# Are VLMs Effective Task Planners for Robotic Object Retrieval in Clutter?

**Anonymous**


## Abstract

Robots should effectively retrieve novel objects in clutter, where a target may not be directly visible or accessible. Such problems involve reasoning about the order of blocking objects to be removed before the target is picked. Engineered solutions in this space focus on identifying object relationships from depth to inform the retraction sequence. Vision-Language-Models (VLMs) have been argued as pretrained solutions that effectively reason about such spatial relationships in images. This paper first aims to evaluate VLMs against engineered solutions for selecting which object to retrieve next. For this purpose, a modular software infrastructure is developed enabling evaluation of alternatives for object selection. Simulated scenes with significant occlusions and clutter are defined as a dataset for sequential object retrieval. These scenes are solvable by the modular software given human-in-the-loop object selection, which also serves as an expert comparison point. The evaluation indicates that both VLMs and engineered solutions do not perform significantly better than random object selection. Yet, they have complementary properties. This motivates hybrid strategies for targeted object retrieval that combine the visual reasoning of VLMs with engineered dependencies via 3D reasoning. The hybrid approaches achieve improved performance. These observations are also confirmed by real world experiments using a robotic arm with a parallel gripper, a stereo camera, and the same software infrastructure.


## Introduction

**T**argeted **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 even visible, (ii) grasping, which specifies how the gripper should attach to an object to achieve closure, (iii) reconstruction that involves obtaining a 3D model for objects to safely perform retrieval, (iv) motion planning, i.e., computing the motions of a robotic arm for picking and retrieval, and (v) task planning, the focus of this work, which involves identifying the sequence of objects that need to be removed from a scene in order for the desired target to be retrieved. Such task planning challenges arise when there is significant clutter and occlusions in the retrieval scene, such as those shown in Figure 1, which comes from an experiment of this work.




**Figure 1:** (Top Left) Bird's eye view from an experiment. (Top Right, Bottom Left/Right) Sequence of segmented images from a robot's viewpoint before a pick is performed until the hidden black box (target) is retrieved. The target is labeled 0 when identified.

**Solution Requirements:** In applications, such as manufacturing and logistics, it is possible to engineer TORC solutions by assuming known models or homogeneous geometry or structuring the workspace to reduce clutter and provide visibility. This comes at the expense of space efficiency and expensive automation equipment. This work aims to address less structured setups, including robots at home, where there is yet a generalizable solution for TORC that handles unknown, heterogeneous objects under significant occlusions.

In particular, a TORC system must: (1) operate in a *model-free manner*, meaning no prior 3D geometric models of objects are available; (2) operate *without significant object clearance* and under *heavy occlusions*. In particular, the target object maybe completely occluded, necessitating interactive physical exploration (or "mechanical search") to discover and retrieve it; (3) balance safety and feasibility. For instance, pushing all objects randomly scatters them and increases the chance of detecting the target over time. This may be preferred by people, who have compliant hands and good knowledge of the objects' physical properties. It is best to avoid, however, with non-highly-compliant robots, as it leads to object damages, and disturbs the scene with unexpected outcomes, such as objects falling off. Thus, this work uses *picking and retrieval actions* for removing objects from the scene with a parallel gripper, which is the most popular type of robot end-effector. (4) solve TORC instances efficiently, where the key metric is *minimizing the number of such actions* until the target is grasped.

**Figure 2:** Real-world experiment of the robot picking an object with ego-centric perception from a torso-mounted stereo camera.

**Task Planning for TORC:** Under this metric, a critical component of TORCis the task planning that identifies which occluding object to remove next to discover the target as fast as possible. To address this, traditional manipulation solutions construct graphical representations from 3D/depth information and adjacency heuristics, referred to here as a Dependency Graph (DG) (alternatively a "Scene Graph") approach. The DG aims to express object relations that result from reachability or visibility constraints ('grasp blocking', 'behind','below'). These engineered approaches, however, can be fragile when object geometric complexity and clutter increase. Given the model-free requirement, an emerging promising solution, corresponds to Visual Language Models (VLMs). The hope is that VLMs can perform effective spatial reasoning directly on images given their internet-scale pretraining.

To evaluate the relative performance of depth-based DG approaches against the image-based VLMs, this work first builds a modular software architecture, which brings together all the components for addressing TORC (semantic segmentation for perception, learning-based grasping, object reconstruction, safe motion planning based on a modified TSDF-based occlusion volume, and task planning). In parallel, it defines a dataset of simulated scenes with significant occlusions and clutter, where multiple object removals need to be performed before the target object is retrieved. These scenes are confirmed to be solvable effectively by the modular software architecture by using a human-in-the-loop approach for selecting the next object to be picked. The human-in-the-loop approach acts as an "expert" solver for TORC task planning. The evaluation of DG-based and VLM-based solutions revealed that both performed on par or slightly better than a naïve alternative of randomly picking an object that is reachable according to the grasp planner.

