Open-Set Object Retrieval in Clutter via Hybrid Vision-Language and Geometric Reasoning

Anonymous submission

***Abstract*— Robots must effectively retrieve novel objects in clutter, where a target may not be directly visible or accessible. In many real-world setups a robot must handle an open set of objects under constrained viewpoints and limited reachability; conditions often simplified in existing laboratory setups. Under these constraints, the robot must generate safe retrieval motions despite kinematic limitations and uncertainty regarding the novel objects' shapes. This work proposes a complete modular pipeline to address such targeted object retrieval problems in clutter. The focus is on evaluating solutions for a critical component of this pipeline: reasoning about the order of blocking objects to be removed before the target is available to be retrieved. Vision-Language-Models (VLMs) have been recently argued as pretrained solutions that can reason about such spatial object relationships given RGB images. Traditional engineered solutions in this space aim to heuristically identify object relationships from depth. This work shows that pretrained VLMs alone (without finetuning) cannot yet outperform even random object selection. Engineering solutions are superior but also lead to a lot of failure cases. This work proposes hybrid strategies for targeted object retrieval that combine the visual reasoning of VLMs with engineered dependencies via 3D reasoning, which improve performance. These observations are confirmed in extensive physics-based simulation experiments and real world experiments for a setup involving a robotic arm with a parallel gripper and a torso-mounted stereo camera. See online [1].**

INTRODUCTION

Targeted **O**bject **R**etrieval in **C**lutter (TORC) arises in manufacturing, logistics and service robotics. It involves multiple challenges: (i) perception, i.e., identifying the target object in the scene and whether it is retrievable, (ii) grasping, i.e., how the gripper should attach to an object to enable stable transfer, (iii) potentially reconstructing the 3D geometry of objects in the scene for safe retrieval, (iv) motion planning, i.e., computing the motions of a robotic arm for picking and retrieval, and (v) task planning, which involves identifying the sequence of objects that need to be removed from a scene in order for the desired target to be retrieved.

In some setups it is possible to engineer TORC solutions by assuming known models or homogeneous geometry or structuring the workspace to reduce clutter and guarantee visibility. This comes at the expense of space efficiency and costly automation equipment. This work focuses on less structured and occluded setups as the one displayed in Fig. 1. For such instances, there is still the need for a generalizable, robust TORC solution, which can handle unknown, heterogeneous objects under significant occlusions that arise due to a camera's viewpoint, and limited robot reachability.

[1]http://2026vlmtaskplanningclutter.github.io/

Fig. 1. (Top Left) Bird's eye view of an experiment. (Top Right, Bottom Left/Right) Segmented images from the robot's view before a pick until the hidden black box (target) is retrieved. The target is labeled 0 when found.

A front view of a cluttered scene creates significant occlusions, which can hide the target object and require the removal of blocking objects until the target is revealed. At the same time, clutter and robot kinematic constraints can restrict the set of reachable grasps and safe retraction paths. In particular, safe retraction paths become uncertain when 3D object models are not available and object reconstruction can result in unreliable shape completions. The situation becomes harder when objects are placed tightly close to each other, or when object stacking or piles are present. In these cases, the removal of an object may cause objects to fall off their support surface.

A complete, modular TORC pipeline: This work aims to address such challenging instances of TORC. For this reason, it first builds a complete, modular pipeline that brings together state-of-the-art tools for object detection and segmentation, grasp generation and motion planning.

Object Selection for TORC: The developed pipeline allows to focus on a key task planning component of TORC, i.e., selecting which occluding object to remove next so as to expose the target object quickly and reliably. Engineered approaches for object selection construct scene graphs or Dependency Graphs (DGs) [1], [2] from depth to encode reachability and visibility relations, such as "behind" or "below" object relationships. While structured and interpretable, these heuristic representations become fragile as clutter and geometric complexity increase. Vision-Language Models (VLMs) have been recently argued [3], [4], [5] as model-free alternatives that can provide spatial reasoning capability by operating over RGB images and given large-scale pretraining.

DG-based and VLM-based task planners operate over different sources of information, depth and RGB images re-

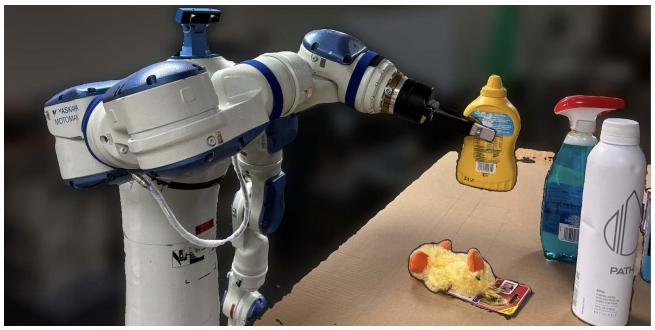

Fig. 2. Real-world experiment of the robot picking an object given a torso-mounted stereo camera viewpoint.

spectively. This motivates looking into hybrid task planners, which use VLMs and RGB information together with DGs and depth. Experiments both on simulated and a real robot show that the most successful task planner arises when fusing the structured rigor of geometric reasoning with the generalized semantic flexibility of modern VLMs. The best performing hybrid planner utilizes the VLM as the final arbiter of object selection, which operates over both RGB data as well as detected relationships over depth information. In particular, the VLM is prompted to apply its semantic and spatial reasoning to correct any inaccurate object relationships due to geometric heuristics, explicitly distinguishing a true "behind" relationship from a "below" relationship.

