2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Language-guided Active Sensing of Confined, Cluttered Environments via Object Rearrangement Planning

Weihan Chen, Hanwen Ren, and Ahmed H. Qureshi

Abstract— **Language-guided active sensing is a robotics subtask where a robot with an onboard sensor interacts efficiently with the environment via object manipulation to maximize perceptual information, following given language instructions. These tasks appear in various practical robotics applications, such as household service, search and rescue, and environment monitoring. Despite many applications, the existing works do not account for language instructions and have mainly focused on surface sensing, i.e., perceiving the environment from the outside without rearranging it for dense sensing. Therefore, in this paper, we introduce the first language-guided active sensing approach that allows users to observe specific parts of the environment via object manipulation. Our method spatially associates the environment with language instructions, determines the best camera viewpoints for perception, and then iteratively selects and relocates the best view-blocking objects to provide the dense perception of the region of interest. We evaluate our method against different baseline algorithms in simulation and also demonstrate it in real-world confined cabinet-like settings with multiple unknown objects. Our results show that the proposed method exhibits better performance across different metrics and successfully generalizes to real-world complex scenarios.**

I. Introduction

Language-guided active sensing is an integrated perception, planning, and control problem wherein a robot, equipped with onboard perception sensors, optimally collects the perceptual information based on given language commands by interacting with the surrounding environment [1]. In confined and narrow-passage environments, these tasks often require adept object manipulation to effectively collect visual data and provide the desired dense perception. These scenarios appear in various settings such as in-home assistance [2], [3] where the disabled and elderly people will rely on robot systems to provide them with the perception of confined environments such as cabinets, drawers, or refrigerators. Likewise, in search and rescue [4], [5], a remote human operator may instruct a robot to provide visuals of specific areas by manipulating the underlying objects. Despite crucial and essential applications of language-guided active sensing, such tasks remain in an unexplored area of research. Most existing work focuses on active sensing of open, table-top-like environments, with an exception of [6], which considers confined environments. Still, none of these methods rearrange objects for dense sensing or account for language instructions.

Therefore, this paper presents a language-guided active sensing system that can effectively provide a dense perception of the given unknown, cluttered environments by manipulating objects to meet a specified requirement of a user given by natural language commands. Our method aims to meet the requirements of a user prompt by exploring an unknown environment with minimal camera movements and object manipulations. The main contributions of our framework are summarized as follows:

- Language to environment spatial correspondence matching module to identify the user's region of interest (ROI) for active perception in the given environment.
- An MPC-style viewpoint planning method that uses Gaussian Mixture Models (GMM) with a novel neural network-based utility function to select the best viewpoints for observing the ROIs.
- An analytical approach to identify ROI view-blocking objects for relocation via manipulation to clear the way for selected viewpoints to observe the user's ROI.
- A unified language-guided active sensing system that combines the above methods for fast selection of viewpoints and relocation of view-blocking objects to complete the user command of sensing a specific portion of the environment.

II. Related work

The most relevant works to our approach are on the topic of autonomous exploration and next-best view selection. The traditional and prominent approaches to autonomous exploration include frontier-based and search-based methods. The frontier-based autonomous exploration method is first introduced in [7], where the robot explores the boundary between mapped and unmapped space. Later work expanded on this technique to achieve a higher coverage along the path of the frontier [8]–[10]. However, these frontier-based methods require overlapping of sequential observations. This would lead to too many viewpoints being generated and not suitable for efficient completion of user's requests. In contrast, the search-based method utilizes a utility function to guide the search, first introduced in [11]. This method is most prominent in 3D reconstruction settings. Further methods derived from this include [12]–[14], which propose different utility measures that can better maximize the coverage. These methods are generally used in single-object reconstruction settings with assumptions about the underlying geometry, which is unfavorable for narrow and cluttered settings.

*This work was supported by the National Science Foundation (NSF) under award no. 2204528.

Weihan Chen, Hanwen Ren, and Ahmed H. Qureshi are with the Department of Computer Science, Purdue University, West Lafayette, IN, USA, 47907. Email {chen3189, ren221, ahqureshi}@purdue.edu

More recent data-driven approaches leverage deep neural networks to approximate the utility function for the next best view generation [15]–[17]. The introduction of neural networks into next-best view estimation eliminates the need to physically move the camera, which reduces sub-optimal movements. These methods, however, only operate on single objects or open spaces, which is unsuitable for exploration in a multi-object, narrow environment. The most recent approach that explores confined, narrow environments is VPFormer [6], which produces results that show minimal viewpoint generation for scene coverage. However, this approach assumes a scene that is observable solely by moving the camera, i.e., without object manipulation.

