MAP INDUCTION: COMPOSITIONAL SPATIAL SUBMAP LEARNING FOR EFFICIENT EXPLORATION IN NOVEL ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans are expert explorers. Understanding the computational cognitive mechanisms that support this efficiency can advance the study of the human mind and enable more efficient exploration algorithms. We hypothesize that humans explore new environments efficiently by inferring the structure of unobserved spaces using spatial information collected from previously explored spaces. This cognitive process can be modeled computationally using program induction in a Hierarchical Bayesian framework that explicitly reasons about uncertainty with strong spatial priors. Using a new behavioral Map Induction Task, we demonstrate that this computational framework explains human exploration behavior better than non-inductive models and outperforms state-of-the-art planning algorithms when applied to a realistic spatial navigation domain.

1 INTRODUCTION

Humans efficiently use spatial reasoning to navigate and forage in new environments. We easily find our way around new buildings, and quickly (without exhaustive search) infer promising foraging locations. Which computational cognitive mechanisms make this possible, and could they be leveraged for better exploration in AI?

In this work we empirically study and computationally model human exploration in novel spaces. We begin by reviewing the literature on human spatial cognition involved in exploration and parallel developments of exploratory algorithms. Next, we present empirical evidence that humans use program induction to predict the structure of unseen space, and show how program induction can be used in a Partially Observable Markov Decision Process (POMDP) planner to optimize exploration.

Distributed non-metric representations. Human exploration relies on approximate spatial representations (Wang & Brockmole, 2003; Warren et al., 2017; Wiener & Mallot, 2003) that support near-optimal planning in daily life (Bongiorno et al., 2021) while sacrificing global metric accuracy (Vuong et al., 2019; Newcombe et al., 1999; Zhu & Levinson, 2015). These representations are acquired by combining redundant observations of local landmarks, such as views from multiple perspectives (Gillner & Mallot, 1998; Foo et al., 2005). The use of distributed spatial representations for cost-effective navigation also has a long history in AI. Examples include a computational theory of wayfinding based on multiple locally unique but globally similar landmarks (Prescott, 1996), which prioritized error recovery over accuracy. The Spatial Semantic Hierarchy model (Kuipers & Levitt, 1988) used distributed spatial graphs as maps for navigation (Kuipers, 2000; Kuipers et al., 2004). Simultaneous Localization and Mapping (SLAM) algorithms circumvent scale limitations by dividing the world into manageable locally metric submaps with topological relationships between submaps (Bosse et al., 2003; Fairfield et al., 2010). By contrast, recent recurrent neural network models of navigation have focused more closely on the metric structure of explored spaces and global map formation, showing how networks trained on path integration can take shortcuts (Banino et al., 2018).

Hierarchical organization. Humans minimize planning costs by representing spaces hierarchically – based on visible spatial boundaries (Kosslyn et al., 1974), geography (Stevens & Coupe, 1978), or otherwise interpreting spaces as composed of sub-regions (Hirtle & Jonides, 1985). Hierarchical spatial organisation is evident in a tendency to plan paths between regions, before planning sub-

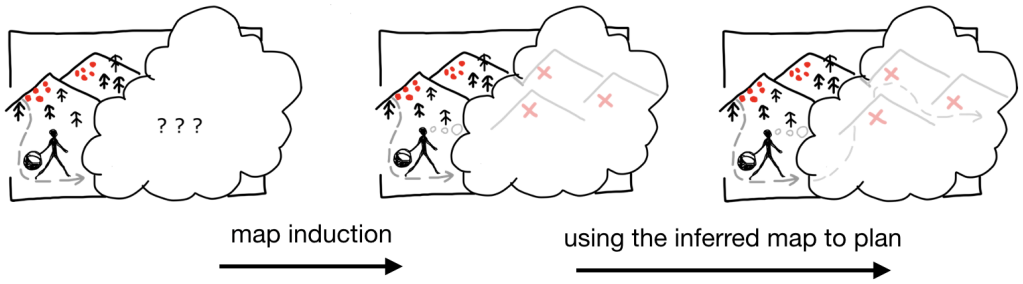

Figure 1: We propose a map induction hypothesis, by which humans infer the structure of unobserved spaces based on partial observations and priors based on familiar environments. Map induction can optimize exploration cost and fill in incomplete or missing observations.

paths within each region (Bailenson et al., 2000; Newcombe et al., 1999; Wiener & Mallot, 2003; Wang & Brockmole, 2003; Balaguer et al., 2016), and in increased reaction times when switching between hierarchy levels during plan execution (Balaguer et al., 2016; Kryven et al., 2021). Hierarchical state-spaces in non-spatial planning domains include drawing (Tian et al., 2020) inverse inference (Shu et al., 2020), and reasoning about topology (Tomov et al., 2020).

Hierarchical representations in AI are often expressed as options – a hierarchical grouping of actions frequently performed together (Sutton et al., 1999). Assuming that a novel Markov Decision Processes (MDP) is sampled from a family of similar MDPs, its rewards can be learned as hierarchically derived from a parent distribution shared by the MDP family (Wilson et al., 2012). Singh et al. (2012) proposed a computational framework for learning efficient state-space representations by recognizing which sequences of actions lead to identical observations – for example, grouping together paths that lead to observing the same landmark. A principle of grouping game-board states based on rotation and reflection symmetries has been used to optimize state-space representations in the game of Go (Silver et al., 2017).

Shared reference frames. From young children to hunter-gatherers, humans spontaneously organize sensory precepts into patterns (Pitt et al., 2021), and use them to generalize between tasks (Tian et al., 2020; Lake & Piantadosi, 2020; Schulz et al., 2018). Generalization and transfer learning in spatial domains include phenomena such as mirror-invariant neural scene representations (Dilks et al., 2011), and reusing of reference frames across similar environments (Marchette et al., 2014). Shared reference frames may occur in other mammals as well, as suggested by evidence of rodents reusing grid-cell maps between similar parts of environments (Derdikman et al., 2009; Carpenter et al., 2015). However, these studies have neither studied the process of discovering structural patterns during spatial exploration, nor considered formal quantitative models of how structured map knowledge may be built and reused.

Map induction. We propose that humans optimize exploration of new spaces by representing maps as composed of reusable reference frames. More specifically, we propose a *map induction* hypothesis – that humans use program induction to infer possible maps of unseen spaces, as made up of submaps encountered in the observed areas. *Program induction* generally refers to inferring a *program* that likely produced a given sequence of observations (Stuhlmuller et al., 2010; Lake et al., 2015). For example, given a sequence of numbers, the inferred program may be an arithmetic progression; given a natural texture the program may be a reaction-diffusion equation (Camazine et al., 2020). Although Bayesian models of concept-learning by probabilistic program induction have been developed for many domains (Stuhlmuller et al., 2010; Lake et al., 2015; Tian et al., 2020), the use of program induction to understand spatial structures and environments is unexplored. Map induction can be used to infer which program(s) could have generated the layout of an environment, given partial observations of this environment, and to generate a hypothetical map of the unseen parts which can then be used for exploration. For example, one can plan a foraging trip by anticipating that a landscape of hills will continue in a previously observed pattern and that the distribution of berries in the unseen space will continue to be similar to the distribution observed so far (see Figure 1).

In this work, we adopt a scientific and engineering goal: (1) to empirically study how humans learn spatial representations using a novel experimental Map Induction Task, and (2) describe a

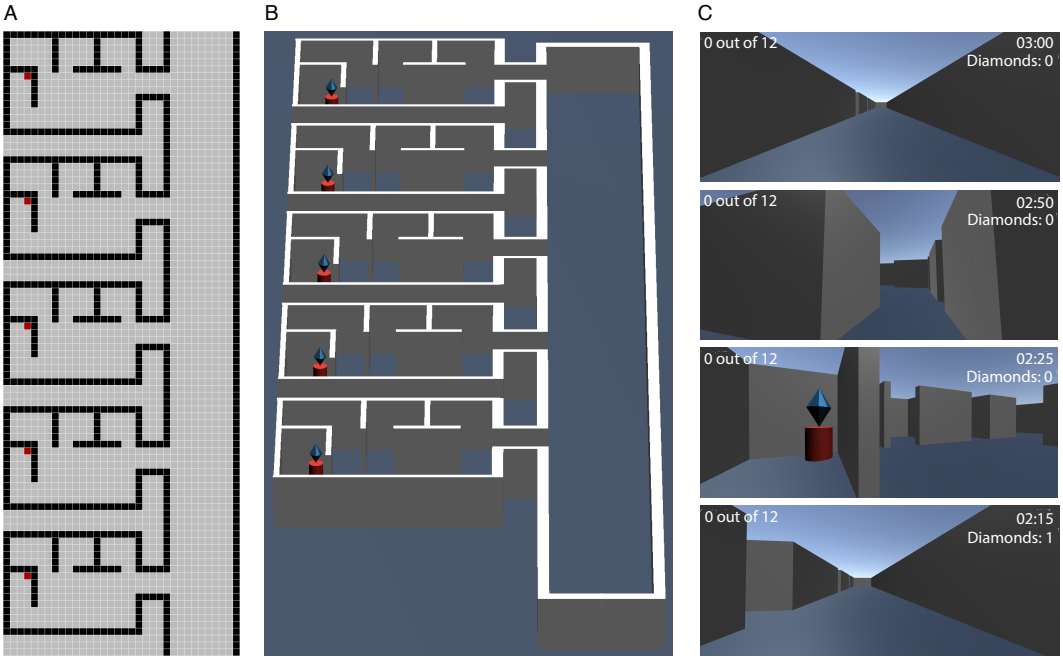

Figure 2: An example of an MIT task. **A.** A 2-dimensional grid discretization showing the map used for modeling exploration trajectory. The black cells indicate walls, and the red cells indicate reward locations. **B.** The 3-dimensional world, in which subjects search for rewards (shown as blue diamonds on a red platform), seen from an overhead perspective in the Unity Design system. **C.** First-person views of the task, as seen by subjects at different times during the experiment.

