2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

Inference Based Multi-Object Reactive Search in a Partially Known Environment with Temporal Logic Specifications

Yaohui Kang, Ziyang Chen, Yanjie Xia, and Zhen Kan

***Abstract*— Efficiently searching for multiple objects in a partially known environment, where only the names and locations of landmarks are available, presents significant challenges. Existing search algorithms in the literature fail to fully utilize prior knowledge to improve search efficiency, and exhibit significantly diminished efficiency when extended to multi-object search. To address these limitations, we propose an inference-based multi-object reactive search framework. This framework utilizes the COMET inference model to reason about co-occurrence values between the target objects and known landmarks, thereby enhancing search efficiency. These co-occurrence values are integrated into a reactive temporal logic motion planning strategy, which allows the robot search for multiple objects with temporal logic constraints specified by LTL and adapt dynamically if the inferred reasoning differs from the actual object arrangement encountered during the search. Extensive simulations were conducted to evaluate the feasibility and efficiency of the proposed motion planning algorithm. Results demonstrate that the integration of commonsense reasoning with reactive temporal logic planning significantly improves multi-object search efficiency. Project website: https://sites.google.com/view/imors.**

I. Introduction

Object search, which involves navigating in an unknown or partially known environment to locate a target object, has numerous applications, such as search and rescue operations [1], security screening tasks [2], and household services [3]. As illustrated in Fig. 1, the robot is assigned the task of searching for multiple objects in unknown locations. Only the names and locations of landmarks in the environment are known a priori. Such a task presents several challenges. First, as the number of objects to be located increases, the complexity and difficulty of the search grow substantially. How can a robot efficiently perform a search of multiple objects with temporal and logic constraints? Second, in environments that are only partially known, how can a robot utilize prior knowledge through commonsense reasoning to improve navigation and search efficiency? Third, given that commonsense reasoning-based knowledge may not always align with new contexts, what methods can support online re-planning to resolve this inconsistency? Motivated by these challenges, this work aims at introducing commonsense reasoning and exploiting reactive temporal logic planning to facilitate multi-object search in a partially known environment.

Y. Kang, Z. Chen, and Z. Kan (corresponding author) are with the Department of Automation at the University of Science and Technology of China, Hefei, Anhui, China, 230026.

Y. Xia is with the School of Information and Artificial Intelligence at the Wuhu Institute of Technology, Wuhu, Anhui, China, 241003.

This work was supported by the National Natural Science Foundation of China under Grant 62173314.

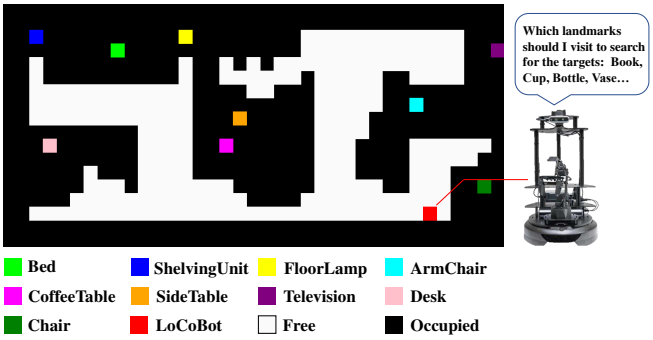

Fig. 1. The robot is tasked to search for multiple target objects with known landmark information.

Recent object search approaches have incorporated prior knowledge of the environment, significantly reducing the exploration space. The most relevant works to ours are [3]–[6], which assume that the locations of landmarks are known a priori and exploit the spatial relationships between the target object and these landmarks to prioritize search areas. Since long-range object search inherently involves continuous decision-making under partial observability, [5] employs co-occurrence probabilities between target objects and landmarks to model the search problem within the framework of a partially observable Markov decision process (POMDP). However, the effectiveness of POMDP-based methods can be limited by their dependence on precise object relevance information, which may not always be available [7]. To derive more accurate real-world object relationships, [4] utilized object co-occurrences derived from image tags. Similarly, [8] used word embedding trained on the Visual Genome dataset [9], employing cosine similarity between embedding vectors to estimate object relationships. Building on this, [3] trained a graph neural network on the Visual Genome dataset [9] to learn object co-occurrence relationships, while [6] trained a graph neural network on the ScanNet dataset [10] to acquire commonsense knowledge about object interactions. In contrast to training a graph neural network from scratch, [11] directly leveraged the COMET model [12], which is trained on the ATOMIC2020 knowledge graph [13]—comprising 1.33 million commonsense knowledge tuples—to generate object relationships. However, these methods are limited to single-object search, whereas our proposed inference based searching method extends it to multi-object search tasks.

