Planning and Explanations with a Learned Spatial Model

Susan L. Epstein
Department of Computer Science, Hunter College and The Graduate Center of The City University of New York, New York, New York 10065, USA
susan.epstein@hunter.cuny.edu

Raj Korpan
Department of Computer Science, The Graduate Center of The City University of New York, New York, New York 10016, USA
rkorpan@gradcenter.cuny.edu

Abstract

This paper reports on a robot controller that learns and applies a cognitively-based spatial model as it travels in challenging, real-world indoor spaces. The model not only describes indoor space, but also supports robust, model-based planning. Together with the spatial model, the controller's reasoning framework allows it to explain and defend its decisions in accessible natural language. The novel contributions of this paper are an enhanced cognitive spatial model that facilitates successful reasoning and planning, and the ability to explain navigation choices for a complex environment. Empirical evidence is provided by simulation of a commercial robot in a large, complex, realistic world.

2012 ACM Subject Classification Computing methodologies → Artificial intelligence; Computing methodologies → Machine learning; Computing methodologies → Modeling and simulation

Keywords and phrases navigation, planning, learning, explanation, spatial model, heuristics

Digital Object Identifier 10.4230/LIPIcs.COSIT.2019.7

Funding This work was supported in part by The National Science Foundation under # CNS-1625843.

Acknowledgements The authors thank Anoop Aroor for his thoughtful comments, Gil Dekel and Sarah Mathew for their pioneering work on doors and hallways, and the anonymous referees for their many constructive suggestions.

1 Introduction

As robots that navigate autonomously among people become increasingly prevalent, the software that controls them must address substantive issues for their control. This paper focuses on effective planning, natural communication, and adaptation to a new environment based on the robot's experience there. The thesis of our work is that a learned cognitive spatial model based on spatial affordances can support robot navigation in built environments (henceforward, *worlds*). In previous work, we demonstrated how a small, inexpensive robot could learn such a model to approximate simple worlds [13]. This paper tackles considerably more challenging worlds, for which it extends the model, plans from it, and uses it to formulate natural explanations of the robot's navigation behavior. The principal results of this paper are that the enhanced spatial model can mitigate the impact of discretization, and that planning with it supports effective navigation, transparent reasoning, and human-friendly communication.

A robot's *controller* is its decision-making software. The context of our work is *SemaFORR*, a robot controller that relies on only its travel experience and a laser range finder to learn a cognitive spatial model for a new world. The next section of this paper provides necessary background and related work. Subsequent sections describe SemaFORR's improved spatial model and how that model supports both planning and Why, SemaFORR's question answerer for its behavior and intentions. The paper then describes and discusses empirical results that support our thesis in a realistic, large-scale world.

2 Background and Related Work

A *robot* is an embodied, artificial, mobile agent whose behavior is produced by a sense-decide-act loop. *ROS* is the state-of-the-art robot operating system [33]. We have written both SemaFORR and Why as ROS modules. Although robots perceive continuous space and their hardware allows a broad range of possible actions, most robot controllers, including SemaFORR, discretize both space and their action set to make computation tractable.

To plan for and communicate about navigation, a robot must both represent and reason about space. Because such communication is simpler when the robot's spatial representation and reasoning are human-like, the ways people represent and reason about space are important. Rather than find common ground between the robot's egocentric perspective of its world and that of a person whose perspective is unknown, we assume an allocentric (i.e., with respect to some external fixed point) perspective.

2.1 Spatial Models and Reasoning for Humans

To represent space, a person learns a *cognitive spatial model*, a mental representation of her world. The earliest clinical evidence for this was from studies of the behavior of rats in mazes [39]. Scientists have since identified neurons in the rat's brain that suggest the existence of a Euclidean spatial model [18]. Because these neurons fire sequentially during sleep or at rest without visual input, scientists hypothesize that the rat uses them to represent space, to learn, and to plan [4]. The wide range of structures and content proposed for human cognitive spatial models includes single designated paths, graphs that record connectivity, *labeled graphs* (with metrics for distances and angles), and *surveys* (precise, allocentric metric maps) [40, 28]. Although it has been suggested that cognitive spatial models use metric distances and angles [16], more recent work indicates that cognitive maps have a non-metric, qualitative topological structure [14]. Other recent work suggests that the model a person learns is not a survey but a labeled graph [7, 8, 41]. The spatial models SemaFORR learns are most similar to labeled graphs where the relative size and position of features are recorded but precise dimensions are not. The exact nature of human cognitive maps, however, remains an important open problem in spatial cognition [42].

To reason about space for navigation, people use a variety of well-documented approaches: reactivity [36], planning [17], and satisficing heuristics [10]. These heuristics are "good enough" rules for decision making in any world, typically triggered by either percepts or an internal signal. Despite pedestrians' individual experiences and physiology, striking regularities appear in the ways that they formulate navigation heuristics [44]. These regularities include how people understand distance and direction, perceive proximity as dependent on context, and view direction as closely related to geometry. SemaFORR's reasoning mechanism incorporates many such heuristics, as well as reactivity and planning.

2.2 Spatial Models and Reasoning for Robots

To represent space for robot navigation, a controller requires some model of the robot's world. If that model must be a survey and one is not provided, the robot can methodically travel its world to create one with the state-of-the-art algorithm SLAM (Simultaneous Localization and Mapping) [30]. *Localization*, the ability to know where one is in the world, is a key challenge, because different locations may provide similar percepts (e.g., when one faces into a corner). Moreover, robots are subject to both sensor error (percepts that provide a noisy version of the ground truth) and actuator error (imprecise command execution). As the robot travels, SLAM localizes while it builds a survey of the obstructions it detects. SLAM is probabilistic, that is, it provides only likelihoods for the robot's location. In addition, a robot that relies on a SLAM-generated metric map must contend with both the errors present during the map's construction and the errors in the robot's current localization and sensing. More recently, to facilitate communication about navigation, semantic mapping has been used to represent space and apply qualitative labels to the environment [22].

Other work on spatial representations for robot navigation has emphasized hierarchical aspects. Prototype, Location, and Associative Networks (*PLAN*) represented a cognitive map with a hierarchical structure from the (egocentric) perspective of the robot [6]. In contrast, the Spatial Semantic Hierarchy (*SSH*) built an allocentric cognitive map with hierarchical metric and topological representations [23]. SSH also incorporated representations of partial knowledge and uncertainty. It was tested as a robot controller in simulation for indoor and outdoor environments, and on a physical robot in an office environment [3]. SemaFORR is hierarchical as well, both in its reasoning structure (described in Section 3.2) and in its ability to combine atomic spatial affordances into higher level ones (described in Section 3.1).

Research has also sought to adapt human-like internal representations of the environment, Thrun's robot controller integrated a grid-based metric map with a topological one [38]. A grid-based map used Bayesian updating to determine the probability that a grid cell was occupied, and the topological map partitioned the grid cells into connected regions at narrow passages, such as doors. Thrun also adapted humans' use of landmarks to guide navigation [37]. His Bayesian approach learned the location of landmarks, trained an artificial neural network to recognize them, and then used them to localize. Another, similar approach used a multi-layer representation: a global metric map, a navigation graph, a topological map, and a conceptual semantic map [43]. This approach used its spatial map for natural language dialogue with a human, and so is closest to our own. SemaFORR, however, does not require a pretrained classifier to build its model.

To move the robot from its current location to some *target* (desired location), the robot's controller must reason about space. SLAM informs a controller but does not navigate. A modern robot first formulates a *plan*, a sequence of locations (*waypoints*) from its current location to its target, in a SLAM-generated map. The robot then travels to each waypoint in turn. The granularity of the planning map, actuator error, or dynamic obstacles, however, often cause the plan to fail. In that case, the controller could repair the plan or construct a new one. Instead, SemaFORR has multiple planners and recourse to multiple heuristics for local search when its plan fails.