In contrast to the above methods, we aim for settings where objects can prevent observation of significant parts of the scene, making selection and relocation of view-blocking objects necessary. Furthermore, we also consider human-in-the-loop settings where they provide language commands to indicate the region of interest in the environment for dense active sensing. Although natural language as commands have been used in various robotic tasks [18]–[20], their role in performing active sensing is unexplored. Similarly, recent studies enable the manipulation of unknown objects using point cloud data gathered from stereo or RGB-D cameras [21]–[23] but not from the perceptive of rearranging scenes for better visual coverage. To the best of our knowledge, this work presents the first framework for language-guided active sensing via object manipulation in confined, narrow environments.

III. Proposed Methods

This section formally presents the main components of our language-guided active sensing via object manipulation system with the overall pipeline shown in Fig. 1.

A. Problem Formulation

Let a confined environment be represented as $S \subseteq \mathbb{R}^3$, whose ranges along the x, y, and z axes are assumed to be known and denoted as d_x, d_y, and d_z, respectively. The value of a coordinate $S(i, j, k)$ is either 1 or 0, where the collection of the former forms the observed region S_o and that of the latter forms the unobserved region S_{ou}. The scene also has a collection of household items $\{o_1, o_2, ...\}$. A robot arm equipped with an in-hand RGB-D camera placed in front of the scene takes an initial image I^0 and presents it to a human user. The user then describes, via language command, a region of interest (ROI), $S_{ROI} \subseteq S$, that requires active visual exploration. In order to measure how well the user's request is fulfilled, a coverage rate of the observed region S_o over the user-specified region S_{ROI} is defined as:

$$\phi(S_o, S_{ROI}) = \frac{\sum_i^{dx}\sum_j^{dy}\sum_k^{dz}(\mathbb{1}_{S_{ROI}\cap S_o}(i,j,k))}{\sum_i^{dx}\sum_j^{dy}\sum_k^{dz}\mathbb{1}_{S_{ROI}}(i,j,k)} \quad (1)$$

The region of interest S_{ROI} is fully explored when $\phi(S_o, S_{ROI}) \approx 1$. At any time step t, the robot can remove a detected view-blocking object o^t from the scene and navigate the camera to a kinematically feasible viewpoint $v^t \in \mathcal{V} \subseteq SE(3)$. Each camera viewpoint v^t after an object o^t relocation leads to an image I^{t+1} and a scene observation S^{t+1} with newly obtained information marked as $S_o^{t+1}(o^t, v^t) \backslash S_o^t$. In all, the objective of the problem is to maximize the overall scene coverage rate $\phi(S_o, S_{ROI})$ leveraging two policy functions π_o, and π_v, that makes the best choice of the object manipulation and viewpoint selection at any given step, resulting in the shortest robot action sequence. The objective can be formally written using the min-max formulation:

$$\max_{v^t\sim\pi_v, o^t\sim\pi_o} \min_T \sum_{t=0}^{T-1} \phi(S_o^{t+1}(o^t, v^t)\backslash S_o^t, S_{ROI}) \quad (2)$$

During the active ROI sensing procedure, all images I^t are sent to the user's end as a response. If the robot agent achieves the objective, it also means that the user can see all the details in the ROI using the minimal number of images, which fulfills the request.

The remainder of the method section discusses our design to optimize the above objective function. Furthermore, to describe various functions of our framework, we use a notation $A_{\{B\}}$ denoting a list A composed of B elements for brevity.

B. Language-guided ROI Generation

The user makes a language prompt to describe a region relative to an observed object for active sensing, e.g., "Show me to the left of the pink cylinder." Our language linker then identifies the anchor object o_i and a relative spatial direction l, as shown by the red-dotted box in Fig. 1. More specifically, we use [24] to parse the user query into part of speech tags, and then o_i is obtained by extracting the tokens with a tag of *pobj* (object of preposition) or *dobj* (direct object). On the other hand, the direction is identified by iterating through the sentence and finding tokens whose synonyms include a directional word. The extracted direction l resides in the set containing {left, right, behind, front}, corresponding to both sides of the X and Y axes. Let the location of the anchor object o_i be represented as $r(o_i) = (x_{o_i}, y_{o_i}, z_{o_i})$, the region of interest S_{ROI} is constructed by expanding r_{o_i} in all axes until the boundaries of the environment following the user-specified spatial direction l.