In summary, this work addresses a challenging class of TORC problems and provides a comparative analysis of different TORC task planners, quantifying their performance based on retrieval success and efficiency across diverse cluttered scenarios in simulation and real experiments. In particular, it **contributes**:

1. A complete, modular pipeline for solving a challenging class of TORC problems involving unknown objects, significant clutter, a single viewpoint, and limited reachability. The framework has been designed to be extendable to different robots, planners, and perception modules. This pipeline enables the systematic evaluation of different task planners for object selection in the context of TORC challenges.

2. A new dataset of heavily cluttered simulated scenes with significant occlusions, built on top of grasping benchmarks [6], where object selection critically affects retrieval success.

3. Hybrid task planners for object selection in TORC that integrate a VLM's spatial reasoning capabilities from images to refine geometrically-derived object relationships.

4. An evaluation of multiple task planners for object selection in the context of TORC that span the space of VLM-based to DG-based and hybrid integrations. For an "expert" baseline, experiments are performed with a human in the loop for object selection.

The evaluation reveals that pretrained state-of-the-art VLM-based solutions - without finetuning - do not yet possess sufficient spatial reasoning capabilities when operating over images. Integrating VLMs, however, with geometric reasoning performed over depth data can result in improved robustness and reliability in sequential action selection for TORC, while reducing the dependence on engineered heuristics of exclusively DG-based solutions.

Related Work

Mechanical Search: When a target object is occluded or inaccessible, an effective strategy is to perform mechanical search, sequentially rearranging objects to expose and retrieve the target. Earlier work takes advantage of the visibility and reachability structure between objects in simple scenes with known object geometry to come up with such a rearrangement plan [7]. Another work proposes searching for a target object by updating the belief state after object manipulations are performed to maximize visibility using a Gaussian process [8], but this approach tries to rearrange the scene to make all objects visible simultaneously, thus not being action-optimal for retrieval tasks. Towards improving the action efficiency, another work uses reinforcement learning to find and retrieve the target object from clutter [9], but the types of objects used were exclusively cuboids. In order to find the target object without explicit relationship reasoning one work learned probability distributions called x-rays [10] and another innovated in end-effector hardware to be able to efficiently pick and use suction with the same tool [11]. It was later extended to reason about stacked objects [12] albeit with rigid assumption on object geometry.

Scene/Dependency Graphs for Manipulation: Task planning literature has used scene graphs to plan object picking for retrieval. For example, one work used hierarchical scene graphs to plan with a Regression Planning Network [1], while another used a POMDP formulation [13]. Similar formulations have been used for object manipulation and target object discovery [14], [15]. Many efforts focus on machine learning approaches for detecting object relationships that can then inform scene graph construction [16], [17], [18].

While scene graphs are object-oriented, other works have constructed robot-oriented graph structures, such as traversability graphs that maintain pairs of object poses reachable by the robot [19]. More recent work combined object relationships and robot reachability estimates to generate a dependency graph keeping track of which objects can be manipulated first resulting in a resolution complete algorithm for object retrieval [2]. Most task and motion planning works assume known object models and employ object pose estimatoion [20], [21].

VLM Planning: Prior efforts have attempted using VLMs directly or with fine-tuning for determining spatial reasoning and object relationships rather than constructing dependency graphs explicitly [22], [23], [24]. Some have attempted to use VLMs for every part of the planning process of object retrieval (perception, grasping, and object selection) [4] or in conjunction with classical task and motion planning for longer horizon tasks [25], [26]. More recent techniques propose constructing dependency or obstruction graphs with privileged information for a large dataset of scenes to construct an expert policy that could be used to finetune a VLM [27], [28]. In contrast, the hybrid methods proposed in this paper use a VLM (without finetuning) to either inform dependency graph generation or be informed by reachability dependencies detected at runtime.

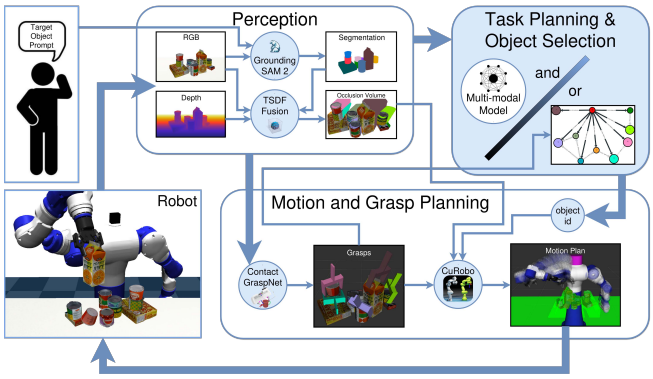

Fig. 3. The modular robot system architecture used in this work to retrieve occluded or hidden objects in clutter. It integrates solutions for perception (top middle), task planning (top right - focus of this paper's evaluation), as well as motion and grasp planning (bottom right).

TORC Problem Setup

Workspace: Let the workspace consist of a tabletop next to a robot arm with a pinch gripper and an RGB-D camera. The tabletop contains a set of n unknown, heterogeneous objects $\mathcal{O} = \{o_1, o_2, ..., o_n\}$, which may not be directly graspable from any orientation. The objects initially rest stably either on a support surface or on top of other objects, or both. A specific object $o_t \in \mathcal{O}$ is designated through natural language as the target object, which may be partially or completely occluded by other objects from the camera. The target object and majority of other objects are presumed to be initially graspable. Let $\mathcal{P}_k \subseteq \mathcal{O}$ be the subset of objects picked and removed from the workspace after the k^{th} action. Let $\mathcal{D}_k \subseteq \mathcal{O}$ refer to objects dropped from the tabletop. Let $\mathcal{R}_k = \mathcal{O} \setminus \mathcal{P}_k \setminus \mathcal{D}_k$ be the objects remaining in the workspace at time k.

