

Commonsense Scene Graph-based Target Localization for Object Search

Wenqi Ge¹, Chao Tang¹, Hong Zhang^{1,*}

Abstract— Object search is a fundamental skill for household robots, yet the core problem lies in the robot’s ability to locate the target object accurately. The dynamic nature of household environments, characterized by the arbitrary placement of daily objects by users, makes it challenging to perform target localization. To efficiently locate the target object, the robot needs to be equipped with knowledge at both the object and room level. However, existing approaches rely solely on one type of knowledge, leading to unsatisfactory object localization performance and, consequently, inefficient object search processes. To address this problem, we propose a commonsense scene graph-based target localization, CSG-TL, to enhance target object search in the household environment. Given the pre-built map with stationary items, the robot models the room-level spatial knowledge with object-level commonsense knowledge generated by a large language model (LLM) to a commonsense scene graph (CSG), supporting both types of knowledge for CSG-TL. To demonstrate the superiority of CSG-TL on target localization, extensive experiments are performed on the real-world ScanNet dataset and the AI2THOR simulator. Moreover, we have extended CSG-TL to an object search framework, CSG-OS, validated in both simulated and real-world environments. Code and videos are available at <https://sites.google.com/view/csg-os>.

I. INTRODUCTION

For household robots frequently interacting with human environments, efficiently locating a target object is essential for completing subsequent tasks, such as object searching for downstream manipulation. For instance, in the scenario where a user instructs a robot to find a wallet, ideally, the robot should prioritize areas where the wallet is likely to be found, such as on a sofa or a chair, rather than inefficiently searching in unlikely places like a refrigerator. Considering the intricate interactions between movable targets (objects that users frequently place) and stationary items at the room-level and object-level across diverse indoor settings, it is a challenge to locate the movable targets by their correlation with known stationary items. To leverage commonsense knowledge of both room-level and object-level for efficient target object localization, similar to human reasoning, we equip the robot with a model that combines the room-level layout and object-level commonsense knowledge. Training this model on real-world scenarios enables the robot to locate targets effectively by utilizing both spatial and contextual insights.

To achieve such a goal, recent efforts focus on leveraging the correlation between target objects and stationary items

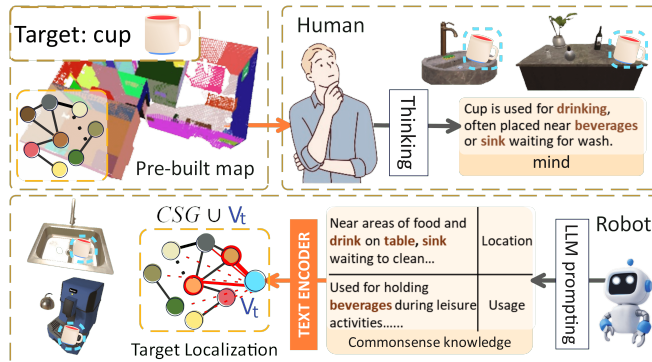


Fig. 1. An overview of target object location strategies comparing human room-level and object-level commonsense-inspired reasoning to our CSG-TL method, which integrates commonsense scene graph (CSG) with room-level and object-level knowledge for efficient target localization.

to enhance target location efficiency, which further benefits object search. Zheng *et al.* [1] primarily utilizes object co-occurrence probabilities to frame object search within the context of a partially observable Markov decision process (POMDP). However, this approach depends on the availability of precise object correlations for specific scenes, which is a lack of general object-level knowledge that is often not feasible. Subsequent study [2] has explored using graph neural networks (GNNs) to infer these correlations from the Visual Genome [3] image dataset. Yet, the image dataset can capture only partial views of scenes, which lacks the room-level knowledge that restricts the models’ ability to understand complete room-level layouts, which is critical for a robot’s comprehensive understanding of its environment.

Concurrently, large language models (LLMs) with vast commonsense knowledge, have shown promise in enhancing robot tasks [4], [5]. Despite their potential, their lack of spatial contextualization hinders directly applying LLMs to target localization in household settings [6], as they struggle to capture the relationships between objects and their surroundings effectively. To leverage LLMs’ strengths and address the limitations of existing works and LLMs, we propose a novel commonsense scene graph-based target localization method, CSG-TL, which integrates object-level commonsense from LLMs with room-level spatial layouts from pre-built maps. In contrast to previous works [1], [2] dependent on statistical correlations without object-level commonsense or struggles with capturing room-level object correlations due to limited viewpoints, our model captures both the room-level spatial layouts from pre-built maps and object-level commonsense knowledge obtained from LLMs. This fusion allows our method to utilize the rich intercon-

¹ Shenzhen Key Laboratory of Robotics and Computer Vision, SUSTech, Shenzhen, China. 12232112@mail.sustech.edu.cn

*corresponding author: Hong Zhang hzhang@sustech.edu.cn

This work was supported by the Shenzhen Key Laboratory of Robotics and Computer Vision (ZDSYS20220330160557001).

nections among objects and benefit the human-like reasoning process for target localization. We integrate CSG-TL into the object search task and contribute to the CSG-OS framework. Specially, in our approach, a robot’s search for a target object in a household environment involves two key phases: target localization (TL) and target object search (OS). During the TL phase, the robot estimates the probable location of the target object by leveraging its correlations at both the room and object levels with stationary items from the pre-built map. Following that, in the OS phase, the robot proceeds to the location predicted by TL to verify the presence of the target object. An illustrative overview of our approach is illustrated in Fig. 1.

