Towards Optimal Correlational Object Search

Kaiyu Zheng†, Rohan Chitnis*, Yoonchang Sung*, George Konidaris†, Stefanie Tellex†

***Abstract*— In realistic applications of object search, robots will need to locate target objects in complex environments while coping with unreliable sensors, especially for small or hard-to-detect objects. In such settings, *correlational information* can be valuable for planning efficiently. Previous approaches that consider correlational information typically resort to ad-hoc, greedy search strategies. We introduce the Correlational Object Search POMDP (COS-POMDP), which models correlations while preserving optimal solutions with a reduced state space. We propose a hierarchical planning algorithm to scale up COS-POMDPs for practical domains. Our evaluation, conducted with the AI2-THOR household simulator and the YOLOv5 object detector, shows that our method finds objects more successfully and efficiently compared to baselines, particularly for hard-to-detect objects such as srub brush and remote control.**

I. Introduction

Object search is a fundamental capability for robots in many applications including domestic services [1, 2], search and rescue [3, 4], and elderly care [5, 6]. In realistic settings, the object being searched for (e.g. pepper shaker) will often be small, outside the current field of view, and hard to detect. In such settings, *correlational information* can be of crucial value. Specifically, suppose the robot is equipped with a prior about the relative spatial locations of object types (*e.g.*, stoves tend to be near pepper shakers). Then, it can leverage this information as a powerful heuristic to narrow down or "focus" the search space, by first locating easier-to-detect objects that are highly correlated with the target object (Fig. 1). Doing so has the potential to greatly improve search efficiency; unfortunately, previous approaches to object search with correlational information tend to resort to ad-hoc or greedy search strategies [7, 8, 2, 9] or assemble a collection of independent components [10], which may not scale well to complex environments.

We follow a long line of work that models the object search problem as a partially observable Markov decision process (POMDP) [7, 11, 12, 13]. This formalization is useful because object search over long horizons is naturally a sequential, partially observed decision-making problem: the robot must (1) search for the target object by visiting multiple viewpoints in the environment sequentially, and (2) maintain and update a measure of uncertainty over the location of the target object, via its belief state. However, existing POMDP-based approaches assume object independence for scalability of maintaining and reasoning about the belief states and do not consider correlational information between objects in the environment during the search process [13, 14, 15].

We introduce COS-POMDP (Correlational Object Search POMDP), a general planning framework for optimal object search with given correlational information. Critically, COS-POMDPs avoid the exponential blow-up of naively maintaining a joint belief about all objects while preserving optimal solutions to this exponential formulation. COS-POMDPs model correlational information using a correlation-based observation model. The correlational information is given to the robot as a factored joint distribution over object locations. In practice, this distribution can be approximated by learning it from data [8, 16] or interpreting human speech [17, 18]. We address scalability by proposing a hierarchical planning algorithm, where a high-level COS-POMDP plans subgoals, each fulfilled by a low-level planner that plans with low-level actions (*i.e.*, given primitive actions); both levels plan online based on a shared and updated COS-POMDP belief state, enabling efficient closed-loop planning.

Fig. 1: We study the problem of object search using correlational information about spatial relations between objects. This example illustrates a desirable search behavior in an AI2-THOR scene, where the robot leverages the detection of a large `StoveBurner` to more efficiently find a small, hard-to-detect `PepperShaker`.

We evaluate the proposed approach in AI2-THOR [19], a realistic simulator of household environments, and we use YOLOv5 [20, 21] as the object detector. Our results show that, when the given correlational information is accurate, COS-POMDP leads to more robust search perfomance for target objects that are hard-to-detect. In particular, for target objects with a true positive detection rate below 40%, COS-POMDP significantly outperforms the POMDP baseline not using correlational information by 42.1% and a greedy, next-best view baseline [2] by 210% in terms of SPL (success weighted by inverse path length) [22], a recently developed metric that reflects both search success and efficiency.

II. Related Work

Object search involves a wide range of subproblems (*e.g.*, perception [7, 12], planning [23, 11], manipulation [24, 25]) and different types of target objects (moving [26] or static [23]). We consider static objects and an environment where the set of possible object locations is given, but we assume no object location is known a priori.

Garvey [27] and Wixson and Ballard [9] pioneered the paradigm of *indirect search*, where an intermediate object

†Brown University, Providence, RI. *MIT CSAIL, Cambridge, MA.
Email correspondence: {kzheng10@cs.brown.edu}

(such as a desk) that is typically easier to detect is located first, before the target object (such as a keyboard). More recently, probabilistic graphical models have been used to model object-room or object-object spatial correlations [2, 7, 8, 28, 29]. In particular, Zeng et al. [2] proposed a factor graph representation for different types of object spatial relations. Their approach produces search strategies in a greedy fashion by selecting the next-best view to navigate towards, based on a hybrid utility of navigation cost and the likelihood of detecting objects. In our evaluation, we compare our sequential decision-making approach with a greedy, next-best view baseline based on that work [2].

Recently, the problem of semantic visual navigation [30, 31, 32, 33, 34] received a surge of interest in the deep learning community. In this problem, an embodied agent is placed in an unknown environment and tasked to navigate towards a given semantic target (such as "kitchen" or "chair"). The agent typically has access to behavioral datasets for training on the order of millions of frames and the challenge is typically in generalization. Our work considers the standard evaluation metric (SPL [22]) and task success criteria (object visibility and distance threshold [31]) from this body of work. However, our setting differs fundamentally in that the search strategy is not a result of training but a result of solving an optimization problem.