computational model of map induction that can be leveraged to optimize exploration in AI. We show that a Hierarchical Bayesian exploration model that implements map induction predicts human exploration and outperforms a naive implementation of a Partially Observable Markov Decision Process (POMDP) that plans without map induction (Experiment 1), showing that map induction may be an important computational principle of human cognition. Moreover, we show that human exploration strategies rely on probabilistic distributions over likely maps-generating programs, as opposed to using just the most likely map (Experiment 2). In the next section, we introduce the Map Induction Task. We then give a detailed description of computational models in Section 3. In Section 4, we describe two human experiments and compare behavioral metrics with results from our models.

2 Map Induction Task

We hypothesize that humans represent survey knowledge as composed of regions, where each region is either recognized as new or categorized as an instance of a previously learned region. For example, someone entering a new unit in an apartment building will likely recognize it as a new region, but after exploring several units, they will be able to predict the remaining floor-plan. One could also consider several possible floor plans consistent with prior observations – for example, the right and the left wing of the building could mirror each other, or two types of units could occur in alteration, etc. In natural spaces, predictable structures occur in patterns of hills and valleys, branching riverbeds, and plants favoring certain features of the environment (see Figure 1).

To study map induction during exploration, we have designed a Map Induction Task (MIT) in which subjects forage for rewards in novel, partially observable 3-dimensional environments. Each environment consists of repeating *units* – structural blocks which occur multiple times throughout the environment by design. All environments contain multiple rewards (diamonds), hidden in specific parts of each unit (see Figure 2B). The subjects have no advance knowledge of environment structure and are instructed to collect all diamonds while navigating the environments from a first-person

perspective (see Figure 2 C). Subjects use a keyboard and mouse (or a trackpad) to navigate, which are standard navigation controls in first-person games. The distribution of rewards is initially unknown to the subjects but can be predicted from the layout of the environment – for example, the rewards are placed in the same part of a unit each time a given unit occurs. While the MIT task can be completed without map induction or even without an explicit map representation – for example, by following a wall – such a naive strategy would take longer than selectively exploring only the parts likely to contain diamonds. Each trial continues either until all diamonds are collected or until a timeout is reached, whichever comes first. The timeouts are chosen to allow subjects enough time to collect all diamonds if searching exhaustively.

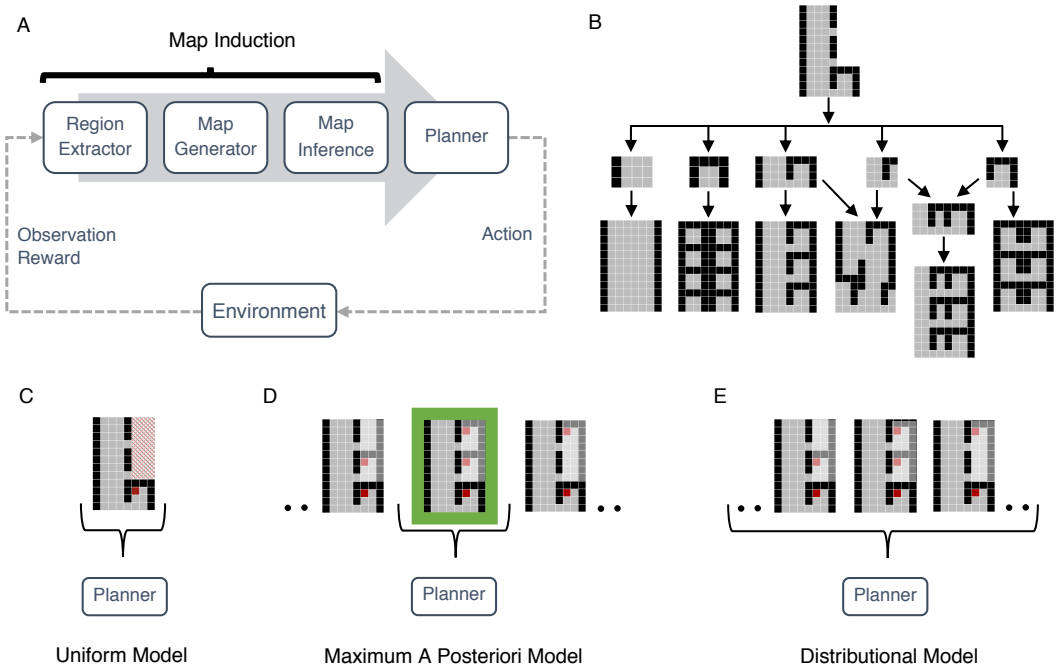

Figure 3: Model architecture. **A.** Four steps required to solve the MIT task and the corresponding computational modules in our framework. **B.** Extracting the candidate regions (submaps) from the observed portion of the environment and generating possible map completions by composing these regions and their simple transformations (e.g., mirror reflections) as defined in the generative grammar of the Map Generator. **C.** Uniform Model assumes a uniform distribution for the unobserved part of the map. **D.** Maximum A Posteriori Model uses the most likely map completion for planning. **D.** Distributional Model uses the entire distribution for planning.

3 Computational Models

Optimizing performance in the MIT task requires several computational steps: (1) inferring that the environment is composed of repeating units (2) inferring the most likely structure(s) of the environment (3) inferring the distribution of rewards within the environment, and (4) planning the shortest route that collects all rewards. Here (1) and (2) involve program induction over the kinds of programs that may have generated the environment, and (3) involves program induction over programs that may have generated the distribution of rewards. For example, the rewards could be placed in the top left corner of every unit, in every odd unit, etc. The reward-generating program could potentially be complex; however, we consider simple short programs and environments in the current work. Because of uncertainty in the environment, this computational pipeline is executed in the context of a replanning loop. Finally, step (4) involves solving a Partially Observed Markov Decision Process (POMDP) that reasons about the distribution of hypothetical environments. While it is possible to do (4) without performing the first three steps, this would entail assuming an uninformed prior that any unobserved cell could equally likely be a wall, an empty space, or a reward – which would result in exploring exhaustively.

Formally, we can model this process by a Bayesian generative framework consisting of four computational modules (see Figure 3A) corresponding to the four steps described above. Each module can be independent of the others and can express different modeling assumptions. In the remainder of this section, we discuss the function of each module in detail and define three specific combinations of module implementations, to express three distinct computational hypotheses formalizing the principle of map induction in different ways.

3.1 MAP INDUCTION THROUGH OBSERVED SPATIAL STRUCTURES

We hypothesize that while navigating novel environments, humans approximately infer a posterior distribution $p(M|\mathcal{D})$ over its possible maps, given a history of observations, and use this distribution to plan. In this section, we will describe the steps taken by our model to estimate this posterior distribution (shown in Figure 3A).

3.1.1 REGION EXTRACTOR:

As humans explore novel spatial environments, they infer the expected maps of these environments based on the parts of the environment that they have observed. The region extractor encapsulates this process by extracting regions (or submaps) M_p from previously explored parts of the map that can be composed to form possible map completions M. Figure 3B illustrates this process with a simple example. The second stage of the hierarchy shows examples of regions extracted from the observed map above. We use the term region to refer to a submap, considered by a model as a hypothetical building block of the environment. This is different from the term unit used in the previous section to describe repeating elements in environment design. It is conceivable that multiple regions can represent a unit or that a region may comprise several units.

3.1.2 MAP GENERATOR:

To develop a space of possible map completions using the extracted regions, we define a probabilistic generative grammar that combines region primitives into a set of possible completed maps. If the map currently being explored has dimensions s_x, s_y, and d_x, d_y are the dimensions of the extracted regions, then the generative grammar for that map is described as shown in Table 1. The third level

Production Rule			Probability
$M_p(d_x, d_y)$	$\rightarrow$	$\text{FLIPH}(M_p(d_x, d_y))$	$\frac{1}{3}$
$M_p(d_x, d_y)$	$\rightarrow$	$\text{FLIPV}(M_p(d_x, d_y))$	$\frac{1}{3}$
$M_p(d_x, d_y)$	$\rightarrow$	$\text{ROTATE90}(M_p(d_y, d_x))$	$\frac{1}{3}$
$M_p(d_{x_1} + d_{x_2}, d_{y_1})$	$\rightarrow$	$\text{HCAT}(M_p(d_{x_1}, d_{y_1}), M_p(d_{x_2}, d_{y_1}))$	$\frac{1}{2}$
$M_p(d_{x_1}, d_{y_1} + d_{y_2})$	$\rightarrow$	$\text{VCAT}(M_p(d_{x_1}, d_{y_1}), M_p(d_{x_1}, d_{y_2}))$	$\frac{1}{2}$
M	$\rightarrow$	$M_p(s_x, s_y)$	1

Table 1: Probabilistic context-free grammar used to generate a distribution of map hypotheses.

of the hierarchy in Figure 3B shows examples of map completions generated using the Map Generator. The probabilities assigned to the rules in the generative grammar encode priors on coverage and consistency of the generated maps.