Our approach, CSG-TL, initiates with creating the commonsense scene graph (CSG), which integrates the room-level spatial relationships of stationary items from pre-built maps with the object-level commonsense knowledge derived from an LLM. This process allows us to represent a target within the CSG as a node, addressing target localization as a link prediction problem. Here, the existence of links between the target and other nodes within the CSG infers potential co-occurrences between the target and respective items, thereby implementing the target localization. CSG-TL is the key component of our proposed object search (CSG-OS) framework, after projecting the likelihood of where the target may be located, as determined by CSG-TL, onto the pre-built map. We then cluster these projections to identify candidate navigation points for search. We trained CSG-TL on the ScanNet dataset, which contains over a thousand real-world scenes with labeled object relationships. To demonstrate its effectiveness and ability to generalize (zero-shot capability), we tested it on both the ScanNet [7] and the AI2THOR environments [8], achieving SOTA results. The overall CSG-OS framework, incorporating CSG-TL, also achieves the SOTA performance. Additionally, we tested our CSG-OS framework on the Jackal mobile robot, successfully demonstrating its real-world applicability.

In summary, our key contributions are as follows:

- We introduce CSG-TL, an innovative method that merges the knowledge at both room-level from pre-built maps and object-level commonsense obtained from an LLM into a commonsense scene graph (CSG), facilitating superior target localization.
- Building on CSG-TL, we propose a commonsense scene graph-based object search (CSG-OS) framework, with the SOTA performance in a simulation study and successful deployment on the real robot.

II. RELATED WORK

Object search involves a wide range of subproblems and different types of target objects. We consider targets as movable objects (users frequently place) and an environment where the location of stationary items is given as in [1], [2].

Object Search Object navigation is traditionally conceptualized as finding a particular category object in an unknown environment [9], [10]. Previous approaches [11], [12] focus on building a semantic map or keeping an episodic memory

during the search, aiming to enhance efficiency by preventing repeated searches in the same area. Nonetheless, these approaches take much time to explore, even searching for different targets in the same scene, rendering them inefficient for household robots equipped with a pre-built map. To improve search efficiency, maps and stationary items can be integrated. [1] employed co-occurrence probabilities between targets and stationary items, framing object search within the context of a partially observable Markov decision process (POMDP). However, the POMDP-based strategies typically presuppose the availability of co-occurrence probabilities. This assumption is not always practical, and they struggle to adapt to new environments due to a lack of object-level commonsense knowledge. Under the same setting, [2] tries to infer the correlations between targets and known stationary items by learning from large image datasets, aiming for a general way to obtain correlation information. Nevertheless, it falls short of capturing the room-level spatial layout due to the limited perspective from the image form dataset, which is critical for adapting room-scale target search challenges.

Unlike previous approaches that fail to capture room-level and object-level knowledge for effective target localization, we introduce CSG-TL. Our approach combines both levels of knowledge, significantly improving the accuracy of target localization and resulting in better search performance.

Scene graph modeling and inference. Scene graphs [13], [14] abstract scenes into structured representations, mapping objects to nodes and their relationships to edges, streamlining complex tasks like image retrieval [15] and visual question answering [16]. However, traditional robotics focuses more on building scene graphs during exploration [17] than leveraging them to guide target localization. [18] addresses this gap by developing spatial commonsense graphs (SCG), which model objects based on their spatial relationships within partially observed environments and integrate commonsense knowledge by connecting concept nodes from ConceptNet [19] to relevant scene items node, trained on dataset to locate objects in unobserved areas. However, this approach faces two main issues. Firstly, many indoor items share the same concept node provided by ConceptNet, which can lead to a fully connected graph that obscures the unique object-level commonsense attributed to each object. Secondly, predicting the specific spatial coordinates of targets based solely on incomplete room-level knowledge from partially observed scenes results in heavy reliance on the training dataset. This can lead to confusion, such as expecting to predict the location of a bed just by giving a general perception of a living room.

To address those limitations, we introduce a commonsense scene graph (CSG), a heterogeneous graph that incorporates a complete room-level layout and object-level commonsense knowledge within its nodes and edges. This model prevents the lack of room-level information, which may cause unstable results. Besides, our object-level commonsense knowledge is obtained from an LLM that avoids the overlap from ConceptNet. This comprehensive model allows a more reasonable inference of target locations based on

our rich correlations with both room-level and object-level knowledge.