Input: The input to the robot includes its kinematic proprioception, a natural language description of the target object, and egocentric images from the single, calibrated RGB-D camera viewing the front of the tabletop $Z_k = (I_{RGB}(k), I_D(k))$ (see Fig. 2 for the camera location relative to the scene). No prior 3D geometric models or poses are provided for any object o_i.

Output: The action space A consists of discrete pick-and-retrieval actions, where each action $a_k = (o_i, g_j)$ involves selecting an object $o_i \in \mathcal{O}$ and outputting a motion plan to retrieve o_i with grasp g_j.

Objective: This work aims to retrieve the selected target object o_t while minimizing the number of pick actions and object drops. Specifically, at any given time step k, given observation Z_k, the goal is to select an object to retrieve $o_i \in \mathcal{R}_k$ to minimize the total number of actions taken α until the target object is picked, i.e., until $o_t \in \mathcal{P}_\alpha$.

A Modular Pipeline for TORC

Fig. 3 visualizes the modular architecture designed to retrieve $o_t \in \mathcal{O}$ without prior object models. The pipeline executes the following sequence for each time step k until the retrieval condition is met:

1) Sensing: The robot captures observation Z_k, generating a point cloud with segments for each visible non-target object $o_i \neq o_t$ and a distinct segment for the target object o_t (if already visible). Additionally, a scene voxelization of the occlusion volume is created.
2) Grasp Planning: Candidate grasps $g_j \in SE(3)$ are sampled for each segmented o_i, then filtered to satisfy reachability and collision constraints.
3) Task Planning: Among the set of graspable objects, the selection policy chooses the next object $o_i \in \mathcal{R}_k$ to be retrieved.
4) Motion Planning: For the selection o_i and its highest scoring grasp g_j, the system computes four motions:
   - Approach: Moving to a pre-grasp pose while avoiding collisions with objects $o \in \mathcal{R}_k$.
   - Grasp: A Cartesian motion from pre-grasp to g_j.
   - Extraction: A Cartesian lift upward, high enough to remove o_i from the clutter.
   - Deposition: Moving o_i to a drop location, which upon success, updates the set of removed objects to $\mathcal{P}_{k+1} = \mathcal{P}_k \cup \{o_i\}$.

Perception: The perception process takes as input an RGB image and a depth image. The perception block of Fig. 3 (top-middle) provides an example of perception outputs given one of the simulated scenes.

On the real system, high quality depth images are obtained using FoundationStereo [29] by feeding the stereo images captured by a ZED camera. Instance segmentation is then performed on the color image via Grounding SAM 2 [30], [31]. To distinguish between the target and other objects, Grounding SAM 2 is called twice: once with a short natural language description of the target object and then prompted just with the word "Object" to segment all objects including occluding and background objects.

Given the segmentation image, depth image, and color images, an occlusion volume of the scene is maintained per object via a modification to an existing implementation of TSDF integration [32]. Rather than keeping track of depth values near the object surface, the TSDF formulation is modified to update occluded voxels to visible voxels as new snapshots are taken from the same viewpoint after every action. Furthermore, in order to determine which object is responsible for which occlusion regions, a voxel mask is maintained where each voxel value is a bit map with the index of a bit representing the segmentation ID of the object at the voxel location. This bit mask volume is implemented by following related work in the literature [33].

Grasp Planning: The goal of the grasp planner is to produce high quality, reachable, and collision free grasps. The resulting grasps inform the task planner which objects are immediately graspable. Reachable grasps in collision inform the task planner about which objects are blocking other objects. Given the surface point cloud and object mask produced by the perception process, a learned grasping planner (Contact-GraspNet) [34] is used to produce candidate grasps g_i for each object o_i. Candidate grasps are first checked for IK feasibility and removed if not feasible. Grasps that are currently feasible are validated. Validated grasps

must have:

- A kinematically-feasible IK solution.
- An IK solution that is collision-free with the tabletop.
- Encompass only the intended o_i within the gripper fingers.
- Collision check between the gripper and the surface points of o_i (though contact with the assigned occlusion volume is permitted).

The grasping process ultimately returns a set of validated grasps and their corresponding robustness scores as evaluated by the Contact-GraspNet [34] grasp generator.

Motion Planning: The motion planning phase plans the motion to retrieve an object o_i provided by the task planner given a set of scored and validated grasps for o_i. If any motion-planning query fails, the next-best grasp candidate g_{j+1} for o_i is attempted. This work leverages CuRobo [35] for the global motion queries and Pink[2] for Cartesian motion planning.

A planned grasp pose is achieved by moving from a pre-grasp to the grasp in a straight line in the workspace to prevent unintended collisions that might be caused by joint-space motion. The first step of the motion planning, the *approach*, motion plans the robot arm to a pre-grasp pose to avoid collisions between the robot and both the surface points and occlusion volume. The *grasp* comes next, which is the Cartesian motion from the pre-grasp to grasp g_j. This Cartesian motion does not check for collisions due to it being unlikely for there to be collisions between a collision free grasp and pre-grasp. After moving to the grasp pose, the gripper closes.

