Evaluating Self-Orienting in Language and Reasoning Models

Eric Bigelow [*1,2] **Zergham Ahmed** [*3] **Tomer Ullman** [1,4]

Abstract

We present a novel evaluation approach based on research in cognitive science, which studies the ability of an agent to *self-orient* (i.e., identify what problem it is solving and which agent it is in the environment). Our task involves a grid-world where the agent needs to navigate to a goal, but does not have prior knowledge of the world, including which entity it controls. Humans solve this task in a two-step process, first figuring out what agent they control, in other words self-orienting, and then navigating to the goal. We ask whether LLMs can accomplish this task. We found that state-of-the-art LLMs (GPT-4o) have the ability to efficiently self-orient with near-optimal performance, but this ability disappears with in-context reasoning (OpenAI o4-mini). However, we find that this ability re-emerges for reasoning models trained with more advanced methods, such as backtracking (o3).

1. Introduction

Large Language Models (LLMs) have demonstrated impressive emergent capabilities, such as planning in complex problems, which can be elicited with few- or even zero-shot learning in the form of prompts. Reasoning LLMs build upon these abilities by generating a sequence of intermediate reasoning steps (i.e., a "chain of thought") before outputting a final response. Considerable work has been devoted to studying the abilities of these models and testing their limits, for example with tasks in planning (Valmeekam et al., 2023) and spatial reasoning (Yamada et al., 2023). However, one question which has not been addressed by prior work is: to what degree can LLMs self-orient? That is, can an LLM adjust its latent world model representation with a semantic marker for the "self", or the specific agent that the LLM controls. This is an important phenomena since it has been suggested that it is one of the core reasons why humans are able to flexibly learn and act (Paul et al., 2023). Consider self-driving cars which may have some sort of self-representation in their environment. However, if this self-representation was made more effective it may allow self-driving cars to be more robust and adapt to situations outside of their training.

Debates about what makes up a "self" are varied and nuanced. They span hundreds of years of study in philosophy, as well as decades of study in fields such as cognitive science and neuroscience. Here, we focus on a recent framework for a minimal version of self representation, which can be rigorously studied:the representation of a particular entity in the world, and tagging it as the agent that is doing the representing, the source of input and output(Paul et al., 2023; De Freitas et al., 2023). Accordingly, the process of identifying where, when, and who one is in the world is referred to as *self-orienting*. In daily life, most people are typically self-orientated. But this process can become momentarily unglued. For example, consider the first few seconds of a player picking up a novel video game. Before they can achieve their goals in the game, they need to figure out *who* they are in the game, which avatar represents them.

Whether machines have the ability to represent themselves long been a topic of interest for computer scientists, psychologists, and philosophers (Hofstadter, 1979; Hofstadter & Dennett, 1981; Minsky, 1986; Damasio, 2012). More recently, the question of self-representation in LLMs has become a compelling philosophical question, with implications for AI evaluation and safety. Recent works have considered, for example, the degree to which LLMs have "situational awareness" and associate details of their implementation such as their architecture and training regimes with their "self" (Laine et al., 2024; Betley et al., 2025). Other work has considered whether LLMs have stable preferences (Lehr et al., 2025), and whether LLMs can "introspect" to accurately predict aspects of their internal hidden states (Binder et al., 2024). However, these questions re-

[*]Equal contribution [1]Department of Psychology, Harvard University [2]CBS-NTT Program in Physics of Intelligence, Harvard University [3]Department of Computer Science, Harvard University [4]Center for Brain Science, Harvard University. Correspondence to: Eric Bigelow <ebigelow@g.harvard.edu>, Zergham Ahmed <zerghamahmed@g.harvard.edu>.

ICML 2025 Workshop on Assessing World Models.

late to the self in a more oblique way and face the same challenges of interpretation as studies of the self in humans. Here, we focus on the more minimal and addressable question of whether LLMs can *self-orient* in a human-like way. This in turn can inform broader capabilities of self-representation, such as those above.