3.1.3 MAP INFERENCE:

We use this probabilistic generative grammar along with the following likelihood over generated maps to form the posterior $p(M|D)$:

$$l(D|M) = \left[1 - \frac{\beta}{s_x \times s_y}\right] \times \left[1 - \frac{\gamma}{s_x \times s_y}\right]$$

where $\gamma/s_x \times s_y$ is the fraction of the map not predicted by a given map completion and $\beta/s_x \times s_y$ is the fraction of mismatched reward locations relative to the history of observations. At a high level, this likelihood function encodes two simple assumptions: (1) humans prefer map completions that are maximally descriptive of unseen portions of the environment; (2) humans prefer map hypotheses that are minimally contradictory with previous observations. The posterior distribution $p(M|D)$ is computed using the above likelihood function, assuming a uniform prior distribution over the generated map hypotheses.[1]

3.1.4 PLANNER

We model the exploration problem as a Partially Observable Markov Decision Process (POMDP). In a POMDP, the state is not directly observable by the agent and can only be inferred through sequences of observations in the environment. A POMDP can be described as a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \Omega, \mathcal{O}, \gamma \rangle$ where $\mathcal{S}$ is the set of possible states in the world, $\mathcal{A}$ is a finite set of actions, $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$ is a stochastic transition function which maps a state action pair to a distribution over possible next states, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function that maps state action pairs to a scalar reward, Ω is a set of observations that an agent can experience in the world, $\mathcal{O} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\Omega)$ is an observation function which maps a state and action pair to a distribution of possible observations after taking the action, and γ is a discount factor such that $0 < \gamma < 1$. An optimal agent in this formulation acts to maximize the expected discounted reward from the environment.

$$\pi^*(s) = \arg\max_a \Big(\mathbb{E}\Big[\sum_{k=0}^{\infty} \gamma^k R(s,a) \Big] \Big)$$

For our task, the state space is defined as $\mathcal{S} = \mathcal{H} \times [0,1]^{|\mathcal{H}|} \times \mathcal{S}_a$ where $\mathcal{H}$ is the set of possible map hypotheses from $p(M|D)$, $[0,1]^{|\mathcal{H}|}$ is the probability associated with each hypothesis, and $\mathcal{S}_a$ is a 3-tuple (a_x, a_y, a_r) with dimensions $(s_x, s_y, 4)$ indicating the index into the grid world state where the agent is positioned and the current direction the agent is facing. The action space is discrete with size 3 for bidirectional rotation and forward movement. We use a reward function R defined by an indicator function over the state space $\mathbb{1}(S_g[a_x, a_y] = \text{reward})$. The transition function $T(s'|s,a)$ is a stochastic function that maps the given action and the previous state to a new state. There is no closed-form expression for T, but it can be expressed using a generative probabilistic sampler that takes in s, samples a grid from the hypothesis distribution in s, deterministically simulates action a in the sampled grid, and returns a new position and orientation of the agent and normalized hypothesis distribution. The observation space $\Omega(s, a; D)$ is defined by a surjective line-of-sight (LOS) observation function $\mathcal{O}$ that maps from state and action to a deterministic observation. This function casts rays within a 90-degree field of view in the direction of the agent's orientation. Lastly, we define $\gamma = 0.90$ for all of our experiments. Since finding an exact solution is intractable due to the size of the problem (Kaelbling et al., 1998), we search through belief space using an approximate online Partially Observable Monte Carlo Planner (POMCP) (Silver & Veness, 2010).

3.2 COMPUTATIONAL HYPOTHESES

We compare human performance to three hypotheses (variants of POMCP) to evaluate whether and how human exploration implements the computational steps described in Figure 3A:

1. **Uniform Model (Uniform-POMCP)**: The Uniform model doesn't use map induction, i.e., it doesn't learn any inductive priors about spatial structures during exploration outlined in steps (1)-(3). Instead, it assumes a uniform distribution over possible settings for each grid cell (reward, empty, etc.) and uses that for planning as shown in Figure 3C.
2. **Maximum A Posteriori Model (MAP-POMCP)**: The MAP model uses the most likely map from the distribution of induced maps $p(M|D)$ to plan (see Figure 3D).
3. **Full Distributional Model (D-POMPC)**: The Distributional model uses the entire set of induced maps in the distribution $p(M|D)$ to plan (see Figure 3E).

[1] Given the relatively small environments, we are able to compute the posterior exactly. However, larger and more complex environments would likely require approximate Bayesian inference methods such as Markov chain Monte Carlo.

4 BEHAVIORAL EXPERIMENTS

Below we describe two human behavioral experiments. Experiment 1 tests whether humans perform map induction, as implemented by the MAP-POMCP and D-POMCP models, in contrast to naive exploration implemented by Uniform-POMCP. The results of the first experiment show that humans indeed rely on map induction and rule out the use of the Uniform model. Experiment 2 tests whether human exploration is best explained by MAP-POMCP or D-POMCP models. This experiment shows that humans plan according to a distribution of hypothetical maps and explicitly gather information to disambiguate those hypotheses.

4.1 EXPERIMENT 1: USING MAP INDUCTION TO OPTIMIZE EXPLORATION

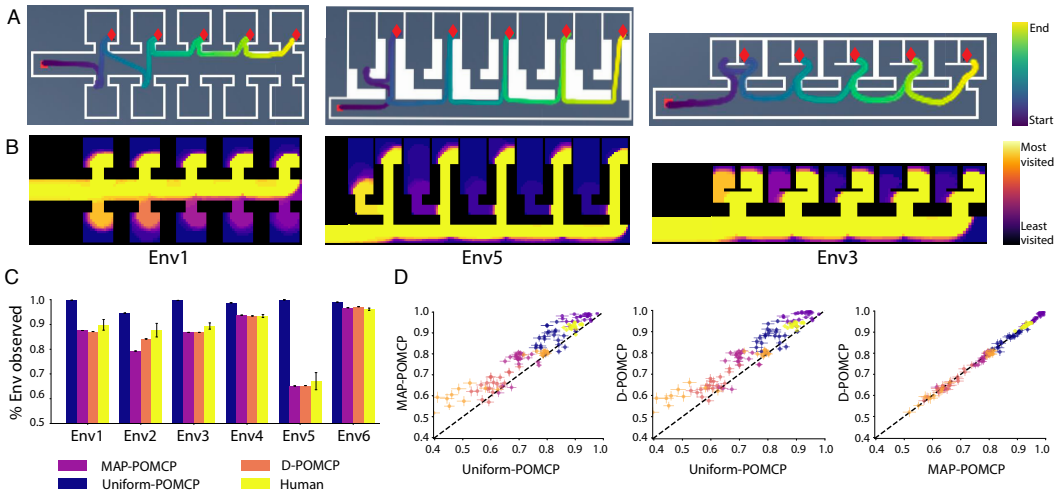

Figure 4: Experiment 1 results. **A**. Exploration trajectories of a representative subject. **B**. Visitation heatmaps of three example environments aggregated across subjects and pairs of reflected environments. The visited area was computed in a 2D grid projection, using a circular radius of five grid cells around the agent. **C**. Fractions of environments observed by the models and humans. Error bars show 95% confidence intervals. **D**. Percent likelihood that the model takes the same actions as the human (details in Appendix section A.3). Each marker is a subject-environment pair, with each color showing a single environment: Env1-Env6 (blue-yellow). Error bars show standard error along each axis.

Method: The experiment was conducted in a web browser, using 3D virtual environments built-in Unity WebGL. Each environment was composed of a corridor connecting five instances of a specific unit. The rewards (diamonds) were placed at identical locations within each unit. An example environment is shown in Figure 2. Each subject performed four practice trials to gain familiarity with the experimental setup and controls, followed by an instruction quiz and 12 test trials in order randomized between subjects. Subjects who failed the instruction quiz repeated the instructions and practice until they answered the quiz correctly. The timer for the trial and the number of diamonds collected were shown on the upper right corner of the screen (see Figure 2C). See Supplementary Materials for full experimental instructions and screenshots.

Stimuli: The 12 test trials comprised six pairs of unique environments and their reflections. We used the reflected environments to increase the number of trials and control for possible left or right biases in navigation.

Subjects: We recruited 30 subjects via Amazon Mechanical Turk. Subjects were paid for 35 minutes of work and received a monetary bonus for each collected diamond.

Results: Out of the 30 subjects, four subjects explored exhaustively, 2 did not complete the experiment and 24 subjects were able to predict reward placement and reduce exploration time, indicating that humans use map induction. Subjects generally exhaustively explored the first two units upon entering a new environment, followed by partial exploration through the remainder of the trial.

Throughout the experiment, the extent to which non-rewarding areas were visited also decreased, indicating that people were increasingly confident about the repeating layout.