After grasping the object, the object needs to be extracted safely from the scene. The *extraction* step first produces a Cartesian motion directly upwards to free the picked object from clutter. This Cartesian motion is not collision-checked, as the upwards motion of the grasped object should only produce slight disturbances to neighboring objects. Then the grasped object geometry is reconstructed. The convex hull of the grasped object segmented point cloud and occlusion volume is attached to the gripper as additional collision geometry. Collision volume neighboring the reconstructed object is removed to enable feasible motion planning while only allowing slight scene disturbance. From the lifted state from *extraction*, the *deposition* step motion plans with this attached reconstructed geometry to move the gripper to a drop location, avoiding collision with both the surface points and occlusion volume. The gripper drops the object o_i then the arm returns to the starting pose to begin the loop again.

TASK PLANNING FOR TORC

TORC task planning must determine a sequence of objects to be picked (i.e., picking an object and removing it from the tabletop workspace) to discover and subsequently retrieve the target object. The target need not be directly visible or pickable from the robot's camera given the initial view. The corresponding solution should also minimize scene disturbance that leads to undesired effects, such as non-target objects falling out of the support surface, i.e., the tabletop. Thus, the primary optimization objective is to minimize the number of performed pick actions until the target object is retrieved, and a secondary objective is to minimize the number of non-target objects dropped out of the tabletop.

A random-choice baseline (Approach 0 below), a VLM approach adapted from a previous work [4] (Approach 1 below), a DG approach adapted from a previous work [2], two proposed hybrid methods (Approaches 3 & 4 below), and an "expert" human demonstration solution are considered for evaluation of different TORC task planners.

Approach 0. RANDOM

The first alternative task planner is a random baseline (RANDOM). That is, of all the objects having valid grasps, an object is chosen randomly to be picked.

Approach 1. VLM-SELECT

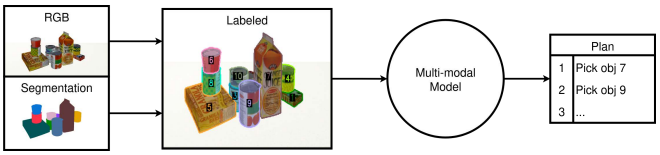

Fig. 4. Overview of the VLM-SELECT approach.

For the VLM-SELECT selection method, a VLM is prompted with a labeled image of the workspace and is asked for a sequence of actions to retrieve the target object, as shown in Fig. 4. Only the first action is executed and then the process repeats. The VLM used in this work is Google DeepMind's Gemini Robotics-ER 1.5 [3], as it is pretrained for robotics tasks. VLM-SELECT does not have knowledge of which objects have valid grasps and is re-queried if a non-graspable object is chosen. This selection strategy is largely inspired from previous work in the literature [4]. The VLM prompts used by the current work are available online.[3]

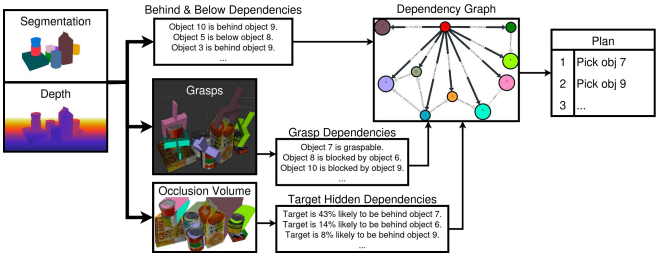

Fig. 5. Overview of the DG-SELECT approach.

Approach 2. DG-SELECT

An engineered selection method using a dependency graph between objects (DG-SELECT) is adapted from previous work in the literature [2], which assumed knowledge of objects' models and poses. The dependencies in the current work are extracted from perceptual information without access to object models.

[2]https://github.com/stephane-caron/pink

[3]http://2026vlmtaskplanningclutter.github.io/prompts

The dependency graph is a directed graph $G(V, E)$ where each node $v \in V$ represents a visible object and each directed edge $e = (v_1 \rightarrow v_2) \in E$ represents a dependency in pick order between objects v_1 and v_2, i.e., object v_1 cannot be picked until v_2 is picked. Dependency relationships include "behind", "below", and "grasps blocked by". These relationships are detected as follows.

To first step is to dilate each object mask and mark two objects as within the set of adjacent object masks M if their dilated masks intersect. For each adjacent pair $(m_a, m_b) \in M$, compare the average depth along their shared boundaries. The object with the larger average depth is labeled as "behind" the other. Some of these pairs are turned into "below" relationships if the following conditions are met. for object a to be marked "below" b the following need to hold for a sufficient number of boundary points (> 50) on a: (1) The points are in close proximity to b in 3D space.(2) The points on a have surface normal vectors pointing within $45°$ of the positive z axis. (3) The points on a are below the closest point on object b.

In addition to the identified set of "behind" object edges H and "below" object edges O, there are also "grasp blocked by" dependency edges B in the dependency graph for objects v_1 that have no collision-free, valid graph. Grasp blocking dependency edges are built for objects v_1 and v_2 where the robot configurations corresponding to IK solutions of grasps of v_1 result in collision with v_2 (or its corresponding occlusion volume). A 'grasp blocking' edge $v_1 \rightarrow v_2$ has a weight equal to $\frac{\text{number of grasps blocked by object } v_2}{\text{total number of (blocked) grasps for } v_1}$. Edges where v_1 has at least one valid, collision-free grasp are not included in B.

