# Cognitive maps are generative programs

**Marta Kryven (marta.kryven@dal.ca)**
Faculty of Computer Science
Dalhousie University

Cole Wyeth (cwyeth@uwaterloo.ca)
Department of Computer Science
University of Waterloo

Aidan Curtis (curtisa@mit.edu)
Computer Science and Artificial Intelligence Laboratory
Massachussetts Institute of Technology

Kevin Ellis (kellis@cornell.edu)
Department of Computer Science
Cornell University

### Abstract

Making sense of the world and acting in it relies on building simplified mental representations that abstract away aspects of reality. This principle of cognitive mapping is universal to agents with limited resources. Living organisms, people, and algorithms all face the problem of forming functional representations of their world under various computing constraints. In this work, we explore the hypothesis that human resource-efficient planning may arise from representing the world as predictably structured. Building on the metaphor of concepts as programs, we propose that cognitive maps can take the form of generative programs that exploit predictability and redundancy, in contrast to directly encoding spatial layouts. We use a behavioral experiment to show that people who navigate in structured spaces rely on modular planning strategies that align with programmatic map representations. We describe a computational model that predicts human behavior in a variety of structured scenarios. This model infers a small distribution over possible programmatic cognitive maps conditioned on human prior knowledge of the world, and uses this distribution to generate resource-efficient plans. Our models leverages a Large Language Model as an embedding of human priors, implicitly learned through training on a vast corpus of human data. Our model demonstrates improved computational efficiency, requires drastically less memory, and outperforms unstructured planning algorithms with cognitive constraints at predicting human behavior, suggesting that human planning strategies rely on programmatic cognitive maps.

**Keywords:** navigation, planning, cognitive maps, computational modeling, large language models.

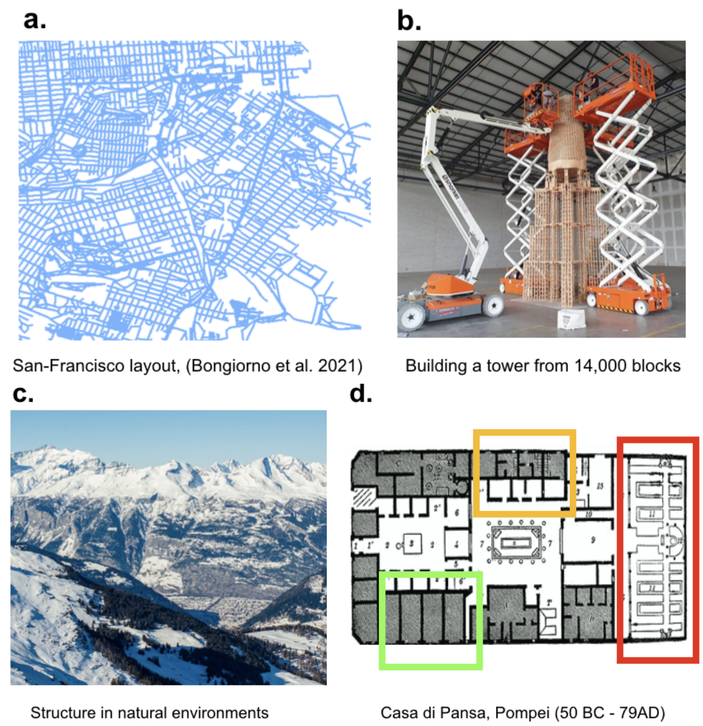


Figure 1: Real-world planning tends to occur in predictably structured domains, such as street networks (a), modular architecture (b,d), and naturally patterned landscapes (c).

## Introduction

Current AI formalizes planning as a search within a decision tree of possible actions and outcomes. This tree can be encoded in various ways, such as a learned neural policy (Liu et al., 2020), an explicit tree structure (Russell & Norvig, 2016), or a neuro-symbolic model (Tang et al., 2024). The size of the underlying decision tree determines the computational cost of the problem, or how difficult it should be. However, the normative difficulty of such models rarely aligns with human experience, as people often solve real-world problems that are theoretically intractable with relative ease. For example, people efficiently navigate cities without knowing exact location of every link in the street network (Fig. 1a) (Bongiorno et al., 2021), and plan construction projects with thousands of actions (Fig. 1b).

