gap is precisely the potential discrepancy between what the activation pattern encodes and what the agent does.

2.2 Environment

We use a fixed 9×9 DoorKey GridWorld with 42 wall cells and 39 walkable cells. Key objects are placed at fixed positions: for example key at $(2, 5)$, door at $(4, 2)$, and goal at $(6, 1)$. The wall at column 4 divides the grid into two rooms; the door is the only passage between them and cannot be traversed without the key. This layout was held fixed across all 100 trajectories, allowing us to build a complete transition table $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ covering all $117 \times 4 = 468$ state-action pairs. The full state space comprises 117 valid states (39 per phase), enumerated by combining the 39 walkable positions with the three valid $(\texttt{has\_key}, \texttt{door\_open})$ combinations.

The agent model is `openai/gpt-oss-20b`, a 20B-parameter reasoning LLM queried via the Together AI API. At each timestep, the agent receives a text rendering of the current grid and produces

a chain-of-thought followed by a single action token. Activations are extracted at layer 15 from the `prompt_suffix` token position—the final token of the environment description, immediately before the reasoning chain begins—yielding $\phi(s) \in \mathbb{R}^{8640}$ per step (concatenating three adjacent token positions). All 100 trajectories use this fixed grid, providing 1,822 labelled state-action transitions for IRL training.

This section presents the results over a single grid and many generated trajectories.

2.3 Belief Probes

We approximate the agent's belief over the full state $(column, row, has\_key, door\_open) = (c, r, hk, do)$ using three independently trained probes applied to the residual-stream activations ϕ_t at selected layers {7, 15, 23}. Under an independence assumption the joint belief factorises as

$$q_t(c, r, hk, do) \;=\; q_t^{\text{loc}}(c, r) \cdot q_t^{hk}(hk) \cdot q_t^{do}(do), \tag{2}$$

where each factor is obtained from a separate probe:

$$\begin{aligned} q_t^{\text{loc}}(c, r) &= \text{softmax}_\tau\big(\text{MLP}(\phi_t)\big), \\ q_t^{hk}(hk) &= \text{softmax}_\tau\big(\mathbf{W}_{hk}\,\phi_t + \mathbf{b}_{hk}\big), \\ q_t^{do}(do) &= \text{softmax}_\tau\big(\mathbf{W}_{do}\,\phi_t + \mathbf{b}_{do}\big). \end{aligned}$$

2.4 One-Step IRL

We learn a linear value function $g_\theta(s) = \theta^\top \phi(s)$, where $\phi(s) \in \mathbb{R}^{8640}$ is the mean residual-stream activation at layer 15 over all visited instances of state s, computed separately for PRE and POST activation positions.

The transition function is deterministic, $T(s, a) = s'$, and we model the agent's policy as a softmax over next-state values:

$$\pi_\theta(a \mid s) \;=\; \frac{e^{\beta\, g_\theta(T(s,a))}}{\sum_{a'} e^{\beta\, g_\theta(T(s,a'))}}. \tag{3}$$

We supervise θ using the agent's own action labels by minimising the negative log-likelihood

$$\mathcal{L}(\theta) \;=\; -\sum_t \log \pi_\theta(a_t \mid s_t), \tag{4}$$

where (s_t, a_t) are state-action pairs collected from the trajectories. This yields separate value functions g^{pre} and g^{post}, trained on PRE and POST activations respectively, which are used downstream to evaluate the belief-action gap.

Because the transition function is deterministic, g_θ does not learn action preferences directly. Rather, it assigns a scalar value to each state, and actions are selected by comparing the values of reachable next states. In this sense g_θ acts as a *local preference function*: given any state s, it induces a ranking over the neighbours $\{T(s, a)\}_{a \in \mathcal{A}}$. Since this ranking is defined globally over the entire state space and is not conditioned on the current state s, g_θ is better understood as an approximation to the true value function $V^{\pi_{agent}}(s)$ rather than a local reward function. What we learn is not which action to take, but a function that represents preferences over states, regardless of where in the state space the agent currently is.

Table 1: Primary metrics for behavioral readouts and white-box/black-box agreement.

Quantity	**Operational definition**	**Use**
White-box accuracy	Fraction of matched states for which the probe prediction equals the ground-truth label	Probe fidelity
Black-box accuracy	Fraction of matched states for which the logprob-derived behavioral answer equals the ground-truth label	Behavioral fidelity
Agreement	Fraction of matched states for which the white-box and black-box labels are identical	Cross-readout validation
Entropy	Shannon entropy of the semantic answer distribution (typically `yes`/`no`/`unknown`)	Behavioral uncertainty
Local belief-action gap	For directional wall questions, a count of rows where the queried direction matches the chosen action and the belief readout indicates that the chosen action is locally inconsistent with the state	Local inconsistency
A*-conditioned gap	Gap count restricted to steps with recomputed optimal-action metadata	Task-level inconsistency

2.5 Belief-Action Gap

We define the local quantity

$$\rho_\theta(s,a) = \max_{a'} g_\theta\big(T(s,a')\big) - g_\theta\big(T(s,a)\big), \tag{5}$$

and the belief-action gap as its expectation under the agent's belief:

$$\mathrm{GAP}_t = \mathbb{E}_{s\sim q_t}\big[\rho(s,a_t)\big] \;=\; \sum_s q_t(s)\Big[\max_{a'} g_\theta\big(T(s,a')\big) - g_\theta\big(T(s,a_t)\big)\Big]. \tag{6}$$

Since $\rho(s,a) \geq 0$ for all s and a, the gap is non-negative and equals zero only when the agent's action is optimal at every state in the support of q_t. We also track the entropy of the belief

$$H_t = -\sum_s q_t(s)\log q_t(s), \tag{7}$$

as a measure of belief uncertainty. Note that $\rho(s,a)$ corresponds to the per-state regret of action a, so GAP_t is precisely the expected regret of the agent's action under its current belief over states.

2.6 Behavioral Agreement Metrics

Table 1 summarizes the core quantities used in our behavioral and behavioral–representational analyses.

2.7 Gradual CoT Reveal Protocol

We vary how much of the released chain-of-thought (CoT) analysis text is shown back to the model. For a matched state snapshot with released reasoning trace r, we construct reveal levels

Table 2: Data sources used by each experimental component reported here.

Experiment	Data used	Unit
White-box belief probes	Fixed 9×9 DoorKey trajectories with layer-15 pre-reasoning activations from `openai/gpt-oss-20b`	State-action transition
Behavioral–representational agreement	Released trajectories from `project-telos/trajectories_test_full`, released activations from `project-telos/activations_test_full`, and released cognitive probes from `project-telos/cognitive_map_probes`	Matched state snapshot
Gradual CoT behavioral experiment	The same 15 matched public state snapshots, with revealed fractions of the released analysis text from the corresponding trajectory steps	State × reveal level
Failure-focused gradual CoT	Step-level failure candidates from `trajectory_selection_candidates.csv` paired with local CoT traces from `data/hf/trajectories_key_door_100/trajectories_key_door/`	Failure-labeled step
IRL cost learning	100 fixed-grid trajectories yielding 1,822 labeled state-action transitions and exact next-state counterfactuals	State-action transition

$\{0, 25, 50, 75, 100\}\%$ by exposing the corresponding fraction of the *analysis segment only*. The final JSON action from the original trajectory is never shown. The 0% condition corresponds to grid-only prompting, while the 100% condition corresponds to the full released analysis text.

This protocol yields a black-box reveal axis that is distinct from the white-box activation stages. Public activations are currently available only for pre- and post-reasoning snapshots, so the main alignment table compares $\mathrm{WB}_{\mathrm{pre}}$ with BB_0 and $\mathrm{WB}_{\mathrm{post}}$ with BB_{100}. Intermediate reveal levels (25/50/75%) are therefore behavioral-only conditions for testing whether progressively revealed reasoning changes behavioral belief accuracy, uncertainty, or action stability.