2.3 Spatial Models and Reasoning for Humans and Robots Together

A *natural explanation* gives transparent, intelligible, human-friendly reasons for behavior in natural language. This enables the robot to gain social acceptance and reduces confusion about the robot's abilities [24]. Explanations compare counterfactual cases, selectively include causes, and recognize that the interlocutor is a social being with her own beliefs and intentions [29].

To generate descriptions of a robot navigator's behavior, many researchers have relied on detailed, relatively opaque logs of the robot's experience [25, 35]. Natural language descriptions of a robot's travelled path have addressed abstraction, specificity, and locality [34, 32], and sought to improve sentence correctness, completeness, and conciseness [2]. Those approaches, however, used a labeled map to generate descriptions and did not explain the robot's reasoning. Other work visually interpreted natural-language navigation commands with a semantic map that showed the robot's resulting action [31]. More recently, some work has selected potentially suboptimal plans [15, 5] or behaviors [20] that are more readily understandable to humans. In contrast, our work with WHY, described in Section 4, answers questions to explain the robot's reasoning and behavior in natural language, but does not influence the robot's decisions.

3 SemaFORR

FORR (FOr the Right Reasons) is a general architecture for learning and problem solving [12]. SemaFORR is a FORR-based robot controller for autonomous navigation, where the *task* of the robot is to travel to a target. SemaFORR currently assumes perfect localization. (Future work could adapt SemaFORR to contend with noisy localization from SLAM.)

The robot's world is indoors, and the robot's sole sensor is a range finder that supports only two spatial dimensions. Thus, at any moment, the robot has a *pose* $< x, y, \theta >$ in an allocentric coordinate system, where (x, y) is the robot's *location* and θ is its *orientation* with respect to the origin. In the work reported here, a simulator provides the robot's current pose, and a range finder gauges distances to the nearest obstruction in multiple directions. SemaFORR's knowledge store holds the robot's target, its learned cognitive spatial model, its plan, a small action repertoire (turns, forward moves, and a pause), and a log of *decision points*, the poses and sensor readings when SemaFORR chose an action or formulated a plan in the current task.

3.1 Learned Cognitive Spatial Model

With SemaFORR as its controller, the robot has no access to a SLAM-generated survey. Instead, as the robot's navigation experience accumulates over a set of tasks, SemaFORR learns a cognitive spatial model on a footprint of a new world, using only its perceptual history and actions. The foundation of this model is a set of atomic *spatial affordances*, static features of the world expected to facilitate navigation there. Affordances generalize over the robot's experience and may not be architecturally precise, that is, the shape of the learned affordances may not match the physical architecture of the environment. Affordances are learned at the end of a task, from the log of the robot's poses and sensor readings, without any reference to the metric map. This cognitively-based approach learns from the robot's egocentric perspective as it travels, and does not assume any global knowledge of the world. The original affordances in the model were paths, regions, trails, conveyors, and a skeleton [13]. This paper introduces two new affordances to the model, doors and hallways, both of which address a laser scanner's limited range and its discrete approximation of continuous space. Examples of all affordances appear in Figure 1.

One important feature of the spatial model is the way atomic features support the development of higher-level ones. For example, a *path* is the ordered sequence of decision points logged for a task. While any contiguous subsequence of a path supports travel, paths are overly specific and may include errors one would want to avoid. A *trail* is a refined version of a path, also represented as a sequence of decision points, but is typically more direct than the path from which it is derived. Trails also facilitate the construction of *conveyors*, cells in a $2m \times 2m$ grid superimposed on the footprint of the world. Conveyors tally the

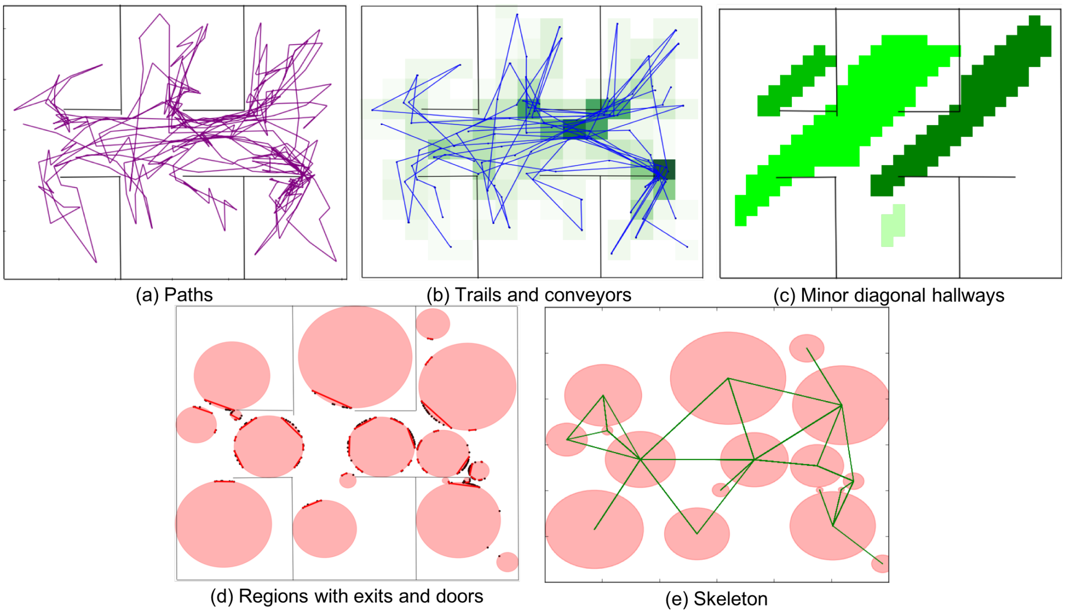


Figure 1 Affordances in a learned spatial model after visiting 40 randomly-assigned locations in a simple world (a) paths taken by the robot (b) trails refined from paths and overlaid on conveyors shaded by their count (c) minor diagonal hallways (d) regions with exits (points on the perimeter) and doors (secants) (e) the skeleton. Horizontal and vertical lines are not part of the model; they represent physical walls.

frequency with which trails have passed through them; those with high counts are likely to facilitate navigation because of the world's topology. A *region* represents unobstructed space as a circle whose center is a decision point and whose radius is the smallest distance to an obstacle detected there. Regions grow and shrink as the robot changes its pose. The *skeleton* is a graph whose nodes represent regions and whose edges represent the ability to move from one region to another. A path or a trail that moves from one region to the next induces an edge in the skeleton. Further details appear in [13].

Another important feature of the model is that many of the affordances are incremental. For example, a new incremental affordance is a *door*, an arc that affords access to a region along its perimeter. (For clarity in Figure 1, doors are drawn as secants on their endpoints.) Each location where a path or a trail crossed a region's perimeter is recorded as an *exit* for that region. To use an exit effectively, however, the robot's heading must align precisely with that exit. As exits accumulate, SemaFORR learns doors, generalizations about the region's circumference. A pair of exits is said to be *nearby* one another when the arc between them is less than ε. Algorithm 1 shows the pseudocode that learns doors. It moves along the circumference of any region with more than one recorded exit until it encounters a consecutive pair of nearby exits. When it finds such a pair, it records the arc between them as a door, and continues to extend the current door as long as the next exit is nearby its most recent addition. Otherwise, the algorithm resumes search for the next door. Doors for a region, along with any unincorporated exits, are recorded in the knowledge store. Data from subsequent tasks adds new exits to existing doors, identifies new doors, and merges them as necessary.