III. PROBLEM FORMULATION

In this work, we tackle the challenge of object search, explicitly focusing on scenarios relevant to household robots equipped with a pre-built map containing stationary items (e.g., furniture, fixtures). Mimicking the human search strategy, we propose the commonsense scene graph-based object search (CSG-OS) framework. This framework enhances the robot’s search efficiency by initially predicting potential target locations based on their correlation at both room and object levels with stationary items (CSG-TL), and subsequently optimizing the search strategy by weighing the spread of these likely locations against the robot’s present location. Fig. 2 shows the pipeline of our framework.

The commonsense scene graph (CSG) is initially constructed from a pre-built map M , with nodes $V = \{v_i | i \in \{1, \dots, N\}\}$ representing the N stationary items within the scene, $E = \{e_{ij} | i \in \{1, \dots, N\}, j \in \mathcal{N}_i\}$ representing the edge between node V_i and V_j , where \mathcal{N}_i denotes the set of neighbors for node V_i . Sec. IV-A introduces how to build the CSG from a pre-built map, and the structure of the commonsense scene graph-based target localization (CSG-TL) model. Following this, Sec. IV-B details the CSG-OS framework.

IV. PROPOSED METHOD

This section provides a detailed discussion of the core component and the whole pipeline of our proposed CSG-OS framework. It includes the methodology for CSG construction, the generation of the dataset for training CSG-based target localization (CSG-TL), and the CSG-based object search (CSG-OS) framework.

A. Commonsense Scene Graph

Data Generation. Our dataset is derived from the ScanNet database [7], which contains 1,513 real-world scenes annotated with spatial relationships between objects. We use these scenes to extract room-level layouts and infuse object-level commonsense knowledge C into the scene graph, using GPT-4 API for LLM-based queries. To construct the Commonsense Scene Graph (CSG), we categorize nodes into stationary items and daily movable objects, the latter being set as target objects for localization. For stationary items, which serve as known nodes in the CSG, we query descriptions of their actual locations and usage, forming the commonsense knowledge $C_{obj_sta} = \{x_{loc}, x_{usage}\}$. For movable objects, which will later be trained as target objects, we similarly query their daily locations and usage to build C_{obj_tar} . For edges, we query descriptions of spatial relationships and functional links between object pairs, generating the commonsense knowledge $C_{edge} = \{e_{geo}, e_{fun}\}$. The prompts used are similar to those in ConceptGraph [21], and the details are provided in our code.

Similar to the previous scene graph setting [2], two objects are considered connected if the Euclidean distance between

them is less than the threshold d_{thre} or if they share a receptacle relationship (e.g., “on” or “inside”). Furthermore, if an object has no other object supporting the above condition, we identify the closest stationary item in the room to serve as its neighboring object. The edge label is created using the following rule:

$$E(V_i, V_j) = \begin{cases} 1 & \|V_i - V_j\| < d_{thre} \\ 1 & \text{receptacle}(V_i, V_j) \\ 1 & \text{if } V_i \text{ has no close or receptacle} \\ & \text{objects and } V_j \text{ is nearest to } V_i \\ 0 & \text{else} \end{cases} \quad (1)$$

According to those, the nodes feature $V = \{x_{cat}, x_{loc}, x_{usage}\}$ is the concation of the object commonsense knowledge $C_{obj} = \{x_{loc}, x_{usage}\}$ with the encoding of the object category x_{cat} , similarly, the edge feature $E = \{e_{geo}, e_{fun}\}$ is the commonsense knowledge C_{edge} . One example is presented in Fig. 3.

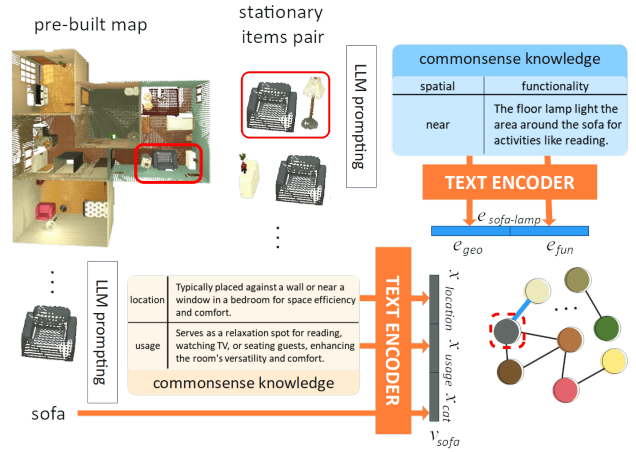


Fig. 3. Illustration of the CSG construction process: The scene graph is initially created by identifying stationary items from the pre-built map based on spatial relationships, as defined in Eq. 1. Subsequently, commonsense knowledge relevant to human inference of correlations is incorporated through LLM prompts.

The Model of CSG-based Target Localization. To facilitate an efficient search process, we aim to identify regions most relevant to the target by utilizing the advanced inferential correlations provided by the commonsense scene graph (CSG) structure. With this objective, we present a target localization model CSG-TL, a key component of our CSG-OS framework.