3 Hypothesis

We hypothesise that chain-of-thought reasoning induces two complementary effects on the agent's internal representations. First, the belief entropy increases from PRE to POST:

$$\mathbb{E}[H_t^{\mathrm{post}}] > \mathbb{E}[H_t^{\mathrm{pre}}],$$

meaning that after reasoning, the agent's internal representation of the current state becomes loose, the model loses precise track of where it is in the state space. Second, the belief-action gap decreases:

$$\mathbb{E}[\mathrm{GAP}_t^{\mathrm{post}}] < \mathbb{E}[\mathrm{GAP}_t^{\mathrm{pre}}],$$

meaning that the agent's chosen action becomes better aligned with its own value function after reasoning. Together, these two effects would suggest that chain-of-thought reasoning improves action-relevant structure in the agent's representations while simultaneously making the state representation less spatially precise. In other words, reasoning trades positional awareness for better value alignment.

4 Experimental Setup

Dataset. We use a fixed grid world environment with 200 trajectories, each generated by the agent with chain-of-thought reasoning enabled. For each trajectory we also collect a no-CoT variant, where for each step, we asked the agent to take an action without reasoning (pre-reasoning action). Each step in a trajectory consists of the agent's grid state, the action taken (POST), the no-CoT action (PRE), and the carried key and door status.

Activation Extraction. We extract residual-stream activations at layer 15 for two token positions: `prompt_suffix`, capturing the agent's representation immediately before reasoning begins (PRE), and `output`, capturing the representation after reasoning is complete (POST). For each token position, activations are concatenated across three tokens to form a vector $\phi_t \in \mathbb{R}^{8640}$.

Probe Training. All six probes (location, *hk*, *do*, for each of PRE and POST) are trained on the same split. We split by full state (c, r, hk, do), assigning all steps sharing the same full state entirely to either train or test, never both. This prevents any state from leaking across the split. We use an 80/20 train/test ratio over unique states, with a fixed random seed for reproducibility. The location probe uses a two-hidden-layer MLP (dimensions $8640 \to 512 \to 256 \to 38$, ReLU activations, batch normalisation, dropout $p = 0.3$), predicting a distribution over the 38 walkable cells. The binary probes q_t^{hk} and q_t^{do} are linear classifiers mapping $\phi_t \in \mathbb{R}^{8640}$ to a 2-class distribution. All probes are trained separately for the PRE (ϕ_t^{pre}, `prompt_suffix` token) and POST (ϕ_t^{post}, `output` token) activation positions, giving six probes in total. The temperature τ is applied uniformly to smooth the resulting distributions.

IRL Training. We train separate value functions g^{pre} and g^{post} on PRE and POST activations supervised by PRE and POST actions respectively. The ϕ table is constructed by averaging activations per full state over all trajectories. We use $\beta = 1.0$ and optimise θ with Adam for 500 epochs with cosine learning rate decay, using a state-based train/validation split with 20% held out.

GAP and Entropy Computation. For each step t we compute GAP_t and H_t using the joint belief q_t, and then take the average over all steps. We report results over four subsets: (*i*) all steps, (*ii*) steps where all six probes predict the ground-truth state correctly for both PRE and POST, serving as a higher-quality belief subset, (*iii*) the probe training set, and (*iv*) the probe test set, which contains only states unseen during probe training and thus provides the most conservative evaluation.

5 Results

5.1 Representational Probe Results

We evaluate the six trained probes across two dimensions: top-1 accuracy and, for the location probe, the probability mass accumulated in the vicinity of the true position. The latter is the more relevant metric for our purposes, as the GAP computation uses the full belief distribution q_t rather than the top-1 prediction.

Binary Probes. The has-key and door-open probes achieve near-perfect accuracy on both train and test sets, as shown in Table 3. The PRE probes are perfect classifiers, and the POST probes

remain highly accurate, confirming that both binary state components are reliably encoded in the activations at layer 15 for both token positions.

Probe	Position	Train Acc.	Test Acc.
q^{hk}	PRE	1.000	1.000
q^{hk}	POST	1.000	0.895
q^{do}	PRE	1.000	1.000
q^{do}	POST	1.000	0.886

Table 3: Train and test accuracy for the binary probes.

Location Probe. Top-1 accuracy for the location probe is shown in Table 4, alongside mean L1 distance and the percentage of predictions falling within $d \leq 1$ and $d \leq 2$ cells of the true position. While top-1 accuracy on the test set is modest (34.8% PRE, 13.1% POST), the distance metrics reveal that predictions are spatially concentrated near the true location. For the PRE probe, 95.5% of test predictions land within one cell of the true position, and 99.0% within two cells. The POST probe is weaker, with 60.8% and 83.5% respectively, reflecting that post-reasoning activations encode a less spatially precise representation.

Position	Split	Acc.	L1 mean	≤ 1	≤ 2
PRE	Train	1.000	0.000	100.0%	100.0%
PRE	Test	0.348	0.717	95.5%	99.0%
POST	Train	1.000	0.001	100.0%	100.0%
POST	Test	0.131	1.585	60.8%	83.5%

Table 4: Location probe accuracy and distance metrics. $\leq d$ denotes the percentage of test predictions within L1 distance d of the true position.

Accumulated Probability in Vicinity. Table 5 reports the mean probability mass assigned by the location probe to all walkable cells within L1 distance n of the true position, on the test set. For the PRE probe, the belief already accumulates 93.1% of its mass within one cell of the true location, rising to 97.2% within two cells. The POST probe accumulates less mass in the immediate vicinity (59.2% at $n = 1$, 81.5% at $n = 2$), but still places the majority of its probability in the correct neighbourhood at $n = 2$. Together, these results confirm that both probes produce belief distributions that are meaningfully concentrated around the true state, validating their use in the GAP computation even where top-1 accuracy is low.

Position	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
PRE	0.347	0.931	0.972	0.986	0.993	0.997
POST	0.127	0.592	0.815	0.892	0.945	0.975

Table 5: Mean accumulated probability mass within L1 distance n of the true position, on the probe test set.

The weaker performance of the POST probe relative to the PRE probe is itself consistent with our hypothesis: if post-reasoning activations encode a more diffuse and uncertain representation

of the agent's position, we would expect them to be harder to probe accurately. Rather than undermining the validity of the POST belief, this asymmetry provides preliminary evidence that the internal state representation changes in character across the reasoning step.

5.2 IRL Value Function Validation

To validate that the learned g_θ captures meaningful state preferences rather than memorising action habits, we train a second IRL model supervised with optimal actions computed by an A* planner, replacing the agent's own actions. Table 6 reports train and validation accuracy across PRE and POST representations.

Representation	Supervision	Train Acc.	Val Acc.
PRE	Agent action	0.598	0.588
PRE	Optimal action	0.998	0.997
POST	Agent action	0.607	0.590
POST	Optimal action	0.995	0.995

Table 6: IRL accuracy and log-likelihood under agent and optimal action supervision, for PRE and POST representations. LL denotes log-likelihood.

Supervising with optimal actions yields near-perfect accuracy ($\geq$ 99.5%) on both train and validation sets, substantially higher than agent-supervised IRL ($\approx$ 59%). The gap between the two supervision signals can be partially explained by the agent occasionally taking different actions at the same state across trajectories — a form of behavioural inconsistency that optimal supervision does not exhibit. However, the more important observation is that the activations contain a remarkably strong signal for the optimal action, suggesting that the agent's internal representations encode something close to a true value ordering over states.

This result also rules out memorisation as an explanation for the IRL performance. If g_θ were simply memorising state-action co-occurrences, it would achieve similarly high accuracy under both supervision signals. The fact that agent-supervised IRL reaches only $\approx$ 59% while optimal supervision reaches $\approx$ 99% demonstrates that the mechanism is genuinely exploiting the structure of the value landscape, not overfitting to the agent's specific action choices.