These two approaches, however, operate over different sources of information (depth vs. RGB) and exhibited complementary properties in terms of object interactions they effectively identified. This motivated the authors to propose hybrid planners, which use VLMs together with DGs. 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, rather than relying entirely on one or the other. The key innovation lies in utilizing the VLM as a Structural Oracle to refine the raw output of fragile geometric calculations. The VLM is prompted to apply its semantic and spatial understanding to correct these noisy candidates, explicitly distinguishing a true 'behind' relationship from a 'below' relationship. This effectively infuses robust common sense into the planning structure without the need of specifically engineering a complex formulation for defining a somewhat semantically ambiguous object relationship.

In summary, this work provides a comparative analysis of different TORC task planners, quantifying their performance based on retrieval efficiency (number of picks) and safety (scene disturbance) across diverse cluttered scenarios in simulation and real experiments. In particular, it **contributes**:

- A complete, modular software to evaluate different task planners for TORC without needing object models, together with a dataset of cluttered simulated scenes with significant occlusions built on top of grasping datasets (Back et al. 2025), where object selection impacts the ability of solving TORC effectively. [1]
- An evaluation of the spatial reasoning capabilities of state-of-the-art robotic VLMs in the context of TORC given appropriate prompts and comparing against more traditional engineered approaches based on depth and geometric reasoning on top of segmentation output.
- Novel hybrid frameworks that leverage a VLM's spatial reasoning capabilities to semantically refine geometrically-derived object relationships. This process results in improved robustness and reliability of sequential action selection in cluttered retrieval tasks.

## Related Work

**Picking an Object under Occlusion:** Picking a target object in the presence of clutter is already challenging, as it involves occlusions and partial views. Earlier work proposed an algorithm which takes advantage of the visibility and reachability structure between objects to come up with such a plan (Dogar et al. 2014). 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 (Poon et al. 2019), however, this approach always tries to rearrange the scene to make all objects visible simultaneously thus using more actions than might be needed to pick just the target. Towards improving the action efficiency, another work used reinforcement learning to find and retrieve the target object from clutter (Novkovic et al. 2020), however, the types of objects used were rather primitive.

Another line of work (Wang et al. 2020) proposed a motion planning solution for object picking using object pose estimation and the Minimum Constraint Removal strategy to minimize collision risk. Critically, this work was limited to motion planning and did not address the necessity of planning a discrete pick sequence to access highly occluded or hidden objects. CaTGrasp (Wen et al. 2022) extends to category-level object picking in dense clutter. However, it focuses on task-oriented grasping and bin clearance instead of target object retrieval, which requires identifying and planning efficient

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

action sequence with minimal disturbance to the remaining objects in the scene.

**Dependency/Scene Graphs for Manipulation:** The very task of identifying object relationships in a scene from images has become a popular task in its own right and many works have attempted to use machine learning to train models to generate such scene graphs (Mota and Sridharan 2018; Li, Meger, and Dudek 2016; Neau et al. 2024). Naturally, some task planning works have used these scene graphs to plan objects pick for retrieval. For example, one work used generated hierarchal scene graph in order to plan with a Regression Planning Network (Zhu et al. 2021) while another used a POMDP formulation (Kumar, Essa, and Ha 2022). Indeed, the POMDP formulation has been used prior for object manipulation and target object discovery even without the aid of scene graphs driving the abstract states (Zhao and Chen 2021; Xiao et al. 2019).

While scene graphs are object-oriented, other works have constructed more robot-oriented graph structures such as traversability graphs which maintain pairs of object poses which the robot is known to be able to transfer objects between (Nam et al. 2021). A more recent work combined object relationships as well as robot reachability estimates to generate a dependency graph keeping track of which objects necessitate manipulation before others (Nakhimovich, Miao, and Bekris 2023). Clever use of this dependency structure enabled a resolution complete algorithm for object retrieval. All the task and motion planning works mentioned thus far assume that object models are known apriori and that poses of visible objects are either explicitly known or computed by object pose estimation (Wen et al. 2024; Mitash et al. 2020).

**Mechanical Search:** Towards the direction of object retrieval without object models, a line of work dubbing the problem as mechanical search was developed initially evaluating simple object selection policies and also included pushing actions in addition to object grasps (Danielczuk et al. 2019). In order to find the target object without explicit relationship reasoning one work learned probability distributions called x-rays (Huang et al. 2021) and another innovated in end-effector hardware to be able to efficiently pick and use suction with the same tool (Huang et al. 2022a). It was later extended to reason about stacked objects (Huang et al. 2022b) albeit with rigid assumption on object geometry.