$G(V, E)$ is made up of all segmented objects as nodes $v \in V$ and edges from O, B, and H with an order of dependence type priority. All edges in O are added to E with weights $1/\text{DEG}(v_1)$. Then a subset of edges in B are added to E where a given edge $v_1 \rightarrow v_2 \in B$ is added if v_1 in $G(V, E)$ does not have an outgoing edge. After adding edges from B, the same process is applied to adding edges from H, but with weights $1/\text{DEG}(v_1)$. If a target object is not visible, a hidden target node is added to V. Then every segmented object is added an edge towards the hidden target node with a normalized weight proportional to the volume of the occlusion region cast by that object.

The method selects the sink node that maximizes the sum over all simple paths to the target, where each path score is the product of its normalized edge weights.

Approach 3. VLM-FIXES-DG

A hybrid approach referred to as VLM-FIXES-DG queries the VLM to directly output candidate dependency relations ("below", "behind") between pairs of objects generated instead of the heuristic process employed by the DG-SELECT approach (fig. 6). The approach relies on the "visual understanding" of the VLM to detect object dependencies but still aims to construct an explicit dependency graph. In this way, it is less dependent on parameter tuning of the heuristics described in the previous section for DG-SELECT.

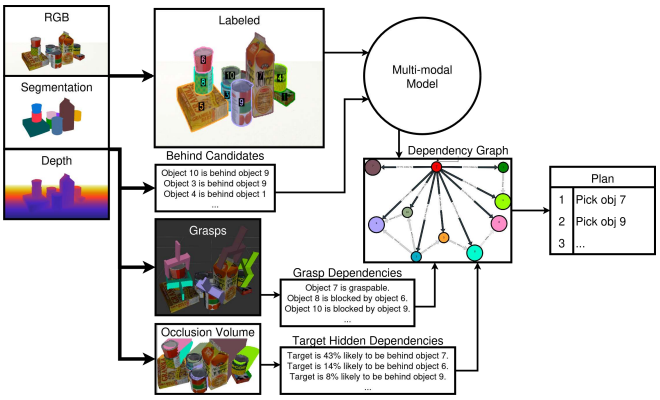

Fig. 6. Overview of the VLM-FIXES-DG approach.

The general prompt outline is similar to that of VLM-SELECT. The output format, however, is now a JSON array of dependencies, and the task specification contains a JSON array of "behind" dependencies (<*candidates*>).

The returned list of dependencies from the VLM are used to construct a Dependency Graph and the next object selected for picking is chosen similarly to DG-SELECT.

Approach 4. VLM+GRASPS

An alternative integration of VLMs with algorithmic reasoning is shown in Fig. 7. The idea for this method is to identify that an object o_1 is blocking another object o_2 when grasps on o_2 are invalid due to collision with o_1. The VLM+GRASPS selection method allows a VLM to make the ultimate choice of which object to be picked next, but prompts the VLM with grasp dependencies as a powerful feature to aid in VLM reasoning. A graph $G(V, E)$ is constructed with all segmented objects as nodes $v \in V$ and edges $E = B$, where the set B is defined as the set of "grasp blocked by" edges in DG-SELECT. If the VLM selects an object that is not directly pickable, then the approach is attempted again.

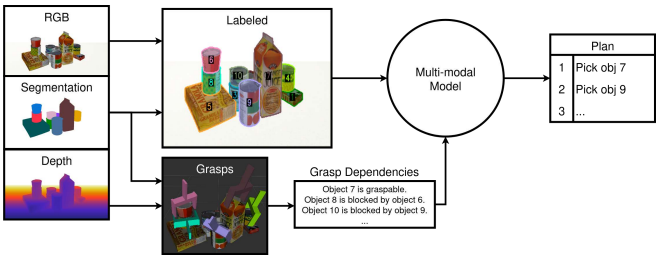

Fig. 7. Overview of the VLM+GRASPS approach.

The overall prompt outline is nearly identical to VLM-SELECT but it is also given a scene description before the task is specified. The $G(V, E)$ is described as a list of grasp dependencies written in natural language sentences of the form "Object A is blocked by object B" or of the form "Object C is graspable" if object C has any valid grasps available.

Fig. 8. Simulated scenes. **Left:** Robot observation of the initial scene setup. **Right:** Same scene visualized from the back. The target is highlighted by a red segment and 0 label.

EXPERIMENTS

Simulated Dataset Generation

An important aspect of the evaluation involved generating useful scenes that are both challenging, i.e., they require multiple object removals till the target is reachable, but are also solvable. Randomly placing objects can frequently lead to scenes without valid grasps. These failure modes may arise due to poor grasp generation or due to complex object-object interactions during retraction, which cannot be trivially resolved even by leveraging privileged perception in simulation with prehensile manipulation. Since the focus of this paper is on evaluating the object selection process, the scenes should be solvable with the developed pipeline and picking operations. Therefore, scenes were generated and manually labeled, tuned, and filtered using hand-specified rules to ensure solvability. This resulted in a dataset of 40 `TORC` scenes, each with a set of objects placed on a tabletop and a target object to be retrieved. See Figure 8 for example simulated scenes.[4]

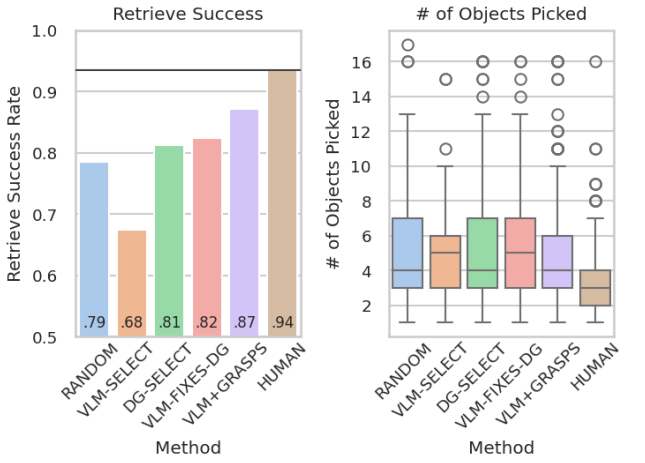

Fig. 9. Simulation results. (Left) Success rate of each method. The horizontal line marks the human expert success rate. (Right) Distribution of number of picks required.