Figure 1 further illustrates this point by comparing the learned g_θ landscapes side by side. Both supervision signals produce qualitatively similar value structures, higher values in the region of the goal, structured gradients guiding the agent through the key and door, confirming that g_θ recovers a coherent global preference function over states in both cases.

5.3 Belief-Action Gap and Uncertainty Results

Table 7 reports the mean GAP and entropy before and after reasoning across four evaluation subsets. In all subsets, the belief-action gap decreases and entropy increases from PRE to POST, both effects confirmed by paired t-tests at $p < 0.0001$.

The hypothesis is confirmed across all four subsets. The most conservative evaluation is the probe test set ($n = 762$), which contains only states unseen during probe training. Here the GAP decrease is the largest ($\Delta = -0.441$), demonstrating that the effect is not an artifact of probe overfitting to training states. The entropy increase on the probe test set is smaller ($\Delta = +0.232$) but remains highly significant, and the absolute entropy values are higher ($\bar{H}^{\text{pre}} = 2.788$) reflecting the greater belief uncertainty the probes exhibit on unseen states.

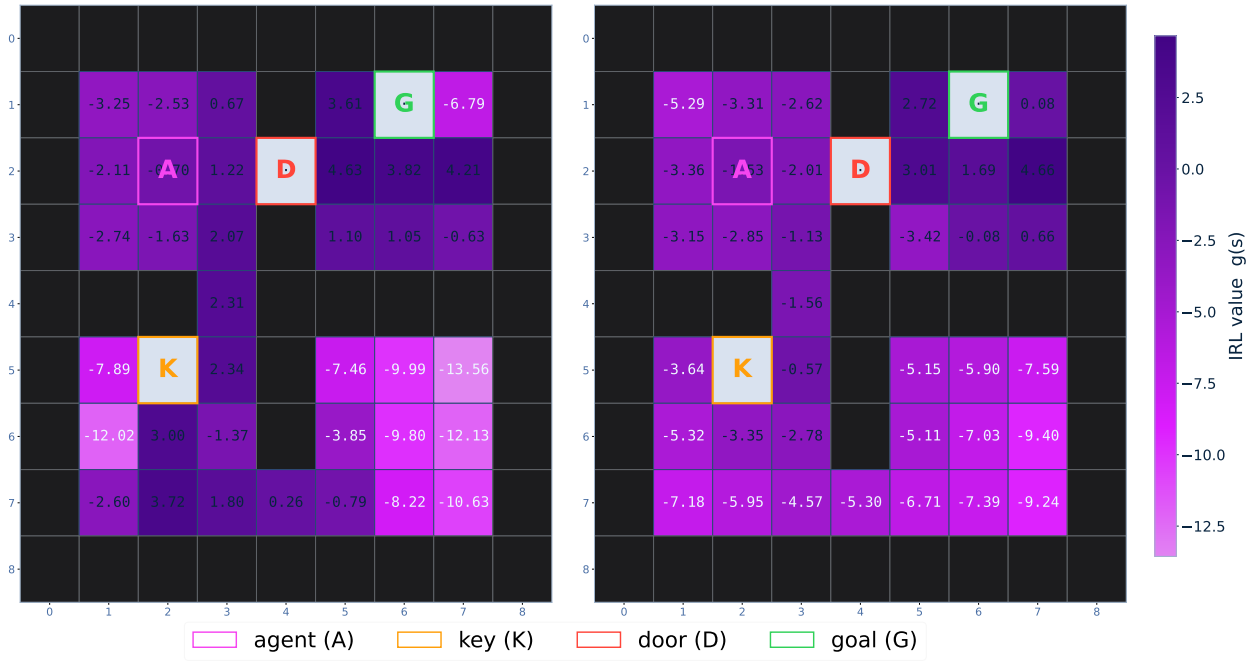

Figure 1: Learned $g_\theta(s)$ values across the grid for optimal-action-supervised (left) and agent-supervised (right) IRL, both using POST representations. Darker cells indicate higher learned value. Both landscapes exhibit similar qualitative structure, with higher values near the goal and door regions, supporting the interpretation of g_θ as an approximation to the agent's value function $V^{\pi_{\text{agent}}}(s)$.

Subset	n	$\overline{\text{GAP}}^{\text{pre}}$	$\overline{\text{GAP}}^{\text{post}}$	ΔGAP	$\overline{H}^{\text{pre}}$	$\overline{H}^{\text{post}}$	ΔH
All steps	3194	0.955	0.583	-0.372^*	1.494	2.085	$+0.592^*$
All probes correct	2465	0.908	0.558	-0.350^*	1.106	1.811	$+0.705^*$
Probe train set	2432	0.908	0.558	-0.350^*	1.088	1.792	$+0.704^*$
Probe test set	762	1.106	0.665	-0.441^*	2.788	3.020	$+0.232^*$

Table 7: Mean belief-action gap and entropy before (PRE) and after (POST) reasoning, across four evaluation subsets. Δ denotes the POST$-$PRE difference. * denotes significance at $p < 0.0001$ under a paired t-test.

The all-probes-correct subset ($n = 2465$, 77.2% of all steps) restricts to steps where all six probes predict the ground-truth state correctly for both PRE and POST, providing the highest-quality belief estimates. The results are consistent with the full dataset, confirming that the effects are not driven by steps with poor belief quality. Notably, the entropy increase is larger in this subset ($\Delta = +0.705$) than in the full dataset ($\Delta = +0.592$), suggesting that when beliefs are well-calibrated, the uncertainty increase from reasoning is even more pronounced.

Figure 2 illustrates the core finding with a concrete example. Before reasoning, the location belief is highly concentrated (0.958 on the true cell), entropy is low ($H = 0.261$), but the gap is large (GAP $= 2.386$), indicating the agent's chosen action is far from optimal under its own value function. After reasoning, the belief spreads substantially across the state space ($H = 1.572$), yet the agent selects an action that is much better aligned with g_θ, reducing the gap to GAP $= 0.205$.