In this work, we present the first empirical test of self-orienting in LLMs. We combine a novel zero-shot learning (i.e., instruction prompt) experiment design, with a task inspired by previous work in cognitive science and reinforcement learning (De Freitas et al., 2023). Following theories proposed in this prior work, we then describe how this task is optimally solved in two phases by a Bayesian learner. In our experiments, we test a combination of reasoning and non-reasoning LLMs. Surprisingly, we find a U-shaped curve in performance, where vanilla LLMs can solve the task, but reasoning greatly harms performance – unless the reasoning model is trained with more advanced techniques such as backtracking and self-verification. We consider this result in light of the framework of self-orienting.

2. Self-Orienting in a Grid World

Our experimental domain is adopted from De Freitas et al. (2023), developed as a minimal test of self-orienting. In these experiments, a (human or AI) participant is instructed that they will play a game, but are given only minimal instructions as to what the point of the game is. For humans, the instructions were simply 'play the game'. We mildly adapt these instructions for LLMs to be controlling an Agent to reach a Goal. The game board is a 9×9 grid world with 4 different types of squares: *Wall* (black or `0`), *Path* (white or `1`), *Agent* (red or `2`), and *Goal* (green or `3`). In the initial

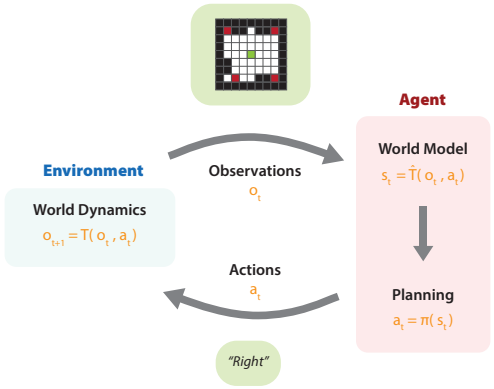

Figure 1. Language model agent framework. We treat LLMs as reinforcement learning agents in a simple grid world domain. The LLM observes grids encoded as lists of digits (visualized here as colors) and produces actions (such as *"Right"*). To solve the task efficiently, an agent must infer a World Model which is then used for Planning.

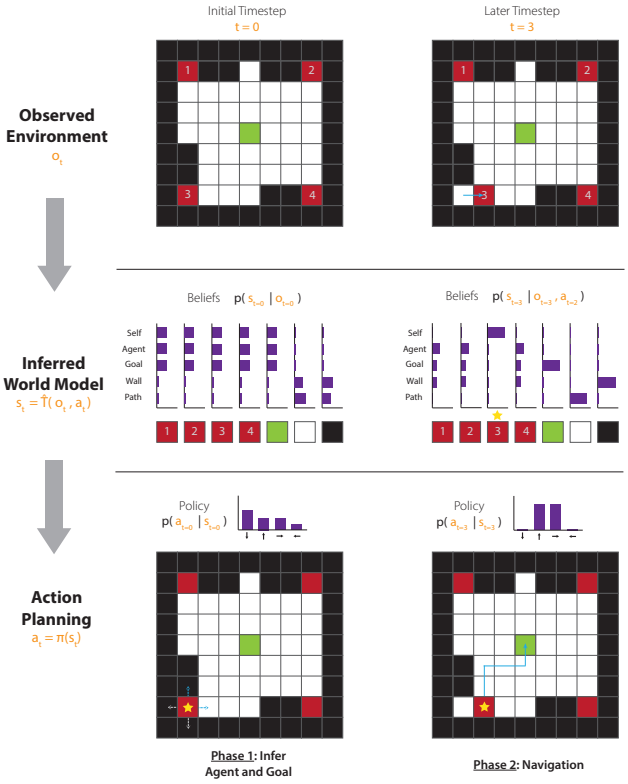

Figure 2. Theory of how to optimally solve our grid world domain. For two times, $t = 0$ (Left column) and $t = 3$ (Right column), we show the observed environment o_t (Top row) along with the agent's inferences. First, the player explores actions until they are able to correctly infer a world model $s_t = \hat{T}(o_t, a_t)$ (Middle row) and then, once the player has identified the "Self" representation, they navigate to the goal by planning $a_t = \pi(s_t)$ (Bottom row).

presentation, it is unclear to the user precisely what each square represents. Additionally, there are 4 markers of the same color as the true *Agent*, and 1 marker with the color of the *Goal*. The game player can take four actions a_t, which control the movements of the true Agent: *Left*, *Right*, *Up*, or *Down*.