This excellent efficiency in real-life planning stands in contrast with often inefficient performance in the laboratory. People deviate from optimal plans in a variety of contexts, including tasks based on multi-arm bandits (Huys et al., 2015; Keramati et al., 2016), games, such as chess (Ferreira, 2013), and many experimental paradigms designed to examine planning behaviors (Callaway et al., 2022; Kryven et al., 2024; Unterrainer et al., 2004). Such deviations from optimality are commonly explained by limiting planning horizon (Ferreira, 2013; Kryven et al., 2024; van Opheusden et al., 2023) and low-level constraints on perception (Kryven et al., 2024). However, most laboratory-based tasks conspicuously lack the predictable and regular problem structure ubiquitous in real-life, such as hills and valleys in nature (Fig.1c) and modular built layouts (Fig.1d). Moreover, classic experimental



In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society* 

paradigms often explicitly avoid creating structure in an effort to isolate variables of interest, such as planning depth.

Here we test the hypothesis that human planning leverages prior knowledge about the world, particularly the expectation that the world is highly structured. Given this, cognitive maps may be represented as *programs*, because programs can capture structures such as symmetries and repeated parts.

Program-structured maps address a key computational problem: Planning in partially-observed environments is intractable (Madani et al., 2003). Approximations are needed. Program-structured maps help tackle hard planning problems because programs that repeatedly generate identical map fragments allow recycling successful policies, essentially decomposing the decision problem. This gives plans that are locally optimal (within code fragments), but globally suboptimal. The key idea is to discover fragments of repeated structure in a partially-observed map, and plan policies only once for each repeated fragment. Then, when each fragment is encountered, we avoid costly belief-space planning by reusing previously computed policies. We probe the extent to which humans similarly recycle successful policies across repeated code fragments, finding that their plans align with this signature of suboptimality. We show that human planning in structured environments is predicted by our generative map model, and can not be explained by alternative models based on unstructured planning with cognitive constrains.

## Background

From young children to hunter-gatherers, people impose structure on the world to solve problems (Lake & Piantadosi, 2020; Pitt et al., 2021). For example, people spontaneously infer correlation between rewards in spatially adjacent locations (Schulz et al., 2018), and represent spaces hierarchically – as divided by visible boundaries (Kosslyn et al., 1974), geography (Stevens & Coupe, 1978), or sub-regions (Hirtle & Jonides, 1985). Behavioral evidence for compositional problem representation extends to auditory (Verhoef et al., 2014), visual (Tian et al., 2020), and abstract concept domains (Schulz et al., 2017) – suggesting that the compositional reasoning may be an evolved adaptation to natural structures encountered in daily life (Johnston et al., 2022).

A recent study found that people form cognitive maps that facilitate planning (Ho et al., 2022), selectively representing only the parts of the map relevant to goal-directed routes. Cognitive maps may also depend on prior expectations about the world, leading people to anticipate regularities in new environments, even when not informed about them in advance (Sharma et al., 2022). In naturalistic setting adaptive planning draws on complex conceptual prior knowledge of the world, including knowledge of how agents and objects interact, often referred to as *core knowledge* (Acquaviva et al., 2022; Dehaene et al., 2006; Spelke & Kinzler, 2007). Learning these natural priors remains an important problem in cognitive AI (Binz et al., 2024; Kumar et al., 2022; Li et al., 2024).

Parallel developments in computing have long tackled the goal of building intelligent systems capable of solving general, procedural tasks (Chollet et al., 2025; Veness et al., 2011). Reinforcement Learning (RL) models abstract problem representations by expressing actions performed together as *options* (Sutton et al., 1999), learning families of similar Markov Decision Process (MDP) with shared rewards (Wilson et al., 2012), and building efficient state-spaces by recognizing actions that lead to identical observations (Singh et al., 2012). Generalized planning frameworks can find algorithm-like policies for solving multiple instances of a task (Curtis et al., 2022), although their ability to handle uncertainty is still limited. A principle of grouping game-board states based on rotation and reflection symmetries was used to optimize representations in the game of Go (Silver et al., 2017), although this work does not consider automatic discovery of structured representations.

The central challenge to both modeling cognition and engineering general intelligence is therefore learning and leveraging natural priors (Feldman, 2013). How can AI learn core knowledge and common sense expectations that make people so efficient in real-life? An emerging line of research leverages LLMs, to make plans in engineering (Tang et al., 2024) and behavioral domains (Correa et al., 2023). While the applications of LLMs to directly generating plans are limited, studies have successfully used LLM's program synthesis abilities, to generate programmatic representations of transition function for planning in the OpenAI Gym domain (Tang et al., 2024; Towers et al., 2024) and to synthesize Planning Domain Definition Language (PDDL) specifications (Xie et al., 2023). Our computational framework uses a similar approach on the assumption that by doing so, we can implicitly access human prior knowledge of the world embedded in language and code used in LLM training.