Finally, our hierarchical planning algorithm for COS-POMDPs differs in not limiting POMDP to local use [10], or assuming navigation tasks for low-level macro-actions [35].

III. Problem Formulation

We present a general formulation of correlational object search as a planning problem, where a robot must search for a target object given correlational information with other objects in the environment. We begin by describing the underlying search environment and the capabilities of the robot, followed by the problem definition.

A. Search Environment and Robot Capabilities

The search environment contains a target object and n additional static objects. The set of possible object locations is discrete, denoted as $\mathcal{X}$. The locations of the target object $x_{\text{target}} \in \mathcal{X}$ and other objects $x_1, \ldots, x_n \in \mathcal{X}$ are unknown to the robot and follow a latent joint distribution $\Pr(x_1, \ldots, x_n, x_{\text{target}})$. The robot is given as input a factored form of this distribution, defined later in Sec. III-B.

The robot can observe the environment from a discrete set of viewpoints, where each viewpoint is specified by the position and orientation of the robot's camera. These viewpoints form the necessary state space of the robot, denoted as $\mathcal{S}_{\text{robot}}$. The initial viewpoint is denoted as $s^{\text{init}}_{\text{robot}}$. By taking a primitive move action a from the set $\mathcal{A}_m$, the robot changes its viewpoint subject to transition uncertainty $T_m(s'_{\text{robot}}, s_{\text{robot}}, a) = \Pr(s'_{\text{robot}}|s_{\text{robot}}, a)$. Also, the robot can decide to finish a task at any timestep by choosing a special action `Done`, which deterministically terminates the process.

At each timestep, the robot receives an observation z factored into two independent components $z = (z_{\text{robot}}, z_{\text{objects}})$. The first component $z_{\text{robot}} \in \mathcal{S}_{\text{robot}}$ is an estimation of the robot's current viewpoint following the observation model $O_{\text{robot}}(z_{\text{robot}}, s_{\text{robot}}) = \Pr(z_{\text{robot}}|s_{\text{robot}})$. The second component $z_{\text{objects}} = (z_1, \ldots, z_n, z_{\text{target}})$ is the result of performing object detection. Each element, $z_i \in \mathcal{X} \cup \{\texttt{null}\}$, $i \in \{1, \ldots, n, \text{target}\}$, is the detected location of object i within the field of view, or `null` if not detected. The observation z_i about object i is subject to limited field of view and sensing uncertainty captured by a *detection model* $D_i(z_i, x_i, s_{\text{robot}}) = \Pr(z_i|x_i, s_{\text{robot}})$; Here, a common conditional independence assumption in object search is made [2, 14], where z_i is conditionally independent of the observations and locations of all other objects given its location and the robot state s_{robot}. The set of detection models for all objects is $\mathcal{D} = \{D_1, \ldots, D_n, D_{\text{target}}\}$. In our experiments, we obtain parameters for the detection models based on the performance of the vision-based object detector (Sec. VI-B).

B. The Correlational Object Search Problem

Although the joint distribution of object locations is latent, the robot is assumed to have access to a factored form of that distribution, that is, n conditional distributions, $\mathcal{C} = \{C_1, \ldots, C_n\}$ where $C_i(x_i, x_{\text{target}}) = \Pr(x_i|x_{\text{target}})$ specifies the spatial correlation between the target and object i. We call each C_i a *correlation model*. This model can be learned from data or, in our case, be given by a domain expert.

Formally, we define the *correlational object search* problem as follows. Given as input a tuple $(\mathcal{X}, \mathcal{C}, \mathcal{D}, s^{\text{init}}_{\text{robot}}, \mathcal{S}_{\text{robot}}, O_{\text{robot}}, \mathcal{A}_m, T_m)$, the robot must perform a sequence of actions, $a_1, \ldots, a_t$, where $a_1, \ldots, a_{t-1} \in \mathcal{A}_m$ and the last action is `Done`. The success criteria depends on the robot state and the target location at the time of `Done`, and the robot should minimize the distance traveled to find the object. In our evaluation in AI2-THOR, we use the success criteria recommended by Batra et al. [31], defined in Sec. VI-A.

IV. Correlational Object Search as a POMDP

In this section, we introduce the COS-POMDP, a POMDP formulation that addresses the correlational object search problem, followed by a discussion on its optimality. We begin with a brief review of POMDPs [36, 37, 38].

A. Background: POMDPs

A POMDP is formally defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{Z}$ denote the state, action, and observation spaces, $T(s', a, s) = \Pr(s'|s, a)$, $O(z, s', a) = \Pr(z|s', a)$ are the transition and observation models, and $R(s, a) \in \mathbb{R}$ is the reward function. At each timestep, the agent takes an action $a \in \mathcal{A}$, the environment state transitions from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ according to T, and the agent receives an observation $z \in \mathcal{Z}$ from the environment according to O.

The agent typically maintains a *belief state* $b_t : \mathcal{S} \to [0, 1]$, a distribution over the states and a sufficient statistic for the history of past actions and observations $h_t = (az)_{1:t-1}$. The agent updates its belief after taking action a and receiving observation z: $b^{z,a}(s') = \eta O(z, s', a) \sum_s T(s', a, s) b(s)$, where η is the normalizing constant [36].