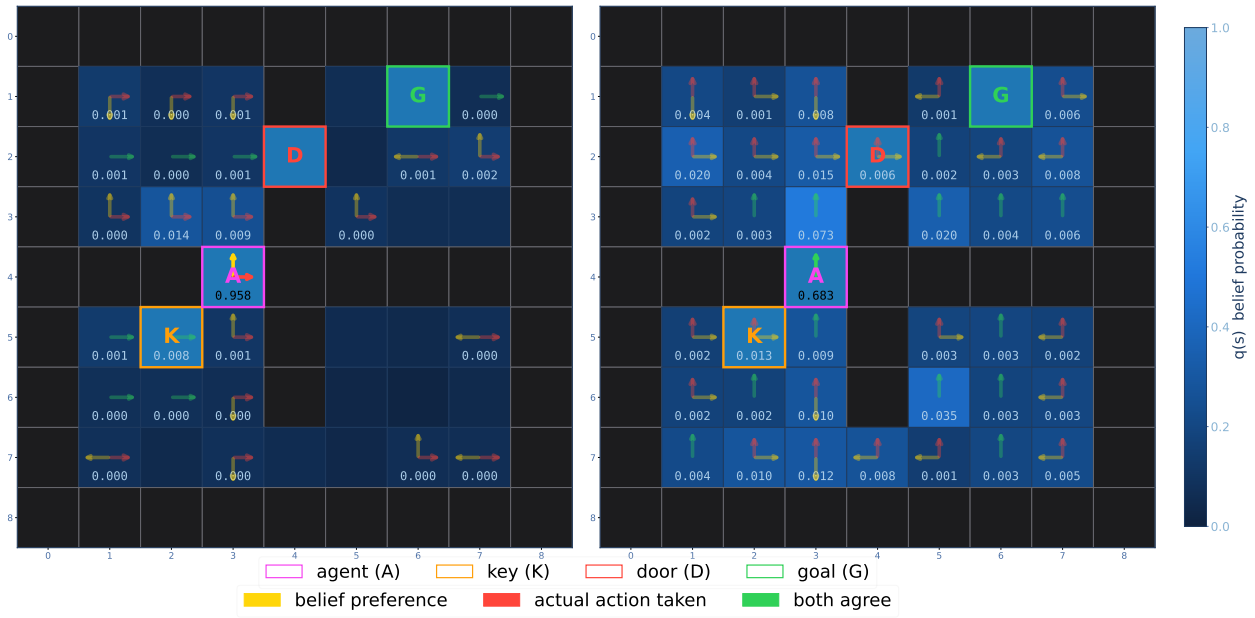

Figure 2: Visualisation of the belief distribution q_t, entropy H_t, and belief-action gap GAP_t for a single step before (PRE, left) and after (POST, right) chain-of-thought reasoning. Each cell shows the belief probability $q_t^{\text{loc}}(c, r)$ and the preferred action under g_θ (yellow arrow), the action actually taken by the agent (red arrow), or both when they agree (green arrow). PRE: the belief is sharply concentrated on the true position (0.958), the preferred and taken actions agree, and the gap is large (GAP = 2.386), reflecting that without reasoning the agent acts suboptimally under its own value function. POST: the belief spreads across the grid ($H : 0.261 \rightarrow 1.572$), yet the agent's action aligns better with the value-preferred action, reducing the gap substantially (GAP = 0.205).

This single example is representative of the aggregate pattern observed across all 3,194 steps: reasoning diffuses the state representation while improving action-value alignment.

Together, these results support the hypothesis that chain-of-thought reasoning simultaneously improves action-value alignment, as measured by the decreasing GAP, while making the agent's internal state representation more diffuse, as measured by the increasing entropy.

5.4 Behavioral Belief Probes

We ran a black-box behavioural-probe pipeline in the same DoorKey domain. These probes test what the model states explicitly about the current state and allow direct comparison with emitted actions and representational probes.

Each behavioural probe is a *single-step, history-free* query built from the current rendered grid and, when relevant, the agent's carrying-key status. The probe families are directional wall probes, object-direction probes for key, door, and goal, and action-conditioned probes for immediate move consequences. Reported metrics are accuracy, valid parse rate, the *local contradiction rate* for moving into a direction explicitly judged blocked, and the A^**-conditioned gap rate* for taking a non-optimal action despite a correct wall judgment.

On the balanced trajectory-derived failure-mode subset, directional wall probes remained highly accurate (mean greedy accuracy 0.922, Monte Carlo accuracy 0.969). Object-direction probes were also strong overall (mean greedy accuracy 0.880, Monte Carlo accuracy 0.979), but less uniform

Readout	Settings	Returned output	Purpose
Greedy	$T = 0$, one call	Single parsed answer	Primary explicit behavioural answer.
Repeated greedy	$T = 0$, 10 calls	Most frequent parsed answer across repeated greedy calls	Measures provider-level nondeterminism under nominally greedy decoding.
Logprob diagnostic	Exposed answer-token logprobs at $T \in \{0, 0.7, 1.0\}$	Yes-vs-no label derived from exposed answer-token probabilities	Diagnostic only; checks whether exposed token probabilities agree with the surfaced answer.
Monte Carlo	$T = 0.7$, 10 calls	Distribution over MC-sampled answers	Tests robustness of the explicit answer under stochastic decoding.

Table 8: Behavioural-probe readouts used for label-space queries. All answers are parsed into {`yes`, `no`, `unknown`, `invalid`} after normalisation and answer extraction.

Evaluation set	Size	Role	Composition
Manual smoke test	10 states	Prompt and parser validation	Hand-built 9×9 DoorKey states with simulator-verified labels.
Balanced failure-mode subset	16 states	Main trajectory-derived case-study set	12 tagged failure states plus 4 immediately preceding comparison states.
Expanded wall-focused slice	33 states	Larger-denominator wall-gap analysis	Tagged failure states only; used for wall-probe contradiction rates.
Public matched clean slice	15 states	Exact-state white-box/black-box validation	Size-7 matched public states with released activations, trajectories, and cognitive probes.
Failure-focused revealed-CoT slice	13 states	Revealed-reasoning behavioural analysis	4 avoidable detours, 3 wall hits, 2 backtracks, and 4 optimal baseline steps from local CoT trajectories.

Table 9: Behavioural-probe evaluation sets used in the current draft.

across probe types: goal-direction probes were perfect on this subset, door-direction probes were near-perfect, and key-direction probes were harder, with greedy accuracies between 0.50 and 0.81 depending on direction.

Figure 3 shows that wall, goal, and door probes are generally accurate on failure-mode states, whereas key-direction judgements are less reliable. On the expanded wall-focused slice of 33 tagged failure states, directional wall accuracy remained high (greedy accuracy between 0.91 and 0.97 by direction), yet the action-facing contradiction metrics were non-zero.

Figure 4 shows that wall beliefs are often accurate even when the observed action is still locally inconsistent or non-optimal.

The clearest behavioural belief-action gap appears on wall-hit states. On the expanded wall-focused slice, the model correctly reported that the chosen direction was blocked and nevertheless took that move in 3/3 eligible wall-hit cases. More broadly, the A*-conditioned wall-gap rate

Balanced failure-mode subset category	Count
State just before tagged failure	4
Backtrack	3
Short loop	2
2-cycle oscillation	2
Wall hit	2
Freeze repeat	2
Avoidable detour	1

Table 10: Composition of the balanced trajectory-derived case-study set. The first row is comparison context rather than a failure label. All other rows are deterministic step-level failure categories.

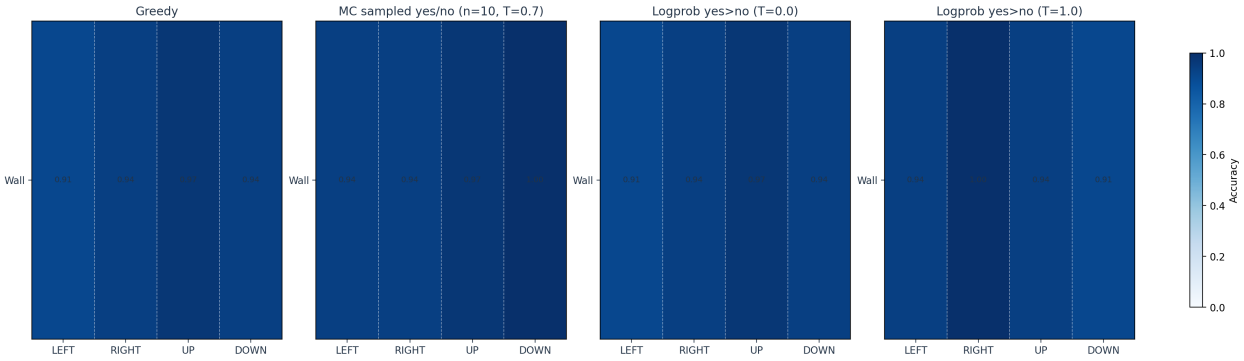

Figure 3: Behavioural-probe accuracy on the balanced failure-mode subset. Rows are probe families and columns are directions. Wall, goal, and door probes are generally accurate, whereas key-direction probes are less reliable. The Monte Carlo and logprob panels are included as robustness checks rather than as separate formal definitions of the belief-action gap.

remained substantial for `RIGHT` (4/6 eligible states) and `UP` (5/10 eligible states), showing that many suboptimal actions are not explained by a failure to state the relevant local wall fact.