## Methods

### Behavioral Experiment

To test people's natural planning strategies, we adapted a version of Maze Search Task (MST) previously used to evaluate computational models of human planning and plan perception in spatial domains (Kryven et al., 2021, 2024). The objective of MST is to navigate a series of partially observable, two-dimensional grid-world mazes, finding exits hidden in each maze. As the mazes are partially observable, the exits are initially placed at a random unobserved location within a maze. Fig.2b shows a sequence of views seen by a participant in MST[1]. People search the mazes by clicking on any unoccupied grid cells adjacent to their character (a round face icon) to move between adjacent cells. The black hidden cells are revealed when they come into the avatar's line of sight. When revealed, the exit becomes visible as a red tile. As soon as the character moves over the exit, the trial ends.

[1]We encourage the reader to try the experiment here: http://18.25.132.241/fragments/int_exp.php

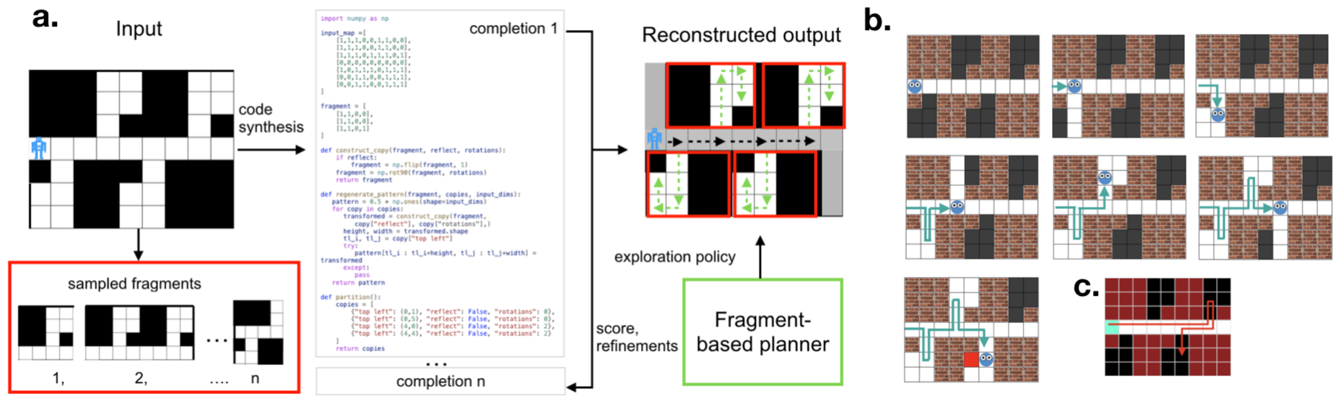


Figure 2: (a) The GMP framework. Generative map representations recovered from the input map constitute a list of fragments, and a program for approximately reconstructing the input from them. The fragments are used by the Planning module to plan a policy for searching each fragment. (b) A series of screens seen by a person in the experiment. Black tiles are initially hidden, and are revealed as they are brought into line of sight. The exit becomes visible in the last screen, shown as a red tile. Arrows indicate the participant's path, searching mazes in a modular way (unit by unit). (c) The optimal path that minimizes the steps needed to reach a randomly placed exit in this maze is non-modular.

**Procedure** Before beginning the experiment people gave informed consent and completed a series of practice mazes, followed by an instruction quiz. Following this, they completed a version of MST with 21 generatively structured mazes, with exit locations randomly chosen at the time of design. In one maze the exit was randomly placed in plain sight, and it was excluded from the analysis. After completing MST people completed a Cognitive Reflection Test (CRT) (Frederick, 2005), previously shown to correlate with allocating cognitive resources to planning (Kryven et al., 2021). Given that the original CRT was used extensively, our version of CRT provides an analogous set of problems in a novel context (Chandler et al., 2014). Lastly, we administered a post-experiment questionnaire in which people were asked to describe any search strategies they used, and provided demographic information. As our goal was to observe people's natural planning, we did not offer performance-based incentives. People were informed that the exit could be in any of the hidden tiles, and instructed to find it in each maze.