Temporal logic motion planning, which incorporates both temporal and Boolean requirements, has emerged as a prominent approach for managing complex and extended

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11127453

sequences of robot tasks [14]–[16]. Its flexibility suggests it could also effectively generalize from a single target to multiple targets through a formal representation of the search task. However, the aforementioned methods generally assume that the robot operates within a known environment or at least known destinations. Consequently, these methods are not applicable to object search tasks in environments that are completely or partially unknown. To address this limitation, several reactive temporal logic planning algorithms have been developed to manage map uncertainty [17]–[20]. Notably, [17] modeled the environment as a partially known initial transition system and design discrete controllers using graph search methods applied to a product automaton. As the environment evolves, the product automaton is updated locally, and new paths are re-planned using graph search on the revised automaton. A similar methodology was employed in [18]. Additionally, [19] proposeed methods for locally patching paths in response to changes in transition systems, ensuring compliance with GR(1) specifications. Furthermore, planning algorithms responsive to Linear Temporal Logic specifications have been introduced in [20], where the robot dynamically adjusts its behavior in response to environmental changes, with task specifications explicitly designed to accommodate this adaptive reactivity. Unlike our approach, these methods are dependent on discrete abstractions of robot dynamics, potentially creating a disconnect between the discrete plans and their physical low-level implementation.

To address the challenges of multi-object search in a partially known environment, the inference-based multi-object reactive search framework is proposed in this work. Specially, the inference model COMET is utilized to reason about the co-occurrence possibility of assorted targets and known landmarks within the workspace. A discrete abstraction of the environment, i.e., the transition system, is then constructed, based on which an initial search plan can be generated using sampling-based approach. During the execution of the multi-object search task, the robot will update the object relationship according to the obtained perceptual information and reactively updates its plan, if necessary, according to the current task progress.

The contributions are summarised as follows. First, temporal logic planning is exploited for multi-object search tasks in a partial known environment (i.e., only the locations of landmarks are known, but the locations of target objects to be searched are not known a priori). Second, an object-level relational reasoning method based on commonsense knowledge graphs is proposed, which enhances the accuracy and reliability of real-world object relations and improving search efficiency by leveraging the spatial relationships between target objects and landmarks. Third, we present a perception-based reactive planning method capable of synthesizing online control strategies for the robot engaged in multi-target search tasks within semantically uncertain environments. Additionally, extensive simulation are performed to validate the feasibility and efficiency of the proposed algorithm.

II. Preliminaries and Problem Formulation

A. Linear Temporal Logic

As a formal language, LTL is defined over a set of atomic propositions AP with Boolean and temporal operators. The syntax of LTL is defined as:

$$\phi := true | ap | \phi_1 \wedge \phi_2 | \neg\phi_1 | X\phi | \phi_1 U \phi_2$$

where $ap \in AP$ is an atomic proposition, *true*, $\neg$ (negation), and $\wedge$ (conjunction) are propositional logic operators, and X (next) and U (until) are temporal operators. Other propositional logic operators such as *false*, $\vee$ (disjunction), $\rightarrow$ (implication), and temporal operators such as G (always) and F (eventually) can also be defined [21].

Let $(2^{AP})^\omega$ denote the set of words that arises from the infinite concatenation of words in 2^{AP}, where 2^{AP} represents the power set of AP. The word $\boldsymbol{\pi} = \pi_0\pi_1\ldots$ is an infinite sequence where $\pi_i \in 2^{AP}$, $\forall i \in \mathbb{Z}_{\geq 0}$. Given a word $\boldsymbol{\pi} = \pi_0\pi_1\ldots$, denote by $\boldsymbol{\pi}[j\ldots] = [\pi_j\pi_{j+1}\ldots]$, $\boldsymbol{\pi}[\ldots j] = [\pi_0\ldots\pi_j]$, and $\boldsymbol{\pi}[i:j] = [\pi_i\ldots\pi_j]$ with $i < j$. The semantics of LTL formulae are interpreted over $\boldsymbol{\pi}$ as:

$$\begin{aligned}
&\boldsymbol{\pi} \models true \\
&\boldsymbol{\pi} \models ap && \Leftrightarrow ap \in \pi_0 \\
&\boldsymbol{\pi} \models \phi_1 \wedge \phi_2 && \Leftrightarrow \boldsymbol{\pi} \models \phi_1, \boldsymbol{\pi} \models \phi_2 \\
&\boldsymbol{\pi} \models \neg\phi && \Leftrightarrow \pi \not\models \phi \\
&\boldsymbol{\pi} \models X\phi && \Leftrightarrow \pi[1\ldots] \models \phi \\
&\boldsymbol{\pi} \models \phi_1 U \phi_2 && \Leftrightarrow \exists i \text{ s.t. } \pi_i \models \phi_2, \forall j \in [0,i), \pi_j \models \phi_1
\end{aligned}$$

More details about LTL syntax, semantics, and model checking are referred to [21]. An LTL formula can be converted to a Non-deterministic Büchi Automata (NBA) [22].