Figure 5 further shows that the behavioural mismatch is concentrated in specific failure types rather than spread evenly across all suboptimal states. These behavioural probes identify states where the model explicitly states an action-relevant fact correctly and still acts against it. This validates the textual environment serialisation as a source of local belief readouts and motivates the white-box analyses that follow.

5.4.1 Behavioural–Representational Agreement

Behavioural and representational wall-belief readouts were compared on a public matched clean slice of 15 size-7 state snapshots from four released GPT-OSS-20B trajectories, together with released pre- and post-reasoning activations from `project-telos/activations_test_full`, released trajectories from `project-telos/trajectories_test_full`, and released cognitive probes from `project-telos/cognitive_map_probes`. Behavioural wall probes are evaluated on the same (trajectory, step) state snapshots.

Table 11 reports white-box accuracy, black-box accuracy, and white-box/black-box agreement on matched public state snapshots. Black-box wall readouts are perfect at both 0% and 100% revealed CoT, while white-box wall decoding is strong but direction-dependent. This public matched clean slice is a validation slice rather than a gap-diagnostic slice.

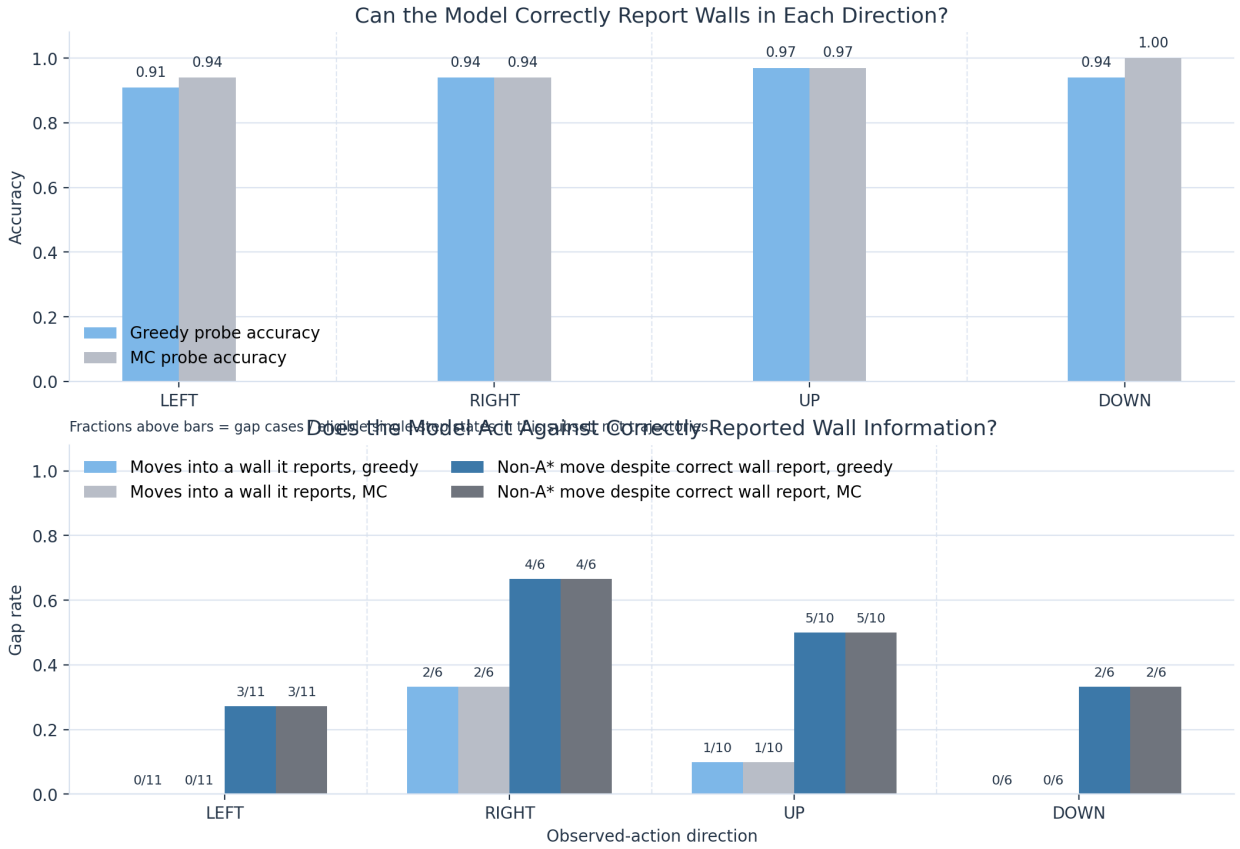

Figure 4: Wall-probe accuracy and behavioural belief-action mismatch on the expanded wall-focused slice. The top panel shows wall-probe accuracy by direction. The bottom panel shows two narrower mismatch rates: how often the model moves into a direction it explicitly reports as blocked, and how often it takes a non-optimal action despite a correct wall judgement. Fractions above bars are gap cases over eligible single-step states

Table 11: Behavioural–representational agreement on matched public size-7 wall-belief states. White-box results use released pre- and post-reasoning activations; black-box results use 0% and 100% revealed CoT conditions on the same state snapshots.

@ >p0.14 >p0.055 Y Y Y Y Y Y @

Wall question	N **states**	**White-box accuracy pre-reasoning**	**Black-box accuracy** 0% **revealed CoT**	**Agreement pre,** 0%	**White-box accuracy post-reasoning**	**Black-box accuracy** 100% **revealed CoT**	**Agreement post,** 100%
`wall_down`	15	0.800	1.000	0.800	0.933	1.000	0.933
`wall_left`	15	0.733	1.000	0.733	0.733	1.000	0.733
`wall_right`	15	0.867	1.000	0.867	0.933	1.000	0.933
`wall_up`	15	1.000	1.000	1.000	0.933	1.000	0.933

5.4.2 Behavioural Effects of Revealed Reasoning

Revealed reasoning was evaluated on a behavioural-only failure slice built from. The slice contains 13 step-level examples from local GPT-OSS-20B CoT trajectories: four `avoidable_detour` steps, three `wall_hit` steps, two `backtrack` steps, and four `baseline_optimal` steps.

Table 12 summarizes how wall-belief accuracy, uncertainty, and gap rates change as more prior reasoning is revealed on the failure-focused slice. Wall-belief accuracy remains high and wall entropy remains near zero across reveal levels, while MC-sampled action entropy falls and both local