**Participants** We recruited 30 (13 female, 17 male, $M(age) = 36,7$, $SD(age) = 13.5$) english-speaking participants on Prolific, who were paid 9£per hour. None were excluded. On average the experiment took 10 minutes to complete. A preliminary pilot experiment revealed a strong effect of modularity, leading us to conclude that a small sample is sufficient to confirm this effect.

## Computational Model

**Problem Formulation** Decision making under partial observability can be modeled by a partially observable Markov decision process (POMDP). Equivalently, it can be viewed as a fully observable search through a space of beliefs, where each belief is a probability distribution over possible states. Solving POMDPs is notoriously hard (Madani et al., 2003), hence understanding how people approach these problems holds deep importance for cognitive science and AI.

Formally, a POMDP is a tuple $\langle \Delta(S), A, \tau, r, b_0, \gamma \rangle$, where $\Delta(S)$ is the space of probability distributions over a state space $S$, $A$ is the set of actions, $\tau$ is the belief update function, $r$ is the reward function, $b_0$ is the initial belief, and $\gamma$ is the discount factor. The belief state evolves deterministically via $\tau$, reflecting both the agent's actions and observations.

In this work, each state $s \in S$ is represented as an $N \times M$ grid whose cells are labeled $\{\text{wall}, \text{empty}, \text{exit}, \text{agent}\}$. The overall state space $S$ consists of all such grids containing exactly one agent and one exit. A belief $b \in \Delta(S)$ is thus a probability distribution over these grids, encoding the agent's uncertainty about the true state. Initially, $b_0$ assumes that the agent and the walls are known, while the exit is uniformly distributed over all valid, unseen cells. The action space $A$ contains four possible movements (up, down, left, right). Observations $o \in O$ reveal the visible subset of the grid around the agent, with each visible cell labeled $\{\text{wall}, \text{empty}, \text{exit}\}$, and any cell outside the agent's visibility range $r$ labeled as *unseen*. Observations are consistent with the grid structure of the true state $s \in S$.

The belief update function $\tau$ is given by

$$b'(s') \propto Z(o \mid s') \sum_{s \in S} T(s', a, s)\, b(s),$$

where $T(s', a, s)$ is the transition function, and $Z(o \mid s')$ is the observation likelihood. The transition function $T(s', a, s)$ specifies the probability of transitioning to state $s'$ from $s$ after executing action $a$. Here, actions that would move the agent into a wall result in the agent remaining in its current position, and transitions to an exit state terminate the process. The observation function $Z(o \mid s')$ encodes the likelihood of observing $o$ given $s'$, where observations reflect the visible subset of the grid within range $r$ of the agent's position. Visibility is blocked by walls, such that cells beyond a wall are labeled as *unseen*. Finally, the reward function $r(b,a)$ is the expected reward under the belief $b$. Since the agent can always see an exit before reaching it, $r(b,a) = 1$ if action $a$ leads the agent to a known exit and 0 otherwise.

**Expected Utility** The optimal policy for a this POMDP can be found through a belief space tree search (Kaelbling et al., 1998). The search is conducted over a tree where each node represents a belief $b \in \Delta(S)$, and edges correspond to action-observation pairs $(a,o)$. Starting from the root node $b_0$, the tree expands by simulating actions $a \in A$ and updating beliefs using the belief update function $\tau$. For each action $a$, the agent considers all possible observations $o \in O$, with the likelihood of each observation determined by the observation function $Z(o \mid s')$. At each node, the value of a belief is computed recursively using the Bellman equation:

$$V(b) = \max_{a \in A} \left[ r(b,a) + \gamma \sum_{o \in O} P(o \mid b,a) V(\tau(b,a,o)) \right], \quad (1)$$

where $P(o \mid b,a)$ is the probability of receiving observation $o$ after taking action $a$ under belief $b$. The optimal policy $\pi^*$ is derived by selecting the action at each belief node that maximizes the expected value. See (Kryven et al., 2024) for further details on this implementation.

Although this is the optimal strategy, human behavior has previously been shown to diverge at times from its predictions (Kryven et al., 2024), where the extent of this divergence varies between individuals in a way that can be explained by the amount of cognitive resources people allocate to planning (Kryven et al., 2021). Previous work with MST, as well as with related non-spatial planning tasks (Huys et al., 2015), has found that people's divergence from the optimal trajectories is most readily explained by a limited planning horizon ( discount factor $\gamma < 1$ in Equation 1). In the the remainder of this section we describe alternative computational hypotheses for how humans could make decisions in this environment by reasoning about structural patterns.