C. ROI-ScoreNet

Due to the irregular shape and various potential locations of the multiple objects in the scene, estimating the new information gained from a given camera pose is nontrivial. Thus, we propose an ROI coverage score function, called ROI-ScoreNet, that predicts the possible scene coverage gain from a given camera viewpoint candidate without physcially repositioning the camera. This function is inspired from the ScoreNet in [6]. However, the original ScoreNet formulation considers the entire scene instead of a ROI. Thus, we proposed a new neural network structure for ROI-ScoreNet. The ROI-ScoreNet can be viewed as a function f_θ with parameters θ that estimates the coverage rate $\hat{c} \in [0, 1]$ over

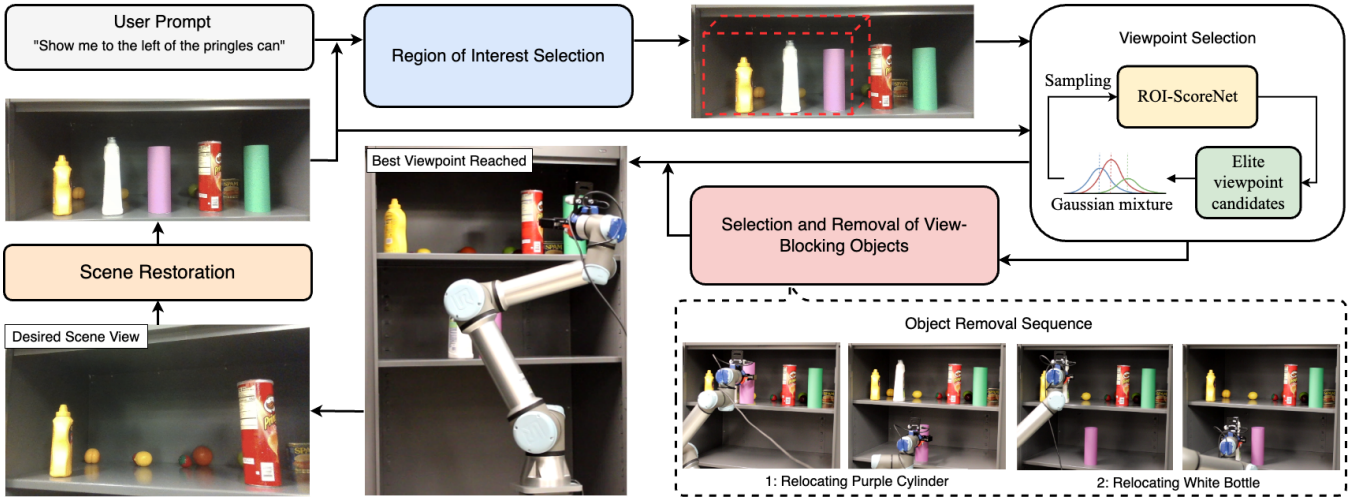

Fig. 1: This figure shows the language-guided active sensing pipeline with an example user prompt. The user prompt and initial scene are processed by the language linking module for the region of interest selection. With the region of interest, the MPC-GMM viewpoint selection algorithm uses ScoreNet and Gaussian mixture to select the optimal viewpoint for the environment setup. After moving to the optimal viewpoint, the view-blocking objects are removed sequentially using the blocking score formulation. Finally, the modified scene view is shown to the user. The system will move on to the next viewpoint if the desired coverage is not reached.

an ROI given the current scene S^t, the region of interest S_{ROI}, and any arbitrary viewpoint v^t.

$$\hat{c} \leftarrow f_\theta(S^t, S_{ROI}, v^t) \tag{3}$$

The ROI-ScoreNet is comprised of a two-channel 3DCNN layer [25] that embeds the scene representation S^t and region of interest S_{ROI} into a latent space Z_S. The viewpoint v^t is passed into a fully connected neural network to generate the latent viewpoint embedding Z_v. These latent features are concatenated and passed into another fully connected neural network to generate the estimated scene coverage $\hat{c}$. The ROI-ScoreNet is trained in a supervised manner using a dataset composed of $(S^t, S_{ROI}, v^t, c^{t+1})$ pairs from scenes with various objects in the simulation environment. The dataset is generated by moving the in-hand camera into a random viewpoint and recording the resulting coverage over the region of interest. During training, the objective is to minimize the Mean Squared Error (MSE) between the predicted coverage rate $\hat{c^{t+1}}$ and the ground truth c^{t+1}, i.e., $\frac{1}{M}\sum_{i=1}^{M} ||c^{t+1} - \hat{c}^{t+1}||^2$.