The solution to a POMDP is a *policy* π that maps a history to an action. The *value* of a POMDP at a history under policy

π is the expected discounted cumulative reward following that policy: $V_\pi(h) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma R(s_t, \pi(h_t)) | h_0 = h]$ where γ is the discount factor. The optimal value at the history is $V(h) = \max_\pi V_\pi(h)$.

B. COS-POMDP Definition

Given an instance of the correlational object search problem defined in Sec. III-B, we define the Correlational Object Search POMDP (COS-POMDP) as follows:

- **State space.** The state space $\mathcal{S}$ is factored to include the robot state $s_{\text{robot}} \in \mathcal{S}_{\text{robot}}$ and the target state $x_{\text{target}} \in \mathcal{X}$. A state $s \in \mathcal{S}$ can be written as $s = (s_{\text{robot}}, x_{\text{target}})$. Importantly, no other object state is included in $\mathcal{S}$.
- **Action space.** The action space is $\mathcal{A} = \mathcal{A}_m \cup \{\texttt{Done}\}$.
- **Observation space.** The observation space $\mathcal{Z}$ is factored over the objects, and each $z \in \mathcal{Z}$ is written as $z = (z_{\text{robot}}, z_{\text{objects}})$, where $z_{\text{objects}} = (z_1, \ldots, z_n, z_{\text{target}})$.
- **Transition model.** The objects are assumed to be static. Actions $a_m \in \mathcal{A}_m$ change the robot state from s_{robot} to s'_{robot} according to T_m, and taking the `Done` action terminates the task deterministically.
- **Observation model.** By definition of z, $\Pr(z|s) = \Pr(z_{\text{robot}}|s_{\text{robot}}) \Pr(z_{\text{objects}}|s)$ where $\Pr(z_{\text{robot}}|s_{\text{robot}})$ is defined by $O_{\text{robot}}(z_{\text{robot}}, s_{\text{robot}})$. Under the conditional independence assumption in Sec. III, $\Pr(z_{\text{objects}}|s)$ can be compactly factored:

$$\Pr(z_{\text{objects}}|s) = \Pr(z_1, \ldots, z_n, z_{\text{target}}|x_{\text{target}}, s_{\text{robot}}) \quad (1)$$

$$= \Pr(z_{\text{target}}|x_{\text{target}}, s_{\text{robot}}) \prod_{i=1}^{n} \Pr(z_i|x_{\text{target}}, s_{\text{robot}}) \quad (2)$$

The first term in Eq (2) is defined by D_{target}, and each $\Pr(z_i|x_{\text{target}}, s_{\text{robot}})$ is called a *correlational observation model*, written as:

$$\Pr(z_i|x_{\text{target}}, s_{\text{robot}}) = \sum_{x_i \in \mathcal{X}} \Pr(x_i, z_i|x_{\text{target}}, s_{\text{robot}}) \quad (3)$$

$$= \sum_{x_i \in \mathcal{X}} \Pr(z_i|x_i, s_{\text{robot}}) \Pr(x_i|x_{\text{target}}) \quad (4)$$

where the two terms in Eq (4) are the detection model $D_i \in \mathcal{D}$ and correlation model $C_i \in \mathcal{C}$, respectively.
- **Reward function.** The reward function, $R(s, a) = R(s_{\text{robot}}, x_{\text{target}}, a)$, is defined as follows. Upon taking `Done`, the task outcome is determined based on $s_{\text{robot}}, x_{\text{target}}$, which is successful if the robot orientation is facing the target and its position is within a distance threshold to the target. If successful, then the robot receives $R_{\text{max}} \gg 0$, and $R_{\text{min}} \ll 0$ otherwise. Taking a move action from $\mathcal{A}_m$ receives a negative reward which corresponds to the action's cost. In our experiments, we set $R_{\text{max}} = 100$ and $R_{\text{min}} = -100$. Each primitive move action (*e.g.*, `MoveAhead`) receives a step cost of -1.

C. Optimality of COS-POMDPs

The state space of a COS-POMDP involves only the robot and target object states. A natural question arises: have we lost any necessary information? In this section, we show that COS-POMDPs are optimal, in the following sense: if we imagine solving a "full" POMDP corresponding to the COS-POMDP, whose state space contains all object states, then the solutions to the COS-POMDP are equivalent. Note that a belief state in this "full" POMDP scales exponentially in the number of objects.

We begin by precisely defining the "full" POMDP, henceforth called the F-POMDP, corresponding to a COS-POMDP. The F-POMDP has identical action space, observation space, and transition model as the COS-POMDP. The reward function is also identical since it only depends on the target object state, robot state, and the action taken. F-POMDP differs in the state space and observation model:

- **State space.** The state is $s = (s_{\text{robot}}, x_{\text{target}}, x_1, \ldots, x_n)$.
- **Observation model.** Under the conditional independence assumption stated in Sec. III, the model for observation z_i of object x_i involves just the detection model: $\Pr(z_i|s) = \Pr(z_i|x_i, s_{\text{robot}})$.

Since the COS-POMDP and the F-POMDP share the same action and observation spaces, they have the same history space as well. We first show that given the same policy, the two models have the same distribution over histories.

Theorem 1. Given any policy $\pi : h_t \to a$, the distribution of histories is identical between the COS-POMDP and the F-POMDP.