Figure 4A shows examples of exploration trajectories of one subject. Subjects generally used identical trajectories to search the reflected pairs of environments, and so we aggregated results for each of the six pairs. Figure 4B illustrates the extent to which three example environments were visited, aggregated across subjects, and across reflected pairs of environments. See the Appendix for a complete plot of all exploration trajectories and heatmaps corresponding to all environments. Figure 4C shows the fractions of each environment observed by humans and models. The areas observed by MAP-POMCP and D-POMCP were similar to humans', while the area observed by Uniform-POMCP was significantly larger. Figure 4D shows the fit of the three models to human behavior, indicating that the MAP-POMCP and D-POMCP models, which implement map induction, explain human behavior better than the Uniform-POMCP model, which makes no predictions about map structure.

4.2 Experiment 2: Distributional sub-map representations

Experiment 2 differentiates between MAP-POMCP and D-POMCP models using color cues to indicate the location of rewards within each unit. This was done by including a 'cue room' within each unit – a small room containing information about the location of the reward. The environments were designed so that MAP-POMCP would take a longer exploration path compared to D-POMCP. A planner guided by the MAP-POMCP model ignores the cue rooms and heads toward the part of a unit most likely to contain a diamond based on previous experience. In contrast, the D-POMCP model visits the cue room to gather information about the location of the diamonds.

Method: Experiment 2 followed a procedure similar to Experiment 1, with a different set of environments. Each subject performed five practice trials, followed by an instruction quiz, 12 test trials in order randomized between subjects, and a test of skills using navigation controls. The controls skill test ensured that subjects could navigate in WebUnity without undue difficulty since Experiment 2 used larger environments. Subjects who failed the controls skill test were paid but excluded from the analysis. The instructions were modified to prompt subjects to consider colors as cues to the placement of the reward – see the Appendix for full details.

Stimuli: The 12 test environments comprised six unique environments and their reflections. The color cues were randomized within each environment so that the reflected pairs of environments were colored in different ways and contained rewards in different locations.

Subjects: We recruited 51 subjects via Amazon Mechanical Turk. Each was paid for 35 minutes of work and received a monetary bonus for each collected diamond.

Results: Out of 51 subjects, 35 completed the controls skill test. Of the 35 subjects, all but nine were able to successfully use color cues, as predicted by the D-POMCP model, indicating that the majority took into account the entire distribution of possible maps in contrast to just the most likely map. Example exploration trajectories of one subject and visitation heatmaps are shown in Figure 5 A and B. See the Appendix for more results. Comparing the fractions of environments observed by humans and models indicates that only the D-POMCP model was comparable to humans (see Figure 5C). D-POMCP model was also best at explaining human behavioral data, as shown in Figure 5D.

5 Discussion

In this work, we proposed a novel *map induction hypothesis* – that humans use program induction to infer maps of novel environments from partial observations – and use the inferred map distribution to optimize exploration. We built a model that conforms to this hypothesis by combining a Bayesian map induction model and an approximate belief-space planner. Lastly, we present the results of two behavioral experiments with a novel Map Induction Task that validate this hypothesis and develop a proof-of-concept computational model that predicts human exploration.

While we explored map induction in a simple environment, it is likely to apply more widely – humans not only forage selectively but also consider only plausible theories or explore only a certain part of human-made structures. This may indicate that humans anticipate the structure of abstract search spaces by noticing repetitions and symmetries to simplify hard computing problems in var-

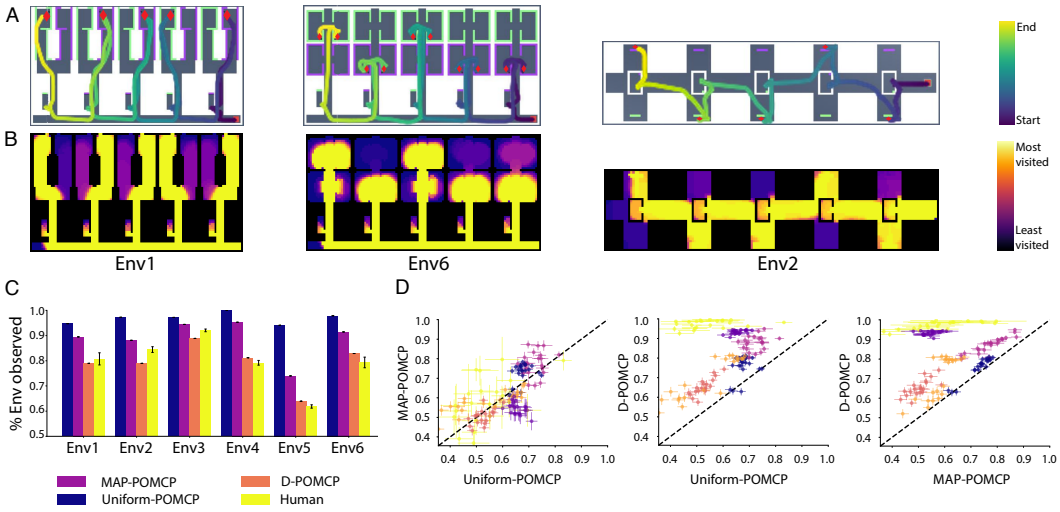

Figure 5: Experiment 2 results. **A**. Exploration trajectories of a representative subject. **B**. Visitation heatmaps of three example environments aggregated across subjects. The visited area was computed in a 2D grid projection, using a circular radius of five grid cells around the agent. **C**. Fractions of environments observed by the models and humans. Error bars show 95% confidence intervals. **D**. Percent likelihood that the model takes the same actions as the human (details in Appendix section A.3). Each marker is a subject-environment pair, with each color showing a single environment: Env1-Env6 (blue-yellow). Error bars show standard error along each axis.

ious domains. Using map induction for exploration may not be unique to humans either – the evidence of rodents reusing grid-cell maps in similar parts of environments suggests that grid-cell remapping may be a neural signature of map induction (Derdikman et al., 2009; Carpenter et al., 2015). Hippocampal reuse of place-cell maps in composite environments (Paz-Villagrán et al., 2004), which are invariant to rotation and scaling (Muller & Kubie, 1987), suggests that animals use landmark cues to guide map induction as well.

In future work we intend to study map induction in larger, more naturalistic environments, where enumerating all possible map completions is not feasible, by imposing priors over map completions - such as prioritizing shorter programs, and maps with realistic statistics.

Our work takes a step toward understanding efficient exploration in novel environments by modelling map induction. We show that map induction can be used to optimize state-of-the art approximate planning, and may support building better exploration algorithms in AI. It may also have potential applications for fast and generalizable map learning in the field of SLAM.

Reproducibility Statement

The source code for the models and the behavioral datasets used for this project are made available at the following GitHub repository, along with instructions on how to use them.
`https://anonymous.4open.science/r/Map-Induction`

The compiled Unity WebGL build packages for both experiments presented in this paper are also provided at the above repository. The builds can be hosted using any web hosting service (e.g., simmer.io) in order to re-run the experiments and reproduce experimental results. The Unity source code used to generate these builds is available on request.

To ensure reproducibility, the behavioral experiments are also explained in detail in the Appendix, and the links to both web hosted experiments are provided below:
Experiment1: `http://18.25.132.202/experiment.html`
Experiment2: `http://18.25.132.202/experiment3.html`

Our institutional IRB approved all experiments.

REFERENCES

Jeremy N Bailenson, Michael S Shum, and David H Uttal. The initial segment strategy: A heuristic for route selection. *Memory & Cognition*, 28(2):306–318, 2000.

Jan Balaguer, Hugo Spiers, Demis Hassabis, and Christopher Summerfield. Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron*, 90(4):893–903, 2016.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

Christian Bongiorno, Yulun Zhou, Marta Kryven, David Theurel, Alessandro Rizzo, Paolo Santi, Joshua Tenenbaum, and Carlo Ratti. Vector-based pedestrian navigation in cities, 2021.

Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An atlas framework for scalable mapping. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pp. 1899–1906. IEEE, 2003.

Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Guy Theraula, and Eric Bonabeau. *Self-organization in biological systems*. Princeton university press, 2020.

Francis Carpenter, Daniel Manson, Kate Jeffery, Neil Burgess, and Caswell Barry. Grid cells form a global representation of connected environments. *Current Biology*, 25(9):1176–1182, 2015.

Dori Derdikman, Jonathan R Whitlock, Albert Tsao, Marianne Fyhn, Torkel Hafting, May-Britt Moser, and Edvard I Moser. Fragmentation of grid cell maps in a multicompartment environment. *Nature neuroscience*, 12(10):1325–1332, 2009.

Daniel D Dilks, Joshua B Julian, Jonas Kubilius, Elizabeth S Spelke, and Nancy Kanwisher. Mirror-image sensitivity and invariance in object and scene processing pathways. *Journal of Neuroscience*, 31(31):11305–11312, 2011.

Nathaniel Fairfield, David Wettergreen, and George Kantor. Segmented slam in three-dimensional environments. *Journal of Field Robotics*, 27(1):85–103, 2010.

Patrick Foo, William H Warren, Andrew Duchon, and Michael J Tarr. Do humans integrate routes into a cognitive map? map-versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):195, 2005.

Sabine Gillner and Hanspeter A Mallot. Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of cognitive neuroscience*, 10(4):445–463, 1998.