D. GMM-MPC Viewpoint Generation

While the ROI-ScoreNet allows fast estimation of scene coverage rate for various viewpoints around the scene, the remaining challenge is to devise an underlying algorithm that can effectively select the optimal one. Thus, we propose GMM-MPC, a bilevel MPC-style viewpoint generation algorithm derived from [6], [26]. Our GMM-MPC extends the bilevel MPC method by replacing the single Gaussian distribution with a Gaussian Mixture Model (GMM). The rationale behind this is the multi-modal nature of the best viewpoint distributions, as there can be many optimal viewpoints for a given ROI. By sampling only around one initial point, the resulting coverage is more likely to be a local maximum. The pseudocode of the GMM-MPC viewpoint generation method is shown in Algorithm 1. In the first stage, GMM-MPC selects N highest scoring viewpoints from uniformly generated samples around the scene as the initial centroids $\mu_1, ..., \mu_N$ of the Gaussian mixture model (lines 2-5). The algorithm then keeps sampling from the existing distribution, selecting the K elite samples determined by the ROI-ScoreNet, and refining each of the N Gaussian distributions in the model for M iterations (lines 6-10). Finally, the best viewpoints from the last round are returned as the candidates of the best next viewpoint (line 12). The optimal number of components N for the Gaussian mixture is chosen via cross-validation and set to seven, which yields better performance in general while maintaining a reasonable time consumption.

Algorithm 1 GMM-MPC viewpoint generation algorithm

1: initialize $\sigma_{\{N\}}$
2: $v^t_{\{S\}} \sim \text{Uniform}(v_s \in S^t_o)_{\{S\}}$
3: $\hat{c}^t_{\{S\}} \leftarrow f_\theta(S^t, S_{ROI}, v^t_{\{S\}})$ ▷ ROI-ScoreNet selection
4: $(v^t, \hat{c})_{\{N\}} \leftarrow \text{Sort}(v^t_{\{S\}}, \hat{c}^t_{\{S\}})[:N]$
5: $\mu_{\{N\}} \leftarrow v^t_{\{N\}}$ ▷ Initial GMM centroids
6: **for** $iter \leftarrow 1$ to M **do**
7: $\quad v^t_{\{B\}} \sim \text{GMM}(\mu_{\{N\}}, \sigma_{\{N\}})$
8: $\quad \hat{c}^t_{\{B\}} \leftarrow f_\theta(S^t, S_{ROI}, v^t_{\{B\}})$
9: $\quad (v^t, \hat{c})_{\{K\}} \leftarrow \text{Sort}(v^t_{\{B\}}, \hat{c}^t_{\{B\}})[:K]$ ▷ Elite samples
10: $\quad \mu_{\{N\}}, \sigma_{\{N\}} \leftarrow Fit(v^t_{\{K\}})$
11: **end for**
12: **return** $v^t_{\{K\}}$ ▷ Best viewpoint

E. View-Blocking Object Selection

In the cluttered environment setting, the ROI cannot be fully covered by simply changing viewpoints but requires object manipulation. In order to determine which object to remove at the current step, we proposed the following view-blocking rate estimation method to guide the process further.

For each observed object o_i, a voxelized point cloud is extracted using the depth image. Then, for each point p_i in the point cloud, a ray r is cast from the current camera viewpoint v^t and shoots towards it. As the ROI is a convex region, the ray will intersect it at two points, denoted p_{i1} and p_{i2}, with p_{i2} being the point further away from the camera. A viewpoint rate h associated with every object is calculated as the sum of line segment lengths:

$$h_{o_i} = \sum_{p_i \in o_i} ||p_i - p_{i2}||_2 \mathbb{1}_{(p_{i2}-p_i)\cdot(v^{t-1}-p_i)<0}(p_i) \quad (4)$$

The indicator term excludes points that are outside the region of interest and therefore does not contribute to blocking the view. Such points would have p_{i2} between the camera and p_i, causing the stated dot product to be positive. The rationale behind the proposed method is that the view-blocking volume is positively correlated to the integration of the line segment lengths from the objects' surface to the boundaries of the ROI following the viewpoint's orientation. We will use the notation `BScore`(S^t, S_{ROI}, v^t) to denote a function that computes the blocking score h_{o_i} for every object o_i that is currently observable. During execution, a max heap is kept during the object manipulation phase. The heap will be initialized with the blocking score of all observed objects with a blocking score greater than a minimum threshold. At any step, the robot arm removes the first object that popped out of the heap, which blocks the most view.