Proof: (Sketch) We prove this by induction. When $t = 1$, the statement is true because both histories are empty. The inductive hypothesis assumes that the distributions $\Pr(h_t)$ is the same for the two POMDPs at $t \geq 1$. Then, by definition, $\Pr(h_{t+1}) = \Pr(h_t, a_t, z_t) = \Pr(z_t|h_t, a_t) \Pr(a_t|h_t) \Pr(h_t)$. Note that $\Pr(a_t|h_t)$ is the same under the given π. We can show the two POMDPs have the same $\Pr(z_t|h_t, a_t)$; the full proof is available in Appendix A.[1] ■

Using Theorem 1, we are equipped to make a statement about the value of following a given policy in either the COS-POMDP or the F-POMDP.

Corollary 1. Given any policy $\pi : h_t \to a$ and history h_t, the value $V_\pi(h_t)$ is identical between the COS-POMDP and the F-POMDP.

Proof: By definition, the value of a POMDP at a history is the expected discounted cumulative reward with respect to the distribution of future action-observation pairs. Theorem 1 states that the COS-POMDP and F-POMDP have the same distribution of histories given π. Furthermore, the reward function depends only on the states of the robot and the target object. Thus, this expectation is equal for the two POMDPs at any h. ■

Finally, we can show that COS-POMDPs are optimal in the sense that we described before.

Corollary 2. An optimal policy π^* for either the COS-POMDP or the F-POMDP is also optimal for the other.

Proof: Suppose, without loss of generality, that π^* is optimal for the COS-POMDP but not the F-POMDP. Let π' be the optimal policy for the F-POMDP. By the definition of optimality, for at least some history h we must have $V_{\pi'}(h) >$

[1] The appendix is available at https://arxiv.org/pdf/2110.09991.pdf

$V_{\pi^*}(h)$. By Corollary 1, for any such h the COS-POMDP also has value $V_{\pi'}(h)$, meaning π^* is not actually optimal for the COS-POMDP; this is a contradiction. ■

V. Hierarchical Planning

Despite the optimality-preserving reduction of state space in a COS-POMDP, directly planning over the primitive move actions is not scalable to practical domains even for state-of-the-art online POMDP solvers [38]. Nevertheless, planning POMDP actions at the primitive level has the benefit of controlling fine-grained movements, allowing goal-directed behavior to emerge automatically at this level. Therefore, we seek an algorithm that can reason about both searching over a large region and searching carefully in a local region.

To this end, we propose a hierarchical planning algorithm to apply COS-POMDPs to realistic domains (Fig. 2). The pseudocode and a detailed description is provided in Appendix B. As an overview: (1) A topological graph is first dynamically generated to reflect the robot's belief in the target location. Nodes are places accessible by the robot, and edges indicate navigability between places [39]. (2) Then, a high-level COS-POMDP is instantiated which plans *subgoals* that can be either navigating to another place, or searching locally at the current place. Both types of subgoals can be understood as viewpoint-changing actions, except the latter keeps the viewpoint the same. (3) At each timestep, a subgoal is planned using a POMDP solver, and a low-level planner is instantiated corresponding to the subgoal. This low-level planner then plans to output an action from the action set $\mathcal{A} = \mathcal{A}_m \cup \{\texttt{Done}\}$, which is used for execution. In our implementation, for navigation subgoals, an A* planner is used, and for the subgoal of searching locally, a low-level COS-POMDP is instantiated whose action space is the primitive movements $\mathcal{A}_m$, and solved using a POMDP planner [40]. (4) Upon executing the low-level action, the robot receives an observation from its on-board object detector. This observation is used to update the belief of both the high-level COS-POMDP as well as the low-level COS-POMDP (if it exists). (5) If the cumulative belief captured by the nodes in the current topological graph is below a threshold (50% in our experiments), then the topological graph is regenerated to better reflect the belief. (6) Finally, the process starts over from step (3). If the high-level COS-POMDP plans a new subgoal different from the current one, then the low-level planner is re-instantiated. Our algorithm plans actions for execution in an online, closed-loop manner, allowing for reasoning about viewpoint changes at the level of both places in a topological graph and fine-grained movements. This is efficient in practice because typical mobile robots can be controlled both at the low level of motor velocities and the high level of navigation goals [41, 42].

VI. Experimental Setup

We test the following hypotheses through our experiments: (1) Leveraging correlational information with easier-to-detect objects can benefit the search for hard-to-detect objects; (2) Optimizing over an action sequence improves performance compared to greedily choosing the next-best view.

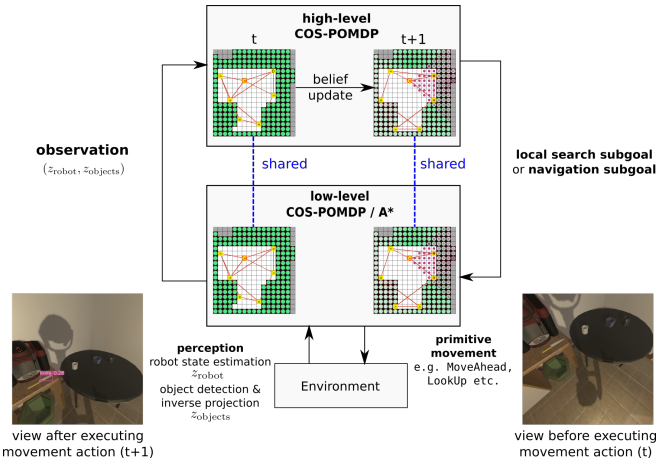

Fig. 2: **Illustration of the Hierarchical Planning Algorithm.** A high-level COS-POMDP plans subgoals fed into a low-level planner that produces low-level actions. The belief state is shared across the levels. Both levels plan with updated beliefs at every timestep.