The CSG-based target localization (CSG-TL) model computes the likelihood of a target’s correlation with objects in CSG using a multi-modal fusion module, which can be represented below:

$$p = \text{CSG-TL}(V_t, V, E) \quad (2)$$

where p is the likelihood between target V_t and the other nodes $V = \{v_i | v \in \{1, \dots, N\}\}$ in CSG, and $E = \{e_{ij} | i, j \in \{1, \dots, N\}\}$ represents the edge in CSG. Fig. 4 details the

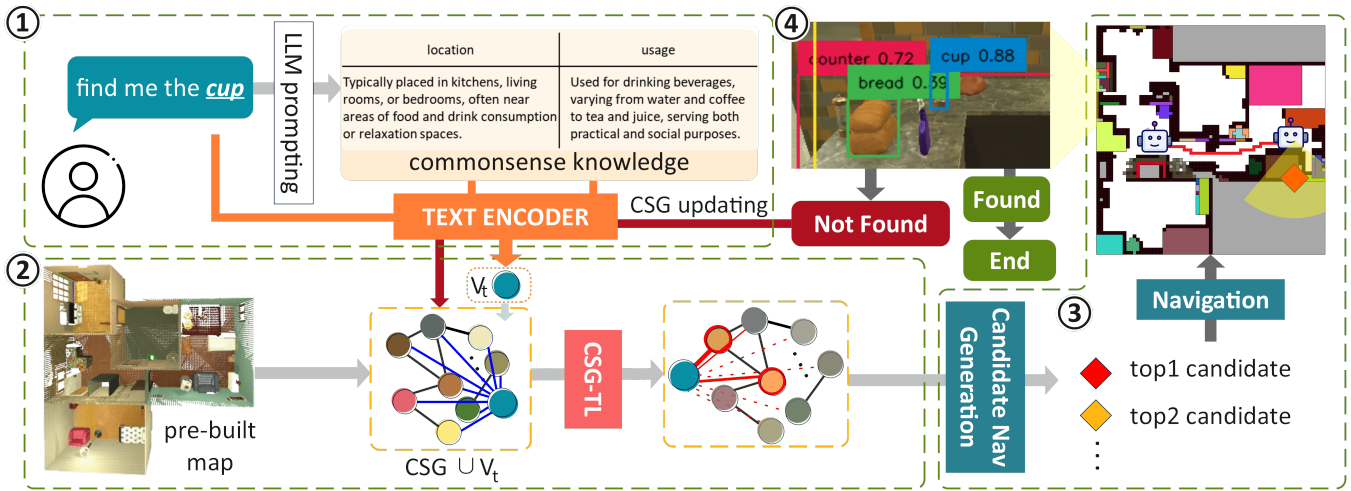


Fig. 2. **CSG-based object search (CSG-OS) pipeline.** Firstly, the user queries the target object, which is then encoded with LLM-derived commonsense knowledge to form the target node V_t . Following this, the CSG is then constructed from a pre-built map of stationary items, incorporating V_t for target localization through CSG-TL, detailed in Sec. IV-A. Thirdly, nodes correlated with the target are clustered based on their locations and the likelihood of their predicted correlations, establishing a set of candidate search points. Finally, the robot navigates to the first candidate points in turn to search for the target. If found, the task is done successfully. Otherwise, the robot updates the CSG by newly detected objects and repeats the search steps until it finds the target or exceeds a preset threshold on the number of steps.

structure of CSG-TL. Node features undergo a two-layer graph attention network (GAN) to fuse their feature v_i'' first, then pass a transformer by utilizing the edge feature as attention in Eq. 3:

$$\alpha_{ij} = \text{ReLU}((Q_i, K_j + e_{ij}) / \sqrt{\|K_j\|}) \quad (3)$$

where Q_i, K_j are corresponding transformer values of node V_i and its neighbors $V = \{V_j | j \in \mathcal{N}_i\}$, respectively. The computed attention weights α are then forward used to fuse the node feature from v_i'' to v_i^* , and finally concatenate the original and fused feature as $\hat{v} = [v^*, v]$ of each node as the final feature. Finally, we use $[\hat{v}_t, \hat{v}_i]$, the concatenation of target node feature \hat{v}_t and all others in CSG to represent the edge feature between them, using an MLP and a sigmoid to normalize the likelihood to $[0, 1]$. If the likelihood exceeds 0.5, it suggests the presence of an edge, indicating that the target is likely associated with the object; otherwise, it is not.

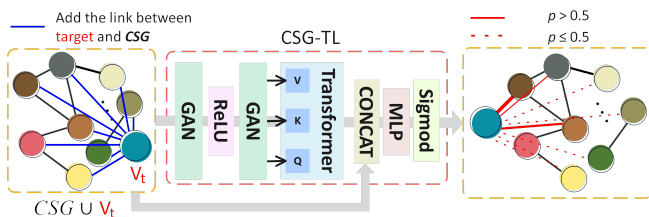


Fig. 4. Illustration of the CSG-TL structure. Given the CSG and a target node V_t , the CSG-TL estimates the probability of correlation between the target and nodes within the CSG. A correlation exists if $p > 0.5$ indicates a likely co-occurrence between the target and the respective object.