Stephen C Hirtle and John Jonides. Evidence of hierarchies in cognitive maps. *Memory & cognition*, 13(3):208–217, 1985.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Stephen M Kosslyn, Herbert L Pick Jr, and Griffin R Fariello. Cognitive maps in children and men. *Child development*, pp. 707–716, 1974.

Marta Kryven, Suhyoun Yu, Max Kleiman-Weiner, and Joshua Tenenbaum. Adventures of human planners in maze search task. *PsyArXiv*, 2021.

Benjamin Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pp. 4845–4851. IEEE, 2004.

Benjamin J Kuipers and Todd S Levitt. Navigation and mapping in large scale space. *AI magazine*, 9(2):25–25, 1988.

Brenden M Lake and Steven T Piantadosi. People infer recursive visual concepts from just a few examples. *Computational Brain & Behavior*, 3(1):54–65, 2020.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Steven A Marchette, Lindsay K Vass, Jack Ryan, and Russell A Epstein. Anchoring the neural compass: coding of local spatial reference frames in human medial parietal lobe. *Nature neuroscience*, 17(11):1598, 2014.

Robert U Muller and John L Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *Journal of Neuroscience*, 7(7):1951–1968, 1987.

Nora Newcombe, Janellen Huttenlocher, Elisabeth Sandberg, Eunhui Lie, and Sarah Johnson. What do misestimations and asymmetries in spatial judgement indicate about spatial representation? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(4):986, 1999.

V Paz-Villagrán, E Save, and Bruno Poucet. Independent coding of connected environments by place cells. *European Journal of Neuroscience*, 20(5):1379–1390, 2004.

Benjamin Pitt, Stephen Ferrigno, Jessica Cantlon, Daniel Casasanto, Edward Gibson, and Steven T Piantadosi. Spatial concepts of number, size, and time in an indigenous culture. *PsyArXiv*, 2021.

Tony J Prescott. Spatial representation for navigation in animats. *Adaptive Behavior*, 4(2):85–123, 1996.

Eric Schulz, Nicholas T. Franklin, and Samuel J. Gershman. Finding structure in multi-armed bandits. *bioRxiv*, 2018. doi: 10.1101/432534. URL `https://www.biorxiv.org/content/early/2018/12/29/432534`.

Tianmin Shu, Marta Kryven, Tomer D Ullman, and Joshua B Tenenbaum. Adventures in flatland: Perceiving social interactions under physical dynamics. In *42d Proceedings of the Annual Meeting of the Cognitive Science Society*, 2020.

David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems, 2010*. Neural Information Processing Systems, 2010.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Satinder Singh, Michael James, and Matthew Rudary. Predictive state representations: A new theory for modeling dynamical systems. *arXiv preprint arXiv:1207.4167*, 2012.

Albert Stevens and Patty Coupe. Distortions in judged spatial relations. *Cognitive psychology*, 10 (4):422–437, 1978.

Andreas Stuhlmuller, Joshua B Tenenbaum, and Noah D Goodman. Learning structured generative concepts. In *32d Proceedings of the Annual Meeting of the Cognitive Science Society*. Cognitive Science Society, 2010.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Lucas Tian, Kevin Ellis, Marta Kryven, and Josh Tenenbaum. Learning abstract structure for drawing by efficient motor program induction. *Advances in Neural Information Processing Systems*, 33, 2020.

Momchil S Tomov, Samyukta Yagati, Agni Kumar, Wanqian Yang, and Samuel J Gershman. Discovery of hierarchical representations for efficient planning. *PLoS computational biology*, 16(4): e1007594, 2020.

Jenny Vuong, Andrew W Fitzgibbon, and Andrew Glennerster. No single, stable 3d representation can explain pointing biases in a spatial updating task. *Scientific reports*, 9(1):1–13, 2019.

Ranxiao Frances Wang and James R Brockmole. Human navigation in nested environments. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(3):398, 2003.

William H Warren, Daniel B Rothman, Benjamin H Schnapp, and Jonathan D Ericson. Wormholes in virtual space: From cognitive maps to cognitive graphs. *Cognition*, 166:152–163, 2017.

Jan M Wiener and Hanspeter A Mallot. 'fine-to-coarse'route planning and navigation in regionalized environments. *Spatial cognition and computation*, 3(4):331–358, 2003.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. Transfer learning in sequential decision problems: A hierarchical bayesian approach. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 217–227. JMLR Workshop and Conference Proceedings, 2012.

Shanjiang Zhu and David Levinson. Do people use the shortest path? an empirical test of wardrop's first principle. *PloS one*, 10(8), 2015.

A Appendix

A.1 Experiment 1

Figure 6 shows the stimuli used in the experiment, with practice stimuli in panel A and test stimuli in panel B. Each of the test stimuli is composed of a corridor connecting five instances of a specific unit. The rewards (diamonds) are placed at identical locations within each unit. The task was to navigate the environments and collect all the diamonds in the least amount of time. At the beginning of each trial, the subject is spawned at a fixed starting location (marked by a red floor-mat) in all environments. The subjects can then use the keyboard and mouse controls to explore the environment and find the diamonds. Note that there are six base test stimuli with the other six as their reflected versions. We use reflected versions to counter-balance the left and right turns required across environments during the experiment and to increase the number of experimental trials while saving effort on environment design. This experimental design can also be used to study how map induction generalizes to transformations of environments (these results are out of the scope of this paper and are therefore not presented). For instance, we noticed that subjects were able to use program induction across environments when they shared similar spatial structures.

Figure 7 shows the scenes displayed during the progression of the experiment. Figure 7A is the welcome scene that subjects see at the beginning of the experiment. Here, we confirm that the subjects consent to participating in the experiment voluntarily, and collect their age and gender information. Figure 7B is the next scene that provides instructions to the subjects about the task and the keyboard/mouse controls that they can use to navigate during the task. Figure 7C shows the first practice trial and is followed by three additional practice trials in the order shown in Figure 6A. At the end of each trial, a "Level Complete" (Figure 7D) or "Out of Time" (Figure 7E) pop-up window is displayed depending on whether the subject collected all the diamonds in the environment or ran out of time before doing so. Both these pop-ups also specify the ratio of the number of diamonds collected to the total diamonds embedded in the environment. This provides feedback to the subjects about their performance in the trial before they continue to the next trial. At the end of the practice trials, subjects are presented with an instruction quiz as shown in Figure 7F, to ensure that they understand the task. If they answer the instruction quiz incorrectly, they are requested to re-read the instructions and re-do the practice trials, and are sent back to the instructions scene in Figure 7B.

If the subjects answer the instruction quiz correctly, they are presented with the twelve test trials (see Figure 6B) in succession. An example test trial is shown in Figure 7G. The test stimuli are grouped into three blocks containing four stimuli each, according to their difficulty level. For each subject, the order of stimulus presentation within blocks is randomized, while the blocks themselves are presented in a fixed order e.g., block1 (simple environments - Env1, Env1R, Env2R, Env2R) is presented first, followed by block2 (moderate complexity - Env3, Env3R, Env4, Env4R), followed by block3 (high complexity - Env5, Env5R, Env6, Env6R). After the test stimuli, a final scene in Figure 7H is shown to thank the subjects for their participation, to get their comments about the strategies they used to solve the task, and their general feedback about the experiment.

Figure 8 shows the results from the experiment. Panel A shows the trajectories of a representative subject on six base test trials (Env1-Env6), shown in the order presented to the subject (top - bot-

tom). Panel B shows the visitation heat-maps indicating the least to most visited regions in each environment, averaged across subjects. The heat-maps are aggregated across the base environments (Env1-Env3) and their reflected versions (Env1R-Env6R) shown in Figure 6B.

Link to the web hosted experiment: `http://18.25.132.202/experiment.html`.