The other new higher-level feature of the spatial model is a hallway. Intuitively, a hallway is a relatively straight, narrow, continuous area with both length and width. Figure 2

Algorithm 1: SemaFORR's door-learning algorithm

```
Input: Regions, Exits
Output: Doors
Doors ← ∅
for each region R with more than one exit do
    Select an exit e
    D_e = {e}
    start ← e
    e′ ← ∅
    while start ≠ e′ do
        Move clockwise from e to the next exit e′
        if e′ is within ε of e then
            D_e ← D_e ∪ {e′}
            e ← e′
        else
            if |D_e| > 1 then
                Doors ← Doors ∪ {D_e}
            e ← e′
            D_e = {e′}
        end
    end
end
return Doors
```

illustrates how horizontal hallways develop in the footprint of a simple world. Algorithm 2 is pseudocode for SemaFORR's hallway-learning algorithm. To begin, the algorithm forms a *segment* from every pair of consecutive poses in a path and the percepts at their endpoints. It then labels each segment (as horizontal, vertical, major diagonal, or minor diagonal), partitions the segments by their label, and performs the same five steps within each subset (e.g., Figure 2(a)). Step 1 identifies segment pairs (*parents*) that are most similar to one another. To do so, it calculates the similarity of each possible pair of segments, based on the distance between their midpoints and the difference in their angles. Parents are those more than three standard deviations above the mean similarity for their common label. (If none are detected, this criterion is iteratively reduced from 3σ by 0.25σ until at least one pair is found or 0 is reached. In our experience, most parents lie above 1.5σ.) Step 2 generates potential building blocks for hallways. Each pair of parents determine a *child* segment midway between them. If the child shares its parents' label, and their percepts indicate that both ends of the child would be visible from their four endpoints (i.e., the child does not pass through a wall), both parents and their child become *candidates* (Figure 2(b)). Step 3 constructs a *heatmap*, a $1m \times 1m$ grid on the footprint of the world. Initially, cells have value 0; each candidate then increments the values in the corresponding grid cells (Figure 2(c)). To smooth the heatmap, the algorithm searches for cells whose neighbors' values indicate that they should join a hallway. If a cell has value 0 but the values in at least 70% of its (at most 8) immediate neighbors meets a threshold τ (here, 1), that cell's value is set to 1 (Figure 2(d)). (Although this process is recursive, in our experience there are rarely more than two iterations.) Step 4 uses depth-first search to find *aggregates*, connected components formed by cells with non-zero values in the heatmap (Figure 2(e)). Step 5 merges any two aggregates

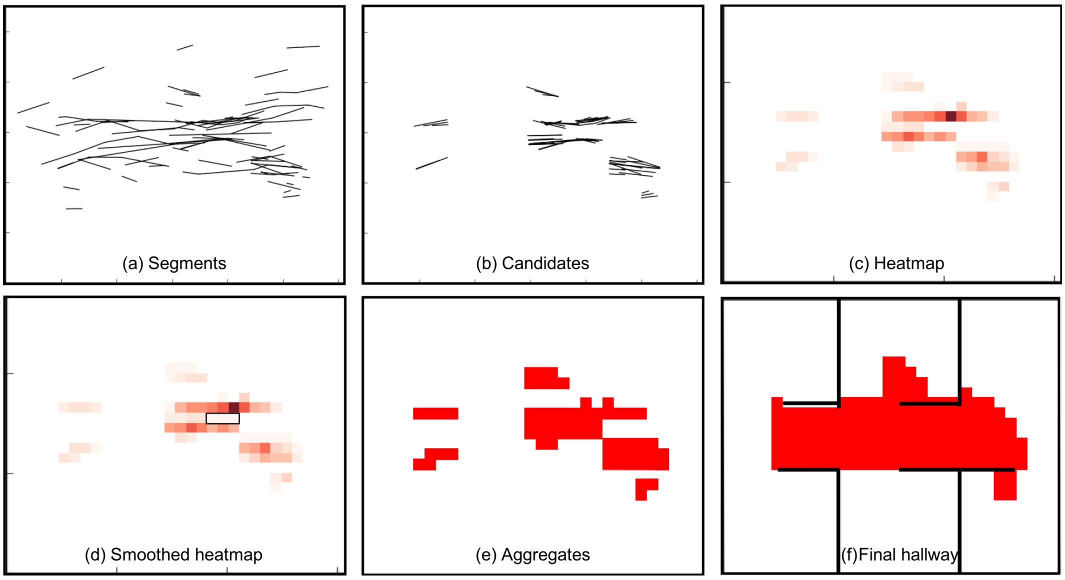


Figure 2 Stages in the development of horizontal hallways in a simple world after 40 tasks (a) segments (b) candidates (c) the heatmap (d) the smoothed heatmap with added cells indicated by the rectangle (e) aggregates (f) final horizontal hallways superimposed on the true map

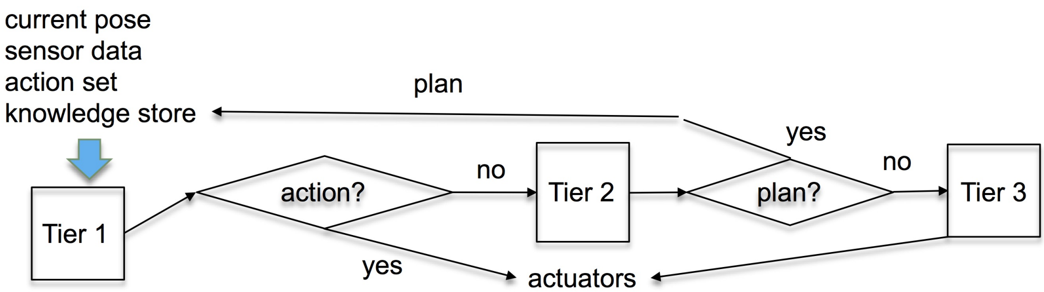


Figure 3 SemaFORR reasons with a hierarchy of Advisors

when each would be visible to the other, and repeats the smoothing process (Figure 2(f)). Finally, the algorithm records in the knowledge store, but does not merge, differently labeled hallways that intersect with one another.

3.2 Reasoning

As a FORR-based system, SemaFORR defines domain-specific "right reasons" called *Advisors*. An Advisor is a procedure that generates *comments*, opinions on how to navigate. Each Advisor has its own *rationale* (e.g.,"avoid walls" or "go to unfamiliar locations"), a narrow perspective on the degree to which a possible action supports or opposes success on the task. Table 1 lists the Advisors used in the work reported here.

FORR represents decision making as a combination of reaction, deliberation, and heuristic choice. To integrate those approaches, SemaFORR organizes its Advisors into the three-tier hierarchy of Figure 3. Advisors in tier 1 are reactive; they respond quickly and are assumed to be correct. A tier-1 Advisor can mandate an action (e.g., move directly to a visible target)

Algorithm 2: SemaFORR's hallway-learning algorithm

Input: paths, laser scan history, smoothing threshold τ
Output: *Hallways*
$LineSegments \leftarrow Segment(\text{paths})$
$CardinalDirections \leftarrow Partition(LineSegments)$
for *each set of segments* $\in CardinalDirections$ **do**
  Calculate pairwise similarity for all segments in the set
  $Parents \leftarrow$ segments with similarity above dynamically-selected threshold
  $Candidates \leftarrow \{\ \}$
  for $pair \in Parents$ **do**
    Compute *child*
    if *child's direction* $=$ *pair's direction* $\wedge$ $Visible(child, pair)$ **then**
      $Candidates \leftarrow Candidates \cup \{child, parent_1, parent_2\}$
  end
  $HeatMap \leftarrow ComputeHeatmap(Candidates)$
  $SmoothedHeatMap \leftarrow Smooth(HeatMap, \tau)$
  $Aggregates \leftarrow ConnectedComponents(SmoothedHeatMap)$
  $MergedAggregates \leftarrow MergeVisible(Aggregates)$
  $SmoothedAggregates \leftarrow Smooth(MergedAggregates, \tau)$
  $Hallways \leftarrow Hallways \cup SmoothedAggregates$
end
return *Hallways*

or veto any number of actions (e.g., those that would move into a wall). Advisors in tier 2 are deliberative; each of them constructs a plan from the robot's current location to its target. Advisors in tier 3 are heuristics that comment on possible actions.