A. AI2-THOR

We conduct experiments in AI2-THOR [19], a realistic simulator of in-household rooms. It has a total of 120 scenes divided evenly into four room types: *Bathroom*, *Bedroom*, *Kitchen*, and *Living room*. For each room type, we use the first 20 scenes for training a vision-based object detector and learning object correlation models (used in some experiments), and the last 10 scenes for validation.

The robot can take primitive move actions from the set: $\{\texttt{MoveAhead}, \texttt{RotateLeft}, \texttt{RotateRight}, \texttt{LookUp}, \texttt{LookDown}\}$. `MoveAhead` moves the robot forward by 0.25m. `RotateLeft`, `RotateRight` rotate the robot in place by 45°. `LookUp`, `LookDown` tilt the camera up or down by 30°. The robot observes the pose of its current viewpoint without noise. To be successful, when the robot takes `Done`, the robot must be within a Euclidean distance of 1.0m from the target object while the target object is visible in the camera frame. The maximum number of steps allowed is 100.

B. Object Detector

We use YOLOv5 [21], a popular vision-based object detector. This contrasts previous works evaluated using a ground truth object detector [33] or detectors with synthetic noise and detection ranges [2, 15]. We collect training data by randomly placing the robot in the training scenes.

Detection Model. Since vision detectors can sometimes detect small objects from far away, we consider a line-of-sight detection model with a limited field of view angle:

$$
\begin{aligned}
&D(z_i, x_i, s_{\text{robot}}) = \Pr(z_i | x_i, s_{\text{robot}}) \\
&= \begin{cases}
1.0 - \text{TP} & s_i \in \mathcal{V}(s_{\text{robot}}) \wedge z_i = \texttt{null} \\
\delta\text{FP}/|\mathcal{V}_E(r)| & s_i \in \mathcal{V}(s_{\text{robot}}) \wedge \|z_i - x_i\| > 3\sigma \\
\delta\mathcal{N}(z_i; x_i, \sigma^2) & s_i \in \mathcal{V}(s_{\text{robot}}) \wedge \|z_i - x_i\| \leq 3\sigma \\
1.0 - \text{FP} & s_i \notin \mathcal{V}(s_{\text{robot}}) \wedge z_i = \texttt{null} \\
\delta\text{FP}/|\mathcal{V}_E(r)| & s_i \notin \mathcal{V}(s_{\text{robot}}) \wedge z_i \neq \texttt{null}
\end{cases}
\end{aligned}
$$

This detection model is parameterized by: TP, the true positive rate; FP, the false positive rate; r, the average

Fig. 3: **Example Sequence.** Top: first-person view with object detection bounding boxes. Bottom: Visualization of belief state corresponding to each view. See Fig. 2 for the legend of the belief state visualization. Our method (COS-POMDP) successfully finds a `CreditCard` in a living room scene, leveraging the detection of other objects such as `FloorLamp` and `Laptop`. For more examples, please refer to the video at `https://youtu.be/RneTq4o0a-A`.

Fig. 4: Visualization of robot trajectory produced by different methods for the example shown in Fig. 3. A gray circle combined with a black line segment indicates a viewpoint.

distance between the robot and the object for true positive detections; σ, the width of a small region around the true object location where a detection made within that region, though not exactly accurate, is still accepted as a true positive detection. We set $\sigma = 0.5$m. The notation $\mathcal{N}(\cdot)$ denotes a Gaussian distribution. The $\mathcal{V}(s_{\text{robot}})$ denotes the line-of-sight field of view with a 90° angle. The $\mathcal{V}_E(r)$ denotes the region inside the field of view that is within distance r from the robot. The weight $\delta = 1$ if the detection is within $\mathcal{V}_E(r)$, and otherwise $\delta = \exp(-\|z_i - s_{\text{robot}}\| - r)^2$.

C. Target Objects

We choose the target and correlated object classes based on detection statistics. The list of target object classes and other correlated classes for each room type is listed below (in no particular order). For detection statistics, please refer to Table I and Table IV (Appendix D).

- **Bathroom.** *Targets* are `Fauct`, `Candle`, `ScrubBrush`; *Correlated* objects are `ToiletPaperHanger`, `Towel`, `Mirror`, `Toilet`, `SoapBar`.
- **Bedroom.** *Targets* are `AlarmClock`, `CellPhone`, `Book`; *Correlated objects* are `Laptop`, `Bed`, `DeskLamp`, `Mirror`, `LightSwitch`.
- **Kitchen.** *Targets*: `Bowl`, `Knife`, `PepperShaker`; *Correlated objects* are `Lettuce`, `LightSwitch`, `Microwave`, `Plate`, `StoveKnob`
- **Living room.** *Targets* are `CreditCard`, `RemoteControl`, `Television`; *Correlated objects* are `Pillow`, `Laptop`, `LightSwitch`, `HousePlant`, `FloorLamp`, `Painting`.