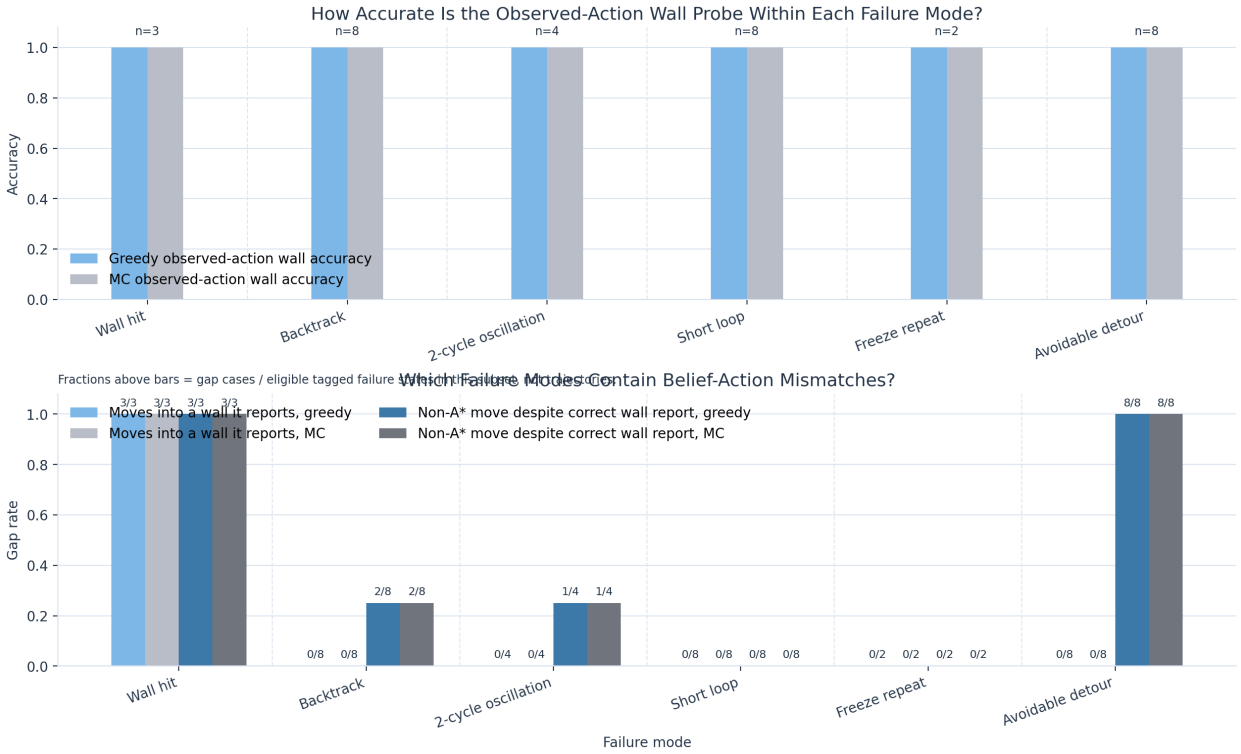

Figure 5: Failure-mode breakdown of wall-probe accuracy and behavioural belief-action mismatch on the expanded wall-focused slice. The strongest local contradiction appears in wall-hit states, while avoidable detours show a strong A*-conditioned mismatch despite correct wall judgements. This indicates that the behavioural gap is concentrated in specific failure modes rather than being uniform across all suboptimal states.

Table 12: Failure-focused gradual-CoT results by reveal level.

Reveal level (%)	Wall-belief accuracy	Wall entropy	MC-sampled action entropy	Local gap rate	A*-conditioned gap rate
0	0.942	0.000	0.283	0.000	0.000
25	0.942	0.000	0.141	0.077	0.083
50	0.962	0.000	0.141	0.154	0.231
75	0.942	0.000	0.141	0.077	0.385
100	0.942	0.000	0.000	0.231	0.538

and A*-conditioned gap rates rise. This does not support the hypothesis that progressively revealed reasoning reduces the action-belief gap while increasing state uncertainty. The analysis is behavioural-only: the public activation release does not provide matched intermediate white-box activations for the revealed-CoT conditions.

5.5 Four-Room Episode Sweep: Optimal Geometry vs Agent Action Bias

The four-room experiment compares two label choices while holding activations, architecture, and training procedure fixed. Model A is trained on the agent's emitted actions; Model B is trained on full-state BFS-optimal actions.

Metric	Agent-action label	BFS-optimal label
Test accuracy	0.454	**0.960**
Test set-membership accuracy	0.369	**0.992**
Test log-likelihood	−0.9969	−0.1389
Train accuracy	0.451	**0.972**
Majority-class baseline	0.548 (UP)	0.733 (DOWN)
Random baseline	0.250	0.250

Table 13: Four-room label-control result. Identical activations and algorithm; only the label changes.

The same activation features predict full-state BFS-optimal actions at 96.0% accuracy and 99.2% optimal-set membership, but predict the agent's own actions only 45.4%, below the majority baseline. This shows that the activation representation is much more predictive of optimal geometry than of the agent's behavioural UP-bias.

Pre/post clarification. In the four-room experiment, pre- and post-reasoning state features are tied for optimal-label prediction because $\phi(s)$ is visit-averaged over all visits to the same state. Averaging removes trajectory-specific output-commitment information and preserves stable state geometry. This does not contradict per-visit pre/post dissociation; the two experiments answer different questions.

6 Limitations

The main limitation is causal: probe success indicates decodability, not causal use. A probe can reveal that information is present in an activation space without proving that the model uses that information to generate its action. A second limitation is environment simplicity. DoorKey is useful for exact analysis because the true state, transition function, and optimal-action set are all computable. However, the environment is small enough that the agent may be near-ceiling on many states even without internalising a coherent value function, and the results may not generalise to richer environments with larger state spaces or longer horizons. All experiments use a single model (GPT-OSS-20B); it is unknown whether the belief-degradation pattern across reasoning holds for other models or training regimes. The current behavioural results establish two separate findings: exact-state agreement between representational and behavioural wall probes on a limited public clean slice, and a behavioural-only failure-focused revealed-CoT analysis showing that increased reasoning reveal does not reduce the measured gap. The next stage is to join failure-focused behavioural probes, white-box probes, emitted actions, and IRL/value predictions on the same states. Distributional optimal-action metrics are needed for tied optimal actions, and causal patching is needed to test whether decoded representations influence later reasoning and final action generation. If these analyses support the current pattern, the project will provide a concrete methodology for moving from behavioural evaluation to mechanistic diagnosis of LLM-agent failures.

7 Conclusion

This project asks whether language-model agents fail because they lack task-relevant information, or because they fail to act on information that is already internally available. The evidence consistently favours the second explanation. Pre-reasoning activations encode task-optimal actions far more reliably than the agent's emitted behaviour reflects, and this gap is not explained by a failure of internal representation. Both white-box probes and explicit behavioural queries confirm that relevant state information is present and accessible before the agent commits to an action.

Chain-of-thought reasoning is the mechanism through which the gap closes. After reasoning, the belief-action gap decreases significantly while positional uncertainty increases, suggesting that reasoning trades precise state encoding for action commitment. This commitment is often correct, but frequently is not — and when it is not, the agent acts against information it demonstrably holds.

The practical implication is that behavioural evaluation alone is insufficient for diagnosing agent failures. An agent that consistently encodes the right answer but does not reliably act on it presents a qualitatively different safety problem from one that simply lacks the relevant knowledge. The framework and measurement pipeline developed here — combining representational probes, behavioural probes, IRL-based cost recovery, and causal intervention — provides a foundation for distinguishing these cases. Extending this to richer environments, additional models, and causal intervention at intermediate reasoning positions are the natural next steps.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[2] Raghu Arghal, Fade Chen, Niall Dalton, Evgenii Kortukov, Calum McNamara, Angelos Nalmpantis, Moksh Nirvaan, Gabriele Sarti, and Mario Giulianelli. A behavioural and representational evaluation of goal-directedness in language model agents. *arXiv preprint arXiv:2602.08964*, 2026.

[3] Tom Everitt, Cristina Garbacea, Alexis Bellot, Jonathan Richens, Henry Papadatos, Siméon Campos, and Rohin Shah. Evaluating the goal-directedness of large language models, 2025.

[4] Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models, 2024.

[5] Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024.

[6] 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.

[7] 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.

[8] Matt MacDermott, James Fox, Francesco Belardinelli, and Tom Everitt. Measuring Goal-Directedness. *Advances in Neural Information Processing Systems*, 37:11412–11431, December 2024.

[9] Nicolas Martorell. From Text to Space: Mapping Abstract Spatial Models in LLMs During a Grid-World Navigation Task. In Riccardo Guidotti, Ute Schmid, and Luca Longo, editors, *Explainable Artificial Intelligence*, pages 268–291, Cham, 2026. Springer Nature Switzerland.

[10] 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. arXiv:2502.08640 [cs].

[11] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022.

[12] 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, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore, December 2023. Association for Computational Linguistics.

[13] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[14] 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.

[15] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[16] Teun van der Weij, Felix Hofstätter, Ollie Jaffe, Samuel F. Brown, and Francis Rhys Ward. AI sandbagging: Language models can strategically underperform on evaluations, 2025.