Loss Function. The binary cross-entropy loss is utilized to measure the discrepancy between the predicted probability of a correlation link's existence, denoted by p and the ground truth label p_{gt} as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N p_{gt,i} \cdot \log(p_i) + (1 - p_{gt,i}) \cdot \log(1 - p_i) \quad (4)$$

where N is the number of total nodes in one graph. In our approach, we set 32 graphs as a batch and minimize the loss \mathcal{L} for all graphs. Additionally, we operate under the assumption that the target will likely co-occur with nodes if their probability of correlation exceeds 0.5.

B. CSG-based Object Search

Leveraging the target localization capabilities of the CSG-TL model, we introduce the CSG-based object search (CSG-OS) framework, designed to efficiently tackle the object search challenge. The overall CSG-OS pipeline is illustrated in Fig. 2.

Target Object Encoding and CSG-based Target Localization. Adapting the encoding strategy from Sec. IV-A, the target is represented as a node V_t , encoded with commonsense knowledge via a prompt LLM. As in dataset and AI2THOR simulator, target category x_{cat} is provided directly, allowing for straightforward encoding as $V_t = \{x_{cat}, x_{loc}, x_{usage}\}$. In practical, real-world scenarios, users can specify their search targets in free-form language $l_{tar-des}$, which typically encompasses the category l_{cat} along with possible additional descriptors or hints, like a general location. Such inputs allow for extracting more detailed commonsense knowledge through LLM prompts, as demonstrated in the TABLE. I. The target node is thus formulated as $V_t = \{l_{cat}, x_{loc}, x_{usage}\}$, accommodating both direct category mentions and more elaborate commonsense knowledge.

After this, we connect the target node V_t to CSG by linking it with all nodes V in CSG to localize the target by CSG-TL.

Target	Location	Commonsense Knowledge
Table	Living room , placed near sofas	Central in social gatherings for food and drinks.
	Kitchen , placed in the center	Dining space for eating meals, a prep area for food preparation, and a gathering spot for ...
Cup in Living room (query by user as target)	Living room , on tables, or other surfaces near seating areas.	Holding beverages during leisure activities, social gatherings, or while relaxing and watching TV.
Cup (generic)	On tables in the living room, kitchen, etc.	A common object in home environments, typically used for holding beverages. Its location often varies based ...

TABLE I. Commonsense knowledge for various inputs via LLM prompts.

Candidate Navigation Point Generation and CSG Updating. The correlation likelihood h_i between the target and each object Obj_i is predicted by the CSG-TL, as detailed in Sec. IV-A. However, the scattered arrangement of these objects does not inherently support an efficient search. To address this, we propose an approach that efficiently guides object search by utilizing the location $\text{loc}_i = [x_i, y_i]$ of each object Obj_i and spreads the likelihood within a defined radius r , decreasing with distance from the object center on the map M , and represented as follows:

$$M[x, y]_i = h_i \cdot \max(0, r - \sqrt{(x - x_i)^2 + (y - y_i)^2})$$

$$M[x, y] = \min(1, \frac{1}{K} \left[\sum_{i=1}^K M_i[x, y] \right] + w \cdot K) \quad (5)$$

where $M[x, y]_i$ indicates the influence of the object i at position $[x, y]$ on the map, $M[x, y]$ is the cumulative likelihood, reflecting the average influence from all K correlated objects with the target, and an award with w by the number of correlated objects. To identify candidate navigation regions, we partition the map M into square sections R_i with a side length of $2r$ and accumulate the aggregated likelihoods within each to calculate W_{R_i} :

$$W_{R_i} = \sum_{[x, y] \in R_i} M[x, y] \quad (6)$$

Candidate regions $\{R_1, R_2, \dots\}$ are ordered by their weights W_{R_i} , and we choose their center as the candidate navigation position $\{p_1, p_2, \dots\}$. Besides this, the distance between robot $p_{\text{robot}} = [x_{\text{robot}}, y_{\text{robot}}]$ and candidate navigation position also should be considered, so we construct a cost function to balance both:

$$C(p_i) = \alpha \cdot \left(1 - \frac{W_{R_i}}{\max\{W_{R_j}\}} \right) + \beta \cdot \left(\frac{D(p_{\text{robot}}, p_n)}{\max\{D(p_{\text{robot}}, p_k)\}} \right) \quad (7)$$

where α and β are weights to balance the importance of the candidate region and the distance to the robot, ensuring

a normalized scale with $\alpha + \beta = 1$. $D(\cdot)$ calculates the distance between two points in the map M by A^* . Upon reaching the candidate search position, the robot looks for the target. If successful, the search concludes. Otherwise, the CSG undergoes an update with new non-stationary objects encountered during the search, repeating the cycle of target localization and navigation until success or the maximum number of steps is reached.

V. EXPERIMENTS

This section discusses the experimental validation of our proposed commonsense scene graph-based object search (CSG-OS) framework and its key component, commonsense scene graph-based target localization (CSG-TL). Both of them outperform current work. Finally, we present an overview of CSG-OS's real-world deployment, highlighting the efficiency and practicality of our method.