D. Correlation Model

We consider a binary correlation model that takes into account whether the correlated object and the target are close or far. Note that our method is not specific to this model. We use this model since it is applicable between arbitrary object classes and can be easily estimated based on object instances.

$$C(x_{\text{target}}, x_i) = \Pr(x_i | x_{\text{target}}) \tag{5}$$

$$= \begin{cases} 1 & \text{Close}(i, \text{target}) \wedge \|x_i - x_{\text{target}}\| < d(i, \text{target}) \\ 0 & \text{Close}(i, \text{target}) \wedge \|x_i - x_{\text{target}}\| \geq d(i, \text{target}) \\ 1 & \text{Far}(i, \text{target}) \wedge \|x_i - x_{\text{target}}\| > d(i, \text{target}) \\ 0 & \text{Far}(i, \text{target}) \wedge \|x_i - x_{\text{target}}\| \leq d(i, \text{target}) \end{cases} \tag{6}$$

where $\text{Close}(\cdot,\cdot)$ and $\text{Far}(\cdot,\cdot)$ are opposite, class-level predicates, $\|\cdot\|$ denotes the Euclidean distance, and $d(\cdot,\cdot)$ is the expected distance between the two object classes. In Sec. VII, we conduct an ablation study where $d(\cdot,\text{target})$ is estimated under different scenarios: **accurate**: based on object ground truth locations in the deployed scene; **estimated** (est): based on instances in training scenes; **wrong** (wrg): same as accurate except we flip the close/far relationship between the objects so that they do not match the scene.

E. Evaluation Metric

We use three metrics: (1) success weighted by inverse path length (SPL) [22]; (2) success rate (SR) and (3) dicounted cumulative rewards (DR). The SPL for each trial i is defined as $\text{SPL}_i = S_i \cdot \ell_i / \max(p_i, \ell_i)$ where S_i is the binary success outcome of the search, ℓ_i is the shortest path between the robot and the target, and p_i is the actual search path. The SPL measures the search performance by taking into account both the success and the efficiency of the search. As a stringent metric, ℓ_i uses information about the true object location, but it does not penalize excessive rotations [31]. Therefore, we also include the discounted cumulative rewards (DR) metric with $\gamma = 0.95$, which takes rotation actions into account.

F. Baselines

Baselines are defined in the caption of Table I. Note that *Greedy-NBV* is based on [2] where a weighted particle belief is used to estimate the joint state over all object locations. During planning, the robot selects the next best viewpoint to navigate towards based on a cost function that considers both navigation distance and the probability of detecting any object. This provides a baseline that is in contrast to the sequential decision-making paradigm considered by COS-POMDPs and the modeling of only robot and target states.

G. Implementation Details

Objects exist in 3D in AI2-THOR scenes. Since the robot can tilt its camera within a small range of angles, all methods (except *Random*) estimate target object height among a discrete set of possible height values, `Above`, `Below`, and `Same`, with respect to the camera's current tilt angle. POMDP-based methods are implemented using the `pomdp_py` [43] library with the POUCT planner [40]. The rollout policy uniformly samples from move actions towards the target or possibly leading to a non-`null` observation about an object.

Method	Bathroom			Bedroom			Kitchen			Living room		
	SPL (%)	DR	SR (%)	SPL (%)	DR	SR (%)	SPL (%)	DR	SR (%)	SPL (%)	DR	SR (%)
Random	0.00 (0.00)	-82.75 (3.43)	0.00	0.00 (0.00)	-81.51 (3.33)	0.00	6.90 (9.81)	-68.51 (15.61)	6.90	0.00 (0.00)	-82.37 (3.62)	0.00
Greedy-NBV	14.34 (9.12)	-19.86 (11.87)	34.48	16.92 (11.70)	-17.52 (7.32)	26.67	11.61 (8.72)	-17.60 (12.41)	31.03	7.13 (7.11)	-21.41 (8.21)	20.00
Target-POMDP	19.88 (9.47)	**-7.37 (12.42)**	**55.17**	19.79 (12.81)	-20.79 (11.29)	26.67	13.80 (8.67)	-20.17 (12.83)	34.48	24.36 (13.28)	-33.58 (11.88)	40.00
COS-POMDP	**30.64 (12.73)**	-14.48 (11.58)	**55.17**	**24.76 (12.95)**	**-15.57 (9.16)**	**40.00**	**20.45 (12.00)**	**-6.55 (12.73)**	**41.38**	**24.99 (13.95)**	**-14.08 (14.22)**	**43.33**
COS-POMDP (gt)	**31.08 (13.31)**	-13.47 (12.67)	**58.62**	**26.67 (13.13)**	**-11.09 (12.07)**	40.00	**35.58 (13.30)**	**-2.75 (14.37)**	**62.07**	**32.88 (14.25)**	**-13.81 (13.22)**	**56.67**
COS-POMDP (est)	17.20 (10.21)	-20.96 (10.75)	41.38	16.78 (11.68)	-31.60 (10.05)	30.00	8.39 (7.94)	-31.36 (13.42)	20.69	14.07 (10.65)	-43.76 (13.30)	26.67
COS-POMDP (wrg)	11.89 (8.14)	-16.55 (10.23)	27.59	14.70 (10.92)	-17.33 (8.38)	23.33	10.51 (8.02)	-20.68 (10.40)	27.59	**31.41 (14.50)**	-15.94 (9.45)	**53.33**

TABLE I: **Main and Ablation Study Results.** Unless otherwise specified, all methods use the YOLOv5 [21] vision detector and are given accurate correlational information. *Target-POMDP* uses hierarchical planning but only the target object is detectable. *Greedy-NBV* is a next-best view approach based on [2]. *Random* chooses actions uniformly at random. The highest value of each metric per room type is bolded. Parentheses contain 95% confidence interval. Ablation study results are bolded if it outperforms the best result from the main evaluation. *COS-POMDP* is more consistent, performing either the best or the second best across all room types and metrics.