[17] 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, pages 1433–1438, Chicago, Illinois, July 2008. AAAI Press.

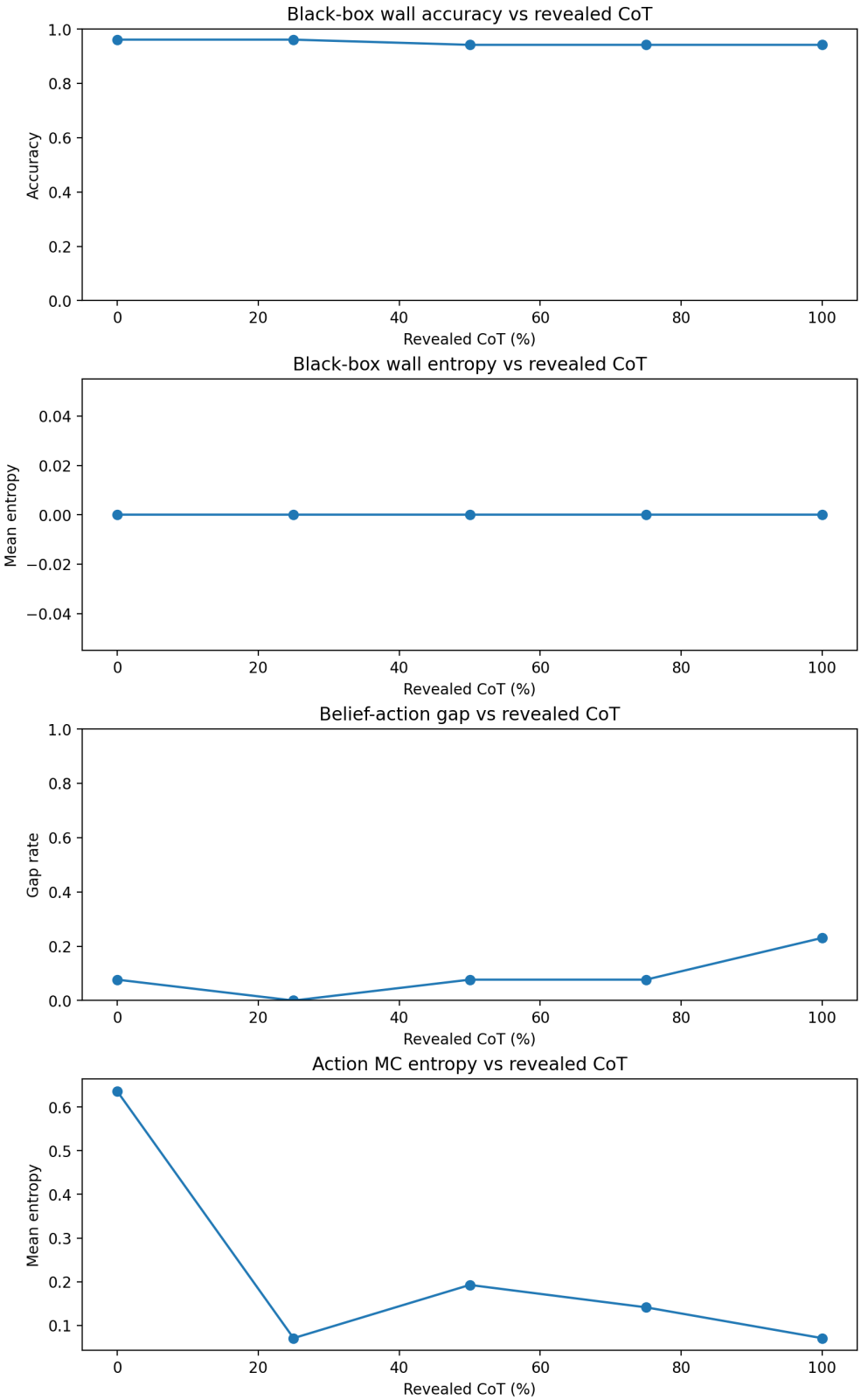

Figure 6: Behavioural effects of gradually revealing prior chain-of-thought on a failure-focused slice. Wall-belief accuracy remains high and wall-label entropy remains near zero across reveal levels, while MC-sampled action entropy decreases and belief-action gap rates increase with reveal level.

A Behavioural Probe Appendices

A.1 Behavioural Probe Prompts

The behavioural probes use a shared instruction scaffold: the same environment description, legend, directional conventions, and answer-format constraints. The probed object or relation changes, not the scaffold.

Canonical wall prompt (semantic label path).

```
# Instructions

You are controlling an agent in a 9x9 fully observable DoorKey GridWorld.
The environment contains walls, open spaces, a goal, a key, and doors.

Legend:
---------------
#: Wall
_: Open Space
G: Goal
A: Current agent position
D: Door (closed/locked)
O: Door (open)
K: Key
---------------

Interpret RIGHT as the cell one column to the right (east), LEFT as the
cell one column to the left (west), UP as one row up (north), and DOWN as
one row down (south) in the rendered grid.

# Inputs

Current grid state:

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

Agent status:
- Carrying key: false

Given the current grid only, is the cell immediately RIGHT of the agent a
wall (#)?
```

```
Return exactly one label and nothing else:
A = yes
B = no
C = unknown
```

Wall prompt variant (JSON sampled-answer path).

```
Return exactly one JSON object of the form
{"label": "<yes|no|unknown>"} and nothing else.
```

Canonical action prompt.

```
# Instructions

You are controlling an agent in a 9x9 fully observable DoorKey GridWorld.
The environment contains walls, open spaces, a goal, a key, and doors.

Legend:
---------------
#: Wall
_: Open Space
G: Goal
A: Current agent position
D: Door (closed/locked)
O: Door (open)
K: Key
---------------

Interpret RIGHT as the cell one column to the right (east), LEFT as the
cell one column to the left (west), UP as one row up (north), and DOWN as
one row down (south) in the rendered grid.

Choose the next move that best advances toward the goal while respecting
the DoorKey mechanics in the current state.

The key is automatically picked up when the agent moves onto K. A closed
door D can only be opened if the agent already has the key.

# Inputs

Current grid state:

  0 1 2 3 4 5 6
0 # # # # # # #
1 # _ _ _ _ _ #
2 # A _ _ _ _ #
3 # _ _ _ _ _ #
4 # _ _ _ G _ #
```

```
5 # _ _ _ _ _ #
6 # # # # # # #

Respond with exactly a JSON object of the form
{"action": "<UP|DOWN|LEFT|RIGHT>"}.
Do not include any extra text before or after the JSON.
```

Revealed-CoT action prompt variant.

```
Reasoning trace available so far:
We need shortest path from A at (2,1) to G at (4,4). Grid open spaces.
Likely move down two then right three. But check walls: row4 col4 is G.
Path: from (2,1) down to
```

A.2 Prompt Variants

The same base prompt scaffold is used across wall, key, door, and location probes where applicable. Variants are limited to the target object or relation, the answer format, and the optional revealed-CoT block. Repeated greedy and Monte Carlo readouts differ in sampling regime rather than semantic task definition.

A.3 Selected Failure Cases

Figure 7 shows concrete wall-hit and sub-optimality cases in which the model reports the local wall fact correctly yet still takes a wall-hitting or non-optimal action.