Definition 1. An NBA is a tuple $B = (S, S_0, \Sigma, \rightarrow_B, S_F)$, where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, $\Sigma = 2^{AP}$ is the finite alphabet, $\rightarrow_B \subseteq S \times \Sigma \times S$ is the state transition, and $S_F \subseteq S$ is the set of accepting states.

Let B_ϕ denote the NBA of the LTL formula ϕ. Let $\Delta : S \times S \rightarrow 2^\Sigma$ denote the set of atomic propositions that enables state transitions in NBA, i.e., $\forall\pi \in \Delta(s,s'), (s,\pi,s') \in \rightarrow_B$. A valid run $\boldsymbol{s} = s_0 s_1 s_2 \ldots$ of B_ϕ generated by the word $\boldsymbol{\pi}$ with $\pi_i \in \Delta(s_{i-1}, s_i)$ is called accepting, if $\boldsymbol{s}$ intersects with S_F infinitely often. An LTL formula can be translated to an NBA by the tool [23]. In this paper, NBA will be used to track the progress of the satisfaction of LTL-based tasks.

B. System Models

The environment model is defined as a tuple $\Omega = (M, O, C, L)$, where $M \subset \mathbb{R}^2$ is a bound workspace, O denotes the set of objects in the workspace, and $C : O \times O \rightarrow \mathbb{R}$ indicates the co-occurrence value between objects. The labeling function $L : M \rightarrow O$ indicates the object $o \in O$ associated to a position $p \in M$, i.e., $L(p) = o$.

The goal of this work is to efficiently search for multiple objects with temporal logic constraints specified by an LTL formula ϕ in a partially known environment. Specially, the set of names of N targets is defined as $O_t \subset O$, while the

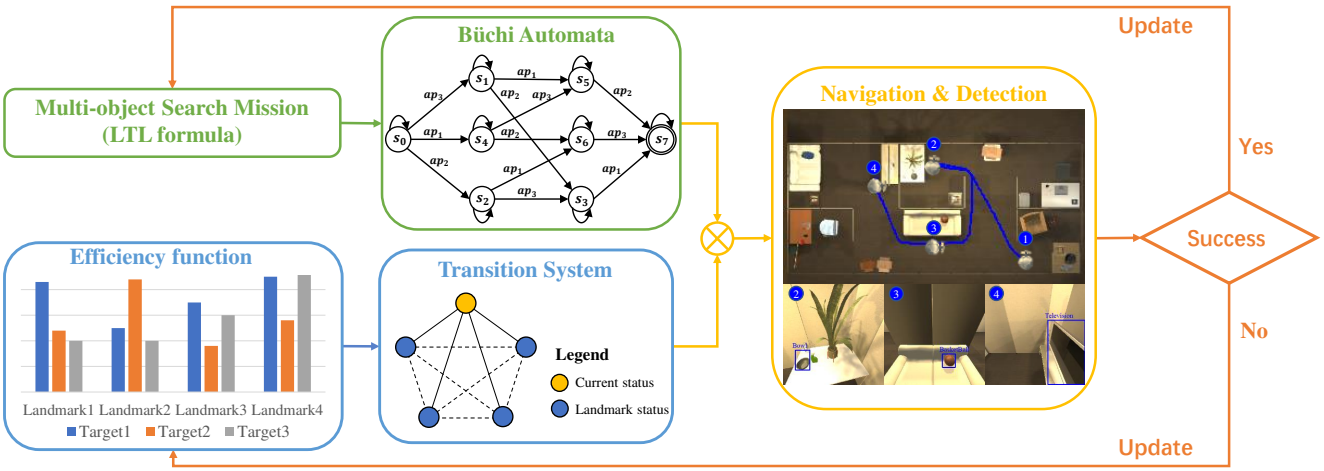

Fig. 2. **Illustration of the proposed inference-based multi-object reactive search algorithm pipeline.** The robot is required to complete a multi-object search task defined by Linear Temporal Logic (LTL), assuming that the locations of landmarks are known a priori. First, the robot determines the most efficient landmarks for target search based on inference, and then constructs a transition system by incorporating the landmark position information. Second, a sampling-based temporal logic planning approach is employed to generate a sequence of landmark locations. Following this, the robot visits landmarks and searches for target objects using the object detection method. Finally, online reactive planning is performed based on the detection results. If the target is successfully detected, the LTL task is updated; otherwise, the co-occurrence value between the target and the landmark is updated. Subsequently, the robot re-plans for a new round of searches.

set of names of K landmarks is defined as $O_l \subset O$. The locations of landmarks are denoted by $P_l \subset M$. It is assumed that the names and locations of landmarks are known a priori. Denote by $L_o : AP \to O_t$ a mapping that indicates the target object $o^t \in O_t$ associated with the atomic proposition ap in ϕ, i.e., $L_o(ap) = o^t$.

The robotic system is defined as $A = (v_r, p_r, O_r)$, where v_r and p_r are the linear speed and position of the robot, respectively. $O_r \subset O$ indicates the objects observed by the robot. To facilitate task planning, the transition system is constructed to relate the workspace and atomic propositions.