Room Type	Target Class	TP	FP	r (m)	Greedy-NBV			Target-POMDP			COS-POMDP		
					SPL (%)	DR	SR (%)	SPL (%)	DR	SR (%)	SPL (%)	DR	SR (%)
Bathroom	Faucet	56.1	8.0	2.16	28.31 (19.58)	0.73 (22.10)	**70.00**	34.67 (22.86)	**8.00 (24.67)**	**70.00**	28.18 (27.25)	-23.27 (24.36)	50.00
	Candle	29.4	2.4	1.81	12.52 (20.12)	-22.81 (20.80)	22.22	16.56 (13.36)	-7.98 (28.99)	**66.67**	**33.89 (21.83)**	**-2.94 (19.08)**	**66.67**
	ScrubBrush	64.3	9.9	1.71	2.00 (4.52)	-37.79 (17.36)	10.00	8.09 (10.79)	-22.18 (13.51)	30.00	**30.18 (25.78)**	**-16.07 (22.13)**	**50.00**
Bedroom	AlarmClock	79.6	7.4	2.77	**39.49 (31.18)**	**-5.54 (18.07)**	**50.00**	14.31 (22.01)	-23.78 (14.43)	20.00	31.57 (30.85)	-15.85 (21.03)	40.00
	Book	62.6	4.9	2.05	8.42 (12.72)	-20.10 (11.71)	20.00	**29.70 (28.85)**	-13.94 (27.69)	40.00	25.92 (22.50)	**-12.56 (16.69)**	**50.00**
	CellPhone	50.0	3.9	1.69	2.85 (6.44)	-26.91 (5.88)	10.00	15.36 (23.21)	-24.64 (22.20)	20.00	**16.80 (21.48)**	**-18.29 (16.16)**	**30.00**
Kitchen	Bowl	60.6	11.5	1.75	19.88 (26.57)	-15.76 (32.76)	33.33	16.33 (16.00)	-10.06 (27.39)	**55.56**	**20.37 (20.70)**	**-3.33 (27.27)**	44.44
	Knife	37.7	8.7	1.68	7.40 (11.42)	-18.94 (23.71)	30.00	4.62 (10.45)	-36.36 (15.51)	10.00	**23.97 (25.58)**	**-2.59 (25.33)**	**50.00**
	PepperShaker	38.1	9.4	1.43	8.39 (10.53)	-17.90 (17.39)	30.00	**20.69 (21.03)**	**-13.07 (27.64)**	**40.00**	17.01 (24.19)	-13.41 (22.95)	30.00
Living room	Television	85.3	5.2	2.59	8.98 (18.36)	-22.86 (13.31)	20.00	**53.60 (26.06)**	**-8.63 (17.97)**	**80.00**	40.08 (32.14)	-12.22 (28.08)	50.00
	RemoteControl	69.6	4.5	1.93	9.24 (13.99)	-13.21 (20.44)	30.00	18.67 (24.17)	-38.38 (18.29)	30.00	**30.14 (28.99)**	**5.81 (25.29)**	**60.00**
	CreditCard	42.9	4.3	1.48	3.18 (7.19)	**-28.15 (11.70)**	10.00	0.82 (1.85)	-53.73 (20.32)	10.00	**4.74 (7.19)**	-35.84 (21.62)	**20.00**

TABLE II: **Detection Statistics and Object Search Results Grouped by Target Classes.** Target objects are sorted by average detection range (r). We estimated the values for TP, FP, and r by running the vision detector at 30 random camera poses per validation scene. *COS-POMDP* performs more consistently and robustly for hard-to-detect objects, such as `ScrubBrush`, `CellPhone`, `Candle`, and `Knife`.

VII. Results and Discussions

Our main results by room type are shown in Table I; results over all room types are in the appendix. The performance of *COS-POMDP* is more consistent compared to other baselines at either the best or the second best for all metrics in the four room types. The performance is broken down by target classes in Table II. *Greedy-NBV* performs well for `AlarmClock` in *Bedroom*; it appears to experience less instability in the particle belief as a result of particle reinvigoration. *COS-POMDP* appears to be the most robust when the target object has significant uncertainty of being detected correctly, including `ScrubBrush`, `CreditCard`, `Candle` `RemoteControl`, `Knife`, and `CellPhone`. An example search trial for `CreditCard` is visualized in Fig. 3. For target objects with a true positive detection rate below 40%, *COS-POMDP* improves the POMDP baseline that ignores correlational information by 42.1% in terms of the SPL metric ($p = 0.028$), and it is more than 2.1 times better than the greedy baseline ($p = 0.023$). Both results are statistically significant. Indeed, when the target object is reliably detectable, such as `Television`, the ability to detect multiple other objects may actually hurt performance, compared to *Target-POMDP*, due to noise from detecting those other objects and the influence on search behavior.

Ablation Studies. We also conduct two ablation studies. First, we equip *COS-POMDP* with a groundtruth object detector, as done in [33], henceforth called *COS-POMDP (gt)*. This shows the performance when the detections of both the target and correlated objects involve no noise at all. We observe better or competitive performance from using groundtruth detectors across all metrics in all room types. The gain over *COS-POMDP* in terms of SPL is not statistically significant ($p = 0.069$).

Additionally, we use correlations estimated using training scenes (*COS-POMDP (est)*) as well as incorrect correlational information that is the reverse of the correct one (*COS-POMDP (wrg)*). Indeed, using accurate correlations provides the most benefit, while correlations estimated through this naive method could offer benefit compared to using incorrect correlations in some cases (*Bathroom* and *Bedroom*), but can also backfire and hurt performance in others. Therefore, properly learning correlations is important, while leveraging a reliable source of information, for example, from a human at the scene, may offer the most benefit.