F. Language-guided Active Sensing Pipeline

Algorithm 2 describes our language-guided active neural sensing via object manipulation approach. The system starts with an initial viewpoint v^0 that looks directly at the scene's center. An overview image I^0 taken from this initial view is presented to the user to obtain a language prompt (lines 2-4). Once the prompt is received, the ROI-Generation module creates the corresponding region of interest S_{ROI} (Line 5). Next, the pipeline actively explores the ROI through visual perception and object manipulation. During this phase, the best viewpoint v^t is predicted by the GMM-MPC viewpoint generation algorithm leveraging the latest observation, the ROI, and the trained ROI-ScoreNet (line 7). Then, the robot moves the in-hand camera to the desired viewpoint v^t using the RRT-Connect motion planner [27] and returns the captured image I^t, the new scene representation S^t and updated ROI coverage score c (line 8). This is done through the `observe`(v) function, which computes the newly observed region through the depth information. Since there are view-blocking objects that occlude the view to the ROI, the system now moves on to the object manipulation phase. Based on the reverse order of the BScore associated with each observed object, the robot arm relocates them to a feasible region outside the ROI before making another observation at the previous viewpoint v^{t-1} (lines 11 - 15). The previous viewpoint is used because it aligns better with our object selection strategy, as the blocking score is calculated based on it. The object manipulation phase ends when all detected objects are removed or the ROI is already fully observed (line 10). All objects removed from the scene will be placed back in their original region so that the scene stays intact. Finally, as long as there are still unobserved regions in the ROI, the pipeline enters the next round, starting from the next best viewpoint selection through the GMM-MPC method(line 7). After the ROI is fully explored, the user is provided the minimal number of images $I_{\{t\}}$, which reveals all the details in the request region so that their request is fulfilled (line 20).

Algorithm 2 Language-guided Active Sensing Approach

1: $S_o^0 \leftarrow \emptyset; c \leftarrow 0; t \leftarrow 1$ ▷ initialization
2: $v^0; f_\theta$ ▷ initial viewpoint and ROI-ScoreNet function
3: $I^0 \leftarrow$ observe(v^0) ▷ Initial image for user to select ROI
4: $prompt \leftarrow$ UserInput(I^o) ▷ ROI description from user
5: $S_{ROI} \leftarrow$ ROI_generation$(prompt)$
6: **while** $t \leq T_{max}$ or $c \leq c_{max}$ **do**
7: $v^t \leftarrow$ GMM-MPC$(S^{t-1}, S_{ROI}, f_\theta)$
8: $(I^t, S^t, c) \leftarrow$ observe(v^t) ▷ new observation
9: $(o, h_o)_{\{N^t\}} \leftarrow$ sorted(BScore(S^t, S_{ROI}, v^t))
10: **while** $c \leq c_{max}$ **and** $N^t > 0$ **do**
11: $t \leftarrow t + 1$
12: $o^t \leftarrow$ pop$((o, h_o)_{\{N_{t-1}\}})$ ▷ view-blocking object
13: object_removal(o^t)
14: $(I^t, S^t, c) \leftarrow$ observe(v^{t-1})
15: $(o, h_o)_{\{N^t\}} \leftarrow$ sorted(BScore(S^t, S_{ROI}, v^t))
16: **end while**
17: scene_restoration() ▷ put removed objects back
18: $t \leftarrow t + 1$
19: **end while**
20: **return** $S_o^t, I_{\{t\}}$ ▷ final observed scene and image sets

G. Implementation Details

In this section, we provide the implementation details of our data collection procedure for the ROI ScoreNet training. First, we set up a virtual shelf environment in the IsaacGym simulator [28]. The shelf has a randomized width of 70-140 cm, height of 25-40 cm, and depth of 50-100 cm. The position of the shelf relative to the robot is also randomized. The shelf can be 30-70 cm away from the robot and 30-60cm above ground level. Inside the shelf, 10-20 different objects are placed with randomized positions and orientations. The objects are further grouped into two categories: large and small. The large objects are placed towards the front for task complexity, i.e., to block a majority of the view for any camera angle. The region of interest S_{ROI} is chosen by selecting a random anchor object in the scene and a random direction relative to said object from {left, right, front, behind}. From these randomly instantiated scenarios, we collect 10,000 different pairs of $(S^t, S_{ROI}, v^t, c^{t+1})$. These pairs are used for training our score function with 0.8, 0.1, and 0.1 data split for the training, validation, and testing set, respectively.

IV. Results

In this section, we present the result and analysis of the following experiments: 1) Comparison experiments to evaluate the performance of our proposed pipeline against multiple baselines in various randomized environments, which also