To control a robot, SemaFORR executes its sense-decide-act loop. Given its knowledge store and the data sensed by its most recent laser scan, SemaFORR moves through the Figure 3 hierarchy. In tier 1, if the target is in view and an action would immediately drive the robot to it, the Advisor VICTORY selects that action. Otherwise, if there is a current plan and the next waypoint in that plan is in view, ENFORCER selects the action that would immediately drive the robot to it. If an action was selected, the decision cycle ends, and SemaFORR sends the selected action to the robot's actuators. Otherwise, AVOIDOBSTACLES and NOTOPPOSITE veto any action that would cause a collision or return the robot to its last heading, respectively, and decision making proceeds to tier 2.

Tier 2 plans only once, at the beginning of a task, and provides waypoints for the entire task. A graph planner has an edge-weighted *cost graph* that reflects the planner's particular objective. The classic example is A*, which builds its cost graph from a grid superimposed on a map of the world, where each node represents the center of a grid cell and an edge represents unimpeded access between two cells with weight equal to the Euclidean distance between their centers. This allows A* to build shortest-path plans. SemaFORR has three planners, each of which exploits a particular category of spatial affordances: regions, hallways, or conveyors. Each planner represents its objective by adjustments to distance-based edge weights in its cost graph. For example, REGIONPLAN starts with the A* cost graph but then modifies each edge weight e between two nodes as described in Table 2. This creates a bias for paths that travel through regions.

Algorithm 3 is pseudocode for tier 2. To resolve conflicts among its planners, each of

Table 1 SemaFORR's Advisors and their rationales.

Tier 1, in order	
Victory	Go toward an unobstructed target
Enforcer	Go toward an unobstructed waypoint
AvoidObstacles	Do not go within ε of an obstacle
NotOpposite	Do not return to the last orientation
Tier 2 planners	
ConveyorPlan	Reduce cost-graph edge weights through conveyors
HallwayPlan	Reduce cost-graph edge weights in hallways
RegionPlan	Reduce cost-graph edge weights in regions and near doors and exits
Tier 3 heuristics	
	Based on commonsense reasoning
BigStep	Take a long step
Curiosity	Go to never visited locations
ElbowRoom	Get far away from obstacles
Enfilade	Go toward recent positions
Explorer	Go to currently unfamiliar locations
GoAround	Turn away from nearby obstacles
Greedy	Go close to the target
VisualScan	Turn in place to examine the world
	Based on the spatial model
Access	Go to regions with many doors
Convey	Go to frequent, distant conveyors
Crossroads	Go to highly connected hallways
Enter	Go into the target's region via an exit
EnterDoor	Go into the target's region via a door
Exit	Leave a region without the target via an exit
ExitDoor	Leave a region without the target via a door
Follow	Use hallways to approach the target
LeastAngle	Leave a region in the target's direction
SpatialLearner	Go to unmodeled locations
Stay	Stay within a hallway
Trailer	Use a trail segment to approach the target
Unlikely	Avoid dead-ends in the skeleton

SemaFORR's planners evaluates the plans of the others from its own perspective. Let C_{ij} be the cost of plan P_i from Advisor A_i as evaluated in Advisor A_j's cost graph. SemaFORR norms C_{ij} values in [0,10] for each i, scores plan P_i as $\sum_j C_{ij}$, selects the plan with the lowest score, places it in the knowledge store, and ends the decision cycle. Figure 4 illustrates this with three plans to travel from the lower left corner to the target (star). Each planner has produced a plan biased toward its particular objective. In Figure 4, when the plans are evaluated in the cost graphs of all three planners, RegionPlan has the lowest total cost because it is also relatively short (A*'s objective) and passes through a hallway (HallwayPlan's objective).

If a plan is in place but multiple possible actions survive tier 1's filter, decision making passes to tier 3's heuristic Advisors. Each Advisor's rationale is deliberately narrow (e.g., "go to unfamiliar locations"), represented as a function that assigns individual *strengths*

Table 2 How RegionPlan exploits its spatial affordances to modify the A* cost graph.

Condition	Modified edge weight
Starts and ends in a region	$0.25e$
Only one end in a region and within 0.5 meters of a door and an exit	$0.50e$
Only one end in a region and within 0.5 meters of a door or an exit	$0.75e$
Only one end in a region but not near a door or exit	$1.00e$
Neither end in a region	$10.00e$

Algorithm 3: SemaFORR's Tier 2 procedure

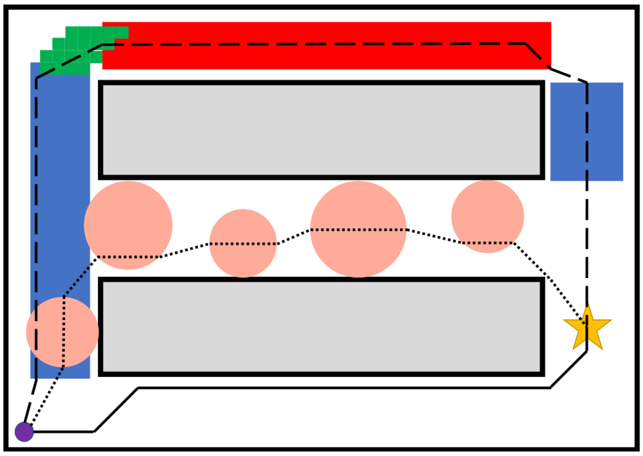
```
Input: current pose, target location, spatial model, A* cost graph
Output: SelectedPlan
for each Advisor A_i ∈ Tier 2 do
    Set A_i's cost graph to a copy of A*'s cost graph
    Update A_i's cost graph based on A_i's objective and the spatial model
    Use A* search to find lowest cost plan P_i in A_i's cost graph
end
for each Advisor A_j do
    for each plan P_i do
        C_ij ← cost of plan P_i in A_j's cost graph
    end
end
Normalize plan costs C_ij in [0,10]
for each plan P_i do
    Score_i = Σ_j C_ij
end
SelectedPlan ← argmin_{i∈Tier2} Score_i
return SelectedPlan
```

normalized in [0,10] to any subset of the remaining actions. Strengths above 5 represent support for the action; those below 5 represent opposition to it. For example, Curiosity supports actions that encourage the robot to travel to places in the environment it has never visited in any task, Explorer supports actions to unvisited locations in the current task, and SpatialLearner supports actions to locations that are not included in regions, conveyors, or hallways. To capitalize on the synergy among multiple heuristics, *voting* selects the action with the maximum total score from all tier-3 Advisors, ends the decision cycle, and sends the selected action to the robot's actuators.

4 Natural Explanations

Why uses SemaFORR's knowledge store and its Advisors' comments at a decision point to explain (and, to a limited extent, discuss) the robot's behavior. Elsewhere we have sketched our general approach to natural explanation, but only when the robot confronts crowds of moving people [21]. This work addresses explanations that reference SemaFORR's spatial model, Advisors, and reasoning structure. To answer questions, Why identifies the Advisors that drove its decisions; their rationales are the reasons for SemaFORR's behavior.

Throughout this section, $\mathcal{N}$ represents functions that map any value to natural language.



	A* cost	Region cost	Hallway cost	Total cost
A* Plan	0	10	10	20
RegionPlan	4	0	6	10
HallwayPlan	10	8	0	18

Figure 4 SemaFORR's tier 2 votes to select the plan with lowest cost in a world with two large obstructions.