A. The Result of CSG-TL

As described in Sec. IV-A, we train the proposed CSG-based target localization (CSG-TL) by ScanNet [7]. We evaluate the trained CSG-TL model's link prediction performance using the ScanNet test split and the AI2-Thor environment. We consider 94 unique object categories in AI2THOR for target identification, consistent with previous studies [2]. Objects are deemed to be connected if they meet the criteria outlined in Eq. 1, with a distance threshold $d_{\text{thre}} = 1$, mirroring the parameters used in earlier studies [2].

In our evaluation, we compared our approach against three established methods:

- **Statistics-based method:** Utilizing statistical correlations between targets and stationary items from the dataset, considering stationary items are linked with the target if their statistical correlation exceeds 0.5.
- **Correlation via Knowledge Graph [20]:** Enhancing statistical correlations with semantic connections from ConceptNet directly applies to visual perception, providing additional contextual insights.
- **GNN model trained on Visual Genome dataset [2]:** Leveraging GNN to predict object correlations based on the Visual Genome image dataset.

To calculate accuracy, we first define a binary indicator function $H(x)$, which distinguishes connections based on a threshold of 0.5 for correlation between the target and other objects. Accuracy is then calculated by averaging the outcomes across all scenes within the dataset G , as shown in Eq. V-A below:

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0.5 \\ 0 & \text{if } x < 0.5 \end{cases} \quad (8)$$

$$\text{Acc} = \frac{1}{G} \sum_{g=1}^G \left(\frac{\sum_{i=1}^N H(\text{CSG}[V_{\text{target}}, V_i]) \wedge \text{GT}[V_{\text{target}}, V_i]}{\text{Num}[V_{\text{target}}, V_i]} \right) \quad (9)$$

The performance results are presented in Table. II. showcase the superior accuracy of our CSG-TL model over other methods within both the ScanNet and AI2THOR



Fig. 5. Visualization of target localization, the \circ indicates the top1 candidate navigation point, \circ and \circ representing the top2 and top3 candidates, respectively (calculation detailed in IV-B). The prior knowledge they utilize is the pre-built map with stationary items only. The target’s actual location in the full 3D map is marked with hollow-red-circle, and the corresponding location on the 2D fixtures-only map is denoted with red-circle. This visualization demonstrates the accuracy of our CSG-TL model in predicting the target’s location against other methods.

Link Pred. Acc	Statistical	ZAVIS [20]	ZSOS [2]	CSG-TL w/o commonsense knowledge	CSG-TL
ScanNet	27.56%	35.63%	/	80.03%	89.73%
AI2-THOR single room	47.32%	59.25%	67.22%	73.11%	81.09%
AI2-THOR multi rooms	31.77%	43.57%	/	65.21%	78.21%
Commonsense knowldge	statistics	statistics/Comet	category similarity	category similarity	category similarity/LLM
Method	statistical	statistical	graph	graph	graph

TABLE II. Comparison of Link Prediction Accuracy across methodologies on the ScanNet dataset and AI2THOR environments.

environments, demonstrating the significant advantage of our proposed CSG, integrating object-level commonsense knowledge via LLM into the scene graph built by room-level layout. As indicated in Table. II, the method bases purely on statistical correlations performs the least effectively. However, integrating semantic and commonsense insights to visual perception [20] significantly improves accuracy, although it still falls short of the outcomes achieved by graph-based inference methods [2] and ours. Notably, even the variant of our model without commonsense knowledge still surpasses the performance of ZSOS [2] relying on training on partial scene views from images. Our complete CSG-TL model, enriched with commonsense knowledge, consistently demonstrates the highest performance, affirming that commonsense reasoning significantly enhances target localization capabilities.

We also provide a visual comparison of target localization across different scenarios in AI2THOR and ScanNet scenar-

ios. Illustrated in Fig. 5, our CSG-TL model pinpoints the most probable candidate positions for the target object.

Scenario	AI2THOR	ProcThor (multi-rooms)	Realworld
w	0.05	0.05	0.1
α	0.4	0.6	0.5
β	0.6	0.4	0.5

TABLE IV. Hyperparameters setting

B. Simulation Result

In this section, we present the efficacy of the proposed commonsense scene graph-based object search (CSG-OS) framework, benchmarked within the AI2THOR [8] simulation. The AI2THOR provides a diverse set of 120 scenarios spanning four distinct environments: kitchens, living rooms, bedrooms, and bathrooms. ProcTHOR [22] also extends this offering with more than 10,000 simulated household scenes, accommodating single and multi-room layouts. They

Model	Kitchen		Bathroom		Bedroom		Living room		Multi rooms	
	SPL	SR	SPL	SR	SPL	SR	SPL	SR	SPL	SR
Greedy-NBV [1]	11.61%	31.03%	14.34%	34.48%	16.92%	26.67%	7.13%	20.00%	/	/
COS-POMDP [1]	20.45%	41.38%	30.64%	55.17%	24.76%	40.00%	24.99%	43.33%	/	/
Target-POMDP [1]	13.80%	34.48%	19.88%	55.17%	19.79%	26.67%	24.36%	40.00%	/	/
ZAVIS [20]	/	/	/	/	30.17%	81.42%	19.69%	75.00%	/	/
ZSOS [2]	23.56%	46.67%	56.62%	86.67%	28.91%	53.33%	35.01%	59.25%	/	/
CSG-OS (ours)	42.92%	63.33%	67.11%	86.67%	59.79%	86.67%	41.36%	76.67%	43.67%	46.67%