In order to solve the game, a player may first infer a world model for the domain, and then use this world model to plan an efficient path to the goal. The arrangement of the *Path* and *Wall* squares is typical for analogous sprites in common video games and grid world environments, and a user with experience seeing other video game environments will likely to quickly infer the meaning of these. However, it is unclear which square is an *Agent* and which is a *Goal*. Further, there is ambiguity as to which of the four possible *Agent* squares corresponds to the "Self" agent that the user actually controls.

Building on De Freitas et al. (2023), we stipulate that efficiently reaching the goal in this domain requires two phases of behavior. First, the user must infer the world model by

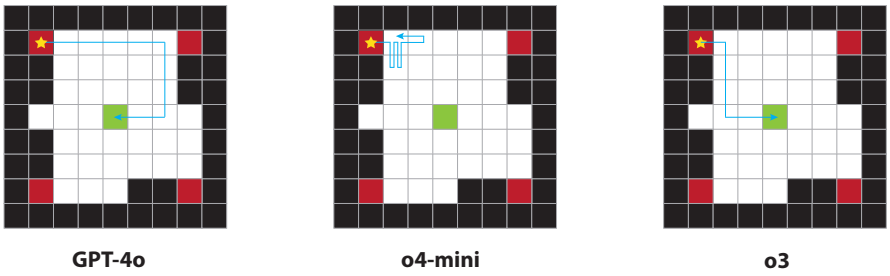

Figure 3. Example path each model takes when attempting to solve the task. Visualizing a single rollout for each model, we see that GPT-4o (Left) and o3 (Right) efficiently navigate to the goal, whereas o4-mini (Center) acts erratically, moving back and forth without reaching the goal (note: this rollout is shortened for readability). Impossible moves, which lead the agent into a Wall, are not shown.

exploring different actions until they take the one action which causes their self-agent to move. Given this action, they must infer the appropriate world model of the game environment. In the second phase, the user must then navigate their agent to the goal. Our environments are designed such that, once the self-agent is identified, they can reach the goal by taking steps in two directions – e.g., if the self-agent is in the bottom-left corner, then the user must take `Up` and `Right` steps until they reach the goal.

In Theory-Based Reinforcement Learning (RL) (Tsividis et al., 2021), planning with a policy $\pi(s_t)$ requires a world model $s_t = \hat{T}(o_t, a_t)$ that can be reasoned over. This world model serves as a theory for both the world and the role that the agent plays in it. In the case of our task, the agent must have a theory of what role is played by different squares on the grid world and in particular have a theory for what square it controls. A working approach to formalizing self-orienting is to frame the problem as a Partially-Observable Markov Decision Process (POMDP) with additional modifications (Paul et al., 2023).

2.1. Problem Formulation

We formulate our problem similar to Ahmed et al. (2025). We start with the classical planning problem, where environment state transitions are defined by a transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ with state space $\mathcal{S}$ and action space $\mathcal{A}$. The goal of the game player (who controls the "Self" agent) is to find a plan $\pi =< a_1, \ldots, a_N >$ that allows it to navigate to the goal square. To do so, the player must find an estimation of the transition function. We denote this estimated transition function and the state it returns as $s_t = \hat{T}(o_t, a_t)$ and refer to it as the inferred world model. Here, o_t refers to the raw observed state of the environment and s_t refers to the player's representation of the state which it will use for planning. Therefore, the player must first find $s_t = \hat{T}(o_t, a_t)$, which we refer to as the self-orienting phase. Then, it must navigate to the goal by finding $\pi(s_t)$, which we refer to as the navigation phase.

3. Experiments

We performed experiments on the "Logic Game", the first of three experiments proposed in De Freitas et al. (2023), which tests self-orienting ability and navigation in static grids. They found that humans were able to solve this task almost immediately, whereas deep RL algorithms took a vastly larger number of trials to learn. Our experiments aimed to answer the following question: How do LLMs and reasoning models perform on a simple self-orienting task?

We evaluated the most recent state-of-the-art LLM by OpenAI, GPT-4o (OpenAI, 2024) as well as their most most recent reasoning models o3 and o4-mini (OpenAI). We choose these models as they are known for their general reasoning and problem solving capabilities.

While De Freitas et al. (2023) proposed additional experimental domains, we focused on the logic games subset of the static grid experiments due to API cost budget constraints.