For example, an action a is described in natural language by $\mathcal{N}(a)$ and the rationale of Advisor A by $\mathcal{N}(A)$. WHY also calculates a variety of metrics that monitor aspects (e.g., confidence or enthusiasm) of the decision process. To explain these real values, WHY maps an ordered partition of each metric's range into natural language (also denoted by $\mathcal{N}$). For example, a metric $m \in (-\infty, +\infty)$, could be partitioned as $\{(-\infty, 0), [0, +\infty)\}$, with $\mathcal{N}(m < 0) \rightarrow$ "a little" and $\mathcal{N}(m \geq 0) \rightarrow$ "a lot." These ranges allow WHY to hedge in its responses, much the way people explain their reasoning when they are uncertain [27].

To generate an explanation, WHY completes templates with its $\mathcal{N}$ functions and appropriate punctuation and conjunctions. All examples in this section were drawn from the experiments described in Section 5. This section first describes WHY for behavior determined by tier 1 or tier 3, and then for plans determined by tier 2.

4.1 Explanations for Single Actions

"Why did you decide to do that?" This questions a particular action a. Algorithm 4 is pseudocode to produce a reply. In response, WHY takes as input the current pose, target location, spatial model, and the Advisors' comments. When SemaFORR makes a decision in tier 1, it is either because VICTORY or ENFORCER mandated it, so that WHY uses the template

I could see our [target/waypoint] and $\mathcal{N}(a)$ would get us closer to it.

or because AVOIDOBSTACLES vetoed all actions but the pause, so that WHY uses the template

I decided to wait because there's not enough room to move forward.

The inherent uncertainty and complexity of a tier-3 decision, however, requires this template's more nuanced explanation:

Although [$\mathcal{N}(\rho_{ai})\mathcal{N}(A_i)$ for A_i that oppose a],
I decided to $\mathcal{N}(a)$ because [$\mathcal{N}(\rho_{ai})\mathcal{N}(A_i)$ for A_i that support a].

WHY includes only those tier-3 Advisors with strong opinions about a, compared to other actions. Let μ_i be the mean comment strength across all actions and σ_i its standard deviation.

Algorithm 4: WHY's explanation procedure for single actions

Input: *current pose, target location, spatial model, Advisors' comments*
Output: *explanation*
switch *mode(decision)* **do**
  case *tier 1 decides action* **do**
    explanation ← sentence based on VICTORY or ENFORCER
  case *only 1 unvetoed action remains after tier 1* **do**
    explanation ← sentence based on vetoes from AVOIDWALLS
  otherwise do
    Compute relative support for tier-3 Advisors' strengths
    Categorize the support level for the chosen action
    Complete template for each Advisor with its support level and rationale
    explanation ← combined completed templates
  end
end
return *explanation*

For comment strength c_{ia} from Advisor A_i on action a, A_i's *relative support* is defined as $\rho_{ia} = (c_{ia} - \mu_i)/\sigma_i$. Because A_i has a strong opinion about a relative to the other actions only if $|\rho_{ia}|$ is large, WHY excludes A_i from its explanation if $\rho_{ia} \in (-0.75, 0.75]$. The first line in the template uses $\mathcal{N}(A_i)$ and $\mathcal{N}(\rho_{ia})$ phrases only if $\rho_{ia} \leq -0.75$; the line is omitted if no Advisors opposed a strongly enough. The second line uses $\mathcal{N}(A_i)$ and $\mathcal{N}(\rho_{ia})$ phrases only for $\rho_{ia} > 0.75$, For example, if GREEDY supports a forward move of $1.6m$ so strongly that $\mathcal{N}(\rho_{ia})$ is "I really want" but EXPLORER opposes that move, and $\mathcal{N}(a)$ is "move forward a lot," WHY would explain "Although I don't want to go somewhere I've been, I decided to move forward a lot because I really want to get close to our target."

"What action would you take if you were* [here]*?" WHY substitutes the alternative pose [here] for the robot's current one, and has SemaFORR recompute its decision from the current spatial model to produce hypothetical comments. WHY then treats this as a "why did you decide" question, but substitutes "I would" for "I decided to."

"How sure are you that this is the right decision?" This asks about the robot's *confidence*, that is, how much it believes its decision will help it reach the target. Decisions in tier 1 are by definition highly confident, so the template for VICTORY or ENFORCER is

> Highly confident, since [our target/the next waypoint in our plan] is in sensor range and this would get us closer to it.

and for AVOIDOBSTACLES the template is

> Highly confident, since there is not enough room to move forward.

Again, tier-3's uncertainty and complexity require more nuanced language. Confidence Λ_a relies on two metrics: γ_a, the extent to which the tier-3 Advisors agree with one another in their opinion of an action, and β_a, SemaFORR's *overall support* for its chosen action compared to other actions. Let $S_a = \sum_{i=1}^{v} c_{ia}$ be the total strength of possible action a when v tier-3 Advisors comment. Then the *level of agreement* on a among all Advisors is the Gini impurity of S_a, $\gamma_a = 2 \cdot (S_a/10v) \cdot (1 - S_a/10v)$, where values near 0 indicate a high level of agreement in support or opposition and values near 0.5 indicate disagreement or lack of a strong opinion [19]. For example, if four Advisors assign equally supportive scores $[10, 10, 10, 10]$ to action a and divergent scores $[0, 0, 10, 10]$ to action a', then $\gamma_a = 0.0$ captures the agreement and $\gamma_{a'} = 0.5$ the disagreement. Overall support β_a for a compared to other actions is $\beta_a = (S_a - \mu_S)/\sigma_S$, where μ_S and σ_S are the mean and standard deviation, respectively, of S_a across all actions a. To gauge the robot's *confidence level* Λ_a, Why weights the level of agreement and overall support equally, with $\Lambda_a = (0.5 - \gamma_a) \cdot \beta_a$, The template is

> I'm $\mathcal{N}(\Lambda_a)$ sure
> because $\mathcal{N}(\gamma_a)$ $\mathcal{N}(\beta_a)$
> even though [$\mathcal{N}$ for whichever of γ_a or β_a is lower than Λ_a],
> [$\mathcal{N}$ for whichever of γ_a or β_a is higher than Λ_a].

To complete it, Why retrieves ordered labels for each of $\mathcal{N}(\Lambda_a)$, $\mathcal{N}(\gamma_a)$, and $\mathcal{N}(\beta_a)$. If γ_a and β_a have the same label as Λ_a, Why uses only the first two lines. For example, "I'm really sure because I've got many reasons for it. I really want to do this the most." If only one of γ_a and β_a match Λ_a's label, Why completes only the first line and the agreeing phrase in the second. For example, "I'm not sure because my reasons conflict." Finally, if neither γ_a nor β_a matches with Λ_a, Why completes the first, third, and fourth lines. For example, "I am only somewhat sure because, even though I've got many reasons, I don't really want to do this the most."

"Why not do* [something else]*?" A person makes decisions with her own mental model of the world. When her decision conflicts with another's, she tries to understand why they made a different decision. To explain SemaFORR's preference for action a over an alternative b, the template for Victory or Enforcer is

> I decided not to $\mathcal{N}(b)$ because [I detect our target/this follows our plan]

and for AvoidObstacles or NotOpposite the template is

> I decided not to $\mathcal{N}(b)$ because [$\mathcal{N}(A_i)$ for A_i that vetoed b].

The other possibility is that b scored lower in tier 3 than a did. How much SemaFORR prefers a to b is based on the difference in the two actions' overall support $\beta_a - \beta_b$. Only tier-3 Advisors with a *clear preference* for a over b (defined by $\rho_{ia} - \rho_{ib} \notin [-1, 1]$) are used to complete this template:

> I thought about $\mathcal{N}(b)$
> because it would let us [$\mathcal{N}(A_i)$ for A_i that prefer b],
> but I felt $\mathcal{N}(\beta_a - \beta_b)$ strongly about $\mathcal{N}(a)$
> since it lets us [$\mathcal{N}(A_i)$ for A_i that prefer a].