TABLE III. Comparative results of search success rate (SR) and success path length (SPL) in AI2THOR simulations of different scenarios.

include hundreds of unique objects with different shapes and textures. The robot is capable of executing basic movement commands, including MoveAhead, RotateLeft, RotateRight. The MoveAhead command propels the robot forward by $0.25m$, while RotateLeft and RotateRight commands turn the robot 45° to the left or right, respectively.

All these methods aim to improve object search by exploring correlations between the target and known stationary items. They employ approaches ranging from statistical distributions [1], extracting categorical similarities from knowledge graphs [20], to leveraging learning models trained on image datasets like Visual Genome [2]. Similar to our method, they also dynamically update the probability of correlation between the target and other recognized objects besides stationary items during the search, allowing for continuous refinement of search strategies. All methods were given equal search steps and tested on identical target object classes to maintain comparative fairness. The YOLOv5 object detector [23] was implemented as the standard perception module for all methods to ensure consistency. A successful search is defined by the robot’s detection of the target object within a $1.0m$ Euclidean distance. Based on these, we set the hyperparameters in Eq.5 and Eq.6 as TABLR IV.

The experimental results are shown in Table III, assessing the search tasks using the Success Rate (SR) and the Success weighted by (normalized inverse) Path Length (SPL). These results demonstrate our proposed CSG-OS’s superiority over existing methods in terms of success rate and search efficiency (SPL). Notably, the traditional statistical correlation approach [1] registers the lowest success rates, while ZAVIS [20] incorporates semantic insights from statistical correlation enhanced with knowledge graphs into visual perception, marking a noticeable improvement. ZSOS [2], which derives correlations from image datasets without relying on predefined statistical knowledge of the environment, demonstrates further progress but falls short of providing a comprehensive understanding of the entire house layout. Our CSG-OS stands out, reaching the SOTA performance even without specific environmental knowledge beforehand. This achievement highlights the importance of incorporating commonsense knowledge into the scene graph, markedly enhancing its inferential capabilities.

C. Real-world Implementation and Results

The practical effectiveness of the commonsense scene graph-based object search (CSG-OS) pipeline is further validated through deployment in a real-world setting. We employed a Jackal unmanned ground vehicle, equipped with a RealSense L515 camera for visual perception and a Livox Mid-360 LiDAR for precise localization. The robot autonomously explored the environment to construct a semantic map, exclusively recording and marking stationary items, as depicted in Fig. 6. Leveraging this semantic map along with commonsense knowledge generated from the same prompted GPT-4 API queries described in Sec. IV-A, we constructed a commonsense scene graph (CSG) to perform target localization, followed by subsequent object search tasks.

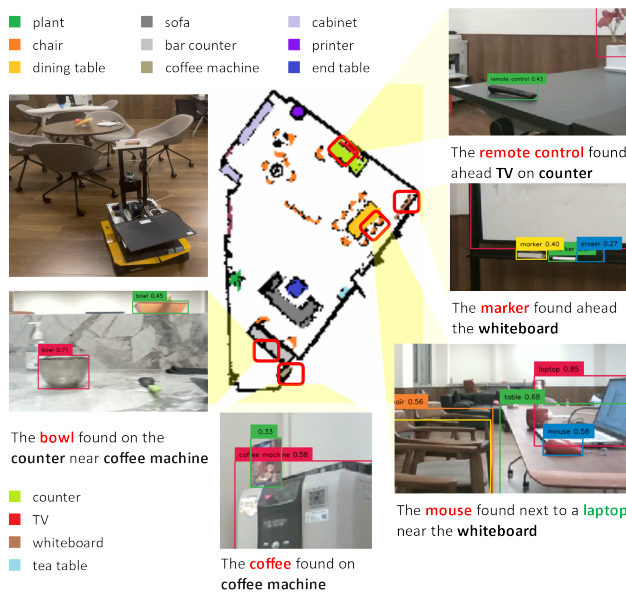


Fig. 6. Real-world Implementation of CSG-OS on the Jackal Robot. The map incorporates stationary items utilized to construct the commonsense scene graph (CSG). Targets are highlighted in red, and are dynamically searched by the robot. Green denotes objects identified during the search that update the initial CSG, enhancing the system’s predictive accuracy for target localization.

To evaluate our approach, we created a home-like environment where seven volunteers were instructed to place five types of commonly used objects in various natural

locations. The Jackal robot was then tasked with locating and retrieving these items by utilizing the CSG-OS framework, effectively demonstrating its ability to perform real-world target localization and search.