Definition 2. The transition system (TS) is defined as a tuple $T = (Q, M, AP, LA, W)$, where $Q \subset M$ is a finite set of interested system states in the workspace M, and $q \in Q$ is set as the position of a landmark, i.e. $L(q) \in O_l$, which can be initialized by P_l. AP is the set of the atomic propositions, $LA : Q \to AP$ indicates the atomic proposition associated with the state $q \in Q$, and $W : Q \times Q \to \mathbb{R}^+$ represents the transition cost, e.g., given $q_i, q_j \in Q$, $W(q_i, q_j) = \|q_i - q_j\|$.

C. Plan

Given the workspace M and the NBA B_ϕ generated by the LTL task ϕ, the plan is defined as a tuple $\Pi = (\boldsymbol{q}, \boldsymbol{s})$, where $\boldsymbol{q} = q_0 q_1 q_2 \ldots$ is the state trajectory of TS with $q_i \in Q$ and q_0 indicating the initial position of the robot, and $\boldsymbol{s} = s_0 s_1 s_2 \ldots$ is the trajectory of B_ϕ states corresponding to $\boldsymbol{q}$ where s_i indicates the automaton state after the atomic task is completed at q_i. By denoting $\Pi_i = (q_i, s_i)$, $i \in \mathbb{N}$, we can rewrite $\Pi = \Pi_0 \Pi_1 \Pi_2, \ldots$ as a series of plan tuples. Given a plan $\Pi = (\boldsymbol{q}, \boldsymbol{s})$, it is said to satisfy the task specification ϕ, if there exists $LA(q_0)LA(q_1)LA(q_2)\ldots \models \phi$, and $L_o(LA(q_i))$ can be detected when the robot reach near the position q_i.

Based on the prefix-suffix structure, the plan Π can be further written in the form of $\Pi = \Pi_{pre}\Pi_{suf}\Pi_{suf}\ldots$, where Π_{pre} and Π_{suf} are finite prefix and finite cyclic suffix plan with the same accepting NBA state at the end, respectively. Therefore, we only need to determine Π_{pre} and Π_{suf} in Π. For the finite plan $\Pi_{finite} = \Pi_{pre}\Pi_{suf}$, its cost is defined as $\mathsf{Cost}(\Pi_{finite}) = \sum_{i=1}^{|\Pi_{finite}|} W(q_{i-1}, q_i)$. Then, the problem can be summarized as follows.

Problem 1. Given the task specification ϕ and the priori knowledge O_l, P_l, how to get a task plan Π_{finite} with minimum cost $\mathsf{Cost}(\Pi_{finite})$.

III. Inference-Based Multi-Object Reactive Search

To address Prob. 1, this section presents an inference-based multi-object reactive search framework. Specially, based on the inference model, the transition system on known landmarks can be constructed and an initial plan can be obtained. While executing the initial plan, the correlation of landmarks and target objects will be updated according to the perceptual information, facilitating the replan of the search strategy. The overall framework of the algorithm is illustrated in Fig. 2.

A. Inference

To improve the efficiency of mobile robots in exploring unknown environments and searching for multiple objects, commonsense knowledge is leveraged to logically infer the potential locations of target objects. In this work, COMET [12] is utilized to construct commonsense knowledge bases. Specifically, the COMET model learns to generate novel and diverse commonsense knowledge tuples by adapting the weights of language models. It subsequently transfers implicit knowledge from deep pre-trained language models to

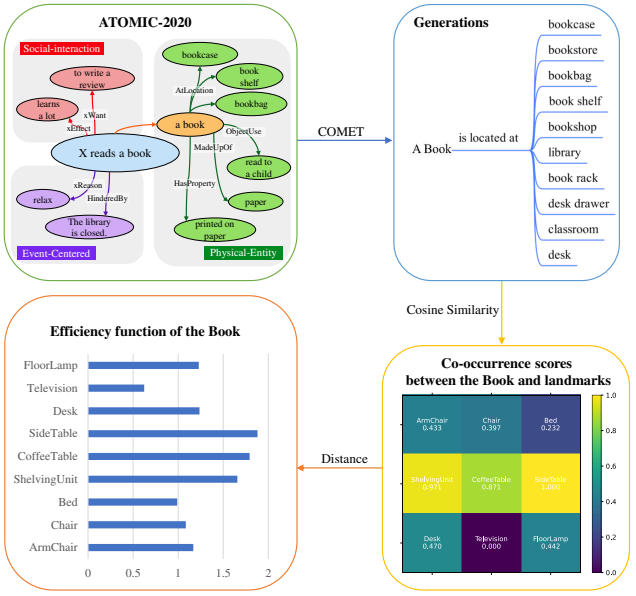

Fig. 3. **Illustration of the commonsense reasoning process.** Taking the target $Book$ as an example, we extract up to 10 generations from the COMET. Following the calculation of the cosine similarity of the word-embedding feature between generations and the landmark, the resulting value is averaged to obtain the co-occurrence value between the target and the landmark. Finally, Eq. 2 is applied to determines the most efficient landmark $SideTable$ for searching the $Book$.

generate explicit knowledge within commonsense knowledge graphs.