[4]More scenes: http://2026vlmtaskplanningclutter.github.io/scenes

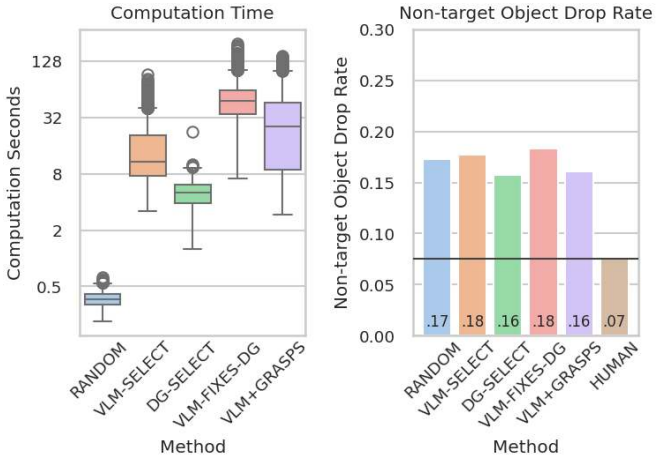

Fig. 10. Simulation results. (Left) Distribution of computation times in seconds. (Right) Drop rate of each method for non-target objects. The horizontal line marks the human expert drop rate.

To initialize the process, scenes were taken from the GraspClutter6D dataset [6]. Each scene from GraspClutter6D consisted of a cluttered arrangement of objects as well as ground-truth grasps for each object. A tabletop support surface is then defined in MuJoCo for the objects to rest on. To produce the retrieval-focused scene dataset, the following steps were executed:

1) A human annotator manually labeled candidate targets in each scene, with preference for occluded objects.
2) Candidate targets that were irretrievable were then pruned: For each target, all objects (1) directly in front, (2) within 15 cm to the left or right, and (3) taller than the target were recorded. If any of these objects possessed no ground-truth grasps that were collision-free and IK-feasible, the target was pruned.
3) Scenes solvable through a single pick were discarded as trivial. Conversely, if the expert was unable to retrieve the target even after attempting to remove the majority of objects, the scene was adjusted until it admitted a solution with a task sequence of at least two steps. Importantly, this includes avoiding inherently infeasible configurations caused by geometric limitations of the pinch gripper, such as objects presenting no graspable region, rather than excluding failures specific to the proposed method.

Simulated Evaluation

Once scenes were validated, 5 different human experts performed object selection on each of the 40 problems, for a total of 200 human trials. The corresponding approach is marked as `HUMAN` and corresponds to an "expert" baseline. Each selection strategy (`RANDOM`, `VLM-SELECT`, `DG-SELECT`, `VLM+GRASPS`, `VLM-FIXES-DG`) was evaluated 10 times on each problem for a total of 400 experiments per approach. In simulated experiments, ground-truth object instance segmentation from MuJoCo is used to isolate the evaluation of object sequencing.

In Figs. 9 and 10 the success rate, number of objects picked until success, computation time and the non-target-object drop rate are reported. The non-target object drop rate

is the proportion of experiments that caused a non-target object to fall off the tabletop during a retrieval. For the number of objects picked and computation time graphs, the distributions are reported over all performed experiments.

The results show that using a VLM out of the box (despite being pretrained for robotics) is even under-performing making random selections among directly graspable objects. The dependency graph approach, DG-SELECT, is more successful than RANDOM and VLM-SELECT but critically depends on the parameters of the heuristics, which were tuned to optimize performance in these experiments. The VLM-FIXES-DG approach, which used the VLM to automatically define the dependency graph achieved similar success rate to DG-SELECTwith some increase in number of objects picked till success was achieved. Feeding the grasp information into the VLM and allowing the VLM to make the object selection as in the VLM+GRASPS approach achieves the highest success rate, while requiring a similar number of objects to be picked as in the DG-SELECT approach. The VLM-based solutions required increased computation time due to the call to the remote Gemini service. All the automated task planners exhibit similar rates of non-target objects rolling off the tabletop during the retrieval process, which in some cases arises due to simulation artifacts.

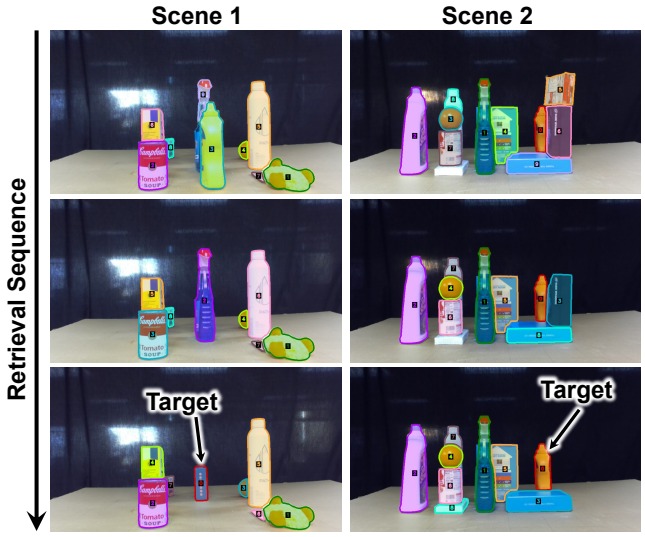

Fig. 11. Example retrieval on 2 different scenes in real-world, human-in-the-loop experiments. Images are taken before each retrieval and object instance segmentation is overlaid. The identified target object is labeled as 0 with red silhouette.