The 4 grids we chose have different initial actions that lead to movement. For example in Fig. 3, if the agent's location is in the top-left, then only taking the action "right" will lead to any movement. Similarly, if we consider all the other possible agents and the initial actions that allow them to move, we will see that only the actions "right", "left", and "up" will yield any movement. We can contrast that with the grid in Fig 2., where only the actions "up" and "down" will yield any movement. We choose 4 diverse grids based on this idea of the initial actions that lead to any movement.

In our experiments, we use the prompts shown in Appendix A to ask the LLM its initial action. Then, we enter this action in the environment which returns the updated observation. We feed this observation back into the LLM and repeat this process, essentially having a closed loop system where the LLM takes actions and sees the updated observation of the environment.

We evaluated the four models—GPT-4o, o4-mini under two reasoning settings (Low and High effort), and o3. Each

model was tested on four distinct grid layouts, and for each layout we ran the agent from each of four possible agent positions, yielding 16 grid-agent combinations. For each combination we performed three independent planning rollouts. We scored a combination as a "success" if at least one of its rollouts reached the goal within the move limit. Finally, accuracy was computed as the fraction of those 16 combinations for which the model succeeded. Fig. 3 reports these success rates across all four models and Fig. 4 shows a heatmap of the aggregated path for each of the rollouts which essentially shows how many time that square was visited. Fig. 5 measures how close the models were to the goal any given point in the interactive prompting loop. This was aggregated over just three of the grids due to the budget constraints of o3. Since we ran the experiment on o3 in the UI, we ran into a weekly usage limit and were unable to get the result to include the last grid. Therefore for Fig. 5, we only tracked the distance on 3 grids with one agent location in each grid since we had to manually extract and record the action sequences. We had 3 rollouts for every model except for o3 we had 3 rollouts for 2 out of the 3 grids, and for the last grid we were only able to perform 1 rollout.

Results In our experiments, we find that GPT-4o is able to solve the task within 20 steps nearly 100% of the time (Figure 4). However, the simple reasoning model we tested (OpenAI o4-mini) consistently fails to solve the task, and this failure rate worsens with higher reasoning effort. Inspecting the available information about reasoning chains, we observe that many reasoning chains infer the wrong world model and are unable to recover, e.g., assuming that the grid square for the goal (green / 3) is actually the agent. While we did not analyze OpenAI o1 for cost reasons, we observed the same pattern of failure with it as well. Finally, with OpenAI's o3 model, this trend reverses and we find a near-100% success rate, marking the final step in this "U-shaped" performance trend. When inspecting the reasoning chains for o3, we find that the model is able to revise its initial world model inferences.

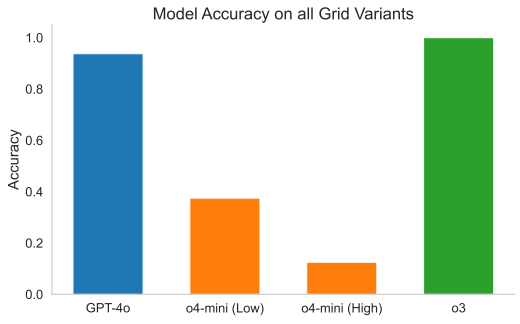

Figure 4. Across all grids and agent locations we tested, GPT-4o scores very high, solving $> 90\%$ of cases within 20 steps. On the other hand o4-mini reasoning models score much lower, and higher reasoning budget (High) negatively impacts performance. o3 scores the highest, solving all grid permutations we tested.

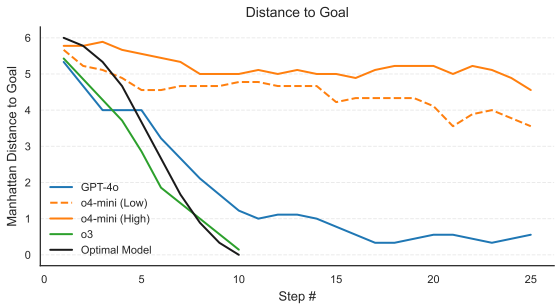

Figure 5. The distance for each is aggregated over three different configurations and averaged over rollouts.