To enhance the reliability of the inferred commonsense knowledge, we follow the approach of [11] and train COMET BART using the ATOMIC2020 dataset [13]. ATOMIC2020 is a high-quality knowledge graph containing 1.33 million commonsense knowledge tuples, covering 23 commonsense relations across social, physical, and other aspects of everyday reasoning. This dataset is particularly well-suited for training knowledge models that can generate accurate and representative knowledge for new, unseen entities and events. In order to obtain the nearest landmark to the target object o^t, the format [Target name] [At Location] [GEN] is applied as the input for COMET, and extract up to k generations $\{G\left(o^t\right)_i\}_{i=1}^k$, indicating the potential locations of the target. Subsequently, the mean word similarity

$$C\left(o^l|o^t\right) = \frac{1}{k}\sum_{i=1}^{k} l\left(w\left(o^l\right), w\left(G\left(o^t\right)_i\right)\right) \tag{1}$$

between the landmark and all generations is taken as the co-occurrence value of the target object o^t and the landmark object o^l, where $l\left(\cdot\right)$ represents the cosine similarity and $w\left(\cdot\right)$ represents the word-embedding feature [24].

As illustrated in Fig. 3, the landmarks used to search for target objects are generated using a greedy algorithm based on an efficiency function. Specifically, given the target object o^t, the efficiency function between the current coordinate p_r and the location of the landmark p_l is defined as

$$E(p_r, p_l) = C\left((o^l|o^t\right) + \gamma(1 - \frac{W(p_r, p_l)}{d_{max}}) \tag{2}$$

where $C\left((o^l|o^t\right)$ is the co-occurrence value between the target object o^t and the landmark object o^l associated to the position p_l, γ is a positive weight of the distance factor, d_{max} is the maximum Euclidean distance between any two objects on the map.

B. Sampling-based Temporal Logic Planning

After identifying the most efficient landmarks for locating target objects via the commonsense reasoning introduced in Sec. III-A, the environment can be abstracted into a transition system T and proceed with path planning using the sampling method described in [25], which is outlined in Algorithm 1. Let F_f denote the set of initial and terminal states of the cyclic suffix segments. First, based on the transition system T and the NBA B_ϕ, the prefix plan $\Pi_{pre}(s)$, with $s \in S_F$ as the terminal state, is generated (lines 1-4). Any accept state without a feasible plan is subsequently removed. Then, lines 5-7 demonstrate how a feasible cyclic suffix segment is obtained, and any accept states that cannot form a state loop are eliminated. Finally, the plan $\Pi_{pre}(s_{min})\Pi^*_{suf}(s_{min})$ with the minimum cost value is selected and returned as the final plan, i.e., the sequence of landmark locations.

Algorithm 1: Get_Plan

Input: TS T, NBA B, current NBA state s_c
Output: $\Pi_{pre}\Pi_{suf}$

```
1  Initialize F_f = S_F;
2  for s ∈ F_f do
3      Π*_pre(s) = Sampling(s_c, s, T, B);
4      if Π*_pre(s) does not exist then
5          Delete s from F_f;
6      end
7  end
8  for s ∈ F_f do
9      Π*_suf(s) = Sampling(s, s, T, B);
10     if Π*_suf(s) does not exist then
11         Delete s from F_f;
12     end
13 end
14 Select Π*_pre(s_min)Π*_suf(s_min) satisfies ∀s ∈ F_f,
     Cost(Π*_pre(s_min)Π*_suf(s_min)) ≤ Cost(Π*_pre(s)Π*_suf(s));
```

Given the path plan, the robot will visit landmarks and search for target objects using the object detection module and various viewpoints obtained by adjusting the camera's horizontal and rotation values.

C. Reactive temporal logic planning

Although our commonsense reasoning approach can establish reliable relationships between objects, in entirely novel environments, the target object's location may be disordered and deviate from commonsense expectations. Consequently, the target object might not be found in the vicinity of landmarks derived from commonsense reasoning. To address this, an online reactive temporal logic planning framework

is developed, as outlined in Alg. 2. First, the NBA B and TS T are initialized, and the current NBA state s_c is set as s_0. The plan can be obtained by the function Get_Plan, and the robot navigate in the environment following Get_Plan. Once the robot reaches the proximity of the landmark within the distance ϵ, we assume that the object O_r can be sensed by the robot. If the current target object $L_o(LA(q_i))$ in the plan can be detected by the sense $O_r(p_r)$, the sub-task can be performed and the NBA state will be updated to the next in the $\Pi_{pre}\Pi_{suf}$. Otherwise, the co-occurrence value between the target and the landmark, and thus the TS, will be updated. Subsequently, re-planning will be performed based on the current NBA state s_i and the updated TS T for a new round of searches. For each target object, the terminal state is defined as either being located or having visited all associated landmarks without success. The search task is deemed complete when all target objects have reached their respective terminal states or when the robot exhausts its distance threshold. If all target objects are detected by the robot within the distance threshold, the search task is considered successful; otherwise it fails.