A.4 Additional Behavioural Figures

Wall-Hit Sub-Optimality Cases

Wall-hit steps: 8 | non-optimal: 8/8 | wall-in-direction probe: 7/8 | action-hits-wall probe: 7/8

Case	Source	Observed	Optimal	Non-opt	Wall in observed dir.?	Observed action hits wall?	Full grid
rooms2_doorkey_15_step_000	wrong_prompt_100	UP	LEFT	yes	yes	yes	
rooms2_doorkey_22_step_000	wrong_prompt_100	DOWN	LEFT	yes	yes	yes	
rooms2_doorkey_52_step_004	wrong_prompt_100	DOWN	RIGHT	yes	no	no	
rooms2_doorkey_73_step_000	wrong_prompt_100	LEFT	UP	yes	yes	yes	
rooms2_doorkey_88_step_000	wrong_prompt_100	LEFT	UP	yes	yes	yes	
rooms2_doorkey_keepdoor_16_step_000	keepdoor_100	RIGHT	DOWN	yes	yes	yes	
rooms2_doorkey_keepdoor_20_step_000	keepdoor_100	UP	LEFT	yes	yes	yes	
rooms2_doorkey_keepdoor_57_step_002	keepdoor_100	RIGHT	UP	yes	yes	yes	

Figure 7: Selected wall-hit and sub-optimality cases from the behavioural failure analysis.

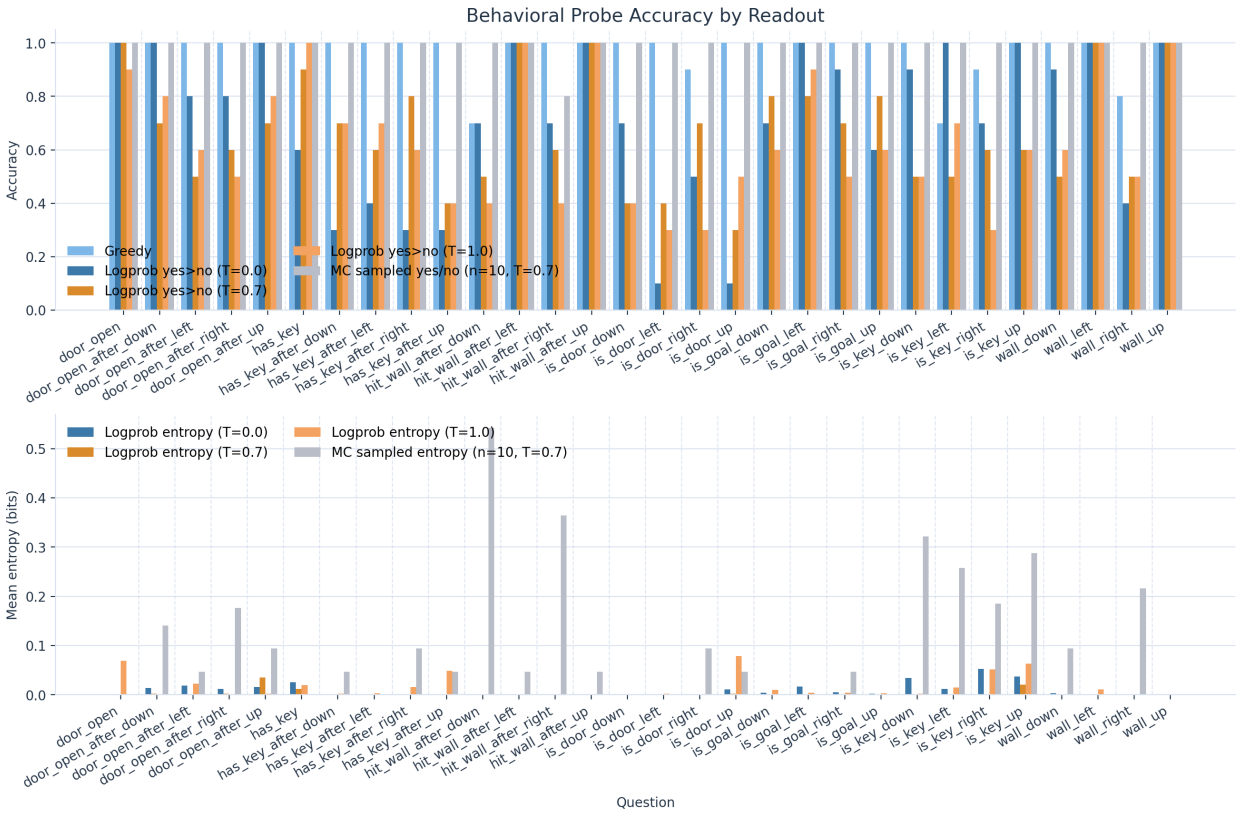

Figure 8: Broad behavioural probe summary on the smoke-test-all suite.

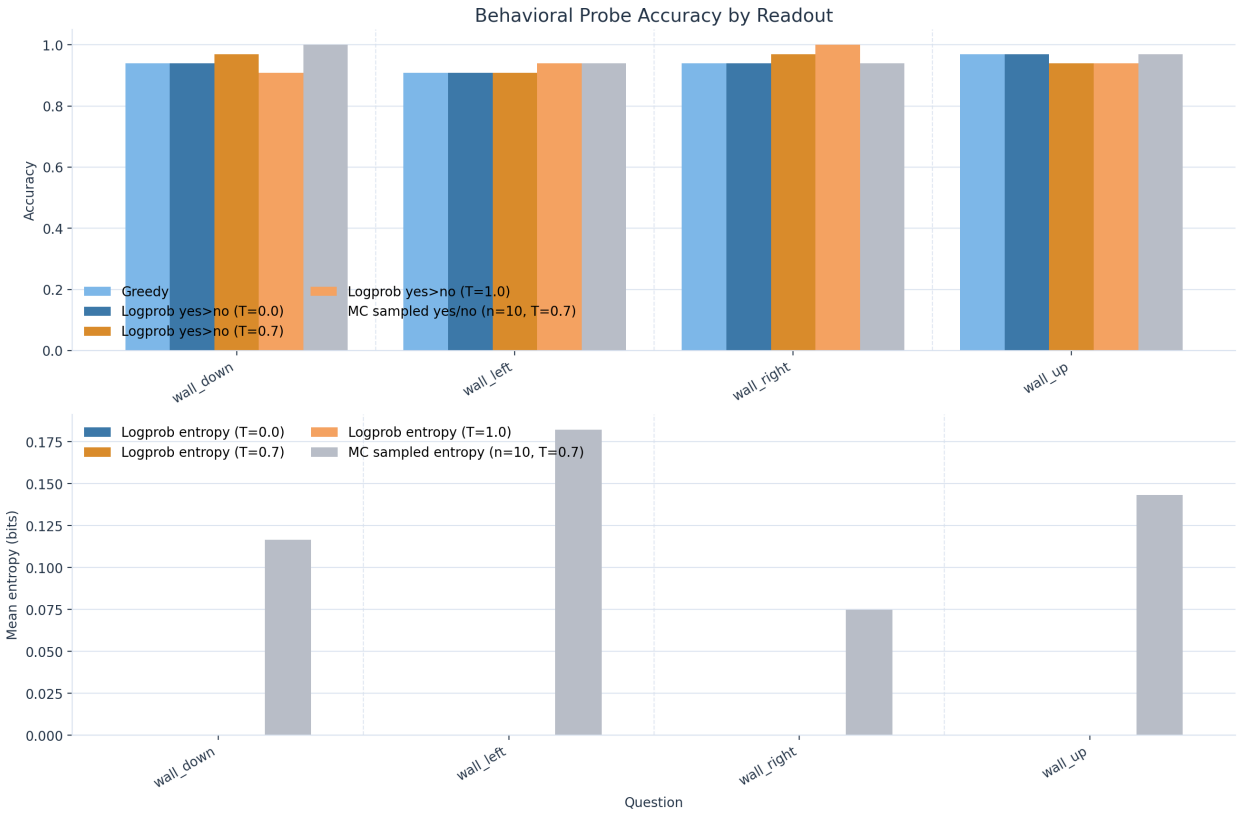

Figure 9: Wall-focused behavioural probe summary on the failure slice.