Real-World Evaluation

The real-world experiments were performed with a single arm of a Yaskawa Motoman SDA10F robot using a Robotiq 85 gripper and a Zed Mini camera as shown in Fig. 2. Grounding SAM 2 is used for object segmentation.

On the real robot, the task planning approaches were evaluated on 3 difference scenes, where each method was repeated 3 times per scene, for a total of 9 experiments per method and 45 total experiments. These scenes were constructed by placing household objects on a large grid on a tabletop. Their positions were recorded for replication to be evaluated across the methods. The scenes were designed such that they required at least 2 picks to retrieve the target. Figs. 1 and 11 show the real-scenes. Results are reported in Figs. 12 and 13 similarly to the simulated scenes.

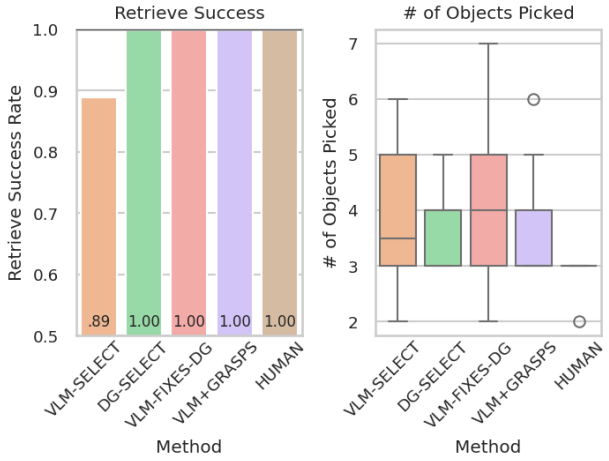

Fig. 12. Real-world results. (Left) Success rate of each method. The horizontal line marks the human expert success rate. (Right) Distribution of number of picks required.

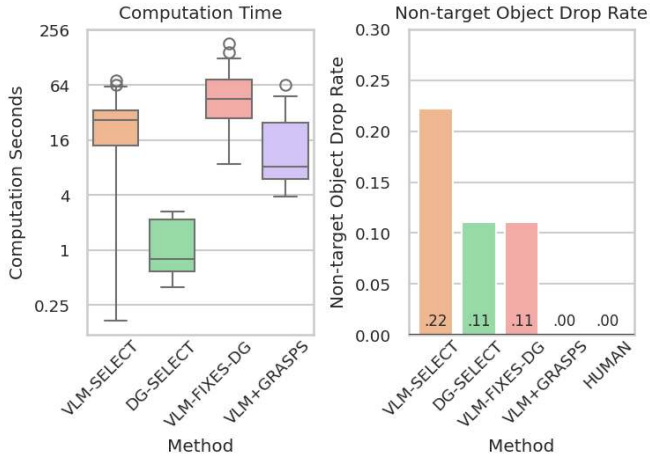

Fig. 13. Real-world results. (Left) Distribution of computation times in seconds. (Right) Drop rate of each method for non-target objects. The horizontal line marks the human expert drop rate.

The real world experiments yield similar observations to the simulated experiments. While the real system consumes more noisy perception data, which can lead to incorrect target detection, success was quite high. A potential advantage of the VLM-based solutions is that they experience a smaller real-to-sim gap given real image input. Still, the VLM-SELECT approach underperforms the alternatives, as it has the lower success rate. The methods that make the final selection via a Dependency Graph, i.e., DG-SELECT and VLM-FIXES-DG, have high success rate but still occasionally dropped objects. The VLM+GRASPS hybrid solution was the best performing method, solving all real scenes without any object drops similar to the human experts.

CONCLUSION

This work evaluates the efficacy of VLMs as task planners for **T**argeted **O**bject **R**etrieval in **C**lutter (TORC) by comparing against and integrating with graphical solutions based on

geometric information. It appears that pretrained VLMs can benefit from additional features about geometric constraints that can be built from algorithmic processes and inputted through language to improve their reasoning capabilities.

While the hybrid strategies proposed demonstrated improved performance, drops of non-target objects were observed in simulation. These arose from the lack of object models and a simple object reconstruction of the picked object (the convex hull of its point cloud segment) as well as from simulation artifacts. This led to collisions with other objects as a picked object is removed from the scene. Furthermore, there were situations that once the scene was disturbed enough, there were no grasps available to proceed. The developed modular pipeline, however, can also adopt improved modules for grasping and reconstruction to decrease the number of non-target dropped objects. Alternatively, non-prehensile manipulation primitives can be introduced.

A potential use of the proposed hybrid solutions is that they can provide examples of effective manipulation sequences for TORC. These can then serve as valuable data for training or finetuning future VLMs as well as VLAs. Then, such improved, finetuned models may be able to make correct object selection choices for targeted picking in cluttered scenes. Future work will focus on leveraging the generated solutions for this purpose. Adopting VLAs in this context can also allow a transition from the current open-loop picking pipeline to a more reactive, closed-loop retrieval.