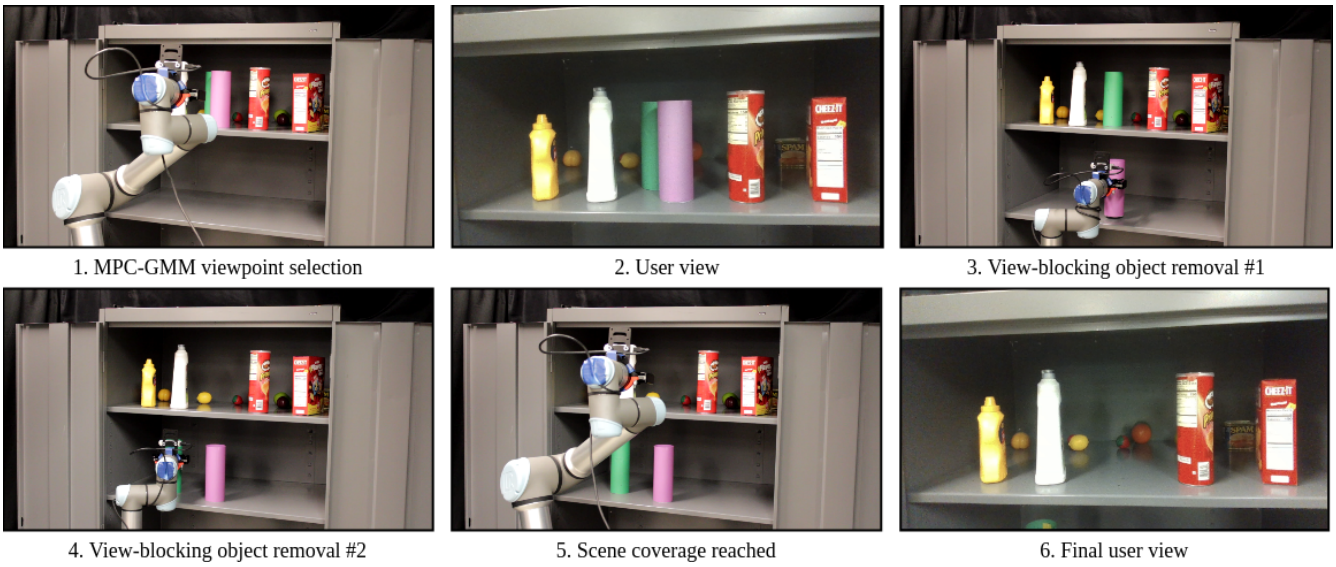

Fig. 2: This figure shows the real-world demonstration of our pipeline. The prompt given to the system is "Show me behind the purple cylinder". From the top left figure, the robot starts by choosing the viewpoint that's roughly in front of the object of interest. The view-blocking object selection algorithm identified two objects to remove. By removing the two objects, the desired coverage is reached and the scene is restored.

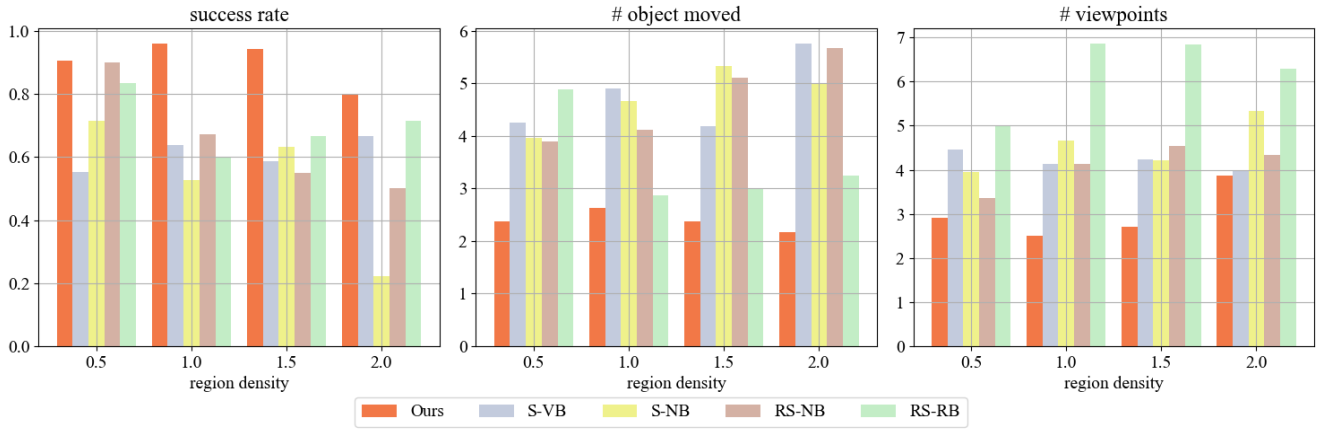

Fig. 3: This figure shows the performance of our method against various baselines under different region densities. Our method outperforms other baselines in all densities in terms of success rate and efficiency.

reveals the importance of each component in our methodology; 2) Real-world experiments to demonstrate the sim2real transfer capabilities.