**Generative Modular Planning (GMP)** We first describe a model that formalizes planning strategies conditioned on automatically discovered latent structure of the state-space. Our model consists of two modules: a Generative Map Module (GMM) and Fragment-based Planning (FP) module. See Fig.2a for a high-level overview of this architecture. The GMM recovers a programmatic representation of the observed state-space, as a composition of fragment units. The FP then uses a planner to plan a piece-wise policy once per-fragment, in contrast to a global policy, saving computing costs. Importantly, this reconstructed programmatic representation is a cognitively-inspired state-space compression. While such a reconstruction may match the ground-truth planning state-space, it does not need to be exact. In theory, the cognitive principle of combining automatic structure discovery with structure-aware planners can apply to any domain, as a proof of concept here we focus on spatial tasks.

**Algorithm 1** Generative Program from 2D Array

**Require:** $I$: Input map, $t$: Threshold, $C$: Number of completions
**Ensure:** $\lambda$ generative program, fragments

$S' \leftarrow 0$
Initialize fragments $\leftarrow$ []
$\lambda \leftarrow$ ""
**while** $S' < t$ **do**
  Generate a prompt from $I$ and fragments
  Send the prompt and receive $C$ completions
  Extract programs $\{\lambda_1, \lambda_2, \ldots, \lambda_C\}$
  **for all** $\lambda_i \in \{\lambda_1, \lambda_2, \ldots, \lambda_C\}$ **do**
    **if** $\lambda_i$ runs successfully **then**
      $S_i \leftarrow S(\lambda_i)$
    **end if**
  **end for**
  $S', i \leftarrow \max(S_i)$
  $\lambda \leftarrow \lambda_i$ with the highest score
  fragments $\leftarrow$ fragments$_i$
**end while**
**return** $\lambda$, fragments

**Generative Map Module** We use LLM-based program synthesis with GPT4 to search for programs that can generate cognitive maps based on input (see Algorithm 1). To do this, we prompt LLM to identify potentially repeating fragments in the input map, and synthesize a Python program that approximately reconstructs the input based on these fragments. The prompt includes Python code with helper functions for admissible fragment transformations, as well as scoring function by which the maps will be evaluated (described below). In our implementation the input map is a grid-world, specified by an numerical array in Python, where each grid cell is associated with a number (e.g. wall=1, floor=0). The result of this operation is a small number of possible programs that approximately reconstruct the map. This reconstruction allows parts of the output to be left undefined.

We score each candidate map completion by a weighted combination of grid-level similarity and the Minimum Description Length (MDL) principle ((Rissanen, 1978)), to produce a ranking of reconstructed maps. The MDL method penalizes fragments by size as both the shape and each entry must be specified. MDL also penalizes each occurrence used to reconstruct the map by the bits necessary to specify the fragment to map transformation: the translation (as position of top left corner), rotations, and reflection. This means that uninformative fragments (for instance, uniform blocks of cells containing a wall or open space) do not form a parsimonious explanation for the map, even though they of course occur many times. This can be viewed as enforcing a preference for simple generative programs, with the addi-

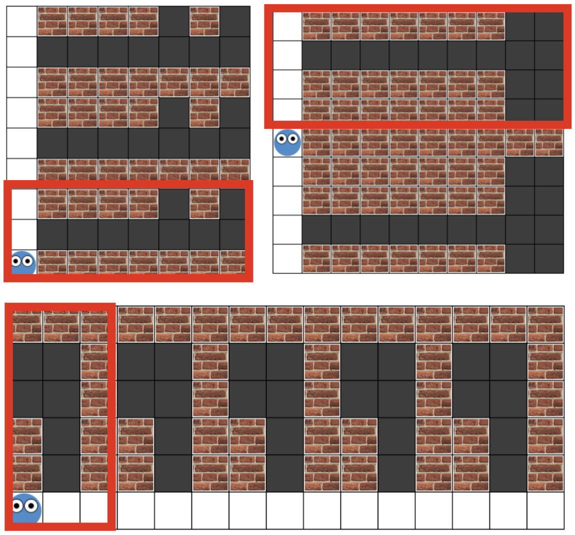

Figure 3: Examples of maps used in the experiment with 3, 2, and 5 structural fragments highlighted.