REFERENCES

[1] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *ICRA*, 2021.

[2] D. Nakhimovich, Y. Miao, and K. E. Bekris, "Resolution complete in-place object retrieval given known object models," in *ICRA*, 2023.

[3] G. R. Team, A. Abdolmaleki, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, A. Balakrishna, N. Batchelor, A. Bewley, J. Bingham, *et al.*, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025.

[4] G. Tziafas and H. Kasaei, "Towards open-world grasping with large vision-language models," in *CoRL*, 2024.

[5] B. Chen1, Z. Xu, S. Kirmani1, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia, "Spatial vlm: Endowing vision-language models with spatial reasoning capabilities," in *CVPR*, 2024.

[6] S. Back, J. Lee, K. Kim, H. Rho, G. Lee, R. Kang, S. Lee, S. Noh, Y. Lee, T. Lee, *et al.*, "Graspclutter6d: A large-scale real-world dataset for robust perception and grasping in cluttered scenes," *RA-L*, 2025.

[7] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa, "Object search by manipulation," *Autonomous Robots*, vol. 36, no. 1, pp. 153–167, 2014.

[8] J. Poon, Y. Cui, J. Ooga, A. Ogawa, and T. Matsubara, "Probabilistic active filtering for object search in clutter," in *ICRA*, 2019.

[9] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *ICRA*, 2020.

[10] H. Huang, M. Dominguez-Kuhne, V. Satish, M. Danielczuk, K. Sanders, J. Ichnowski, A. Lee, A. Angelova, V. Vanhoucke, and K. Goldberg, "Mechanical search on shelves using lateral access x-ray," in *IROS*, 2021.

[11] H. Huang, M. Danielczuk, C. M. Kim, L. Fu, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg, "Mechanical search on shelves using a novel "bluction" tool," *ICRA*, 2022.

[12] H. Huang, L. Fu, M. Danielczuk, C. M. Kim, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg, "Mechanical search on shelves with efficient stacking and destacking of objects," *arXiv preprint arXiv:2207.02347*, 2022.

[13] K. N. Kumar, I. Essa, and S. Ha, "Graph-based cluttered scene generation and interactive exploration using deep reinforcement learning," in *ICRA*, 2022.

[14] W. Zhao and W. Chen, "Hierarchical pomdp planning for object manipulation in clutter," *Robotics and Autonomous Systems*, vol. 139, p. 103736, 2021.

[15] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *ICRA*, 2019.

[16] T. Mota and M. Sridharan, "Learning the grounding of expressions for spatial relations between objects," in *Workshop on Perception, Inference and Learning for Joint Semantic, Geometric and Physical Understanding at ICRA 2018*, 2018.

[17] J. Li, D. Meger, and G. Dudek, "Learning to generalize 3d spatial relationships," in *ICRA*, 2016.

[18] M. Neau, P. E. Santos, A.-G. Bosser, and C. Buche, "React: Real-time efficiency and accuracy compromise for tradeoffs in scene graph generation," 2024. [Online]. Available: https://arxiv.org/abs/2405.16116

[19] C. Nam, S. H. Cheong, J. Lee, D. H. Kim, and C. Kim, "Fast and resilient manipulation planning for object retrieval in cluttered and confined environments," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1539–1552, 2021.

[20] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.

[21] C. Mitash, B. Wen, K. Bekris, and A. Boularias, "Scene-level pose estimation for multiple instances of densely packed objects," in *CoRL*, 2020.

[22] W. Ma, Y.-C. Chou, Q. Liu, X. Wang, C. de Melo, J. Xie, and A. Yuille, "Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning," in *NeurIPS*, 2025.

[23] N. Zantout, H. Zhang, P. Kachana, J. Qiu, G. Chen, J. Zhang, and W. Wang, "Sort3d: Spatial object-centric reasoning toolbox for zero-shot 3d grounding using large language models," in *IROS*, 2025.

[24] C. H. Song, V. Blukis, J. Tremblay, S. Tyree, Y. Su, and S. Birchfield, "Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics," in *CVPR*, 2025.

[25] T. Lee, G. Kang, B. Wen, Y. Kim, S. Back, I. S. Kweon, D. H. Shim, and K.-J. Yoon, "Delta: Demonstration and language-guided novel transparent object manipulation," *arXiv preprint arXiv:2510.05662*, 2025.

[26] Z. Yang, C. Garrett, D. Fox, T. Lozano-Pérez, and L. P. Kaelbling, "Guiding long-horizon task and motion planning with vision language models," in *ICRA*, 2025.

[27] Y. Feng, J. Han, Z. Yang, X. Yue, S. Levine, and J. Luo, "Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation," *arXiv preprint arXiv:2502.16707*, 2025.

[28] R. Jiao, M. Bortolon, F. Giuliari, A. Fasoli, S. Povoli, G. Mei, Y. Wang, and F. Poiesi, "Obstruction reasoning for robotic grasping," *arXiv preprint arXiv:2511.23186*, 2025.

[29] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *CVPR*, 2025.

[30] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *ECCV*, 2024.

[31] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," in *ICLR*, 2025.

[32] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.

[33] M. Grinvald, F. Tombari, R. Siegwart, and J. Nieto, "Tsdf++: A multi-object formulation for dynamic object tracking and reconstruction," in *ICRA*, 2021.

[34] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *ICRA*, 2021.

[35] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "curobo: Parallelized collision-free minimum-jerk robot motion generation," 2023.