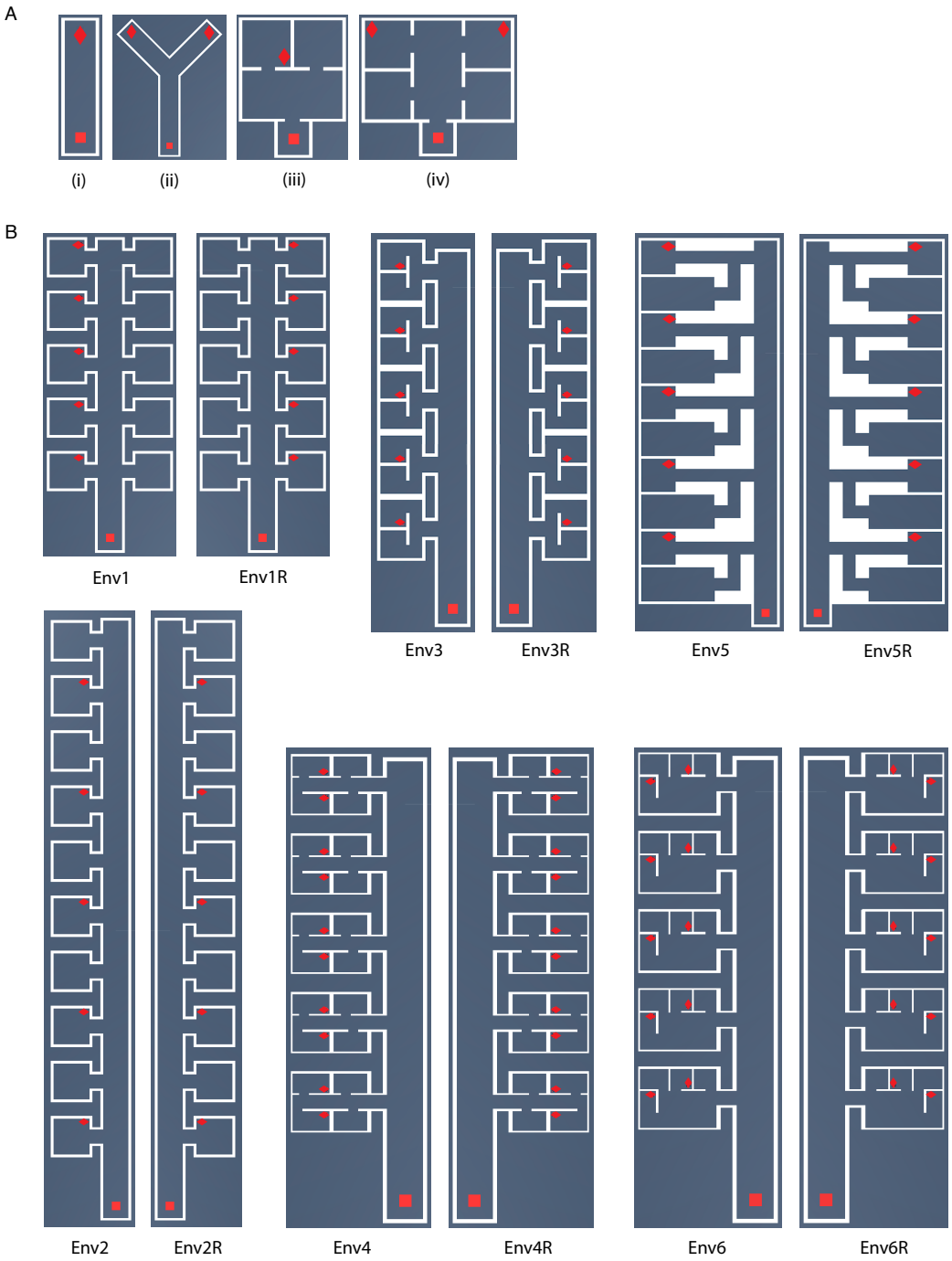

Figure 6: Top-down views of the stimuli used in Experiment1. **A.** Practice stimuli. **B.** Test stimuli. 'R' indicates reflected. The red square is a floor mat indicating the starting location. The red diamonds are the rewards.

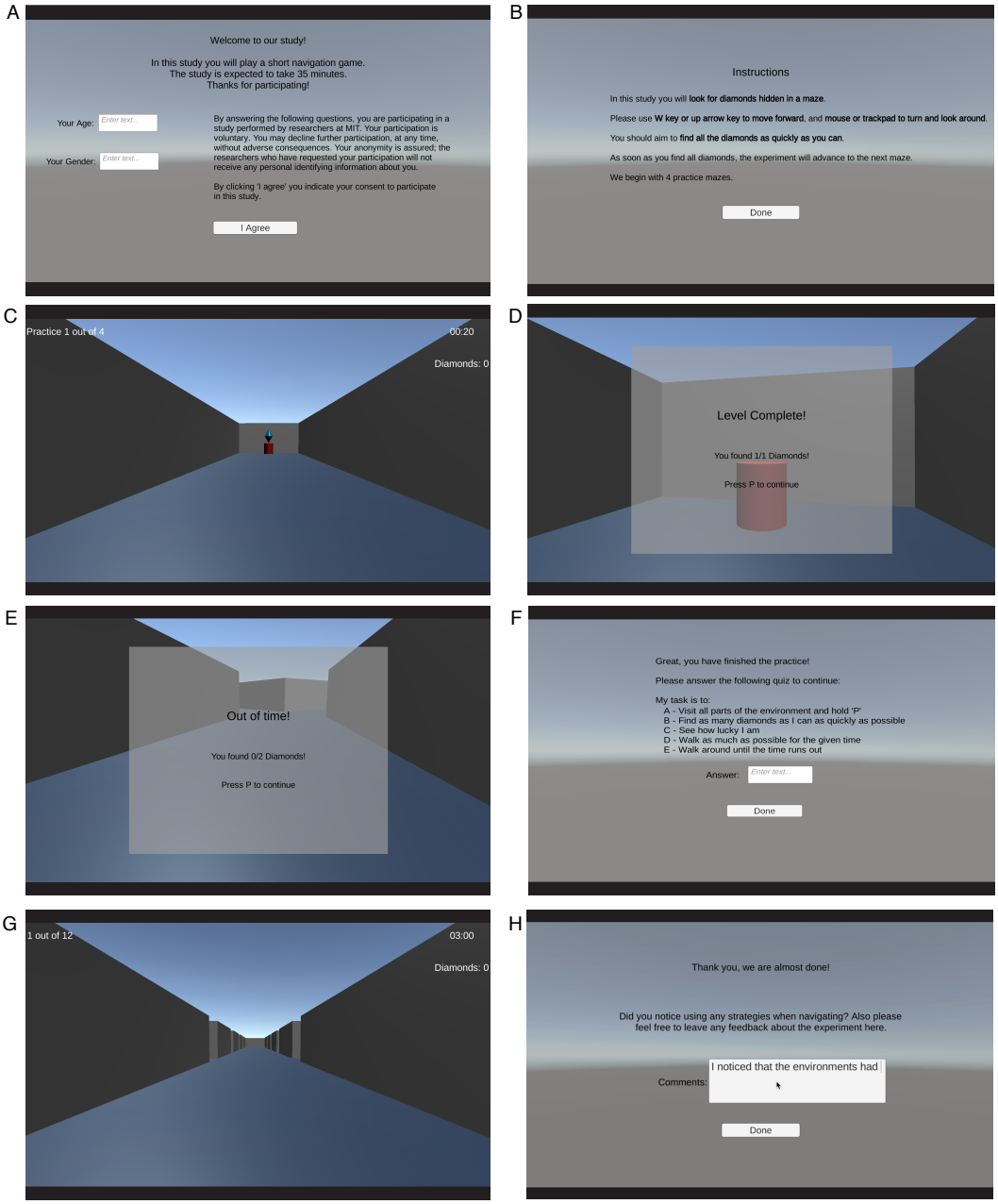

Figure 7: Screenshots from Experiment1. **A.** Welcome scene. **B.** Instructions. **C.** First practice trial. **D.** Pop-up window displayed at the end of a successful trial. **E.** Pop-up window displayed when the subject runs out of time before collecting all diamonds. **F.** Instruction Quiz. **G.** Sample first test trial. **H.** Concluding scene at the end.

A.2 EXPERIMENT 2

Figure 9 shows the stimuli used in this experiment, with practice stimuli in panel A and test stimuli in panel B. The practice stimuli have color cues at the entrance indicating the location of diamonds. Each of the test stimuli is composed of a corridor connecting five instances of a specific unit. Unlike Experiment 1, the diamonds are placed at different locations within each unit determined by the color cue in the cue room of the unit. Relative to the base test environments (Env1-Env6), the reflected versions (Env1R - Env6R) have a reflected geometry, however unlike Experiment 1, they

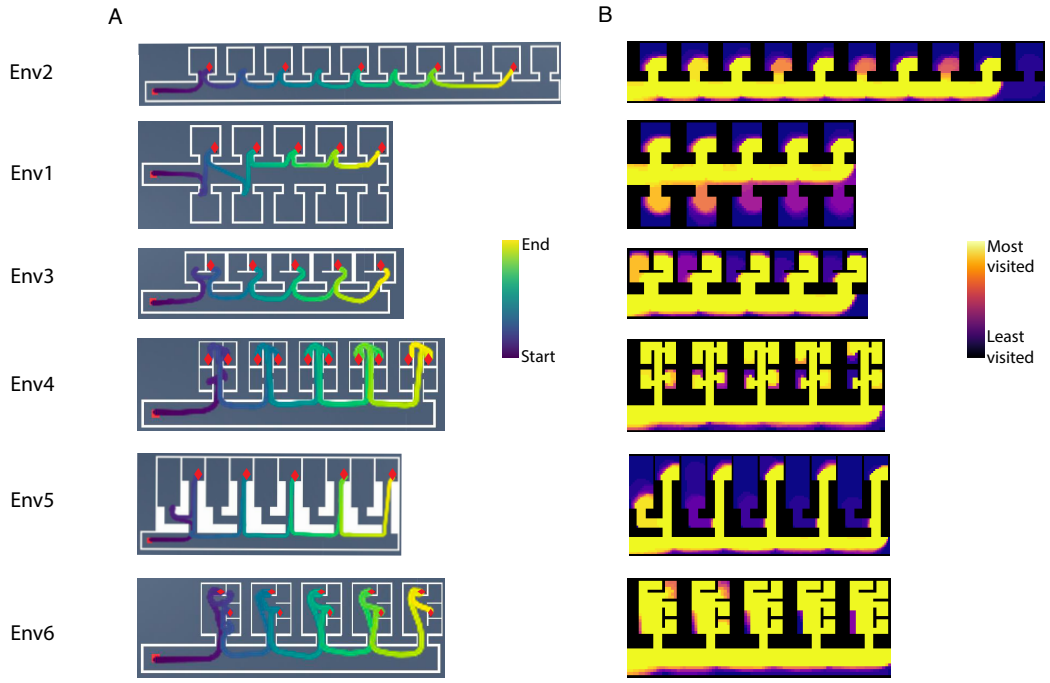

Figure 8: Trajectories and heat-maps from Experiment 1. **A**. Exploration trajectories of a representative subject. Stimuli are shown in the order presented to the subject (top - bottom). **B**. Visitation heat-maps aggregated across subjects. Each subject's visited area was computed in a 2D representation of the environment, using a circular radius of five grid cells centered at the cells traversed by the subject.

have a different reward distribution. The task was to navigate the environments and collect all the diamonds in the least amount of time. At the beginning of each trial, the subject is spawned at a fixed starting location (marked by a red floor-mat) in all environments. The subjects can then use the keyboard controls to explore the environment and find the diamonds.