The second line is included only if any Advisors showed a clear preference for b. For example, if Greedy preferred a, while Explorer preferred b, one explanation is "I thought about b because it would let us go somewhere new, but I felt slightly more strongly about a since it lets us get closer to our target."

Algorithm 5: WHY's explanation procedure for plans

Input: *robot's pose, target location, Advisors' comments, objectives* $\mathcal{O}_s$ *and* $\mathcal{O}_q$
Output: *explanation*
Compute plans: P_q based on $\mathcal{O}_q$ and P_s based on $\mathcal{O}_s$
Compute perspectives: $\Delta_q = C_{sq} - C_{qq}$ and $\Delta_s = C_{ss} - C_{qs}$
switch *mode(*Δ_q, Δ_s*)* **do**
  case $\Delta_q = \Delta_s = 0$ **do**
    explanation ← sentence based on template for equivalent plans
  case $\Delta_s < 0$ *and* $\Delta_q = 0$ **do**
    explanation ← sentence based on $\mathcal{O}_s$ (e.g., follows hallways)
  case $\Delta_s < 0$ *and* $\Delta_q > 0$ **do**
    explanation ← sentence based on $\mathcal{O}_s$ and $\mathcal{O}_q$ (e.g., follows hallways and length)
end
return *explanation*

4.2 Explanations for Plans

Explanations for a plan assume an alternative objective. Assume SemaFORR's current plan was produced by planner P_s with objective $\mathcal{O}_s$ in its cost graph, and that the questioner reasons instead with P_q and $\mathcal{O}_q$. Let C_{ij} be the cost of planner P_i's plan in the cost graph of planner P_j. WHY addresses the differences in the perspectives of P_s and P_q as $\Delta_q = C_{sq} - C_{qq}$ and $\Delta_s = C_{ss} - C_{qs}$. WHY's responses are based on the robot's pose, the Advisors' comments, the target, and objectives $\mathcal{O}_s$ and $\mathcal{O}_q$. As a running example, assume $\mathcal{O}_q$ is "take the shortest path" and $\mathcal{O}_s$ is "take the hallways." WHY translates objective $\mathcal{O}$ as $\mathcal{N}(\mathcal{O})$; in the example, this would be "short"and "follows hallways," respectively.

"Why does your plan go this way?" could be asked anywhere along the robot's intended path. Algorithm 5 is pseudocode for WHY's explanation procedure. Based on the values for Δ_q and Δ_s, there are several possible cases, each with its own language template. If both are 0, then the plans equally address the two objectives, and WHY explains:

I decided to go this way because I think it's just as $\mathcal{N}(\mathcal{O}_s)$ and equally $\mathcal{N}(\mathcal{O}_q)$.

Otherwise, the plans differ with respect to one or both objectives. If Δ_s is negative (e.g., P_s is more aligned with hallways), then WHY uses the template

Although there may be a $\mathcal{N}(\Delta_q)$ $\mathcal{N}^*(\mathcal{O}_q)$ way,
I think my way is $\mathcal{N}(\Delta_s)$ $\mathcal{N}^*(\mathcal{O}_s)$.

where $\mathcal{N}^*(\mathcal{O})$ is a comparator for $\mathcal{O}$ (e.g., "shorter" or "better at following hallways"). For example, an explanation could be "Although there may be a somewhat shorter way, I think my way is a lot better at following hallways." WHY omits the first line in the template if $\Delta_q = 0$. Other cases, where Δ_q is negative or Δ_s is positive, cannot occur because each planner is optimal with respect to its own objective.

"What makes your plan better than mine?" If Δ_q and Δ_s are both 0, then WHY replies, "I think both plans are equally good." Otherwise, WHY responds with the template

I think my way is better because it's $\mathcal{N}(\Delta_s)$ $\mathcal{N}^*(\mathcal{O}_s)$.

For example, an explanation could be "I think my way is better because it's a lot better at following hallways."

"What's another way we could go?" In response, Why applies the template

We could go that way since it's $\mathcal{N}(\Delta_q)$ $\mathcal{N}^*(\mathcal{O}_q)$ but it could also be $\mathcal{N}(\Delta_s)$ $\mathcal{N}'(\mathcal{O}_s)$.

where $\mathcal{N}'$ denotes an opposite comparator (e.g., "longer" or "farther from known hallways"). For example, an explanation is "We could go that way since it's somewhat shorter but it could also be a lot farther from known hallways."

"How sure are you about your plan?" Why analyzes and explains its confidence in its objective with the template

I'm $\mathcal{N}(P_s)$ sure because
my plan is $\mathcal{N}(\mathcal{O}_s)\mathcal{N}^*(\mathcal{O}_s)$ and only $\mathcal{N}(\mathcal{O}_q)\mathcal{N}'(\mathcal{O}_q)$ than your plan.
even though my plan is $\mathcal{N}(\mathcal{O}_s)\mathcal{N}^*(\mathcal{O}_s)$, it is also $\mathcal{N}(\mathcal{O}_q)\mathcal{N}'(\mathcal{O}_q)$ than your plan.
my plan is $\mathcal{N}(\mathcal{O}_q)\mathcal{N}'(\mathcal{O}_q)$ and only $\mathcal{N}(\mathcal{O}_s)\mathcal{N}^*(\mathcal{O}_s)$ than your plan

Why retrieves $\mathcal{N}(P_s)$, its confidence in SemaFORR's plan P_s based on Δ_s and Δ_q. To compute confidence, the values for Δ_s and Δ_q are first partitioned into three intervals each. The Cartesian product of the two partitions results in nine possible combinations. Finally, $\mathcal{N}(P_s)$ applies one of the labels ["really", "only somewhat", "not"] to each intersection. If $\mathcal{N}(P_s)$ = "really," Why uses the second line in the template; if $\mathcal{N}(P_s)$ = "only somewhat," it uses the third line; otherwise it uses the fourth. For example, "I'm really sure because my plan is a lot better at following hallways and only somewhat longer than your plan."

5 Empirical Design and Results

The results reported here were run in simulation with Fetch Robotics' robot Freight, whose laser range finder reports 660 distances within $25m$, along a 220° arc at a rate of 15 times per second. The robot's world, shown in Figure 5, was the fifth floor of a building that occupies an entire Manhattan block (approximately $110m \times 70m$). It includes the jogs, narrow doorways, and support columns (which appear as small circles) of the original architectural floorplan. Moreover, Figure 5's four horizontal parallel hallways, and its three parallel vertical ones, provide multiple alternate routes to most targets. Nonetheless, the extent and accuracy of SemaFORR's model will be dependent upon where the robot has traveled. An example of the model learned after 40 tasks in this world appears in Figure 6.

During this experiment, the simulator localizes the robot directly within Figure 5 and reports the percepts it would experience; SLAM is not used. An *experiment* was a sequence of 40 preselected, randomly chosen targets to visit (*tasks*). To encourage a variety of challenges, there were 5 such experiments, each with a different set of 40 targets. The robot always began an experiment in the same pose and addressed its tasks in their given order. Each task after the first began wherever the previous one had ended. If the robot did not reach its target after 500 decision steps, it *failed* that task and began to address the next task from its current pose. Evaluation metrics were total (wall clock) travel time in seconds, total travel distance in meters, percentage of successful tasks, and *coverage*, the fraction of the world's footprint covered by the spatial model, as evaluated in a $1m \times 1m$ grid.