Algorithm 2: Controller

Input: Ω
Output: Π
Initialize B, T based on ϕ, Ω;
Initialize the task state $s_c = s_0$;
$\Pi_{pre}\Pi_{suf} = \mathsf{Get\_Plan}(B, T, s_c)$;
while *1* **do**
  $\Pi_{pre}\Pi_{suf} = \mathsf{Get\_Plan}(B, T, s_i)$;
  $\mathsf{Control}(p_{target}, A)$, where $p_{target} = q_i$;
  Obtain $O_r(p_r)$;
  if *near target landmark, i.e.* $\|p_r - p_{target}\| < \epsilon$ **then**
    if *target object* $o = L_o(LA(q_i)) \in O_r(p_r)$ **then**
      Update $p_{target} \to q_{i+1}$ based on $\Pi_{pre}\Pi_{suf}$;
    else
      for $o^* = L(q)$, $C(o, o^*) = 0$;
      Update LA in T based on C;
    end
  end
end

IV. Experiments and Results

In this section, we present simulations to validate the effectiveness of our inference-based multi-object reactive search algorithm. The near-realistic, interactive AI2-THOR framework [26] is employed in the simulation, which provides diverse scenarios across multiple rooms, as depicted in Fig. 4. The results demonstrate that (1) leveraging object co-occurrence relationships significantly improves the efficiency of object navigation; and (2) integrating object navigation with reactive temporal logic planning facilitates the successful and efficient search of multiple objects.

A. Simulation

An example simulation is conducted to validate the effectiveness of the proposed inference-based multi-object reactive search method. In this scenario, the robot knows the names and locations of four landmarks and is tasked with searching for three objects. The set of landmark is defined as $O_l = \{SideTable, DiningTable, CoffeeTable, Sofa\}$, and the corresponding locations are $P_l = \{(8.93, -3.36), (4.72, -1.60), (1.38, -3.50), (1.18, -4.20)\}$. The multi-target search task is formalized using the following LTL formula: $\phi := Fap_1 \wedge Fap_2 \wedge Fap_3$, where ap_1 represents the task of locating a $Book$, ap_2 represents the objective of finding a $HousePlant$, and ap_3 represents the task of identifying an $Apple$. The co-occurrence scores between the landmark objects and target objects are computed based on commonsense reasoning, as presented in Table I.

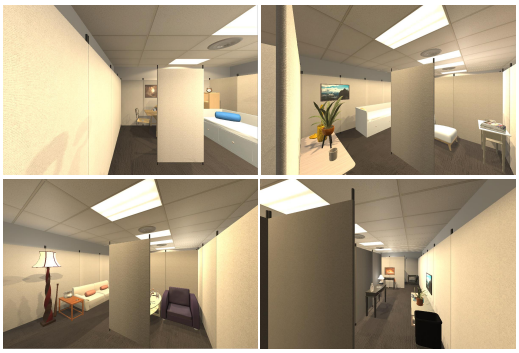

Fig. 4. **Simulated Scenes.** The images are from the FloorPlan_Val{2}_{2:5} scenes in RoboTHOR, which is an environment within the AI2-THOR framework.

TABLE I
Co-occurrence Scores between Targets and Landmarks

	SideTable	DiningTable	CoffeeTable	Sofa
Book	1.0	0.861	0.0	0.588
HousePlant	1.0	0.334	0.161	0.0
Apple	0.252	0.678	1.0	0.0

Fig. 5 illustrates the robot's trajectory in the simulated scenario. Notably, after locating the $Book$ on the $SideTable$ at $t = 22$, the robot's next objective is to search for the $HousePlant$. Despite being closest to the $Sofa$, the robot bypasses this landmark due to its low co-occurrence score with the $HousePlant$. Similarly, following the unsuccessful detection of the $Apple$ on the $SideTable$ at $t = 245$, the system adopts the same strategy, skipping the $Sofa$ and instead directing its search to the $CoffeeTable$, which has a higher co-occurrence score with the $Apple$, thus optimizing the robot's path.