Figure 10 shows the scenes displayed during the progression of the experiment. Figure 10A is the welcome scene - same as in Experiment1. Figure 10B is the next scene that provides instructions to the subjects about the task. The next scene is the instruction quiz shown in Figure 10C that ensures that subjects have read the instructions. If they get the quiz wrong, they are sent back to the instructions page, and requested to re-read them. This is followed by five practice trials in the order shown in Figure 9A. As an example, the second practice trial is shown in Figure 10D. At the end of each trial, a "Great Job" (Figure 10E) or "Out of Time" (Figure 10F) pop-up window is displayed depending on whether the subject collected all the diamonds in the environment or ran out of time before doing so. Both these pop-ups also specify the ratio of the number of diamonds collected to the total diamonds embedded in the environment. This provides feedback to the subjects about their performance in the trial before they continue to the next trial. At the end of the practice trials, subjects are presented with another instruction quiz as shown in Figure 7G, to ensure that they understand the task. If they answer the instruction quiz incorrectly, they are requested to re-read the instructions and re-do the practice trials, and are sent back to the instructions scene in Figure 7B.

If the subjects answer the instruction quiz correctly, they are shown additional instructions (Figure 10H) and another instruction quiz (Figure 10I) before moving onto the test trials. Next, the twelve test trials (see Figure 9B) are presented in succession. An example test trial is shown in Figure 10J. The test stimuli are grouped into two blocks containing six stimuli each, according to their difficulty level. For each subject, the order of stimulus presentation within blocks is randomized, while the blocks themselves are presented in a fixed order e.g., block1 (relatively simple environments - Env1, Env1R, Env3, Env3R, Env5, Env5R) is presented first, followed by block2 (high complexity - Env2, Env2R, Env4, Env4R, Env6, Env6R).

After the test stimuli, subjects are subjected to a controls skill test unbeknownst to them, to test their skills using navigation controls. This test is introduced to ensure that subjects are able to navigate in WebUnity without undue difficulty, given the larger environments, and a relatively higher cognitive load in this experiment. The test is introduced at the end rather than at the beginning to account for the deterioration of control skills due to fatigue. Subjects are shown the scene in Figure 11A that tells them that they are about to enter an environment with ten diamonds, one in every room. They are instructed to visit every room to collect all the diamonds. Figure 11B shows the controls skill test trial with the top-down view of the stimulus shown in Figure 11C. To pass the test, subjects must collect all the diamonds in this trial. The test helps disqualify two types of subjects: i) subjects who do not complete the experiment i.e., wait at the entrance without exploring the spatial environments, in the hope of getting the base payment from the study; ii) subjects who have difficulty navigating in WebUnity due to inexperience/fatigue or difficulty with controls since they may not have the appropriately functioning control devices (mouse/keyboard). Subjects who failed the controls skill test were paid, but excluded from the analysis. Figure 11D shows the trajectory of a representative subject who failed the test. After the controls skill test trial, a final scene is shown to thank the subjects for their participation, and to get their comments/feedback.

Figure 12 shows the results from the experiment on the six base test stimuli (Env1-Env6). Panel A shows the trajectories of a representative subject. The stimuli are shown in the order presented to the subject (top - bottom). Panel B shows the heat-maps indicating the least to most visited regions in each of the base test environments, averaged across subjects.

Link to the web hosted experiment: `http://18.25.132.202/experiment3.html`

A.3 Model Likelihood Comparisons

Our model likelihood analysis aims to compare the similarity of human decisions to model decisions. Specifically, we wanted to determine which models made the most human-like decisions. Because action-level similarity was too granular, we decided to compare room visitation similarity. We accomplished this by manually breaking each map into a set of convex rooms. Using this map decomposition, we obtained an ordering of visited rooms for each human on each map. For each point on the human trajectory, we provided each model with the current human initial state and history of human observations. We asked to model to find an optimal plan from this state. The result of such a query to the models is a policy tree of the branches on observations. The most significant statistic of a POMDP policy tree is visitation frequency of nodes on the tree, as this serves as a surrogate for the value of that state according to the model. We look at all trajectories in the policy that led to new rooms and summed and normalized their state visitation frequency in the policy tree. This result for a single point in the human trajectory is a distribution of rooms that the model is likely to visit given the human's experience. The model likelihood for a particular human is defined as the average probability of the human's next room in the model's visitation tree.

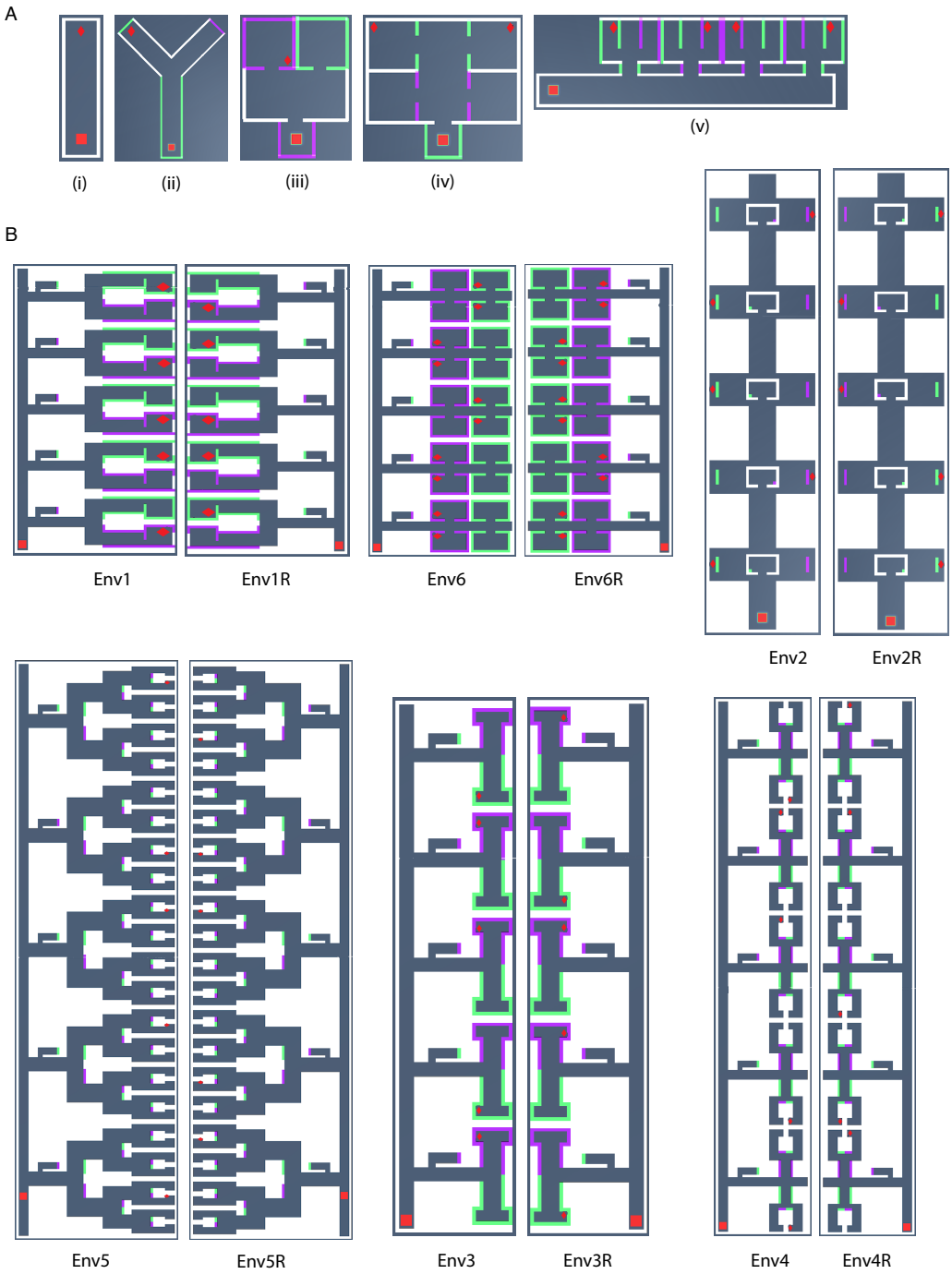

Figure 9: Top-down views of the stimuli used in Experiment 2. **A.** Practice stimuli. **B.** Test stimuli. 'R' indicates reflected. The red square is a floor mat indicating the starting location. The red diamonds are the rewards.