We tested SemaFORR with the full spatial model, all the Advisors in Table 1, and the procedure to select a plan in Algorithm 3. We also tested ablated versions that kept tier 1 but dropped Advisors from other tiers. The *model-free* version had only an A* planner in tier 2 and the commonsense Advisors in tier 3; it entirely ignores spatial models. Two other versions, RegionFocused and HallwayFocused, had only the planner for their affordance

Figure 5 Test world for SemaFORR and WHY

Table 3 Performance of SemaFORR and ablated versions

Navigator	Travel Time	Distance	Success Rate	Avg. Coverage	Final Coverage
Model-free	3538.61	3475.35	89.40%	——	——
RegionFocused	**2846.13**	**3087.67**	**95.00%**	7.69%	12.36%
HallwayFocused	**3142.10**	3317.62	**92.90%**	6.21%	10.92%
SemaFORR	**2791.26**	**3194.94**	**93.90%**	10.95%*	17.65%*

(REGIONPLAN or HALLWAYPLAN), the commonsense Advisors, and any tier-3 Advisors that used their affordance (regions, doors, and exits for RegionFocused, and hallways for HallwayFocused). Both planning versions use affordances to represent unobstructed space in the environment and the connectivity of the space. By ablating these versions, the experiment is able to tease out the difference in these approaches, their ability to represent connectivity, and their usefulness for planning.

SemaFORR averaged 137.48 decisions per task, and each decision required 0.04 seconds. After its one-time planning, SemaFORR made about 64% of its decisions in tier 1 and 36% in tier 3. Tier 2 selected on average 38.46% of its plans from REGIONPLAN, 28.21% from HALLWAYPLAN, and 35.90% from CONVEYORPLAN. The spatial model required about 9 seconds to learn and revise at the end of each task.

The results in Table 3 report average performance across 25 runs (5 iterations on 5 sets of 40 targets each). Data in boldface indicates statistically significant improvements compared to the model-free version. Both RegionFocused and SemaFORR produced plans that allowed the robot to travel a shorter distance than the model-free version. All three alternatives to the model-free version enabled the robot to reach its targets more quickly and succeed more often ($p = 0.05$). The only statistically significant differences (denoted by an asterisk) between SemaFORR and RegionFocused lie in their coverage: SemaFORR's coverage is greater than that of RegionFocused, both during an experiment (measured after each of the 40 tasks and averaged) and at its completion (after 40 tasks). Although some region-related Advisor is deemed supportive in 63.17% of all explanations, SemaFORR draws on a richer set of reasons from its full spatial model and, in the end, has learned more about its world.

WHY's tables for $\mathcal{N}$ generate distinct natural explanations that simulate people's ability to vary their explanations based on their context [26]. To examine its explanations, we ran an experiment with HALLWAYPLAN for P_s and A* for P_q. The system learned the full spatial model as it navigated to 80 targets and answered every question described in Section 4 at each decision point. WHY averaged less than 7 msec to compute each explanation. The results in Table 4 show that this approach is also nuanced, with many unique explanations per question.

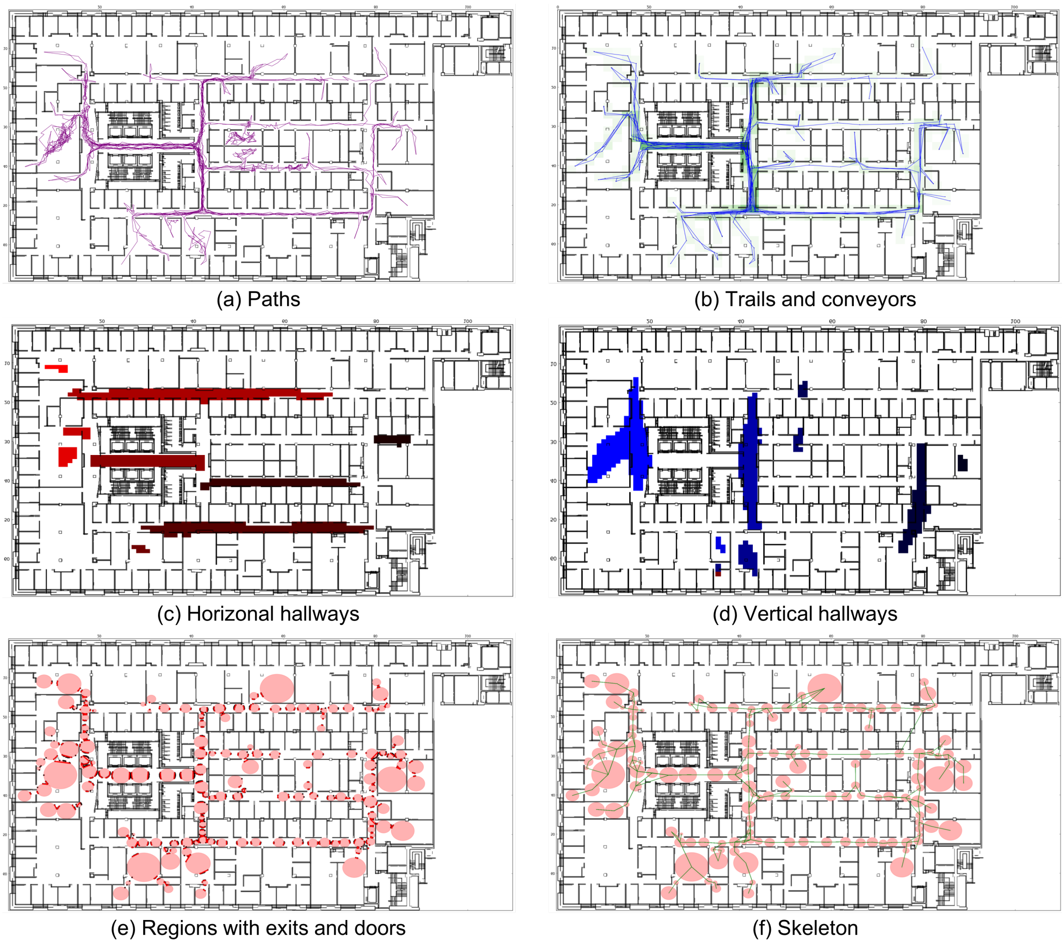


Figure 6 An example of SemaFORR's learned spatial model in the test world

The Coleman-Liau index measures text readability [9]; it gauged WHY's explanations at approximately a sixth-grade level, and thus readily understandable to a layperson.

6 Discussion

SemaFORR can serve as a robot controller for autonomous navigation in simulation, as it was used here, or on the floor. It can also merely observe and comment upon the behavior of a robot that has a range sensor but navigates with a different controller. Moreover, it can be used in dynamic worlds where it learns and exploits crowd models, in tiers 2 and 3 [1].

Learning a cognitive spatial model takes experience. If the robot does not travel within sensor range of an area, it will have no model for it. For example, in preliminary work we implemented TRAILPLAN, a planner that that relied only on trails in tier 2, and tested an ablated version called TrailFocused. That approach quickly preferred to reuse just a few early trails and therefore explored, and learned, very little. This considerably degraded the coverage of its learned spatial model; TrailFocused repeatedly failed and was eliminated from the study. Although CONVEYORPLAN developed more credible plans, the ablated version, ConveyorFocused, experienced similar difficulties and so was not evaluated separately.

Instead of learning from navigation experience, one could simply position the robot

Table 4 Analysis of explanation results by tier

	Tier 1	Tier 2	Tier 3
Number of explanations	100,797	9,796	23,966
Average computation time (msec)	0.62	5.59	0.50
Number of unique phrasings			
Why did you do that/Why does your plan go that way?	13	6	2,655
How sure are you?	3	6	7
Why not do something else/What's another way to go?	20	6	12,592
What makes your plan better than mine?	—	4	—
Total	36	22	15,254
Average readability			
Why did you do that/Why does your plan go that way?	4.32	5.89	5.15
How sure are you?	7.49	7.30	8.36
Why not do something else/What's another way to go?	5.42	5.35	6.55
What makes your plan better than mine?	—	7.25	—
Overall	5.48	6.45	6.60

in multiple locations throughout an architectural drawing or a SLAM-based map. The resultant model, however, may not detect useful, task-oriented affordances. In contrast, SemaFORR's model reflects the robot's experience, and the controller can resort to its commonsense heuristics in areas without coverage. Alternatively, an offline process could initialize SemaFORR's model and then be augmented and modified as the robot travels.