A. Simulation Experiments

The simulated experiments against the baselines are conducted in 100 randomly generated scenes. The parameters of the environment that are randomized include the following: the robot's distance from the scene, scene dimensions, the objects placed in the scene, and their poses. For each method, the ROI exploration ends either when the ROI is fully observed or the number of unique viewpoints exceeds a certain threshold, where the former is considered successful while the latter is a failure. The following metrics are collected and utilized to evaluate the methods :

- Success Rate (SR): An experiment is considered successful if the ROI coverage rate exceeds the threshold using less than 8 unique viewpoints.
- Number of objects moved (# Object-moved): It tracks the number of objects that are removed from the scene before reaching the coverage threshold.
- Number of viewpoints (# Viewpoints): It records the number of unique viewpoints the robot has moved to before reaching the coverage threshold.
- Time: It shows the time consumed before reaching the coverage requirement, including the planning time of the various components and the time for the robot to move to the different viewpoints.

The ROI coverage threshold c_{max} is set to be 80% based on empirical observations. With an 80% coverage, the scene and the objects within it can be seen with sufficient clarity. The remaining 20% is attributed to the lack of information inside large objects and the remaining regions near the back surface of the ROI where no other objects exists.

The following baseline algorithms are designed as comparisons to highlight the performance of various modules.

Note that since our approach is the first method to solve language-guided active sensing via object manipulation problem, we utilize some heuristics like nearest object first or random related methods in the baselines. The baselines differ from our approach regarding the ROI coverage score prediction method and the object removal strategy. We use the following shorthand notation to describe the various aspects of the baselines: For the coverage rate prediction network, (RS) represents the proposed ROI ScoreNet while (S) is the original ScoreNet formulation from [6]. On the other hand, for the object removal strategy, (VB) stands for the proposed blocking-score-based method, (RB) indicates a random object selection method, and (NB) greedily chooses the nearest object from the camera viewpoint. Hence, the four constructed baseline methods include ROI-ScoreNet with random object selection (RS-RB), ROI-ScoreNet with nearest object selection (RS-NB), ScoreNet with blocking-score-based object removal (S-VB), and ScoreNet with nearest object selection (S-NB). Furthermore, all methods mentioned above use our GMM-MPC for viewpoint generation.

Table I shows the performance of our approach against the baselines. In all, our method achieves the highest success rate and lowest number of object manipulations and viewpoints, outperforming the baselines in all metrics. The random object baseline (RS-RB) significantly underperformed, highlighting that random object removals from ROI do not yield the best observability of the scene. The nearest object selection method (RS-NB) failed to achieve a higher success rate, indicating that the nearest object to the in-hand camera is not always the view-blocking object. Furthermore, we observed that in the GMM-MPC with ROI-Scorenet, the viewpoints are generated relatively farther away from the ROI with appropriate angles, leading to better scene coverage. Hence, the objects closer to the camera are often not the view-blocking objects. Thus, removing them does not help to increase the ROI coverage rate. On the other hand, the original ScoreNet formulation in [6] yields a low success rate since it considers the whole scene instead of just the ROI. Even with our proposed object selection algorithm, the coverage can hardly be improved because the coverage of the viewpoint generated did not overlap with the ROI. Lastly, when both viewpoint generation and object selection methods are replaced, i.e., (S-NB), the success rate and other metrics are further degraded as expected.

Figure 3 shows the visual comparison between our method and the baselines under varying region densities. Since the scene size, region of interest, and objects are randomized in each experiment trial, differentiating these factors can provide more insight into the performance of the various methods. Thus, we decided to stratify the results using region density, which is defined as the ratio of the number of objects in ROI and the ROI volume. As apparent from the graph, the difficulty of active ROI sensing increases with region density since more objects can block the view. It can also be seen from the graph that our method outperforms other baselines in terms of success rate, number of object manipulations, and number of viewpoints under all region densities.

Algorithms	Performance Metrics			
	SR (%) ↑	# Obj-moved ↓	# Viewpoints ↓	Time (s) ↓
Ours	**93**	$\mathbf{2.69 \pm 1.72}$	$\mathbf{2.79 \pm 2.42}$	$\mathbf{61.3 \pm 44.4}$
RS-RB	66	3.62 ± 3.54	6.60 ± 3.67	265.11 ± 190.35
RS-NB	65	4.78 ± 2.03	4.16 ± 1.46	145.54 ± 62.93
S-VB	60	4.68 ± 1.61	4.25 ± 1.46	152.59 ± 58.76
S-NB	58	4.84 ± 1.83	4.39 ± 1.48	159.40 ± 61.89

TABLE I: This table shows the results of the experiments. Our proposed method can be seen to have the highest success rate and is most efficient in object removal and viewpoint generation. Due to this efficiency, the processing time of our method is significantly lower than the other baselines.

B. Real Robot Experiments

We created three scenes with different user prompts to test the effectiveness of our pipeline in real-world settings. In real settings, we utilize an image segmentation model to generate the object point clouds. The image segmentation model we use is grounding-SAM [29]. This model utilizes the segment-anything [30] model and grounding-DINO [31] to produce segmentation masks from language inputs.