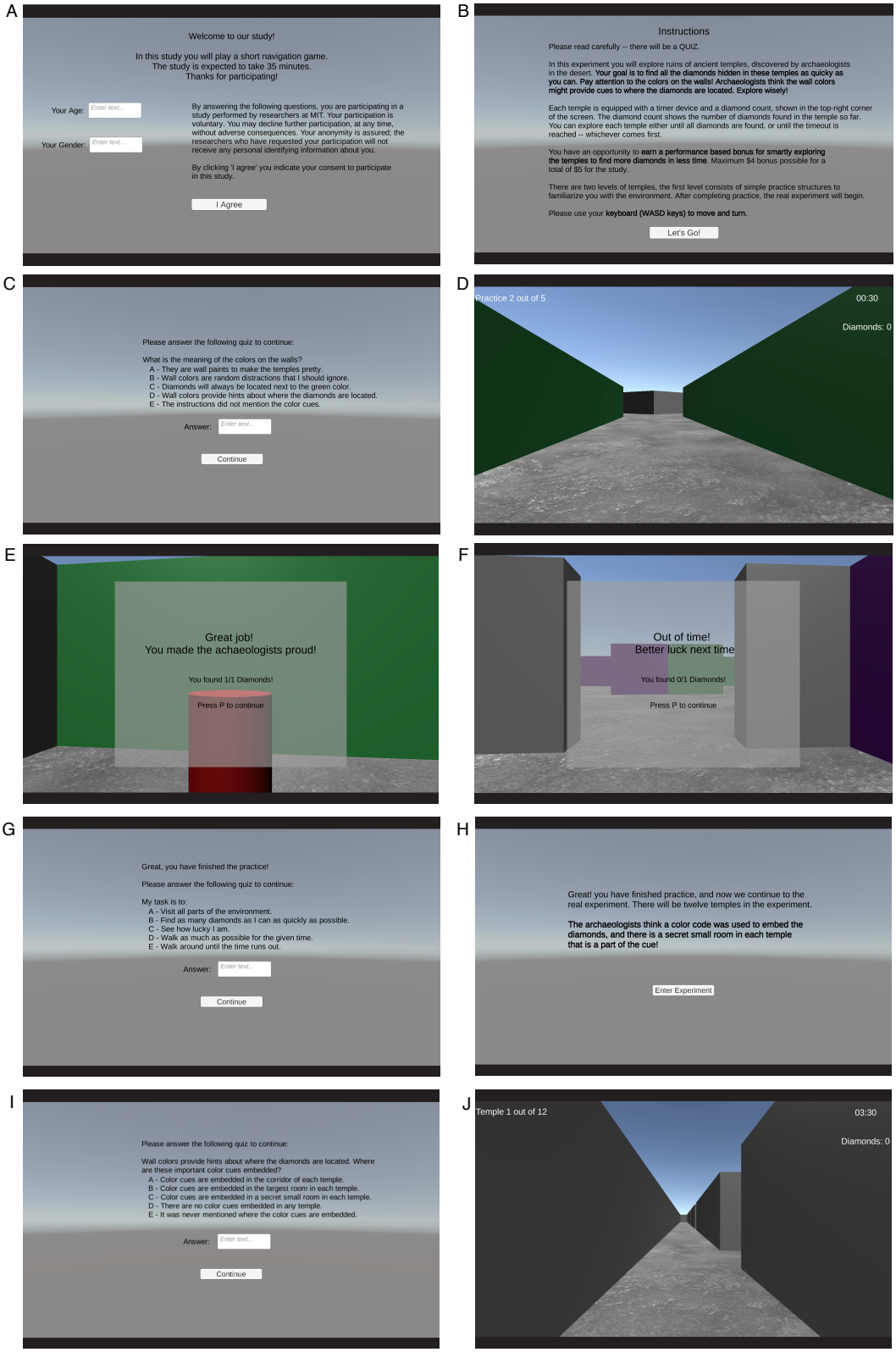

Figure 10: Screenshots from Experiment 2. **A.** Welcome scene. **B.** Instructions. **C.** Instruction Quiz 1. **D.** Second practice trial. **E.** Pop-up window displayed at the end of a successful trial. **F.** Pop-up window displayed when the subject runs out of time before collecting all diamonds. **G.** Instruction Quiz 2. **H.** Instructions for test trials. **I.** Instruction quiz 3. **J.** Sample first test trial.

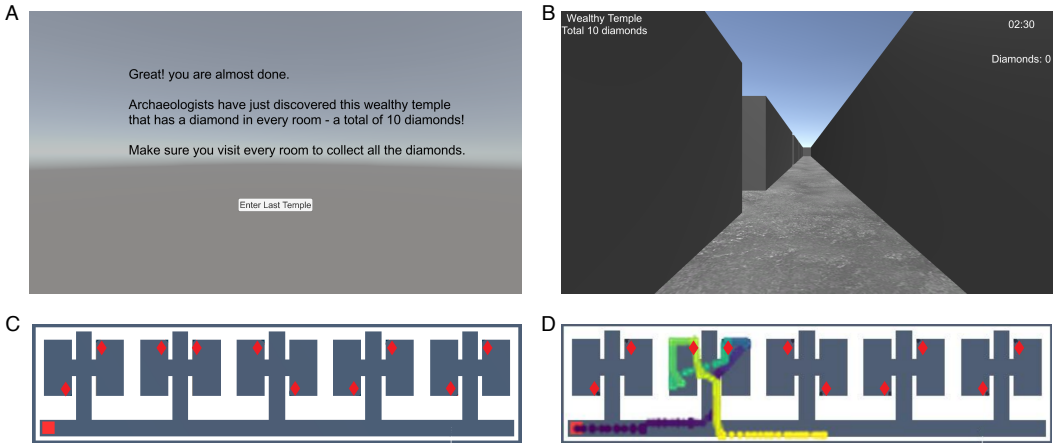

Figure 11: Controls skill test in Experiment 2. **A.** Instructions about the Controls skill trial. **B.** Controls skill trial. **C.** Top-down view of the stimulus used for the test. **D.** Exploration trajectory of a subject who failed the test.

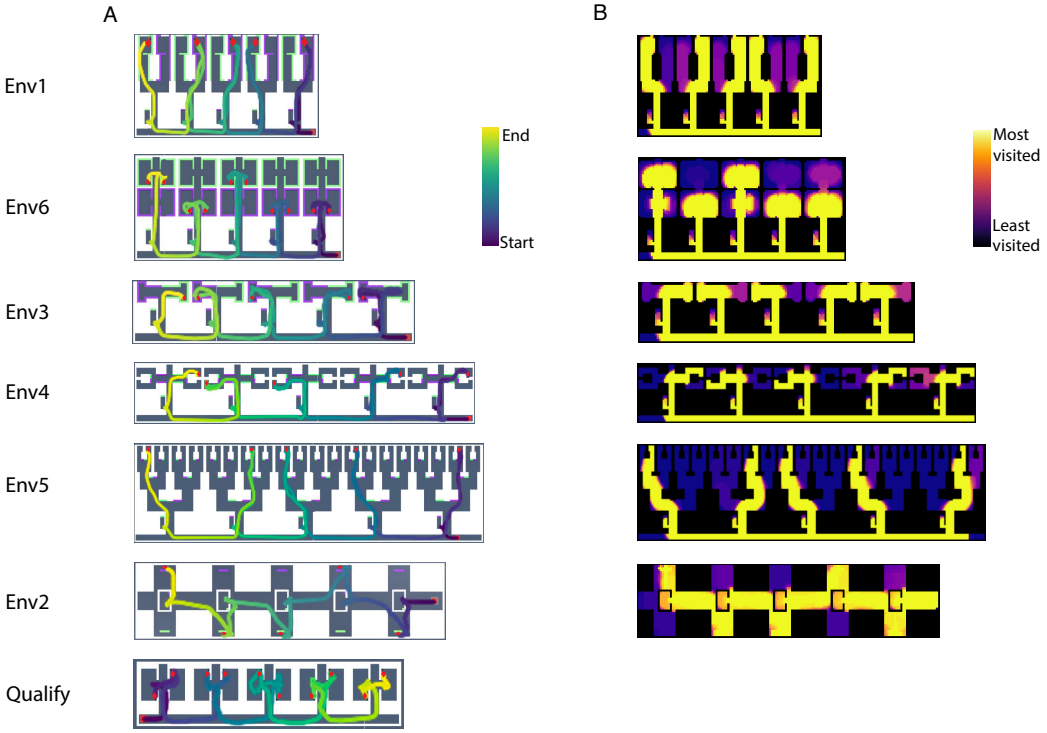

Figure 12: Trajectories and heat-maps from Experiment 2. **A**. Exploration trajectories of a single representative subject. Stimuli are shown in the order presented to the subject (top - bottom). **B**. Visitation heat-maps aggregated across subjects. Each subject's visited area was computed in a 2D representation of the environment, using a circular radius of five grid cells centered at the cells traversed by the subject.