This pattern is further confirmed by considering the specific paths that agents take in moving through the grid (Figure 3). We find that GPT-4o and OpenAI o3 efficiently navigate to the goal, although not always following the most optimal path. These models both distinct biases in their moves, in particular the earlier exploratory of identifying the agent. Depending on where the agent is located on the grid, either model may be more or less efficient in identifying the agent or in navigating to the goal. For the o4-mini model, however, we observe a very different pattern of behavior: the model moves erratically and without clear purpose, in some cases simply going in circles (e.g., o4-mini (High) in Figure 3). Aggregating movement patterns across all grids and agent locations, we see that GPT-4o and o3 perform nearly as well as our Optimal Model (Figure 5), efficiently reducing the distance between the agent and the goal. Conversely, both o4-mini models only slightly reduce the distance to the goal even after many steps.

4. Discussion

We have presented the first study, to our knowledge, of self-orienting in LLMs. Adopting an experimental domain from cognitive science, we have shown that state-of-the-art LLMs are able to solve this task with near-optimal efficiency, suggesting these models are capable of both inferring the correct world model and then navigating to the goal. We also find a striking pattern where reasoning (o4-mini) harms performance, which we attribute to the model inferring the wrong world model in its reasoning chain and being unable to self-correct. However, we also find that a much more advanced reasoning model (o3) can solve the task nearly optimally. While implementation details of o3 are not publicly available, we speculate that the key ingredient is backtracking, or the model's ability to change its beliefs over inferred world models after they have been explicitly stated.

References

Ahmed, Z., Tenenbaum, J. B., Bates, C. J., and Gershman, S. J. Synthesizing world models for bilevel planning. *arXiv preprint arXiv:2503.20124*, 2025.

Betley, J., Bao, X., Soto, M., Sztyber-Betley, A., Chua, J., and Evans, O. Tell me about yourself: Llms are aware of their learned behaviors. *arXiv preprint arXiv:2501.11120*, 2025.

Binder, F. J., Chua, J., Korbak, T., Sleight, H., Hughes, J., Long, R., Perez, E., Turpin, M., and Evans, O. Looking inward: Language models can learn about themselves by introspection. *arXiv preprint arXiv:2410.13787*, 2024.

Damasio, A. *Self comes to mind: Constructing the conscious brain*. Vintage, 2012.

De Freitas, J., Uğuralp, A. K., Oğuz-Uğuralp, Z., Paul, L., Tenenbaum, J., and Ullman, T. D. Self-orienting in human and machine learning. *Nature Human Behaviour*, 7(12): 2126–2139, 2023.

Hofstadter, D. R. *Gödel, Escher, Bach: an eternal golden braid*. Basic books, 1979.

Hofstadter, D. R. and Dennett, D. C. The mind's i: Fantasies and reflections on self and soul. 1981.

Laine, R., Chughtai, B., Betley, J., Hariharan, K., Balesni, M., Scheurer, J., Hobbhahn, M., Meinke, A., and Evans, O. Me, myself, and ai: The situational awareness dataset (sad) for llms. *Advances in Neural Information Processing Systems*, 37:64010–64118, 2024.

Lehr, S. A., Saichandran, K. S., Harmon-Jones, E., Vitali, N., and Banaji, M. R. Kernels of selfhood: Gpt-4o shows humanlike patterns of cognitive dissonance moderated by free choice. *Proceedings of the National Academy of Sciences of the United States of America*, 122(20): e2501823122, 2025.

Minsky, M. *Society of mind*. Simon and Schuster, 1986.

OpenAI. Introducing openai o3 and o4-mini. URL https://openai.com/index/introducing-o3-and-o4-mini/.

OpenAI. Gpt-4o system card, 2024. URL https://openai.com/index/gpt-4o-system-card/. Accessed: 2024-02-03.

Paul, L., Ullman, T., De Freitas, J., and Tenenbaum, J. Reverse-engineering the self. 2023.

Tsividis, P. A., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S. J., and Tenenbaum, J. B. Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*, 2021.

Valmeekam, K., Marquez, M., Sreedharan, S., and Kambhampati, S. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Yamada, Y., Bao, Y., Lampinen, A. K., Kasai, J., and Yildirim, I. Evaluating spatial understanding of large language models. *arXiv preprint arXiv:2310.14540*, 2023.