B. Ablation Study and Analysis

To better understand the importance of the two core components in the inference-based multi-object reactive search method, additional ablation simulations were conducted. In these simulations, each experimental episode involved searching for nine target objects under the condition that the names and locations of 19 landmark objects were known a priori. With respect to the reactive temporal logic planning

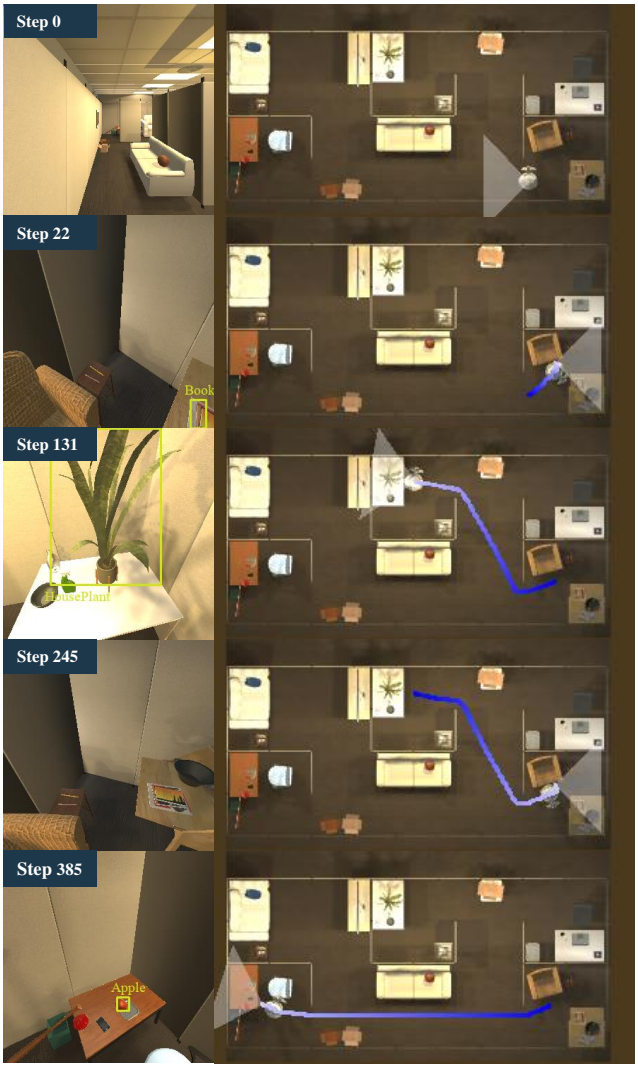

Fig. 5. **A rollout of an search episode.** A top-down view and RGB observation are provided for visualization. Following the proposed methodology, the robot's initial objective is to navigate to the $SideTable$ in search of the $Book$. At $t = 22$, the robot successfully locates the $Book$. The task state is then updated, and the robot continues searching for the $HousePlant$ on the $SideTable$, but no detections are made. The co-occurrence score is updated, and the transition system (TS) is reconstructed. At $t = 131$ the robot successfully finds the $HousePlant$ on the $DiningTable$, triggering another task state update. At $t = 245$ the robot moves to the $SideTable$ to search for the $Apple$ but is unsuccessful. The robot then proceeds to the $CoffeeTable$, where it locates the $Apple$, successfully concluding the search task.

algorithm, a sequential search strategy that follows the order of the targets was employed. Additionally, the impact of the proposed object-relational reasoning approach on search efficiency was evaluated by comparing it to methods based on word similarity [24] and nearest-distance search. To assess the effectiveness of the proposed methods, two commonly used object search evaluation metrics were employed: SPL (Success weighted by Path Length) [27] and Progress (the proportion of target objects successfully located) [28].

As shown in Table II, the proposed method for inferring object relationships using COMET outperforms the other two inference methods, showing an average improvement of

TABLE II

ABLATION STUDIES AND METHODOLOGICAL ANALYSIS

Ablations	SPL	Progress(%)
Relational Inference(Distance-Nearest)	0.161	60.9
Relational Inference(Word Similarity [24])	0.283	68.7
w/o Reactive Temporal Logic Planning	0.114	**75.9**
Inference-Based Multi-Object Reactive Search	**0.344**	72.6
Oracle Relationships	0.438	79.2

0.122 in SPL and 15.6% in Progress. This indicates that the proposed inference approach more accurately captures object-landmark relationships, resulting in more efficient searches. Although the Progress metric of the proposed method without reactive temporal logic planning showed only slight improvement, its search efficiency dropped significantly, with a 66.9% reduction in SPL. This underscores the critical role of global temporal logic planning in enhancing multi-objective search efficiency, more so than relational inference alone.

The proposed model demonstrates sub-optimal performance, primarily due to two factors: inaccurate object relationships and the absence of optimal global planning. To quantify the impact of these issues, oracle object relationships were derived using information provided by the AI2-THOR simulator for each object. This resulted in an SPL of 0.438 compared to 0.344, suggesting that approximately 27.3% of the performance could be improved by implementing more accurate object relationships.