Succ.	bowl	coffee	mouse	marker	remote control
CSG-OS on Jackal	100.00%	85.71%	85.71%	71.43%	71.43%

TABLE V. Real-world object localization success rates by CSG-OS

The results of the Jackal robot’s performance in searching each object are presented in Table V, demonstrating the effectiveness of the CSG-OS in real-world applications. The task of *bowl* searching achieved a 100% success rate, attributable to its predictable placement on counters or tables and easy detection. The *mouse* followed closely, benefiting from its strong association with laptops, which can be detected easily, the similar conclusion of *coffee*. However, the success rates for the *marker* and *remote control* are relatively low. Although these items are often associated with whiteboards and TVs, they present greater challenges when displaced from their usual settings, and they are difficult to detect because of their small size and colors that easily blend with their surroundings.

VI. CONCLUSION

In this study, we introduced an innovative Commonsense Scene Graph (CSG) that utilizes commonsense knowledge at both the room and object levels to improve scene modeling and target localization, mirroring human reasoning processes. Our approach markedly improves upon traditional object search strategies, which rely on correlation-based methods or learn from partial image views limited to a single type of knowledge. Our CSG-based object search (CSG-OS), facilitated by CSG-based target localization (CSG-TL), demonstrates effective zero-shot object search capabilities both in the AI2THOR simulator and real-world scenarios. While our framework demonstrates significant advancements, it encounters challenges in adapting to unconventional household environments like laboratories or certain public areas and struggles to interpret complex language-based goals within a scene. Our future work will aim to expand the adaptability of our approach to a wider range of scenarios, where language understanding will be pivotal in guiding robotic tasks effectively.

REFERENCES

- [1] Zheng K, Chitnis R, Sung Y, et al. Towards optimal correlational object search[C]//2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 7313-7319.
- [2] Chen H, Xu R, Cheng S, et al. Zero-Shot Object Searching Using Large-scale Object Relationship Prior[J]. arXiv preprint arXiv:2303.06228, 2023.
- [3] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

- [4] Kapelyukh I, Vosylius V, Johns E. Dall-e-bot: Introducing web-scale diffusion models to robotics[J]. IEEE Robotics and Automation Letters, 2023.
- [5] Tang C, Huang D, Ge W, et al. Grasppt: Leveraging semantic knowledge from a large language model for task-oriented grasping[J]. IEEE Robotics and Automation Letters, 2023.
- [6] Hu Z, Shu T. Language models, agent models, and world models: The law for machine reasoning and planning[J]. arXiv preprint arXiv:2312.05230, 2023.
- [7] Dai A, Chang A X, Savva M, et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5828-5839.
- [8] Kolve E, Mottaghi R, Han W, et al. Ai2-thor: An interactive 3d environment for visual ai[J]. arXiv preprint arXiv:1712.05474, 2017.
- [9] Aydemir A, Pronobis A, Göbelbecker M, et al. Active visual object search in unknown environments using uncertain semantics[J]. IEEE Transactions on Robotics, 2013, 29(4): 986-1002.
- [10] Batra D, Gokaslan A, Kembhavi A, et al. Objectnav revisited: On evaluation of embodied agents navigating to objects[J]. arXiv preprint arXiv:2006.13171, 2020.
- [11] Gadre S Y, Wortsman M, Ilharco G, et al. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 23171-23181.
- [12] Zhou K, Zheng K, Pryor C, et al. Esc: Exploration with soft commonsense constraints for zero-shot object navigation[C]//International Conference on Machine Learning. PMLR, 2023: 42829-42842.
- [13] Li Y, Ouyang W, Zhou B, et al. Scene graph generation from objects, phrases and region captions[C]//Proceedings of the IEEE international conference on computer vision. 2017: 1261-1270.
- [14] Xu D, Zhu Y, Choy C B, et al. Scene graph generation by iterative message passing[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5410-5419.
- [15] Yoon S, Kang W Y, Jeon S, et al. Image-to-image retrieval by learning similarity between scene graphs[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(12): 10718-10726.
- [16] Hildebrandt M, Li H, Koner R, et al. Scene graph reasoning for visual question answering[J]. arXiv preprint arXiv:2007.01072, 2020.
- [17] Hughes N, Chang Y, Carlone L. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization[J]. arXiv preprint arXiv:2201.13360, 2022.
- [18] Giuliani F, Skenderi G, Cristani M, et al. Leveraging commonsense for object localisation in partial scenes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.
- [19] Speer R, Chin J, Havasi C. Conceptnet 5.5: An open multilingual graph of general knowledge[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).
- [20] Park J, Yoon T, Hong J, et al. Zero-shot active visual search (zavis): Intelligent object search for robotic assistants[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023: 2004-2010.
- [21] Gu Q, Kuwajerwala A, Morin S, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024: 5021-5028.
- [22] Deitke M, VanderBilt E, Herrasti A, et al. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation[J]. Advances in Neural Information Processing Systems, 2022, 35: 5982-5994.
- [23] Jocher G, Chaurasia A, Stoken A, et al. ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation[J]. Zenodo, 2022.