A. Prompts

Here we provide the precise prompts used in our experiment. The first prompt is used as an initial chat message to the LLM, with a grid encoded as a comma-separated nested list of values $(0 \mid 1 \mid 2 \mid 3)$. Below, we provide an example initial grid.

Instruction Prompt

```
You are going to play a game where you will control an Agent and navigate to a Goal.
     You will make a series of moves where you can respond with 4 different
    directions: "up", "down", "left", or "right". Each of your responses will be a
    single word, which is one of these directions. After each move, you will receive
     an updated version of the game environment. You will receive a reward if you
    complete the game successfully.

Respond only with a single action "up", "down", "left", or "right". Only reply with
    1 word, and no other text.

Now, consider the following 2-dimensional grid:
```
[
   [0, 0, 0, 0, 0, 0, 0, 0, 0],
   [0, 2, 0, 0, 1, 0, 0, 2, 0],
   [0, 1, 1, 1, 1, 1, 1, 1, 0],
   [0, 1, 1, 1, 1, 1, 1, 1, 0],
   [0, 1, 1, 1, 3, 1, 1, 1, 0],
   [0, 0, 1, 1, 1, 1, 1, 1, 0],
   [0, 0, 1, 1, 1, 1, 1, 1, 0],
   [0, 2, 1, 1, 1, 0, 0, 2, 0],
   [0, 0, 0, 0, 0, 0, 0, 0, 0]
]
```

What is your first move?
```

Next, we provide the Follow-Up Prompt used to provide an updated observation of the environment given an action response from the LLM. At each step, the environment and the history of previous actions is encoded as a chat conversation history of actions from the agent and responses from the environment. The example updated grid shown here is what an LLM would receive, if it was given the same grid example in the first Instruction Prompt (with the `2` in the bottom-left being the agent) and the LLM replied with the action *"right"*.

Follow-Up Prompt

````
Here is an updated grid following your last move:

```
[
   [0, 0, 0, 0, 0, 0, 0, 0, 0],
   [0, 2, 0, 0, 1, 0, 0, 2, 0],
   [0, 1, 1, 1, 1, 1, 1, 1, 0],
   [0, 1, 1, 1, 1, 1, 1, 1, 0],
   [0, 1, 1, 1, 3, 1, 1, 1, 0],
   [0, 0, 1, 1, 1, 1, 1, 1, 0],
   [0, 0, 1, 1, 1, 1, 1, 1, 0],
   [0, 1, 2, 1, 1, 0, 0, 2, 0],
   [0, 0, 0, 0, 0, 0, 0, 0, 0]
]
```

What is your next move?
````

B. Full List of Stimuli Used

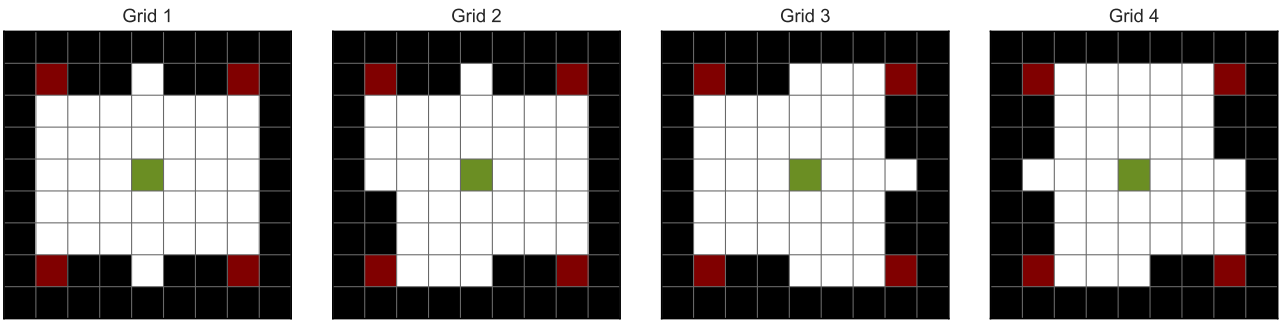

Figure 6. We used the following grid configurations for our experiments. For each grid, we evaluated the agent on each of the 4 possible agent locations. These grids selected from the twelve grid stimuli used in De Freitas et al. (2023), Experiment 1.