tional structural requirement that *the generative programs use primitives, in this case transformations, encoding symmetries which are relevant to planning.* The highest-ranking map program is then used by the planning module to generate a policy. Formally, the score is

$$S = \frac{w_1}{M \cdot N} \sum_{x=1}^{M} \sum_{y=1}^{N} \left(I(x,y) - O(x,y)\right)^2 - w_2 |\lambda| \tag{2}$$

Here $N, M$ are map dimensions, $I$ is the input, $O$ is the output (reconstructed) map, $\lambda$ is the program, and $w_1, w_2$ are weights, free parameters of the model. In the general case, input $I$ and output $O$ are real-valued 2D image arrays. In our current implementation input $I$ takes values 0 and 1, and the output $0 \leq O(x,y) \leq 1$.

Instead of using the raw Python program for $\lambda$ to measure of its complexity, we use a compressed encoding of the fragments, and the transformations used to reconstruct the map. Here compressing LLM-generated fragments and transformations is analogous to refactoring the synthesized programs. As the length of LLM-synthesized code may be noisy, due to injected comments an code redundancies, we refactoring the output obtains a denoised metric of complexity. For example, just two bits are necessary to specify a copy's rotation and one bit necessary to specify whether it is reflected.

Fig.3 shows examples of three maps used in the experiment with 4, 2 and 5 structural fragments, shown highlighted in red.

**Fragment-based Planning Module** In the current implementation we adapt the optimal Expected Utility model (an implementation of $\pi^*$) from (Kryven et al., 2024) to plan within fragments. The implication of this modeling choice is to assume that people are locally optimal but globally suboptimal, as is a natural consequence of a model that can exactly solve small problems and then reuse them. To plan between fragments (in parts of the output left undefined) FP moves toward the closest fragment by locally solving a Markov Decision Process (using value iteration), on the assumption that the prior probability of finding the exit is uniform across fragments. When such priors are non-uniform, the problem of planning between fragments can be solved by any planning algorithm.

This approach is resource-efficient, as our planner computes a decision (sub)tree only once per fragment, and subsequently reuses it every time the fragment is encountered. Instead of computing a decision-tree for the entire map, our model maintains several smaller decision-trees, along with a generative program that describes how the global map can be reconstructed from them. In principle, as the map reconstruction is approximate the planner may encounter observations inconsistent with the reconstructed map. When this happens, revert to non-modular planning (e.g. using the Expected Utility planning (Kryven et al., 2024)) to ensure robustness of the algorithm. While our choice of planner was motivated by comparing results to prior work, our framework does not critically depend on the choice of the internal non-modular planner, and can integrate with other implementations.

**Alternative models** We compare GMP to an existing set of models previously used to explain human behavior in MST (Kryven et al., 2024). The models treat maze search as a path minimization problem - an implicit goal spontaneously reported by participants in the post-experiment questionnaire. Here we compared people to a subset of models from pervious work – the Expected Utility (optimal planner) and the Discounted Utility (an overall best-fitting planner, assuming $\gamma = 0.7$ which plans with a limited horizon)(Kryven et al., 2024). To facilitate this comparison, we designed the environments so that these existing models either predict non-modular policies, or are indifferent between modular and non-modular search. Whenever an existing model is indifferent between modular and non-modular paths in an environment, the probability of such a model taking a modular path is at most 0.25.

## Results

We introduce the following metrics for comparing behavior to our models. First, we define a conservative definition of a **modular path** as a path that visits all fragments in order, always moving to the closest subsequent fragment. This definition of modularity is consistent with our implementation of GMP, however, it underestimates the true rates of modularity since people who represent a map as modular may have non-uniform prior beliefs about which fragments are 'rewarding'. For example, people could assume that the exit tends to be in the second or third fragment, skipping some parts of the map. We use this definition to compute **modularity** – that is, the fraction modular paths for each individual (Fig.4a) and for each environment (Fig.4b). Second, we define as a **discriminating decision** as any state of the environment in which GMM and an alternative model predict a different