The scene settings and the prompts in the real experiments are crafted with the intent of testing the different capabilities of the proposed method. The prompt in the first scene, as can be seen in Fig. 1, is "Show me to the left of the Pringles can." Our pipeline relocates two objects to reveal the majority of the requested area. In the second experiment shown in Fig. 2, the robot is asked to explore the region behind the purple cylinder. This scene is significant in showing the effectiveness of our pipeline since the green cylinder is only partially visible. It can only be fully seen and identified as a view-blocking object after removing the purple one. The complete execution processes of all real experiments are available in the supplementary videos. In all, these successful real-world experiments demonstrate the robustness of our system under adverse lighting and can generalize to environments with daily-life objects with arbitrary placements.

V. Conclusion

This paper presents a novel method for language-guided active sensing via object manipulation in unknown, cluttered environments. Our approach first identifies an ROI given by a user via natural language. Then, it explores the ROI through a unique neural network-based viewpoint generation algorithm and a specially designed view-blocking object manipulation strategy. The efficacy of our method is first tested in the simulation environments and further verified in real-world scenarios. Comparative analyses indicate that our strategy surpasses other benchmark methods regarding the success rate and the number of object interactions. The real-world experiments justify the sim2real transfer capability of our pipeline, making it suitable for direct deployment. In our future work, we plan to build an end-to-end deep learning approach that optimizes viewpoints and object relocations to maximize the ROI coverage. We believe such an approach can realize even faster execution for real-time applications.

REFERENCES

[1] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Coordinated decentralized search for a lost target in a bayesian world," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 1, 2003, pp. 48–53 vol.1.

[2] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos, "A survey on the application trends of home service robotics," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018, pp. 1999–2006.

[3] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1396–1402, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:31771638

[4] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[5] S. Sarıel and H. L. Akın, "A novel search strategy for autonomous search and rescue robots," in *RoboCup 2004: Robot Soccer World Cup VIII*, D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 459–466.

[6] H. Ren and A. H. Qureshi, "Robot active neural sensing and planning in unknown cluttered environments," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2738–2750, 2023.

[7] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge university press, 2010.

[8] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1071–1078.

[9] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[10] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.

[11] C. I. Connolly, "The determination of next best views," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 432–435, 1985. [Online]. Available: https://api.semanticscholar.org/CorpusID:42940303

[12] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View planning for 3d object reconstruction with a mobile manipulator robot," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4227–4233, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14192829

[13] ——, "View/state planning for three-dimensional object reconstruction under uncertainty," *Autonomous Robots*, vol. 41, pp. 89–109, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:25430392

[14] S. Isler, R. Sabzevari, J. A. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3477–3484, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:15468346

[15] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta, "Supervised learning of the next-best-view for 3d object reconstruction," *Pattern Recognition Letters*, vol. 133, pp. 224–231, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865518305531

[16] B. Hepp, D. Dey, S. N. Sinha, A. Kapoor, N. Joshi, and O. Hilliges, "Learn-to-score: Efficient 3d scene exploration by predicting view utility," Cham, pp. 455–472, 2018.

[17] D. Gallos and F. P. Ferrie, "Active vision in the era of convolutional neural networks," *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 81–88, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:199441595

[18] T. Kollar, S. Tellex, D. K. Roy, and N. Roy, "Toward understanding natural language directions," *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 259–266, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:276090

[19] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. M. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247939706

[20] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *AAAI Conference on Artificial Intelligence*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:220828823

[21] Y. Kakiuchi, R. Ueda, K. Kobayashi, K. Okada, and M. Inaba, "Working with movable obstacles using on-line environment perception reconstruction using active sensing and color range sensor," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1696–1701.

[22] M. K. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1647–1654, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232290807

[23] T. Kollar, M. Laskey, K. Stone, B. Thananjeyan, and M. Tjersland, "Simnet: Enabling robust unknown object manipulation from pure synthetic data via stereo," *ArXiv*, vol. abs/2106.16118, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235683469

[24] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2015, pp. 4489–4497. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.510

[26] H. Bharadhwaj, K. Xie, and F. Shkurti, "Model-predictive control via cross-entropy and gradient-based optimization," in *Conference on Learning for Dynamics & Control*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:215827996

[27] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[28] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," in *Conference on Robot Learning*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53084610

[29] IDEA-Research, "Idea-research/grounded-segment-anything: Grounded-sam: Marrying grounding dino with segment anything; stable diffusion; recognize anything - automatically detect , segment and generate anything." [Online]. Available: https://github.com/IDEA-Research/Grounded-Segment-Anything

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick, "Segment anything," *ArXiv*, vol. abs/2304.02643, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257952310

[31] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," 2023.