In realistic worlds, planners are essential. SemaFORR without any planners failed on most tasks in Figure 5, and so learned little or no spatial model. A graph-based planner with too coarse a grid can also fail, because sequences of waypoints in its cost graph become less reliable. Our robot is nearly as broad as some doorways; it can only leave a room if it approaches the door at just the right angle. As a result, we used a relatively fine grid, which produces a large graph (approximately 85,000 vertices and 170,000 edges). A* is optimal because its heuristic is admissible and consistent. Without such a heuristic, SemaFORR's model-based planners use Dijkstra's algorithm [11], whose theoretical time complexity is the same as A*'s, but whose average case performance is worse. As a result, the model-based planners required significantly more time (about 1 minute versus 15 seconds) than A*. Nonetheless, navigation with them proved more successful.

SemaFORR's affordance-based planners consider distance but do not assume that all unobstructed grid cells have identical features. In our experiments, A* plans tended to hug the walls and travel through tight spaces (e.g., narrow hallways), where turns were difficult and the robot often became stuck. For a robot with fragile or unstable cargo, the smoothness of a hallway or the range of available actions within a region may also be important.

SemaFORR's spatial model is hierarchical, graph-oriented, and has well-defined semantics, all features observed in the models that people generate. There are, however, no landmarks and its graphs are not labeled. Current work investigates ways to accelerate model-based planning, including admissible heuristics that would support A* in model-based cost graphs. Future work includes landmarks, other sensors, extended dialogue (e.g., queries to the user), and human subjects to gauge the quality of WHY's explanations and the reasonableness of its current values for $\mathcal{N}$. Meanwhile, SemaFORR demonstrates the power of a cognitive spatial model to inform both planning and user-friendly explanations, and to support autonomous navigation through the complexities of a large realistic world.

References

1 A. Aroor, S. L. Epstein, and R. Korpan. Online learning for crowd-sensitive path planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1702–1710, 2018.

2 D. P. Barrett, S. A. Bronikowski, H. Yu, and J. M. Siskind. Driving Under the Influence (of Language). *IEEE Trans. on Neural Networks and Learning Systems*, pages 1–16, 2017.

3 P. Beeson, J. Modayil, and B. Kuipers. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *The International Journal of Robotics Research*, 29(4):428–459, 2010.

4 L. Buhry, A. H. Azizi, and S. Cheng. Reactivation, replay, and preplay: How it might all fit together. *Neural Plasticity*, 2011, 2011.

5 T. Chakraborti, S. Sreedharan, S. Grover, and S. Kambhampati. Plan explanations as model reconciliation. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 258–266. IEEE, 2019.

6 E. Chown, S. Kaplan, and D. Kortenkamp. Prototypes, location, and associative networks (PLAN): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995.

7 E. R. Chrastil and W. H. Warren. Active and passive spatial learning in human navigation: Acquisition of survey knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cgnition*, 39(5):1520–1537, 2013.

8 E. R. Chrastil and W. H. Warren. From cognitive maps to cognitive graphs. *PloS One*, 9(11):1–8, 2014.

9 M. Coleman and T. L. Liau. A Computer Readability Formula Designed for Machine Scoring. *Journal of Applied Psychology*, 60(2):283–284, 1975.

10 J. A. Conlin. Getting around: making fast and frugal navigation decisions. *Progress in Brain Research*, 174:109–117, 2009.

11 E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

12 S. L. Epstein. For The Right Reasons: The FORR Architecture For Learning In A Skill Domain. *Cognitive Science*, 18(3):479–511, 1994.

13 S. L. Epstein, A. Aroor, M. Evanusa, E. I. Sklar, and S. Parsons. Learning Spatial Models for Navigation. In *Proceedings of the 12th International Conference on Spatial Information Theory - Volume 9368*, COSIT 2015, pages 403–425. Springer-Verlag New York, Inc., 2015.

14 P. Foo, W. H. Warren, A. Duchon, and M. 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):195–215, 2005.

15 M. Fox, D. Long, and D. Magazzeni. Explainable planning. In *IJCAI-17 Workshop on Explainable AI (XAI)*, 2017.

16 C. R. Gallistel. *The organization of learning.* The MIT Press, 1990.

17 R. G. Golledge. Human wayfinding and cognitive maps. In *Wayfinding Behavior: Cognitive mapping and other spatial processes*, pages 5–45. Hopkins University Press, 1999.

18 T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436:801–806, 2005.

19 T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2nd edition, 2009.

20 S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan. Enabling robots to communicate their objectives. *Autonomous Robots*, 43(2):309–326, 2019.

21 R. Korpan, S. L. Epstein, A. Aroor, and G. Dekel. WHY: Natural Explanations from a Robot Navigator. In *AAAI Fall Symposium on Natural Communication for Human-Robot Collaboration*, 2017.

22 I. Kostavelis and A. Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015.

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

24 T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W. K. Wong. Too Much, Too Little, or Just Right? Ways Explanations Impact End Users' Mental Models. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 3–10. IEEE, 2013.

25 C. Landsiedel, V. Rieser, M. Walter, and D. Wollherr. A Review of Spatial Reasoning and Interaction for Real-World Robotics. *Advanced Robotics*, 31(5):222–242, 2017.

26 B. F. Malle. How People Explain Behavior: A New Theoretical Framework. *Personality and Social Psychology Review*, 3(1):23–48, 1999.

27 R. Markkanen and H. Schröder. *Hedging and Discourse*, volume 24 of *Research in Text Theory*. Walter de Gruyter, 1997.

28 T. P. McNamara. Mental representations of spatial relations. *Cognitive psychology*, 18(1):87–121, 1986.

29 T. Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial Intelligence*, 2018.

30 M. Montemerlo and S. Thrun. Fastslam 2.0. *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*, pages 63–90, 2007.

31 J. Oh, T. M. Howard, M. R. Walter, D. Barber, M. Zhu, S. Park, A. Suppe, L. Navarro-Serment, F. Duvallet, A. Boularias, et al. Integrated Intelligence for Human-Robot Teams. In *International Symposium on Experimental Robotics*, pages 309–322. Springer, 2016.

32 V. Perera, S. P. Selveraj, S. Rosenthal, and M. Veloso. Dynamic Generation and Refinement of Robot Verbalization. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 212–218. IEEE, 2016.

33 M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: An open-source robot operating system. In *Proceedings of the ICRA Workshop on Open Source Software*, volume 3, page 5. IEEE, 2009.

34 S. Rosenthal, S. P. Selvaraj, and M. Veloso. Verbalization: Narration of Autonomous Mobile Robot Experience. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 16, pages 862–868, 2016.

35 R. Scalise, S. Rosenthal, and S. Srinivasa. Natural Language Explanations in Human-Collaborative Systems. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 377–378. ACM, 2017.

36 H. J. Spiers and E. A. Maguire. The dynamic nature of cognition during wayfinding. *Journal of Environmental Psychology*, 28(3):232–249, 2008.

37 S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 2, pages 958–963. IEEE, 1998.

38 S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

39 E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

40 B. Tversky. Cognitive maps, cognitive collages, and spatial mental models. In *European Conference on Spatial Information Theory*, pages 14–24. Springer, 1993.

41 W. H. Warren, D. B. Rothman, B. H. Schnapp, and J. D. Ericson. Wormholes in virtual space: From cognitive maps to cognitive graphs. *Cognition*, 166:152–163, 2017.

42 S. M. Weisberg and N. S. Newcombe. How do (some) people make a cognitive map? routes, places, and working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(5):768–785, 2016.

43 H. Zender, O. M. Mozos, P. Jensfelt, G. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

44 B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936. IEEE, 2009.