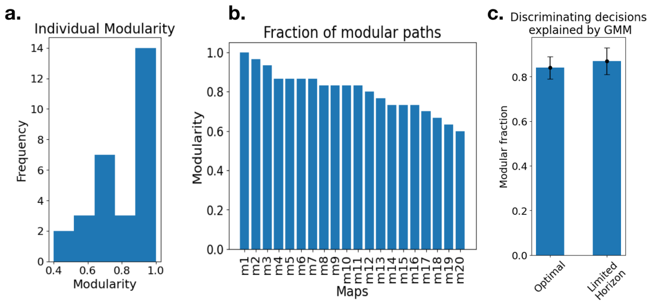


Figure 4: People are highly modular, according to the conservative definition. (a) Histogram of modularity across participants. (b) Fraction of people who searched using modular trajectories, for each of the 20 maps. (c) Aggregating across individual decisions and people. Each bar shows the fraction of discriminating decisions per participant where a participant's choice was predicted by our model, but and not by the alternative model. Error bars are 95% CI over people.

most likely path. As different models can predict different paths, the set of discriminating decisions will different for any given pair of models, given the same set of environments. By measuring how well the alternative models predict people in discriminating decisions, we account for the possibility that people could be using several planning strategies within one environment, including switching between Modular and non-modular strategies.

Examination of individual and map-specific modularities in Figures 4a and 4b shows that people are highly consistent with our model, demonstrating signatures of modular planning across all environments and participants. In contrast, by our experiment design the alternative models predict that the mazes should be searched in a non-modular way. Further, GMM predicts people significantly better than the optimal planner, or planning with limited horizon perviously shown explain human behavior in MST (Fig.4).

To test the hypothesis that modular planning is a resource-efficient adaptation for reducing cognitive costs, we compute a linear regression of people's CRT score against their modularity rate. While we found a negative correlation between these scores during pilots, it was not significant in the current results.This lack of replication is likely due to our conservative definition of modularity underestimating the true rate whenever people searching fragments in a different order.

## Discussion

We test the hypothesis that people may be representing cognitive maps by generative programs, conditioned on conceptual prior knowledge of the world. We leverage an experimental task previously used to study human planning, with partially observable environment layouts designed to differentiate between Modular planning, planning that follows optimally computed Expected Utility, and the model previously shown to describe planning by a limited horizon. We found that people follow highly modular strategies, that generalize across environments and individuals in support of our hypothesis. We interpret our results by a computational model that combines LLM program synthesis for map generation with modular planning, showing that this model is highly accurate at predicting behavior. This result suggests that people deviate from the optimal plans in structured environments at least in part due to reasoning about the environmental structure.

We makes three contributions in the scientific and engineering domains. First, we make a scientific contribution, showing that people make adaptive plans consistent with program-like representations of cognitive maps. Second, we advance the understanding of latent cognitive abilities of LLMs by showing how to elicit human inductive biases about spatial decision-making. Third, we contribute an implementation showing how to actually build these principles into a working system. Our GMP model differs from existing hierarchical models (which divide a complex problem into manageable chunks), as it can additionally reuse computations across repeating fragments. It also differs from existing generalizable task-and-motion models specified in PDDL, as it tolerates uncertainty. By using LLMs for program-synthesis, our work is more practical and scalable than the traditional enumerative search (brute-force) methods (Sharma et al., 2022; Veness et al., 2011) for discovery of structural form.

Notably, we find variability in behavior between individuals and between maps that is not accounted for by our conservative definition of Modular planning. Much of this variability arises from people skipping fragments, suggesting that future versions of the model should include non-uniform priors over which fragments people believe to be rewarding. Future studied should also consider how latent cognitive factors, such as learning, attention, and available cognitive resources, may affect how people modularize and represent maps.

As map reconstructions are approximate, the planner may encounter observations inconsistent with the reconstructed map. We currently handle such scenarios by reverting to a non-modular planner, however the alignment of this solution with people remains to be evaluated experimentally. More studies are needed to understand how people process discrepancies between their cognitive map and reality, when their expectations fail. Future studies should also examine the extent of variation between fragments perceived as instances of the same class in the light of possible goals, as cognitive maps are likely to be goal-dependent (Ho et al., 2022).

While planning cognition has been studied extensively, the domain of real-world planning remains underexplored. Our work contributes a formal computational insight into how human prior knowledge may guide resource-efficient plans, as well as contributes to the emerging investigation of whether LLMs can capture naturalistic inductive biases (e.g., (Tang et al., 2024)), offering insights into this question in the spatial domain. By expressing our findings in computational terms, our GMP model moves toward translating cognitive mechanisms behind human planning into AI applications.