VIII. Conclusion and Future Work

In this paper, we formulated the problem of correlational object search as a POMDP (COS-POMDP), and proposed a hierarchical planning algorithm to solve it in practice. Our results show that, particularly for hard-to-detect objects, our approach offers more robust performance compared to baselines. Directions for future work include investigating different correlation models, searching in more complex settings that involve *e.g.*, container opening and dynamic objects, and evaluating on a real robot platform.

Acknowledgements: This work was supported by awards from Echo Labs, STRAC, and ONR under grant number N00014-21-1-2584. We sincerely thank Leslie Kaelbling and Tomás Lozano-Pérez for their critical and invaluable advice. We also thank Mitchell Wortsman, Yiding Qiu, and Anwesan Pal, and the AI2-THOR developers for helpful clarifications.

REFERENCES

[1] D. Sprute, A. Pörtner, R. Rasch, S. Battermann, and M. König, "Ambient assisted robot object search," in *International Conference on Smart Homes and Health Telematics*. Springer, 2017.

[2] Z. Zeng, A. Röfer, and O. C. Jenkins, "Semantic linking maps for active visual object search." IEEE, 2020, pp. 1984–1990.

[3] M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, "Automated hyperspectral cueing for civilian search and rescue," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1031–1055, 2009.

[4] J. Sun, B. Li, Y. Jiang, and C.-y. Wen, "A camera-based target detection and positioning uav system for search and rescue (SAR) purposes," *Sensors*, vol. 16, no. 11, p. 1778, 2016.

[5] I. Idrees, S. P. Reiss, and S. Tellex, "Robomem: Giving long term memory to robots," *arXiv preprint arXiv:2003.10553*, 2020.

[6] M. R. Loghmani, T. Patten, and M. Vincze, "Towards socially assistive robots for elderly: An end-to-end object search framework," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2018.

[7] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 986–1002, 2013.

[8] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2168–2173.

[9] L. E. Wixson and D. H. Ballard, "Using intermediate objects to improve the efficiency of visual search," *International Journal of Computer Vision*, vol. 12, no. 2-3, pp. 209–230, 1994.

[10] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2818–2824.

[11] J. K. Li, D. Hsu, and W. S. Lee, "Act to see and see to act: POMDP planning for objects search in clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[12] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *Proceedings of the International Conference on Robotics and Automation*, 2019.

[13] K. Zheng, Y. Sung, G. Konidaris, and S. Tellex, "Multi-resolution POMDP planning for multi-object search in 3D," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[14] A. Wandzel, Y. Oh, M. Fishman, N. Kumar, and S. Tellex, "Multi-object search using object-oriented POMDPs," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[15] L. Holzherr, J. Förster, M. Breyer, J. Nieto, R. Siegwart, and J. J. Chung, "Efficient multi-scale POMDPs for robotic object search and delivery," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6585–6591.

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[17] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010.

[18] K. Zheng, D. Bayazit, R. Mathew, E. Pavlick, and S. Tellex, "Spatial language understanding for object search in partially observed cityscale environments," in *International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2021.

[19] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[21] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[22] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[23] Y. Ye and J. K. Tsotsos, "Sensor planning in 3d object search: its formulation and complexity," in *The 4th International Symposium on Artificial Intelligence and Mathematics, Florida, USA*. Citeseer, 1996.

[24] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Manipulation-based active search for occluded objects," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2814–2819.

[25] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8338–8344.

[26] S. S. Brown, "Optimal search for a moving target in discrete time and space," *Operations research*, vol. 28, no. 6, pp. 1275–1289, 1980.

[27] T. D. Garvey, "Perceptual strategies for purposive vision," *Thesis Ph.D. Stanford University*, 1976.

[28] M. Lorbach, S. Höfer, and O. Brock, "Prior-assisted propagation of spatial information for object search," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.

[29] S. Amiri, K. Chandan, and S. Zhang, "Reasoning with scene graphs for robot planning under partial observability," *IEEE Robotics and Automation Letters (accepted)*, 2022.

[30] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[31] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[32] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] Y. Qiu, A. Pal, and H. I. Christensen, "Learning hierarchical relationships for object-goal navigation," in *2020 Conference on Robot Learning (CoRL)*, 2020.

[34] B. Mayo, T. Hazan, and A. Tal, "Visual navigation with spatial attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 898–16 907.

[35] Y. Lee, P. Cai, and D. Hsu, "MAGIC: Learning Macro-Actions for Online POMDP Planning ," in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[36] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[37] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, 2013.

[38] H. Kurniawati, "Partially observable markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, 2022.

[39] K. Zheng and A. Pronobis, "From pixels to buildings: End-to-end probabilistic deep networks for large-scale semantic mapping," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3511–3518.

[40] D. Silver and J. Veness, "Monte-carlo planning in large POMDPs," in *Neural Information Processing Systems*, 2010.

[41] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2718–2725.

[42] K. Zheng, "ROS Navigation Tuning Guide," in *Robot Operating System (ROS)*, A. Koubaa, Ed. Springer International Publishing, 2021, ch. 6, pp. 197–226.

[43] K. Zheng and S. Tellex, "pomdp_py: A framework to build and solve POMDP problems," in *ICAPS 2020 Workshop on Planning and Robotics (PlanRob)*, 2020.