V. CONCLUSIONS

In this paper, an inference based reactive motion planning strategy is developed for a robot to navigate to multiple objects in a partially known environment with specified temporal and logic constraints. By continuous reasoning and update on the relationship between the targets and the landmark objects, we show that the integration of commonsense reasoning and reactive temporal logic planning can significantly improve the efficiency of multi-object navigation. While the proposed inference-based planning method effectively addresses the multi-object search problem, it also presents certain limitations. The robot is reliant on prior information regarding landmarks, which can prove challenging to obtain in certain scenarios. In future work, our objective is to enable the robot to autonomously detect landmarks and utilise this information to infer target locations and optimise search paths. Furthermore, real-time target detection represents a promising way for improving search efficiency. Other interesting future directions are to extend the framework to accommodate open-set objects search and specifying targets using free-form language, which will enhance the scalability and intelligence of our approach.

REFERENCES

[1] M. B. Bejiga, A. Zeggada, and F. Melgani, "Convolutional neural networks for near real-time object detection from uav imagery in avalanche search and rescue operations," in *IEEE Int. Geosci. Remote Sens. Symp.* IEEE, 2016, pp. 693–696.

[2] A. T. Biggs and S. R. Mitroff, "Improving the efficacy of security screening tasks: A review of visual search challenges and ways to mitigate their adverse effects," *Appl. Cogn. Psychol.*, vol. 29, no. 1, pp. 142–148, 2015.

[3] H. Chen, R. Xu, S. Cheng, P. A. Vela, and D. Xu, "Zero-shot object searching using large-scale object relationship prior," *arXiv preprint arXiv:2303.06228*, 2023.

[4] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2009, pp. 2168–2173.

[5] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Tellex, "Towards optimal correlational object search," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2022, pp. 7313–7319.

[6] W. Ge, C. Tang, and H. Zhang, "Commonsense scene graph-based target localization for object search," *arXiv preprint arXiv:2404.00343*, 2024.

[7] 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 *IEEE Int. Conf. Robot. Autom.* IEEE, 2021, pp. 6585–6591.

[8] R. Druon, Y. Yoshiyasu, A. Kanezaki, and A. Watt, "Visual object search by learning spatial context," *IEEE Robotics Autom. Lett.*, vol. 5, no. 2, pp. 1279–1286, 2020.

[9] 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," *Int. J. Comput. Vis.*, vol. 123, pp. 32–73, 2017.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[11] J. Park, T. Yoon, J. Hong, Y. Yu, M. Pan, and S. Choi, "Zero-shot active visual search (zavis): Intelligent object search for robotic assistants," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2023, pp. 2004–2010.

[12] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, and Y. Choi, "Comet: Commonsense transformers for automatic knowledge graph construction," *arXiv preprint arXiv:1906.05317*, 2019.

[13] J. D. Hwang, C. Bhagavatula, R. Le Bras, J. Da, K. Sakaguchi, A. Bosselut, and Y. Choi, "(comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs," in *AAAI Conf. Artif. Intell.*, vol. 35, no. 7, 2021, pp. 6384–6392.

[14] M. Cai, M. Hasanbeig, S. Xiao, A. Abate, and Z. Kan, "Modular deep reinforcement learning for continuous motion planning with temporal logic," *IEEE Robotics Autom. Lett.*, vol. 6, no. 4, pp. 7973–7980, 2021.

[15] M. Cai, S. Xiao, Z. Li, and Z. Kan, "Optimal probabilistic motion planning with potential infeasible LTL constraints," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 301–316, 2023.

[16] Z. Chen and Z. Kan, "Real-time reactive task allocation and planning of large heterogeneous multi-robot systems with temporal logic specifications," *Int. J. Robot. Res.*, p. 02783649241278372, 2024.

[17] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local ltl specifications," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 218–235, 2015.

[18] M. Lahijanian, M. R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, and M. Y. Vardi, "Iterative temporal planning in uncertain environments with partial satisfaction guarantees," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 583–599, 2016.

[19] S. C. Livingston, R. M. Murray, and J. W. Burdick, "Backtracking temporal logic synthesis for uncertain environments," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2012, pp. 5163–5170.

[20] J. Alonso-Mora, J. A. DeCastro, V. Raman, D. Rus, and H. Kress-Gazit, "Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles," *Auton. Robots*, vol. 42, pp. 801–824, 2018.

[21] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[22] M. Y. Vardi and P. Wolper, "An automata-theoretic approach to automatic program verification," in *IEEE Symp. Log. Comput. Sci.* IEEE Computer Society, 1986, pp. 322–331.

[23] P. Gastin and D. Oddoux, "Fast ltl to büchi automata translation," in *Int. Conf. Comput. Aided Verif.* Springer, 2001, pp. 53–65.

[24] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] Y. Kantaros and M. M. Zavlanos, "Stylus*: A temporal logic optimal control synthesis algorithm for large-scale multi-robot systems," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 812–836, 2020.

[26] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[27] 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.

[28] S. Wani, S. Patel, U. Jain, A. Chang, and M. Savva, "Multion: Benchmarking semantic map memory using multi-object navigation," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 9700–9712, 2020.