**VLM Planning** With the rise of Visual Language Models (VLMs), many have attempted using them directly or with fine-tuning for determining spatial reasoning and relationships between objects in a scene (Ma et al. 2025; Zantout et al. 2025; Song et al. 2025) rather than learning scene graphs explicitly. Moreover, many have attempted to use VLMs for every part of the planning process of object retrieval (perception, grasping, and object selection) (Tziafas and Kasaei 2024) or in conjunction with classical task and motion planning to achieve longer horizon plans for more complex tasks (Lee et al. 2025; Yang et al. 2025). The most recent techniques propose using expert constructed dependency graphs or obstruction graphs for a large dataset of scenes to construct an expert policy that could be used to finetune a VLM (Feng et al. 2025; Jiao et al. 2025). 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.

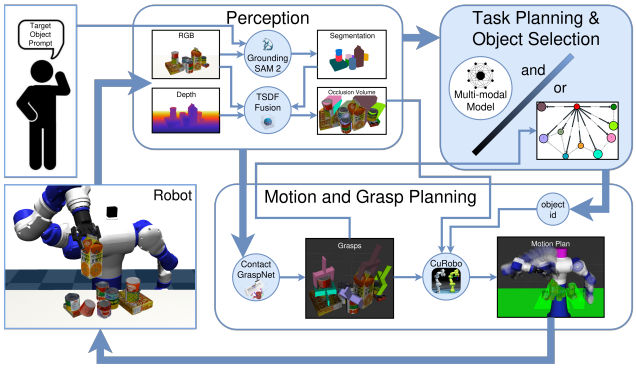


**Figure 3:** The modular robot system architecture used in this work to retrieve occluded or hidden objects in clutter. It integrates solutions for perception (top middle), task planning (top right - focus of this paper), as well as motion and grasp planning (bottom right).

## TORC Problem Setup

Consider an environment with objects for which 3D models *aren't* provided. The objects initially rest stably on a support surface, on top of other objects, or both. Objects are allowed to occlude each other given the camera's view, which is from an RGB-D sensor attached to the robot's torso for ego-centric perception as in Fig. 2. No multi-view observation is available to perform full scene reconstruction due to the static robot base.

## A Modular Pipeline for TORC

Fig. 3 visulizes the components brought together for addressing TORC without access to object models. The steps executed for a single object 'pick' action are the following:
(1) The scene is sensed with an RGB-D camera, which produces a segmented point cloud and a uniform-resolution scene voxelization of the current occlusion volume.
(2) The grasp planner samples and scores grasps per object on the segmented point cloud.
(3) A task planner selects the next object to be picked.
(4) Then, given a selected object and the highest scoring valid grasp, four motion planning queries are computed: (i) one approaching a pre-grasp pose, (ii) a cartesian motion from pre-grasp to the grasp, (iii) another cartesian motion to lift up the object away from the clutter, and (iv) a motion query to move to object to a drop location outside the workspace.

The first and last motion queries of step (4) avoid collisions between the robot and the workspace. Collision checks are performed against both the visible object point cloud and the occlusion volume of step (1). The cartesian motion queries do not check for collisions. Additionally, the last motion query attaches an estimated collision volume of the picked object (the convex hull of its point cloud segment) to avoid undesirable collisions. If any of these motion planning queries fail, the next best grasp for that object is attempted.

The above four processes are repeated until the target object is retrieved from the workspace. The occlusion volume used for collision checking is continuously updated given

subsequent observations while the surface point cloud used for grasp and motion planning remains fixed until the next object retrieval.

**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 (Wen et al. 2025) by feeding the stereo images captured by a ZED camera. Instance segmentation is then performed on the color image via Grounding SAM 2 (Liu et al. 2023; Ravi et al. 2024). 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. In simulation, the same process can be used but there is also access to privileged depth and object instance segmentation from MuJoCo (Todorov, Erez, and Tassa 2012). The privileged information is used in the accompanying simulated experiments to focus the evaluation on the performance of task planning.

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 (Zeng et al. 2017). 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 the literature (Grinvald et al. 2021).

**Grasping** Given the surface point cloud and object mask produced by the perception process, ContactGraspnet (Sundermeyer et al. 2021) is leveraged to produce candidate grasps for each object. These grasps are checked for IK feasibility and removed if not feasible.

In order to produce a list of grasp dependencies (used in some of the object selection approaches described in the Task Planning Solutions section), a weighted grasp collision graph is first computed with object IDs as nodes and the number of grasps in collision with another object as the directed weighted edges. This graph is computed by collision checking the robot arm, which is placed at an IK solution for each grasp, against the surface and occlusion volume corresponding to each other object. After these reachability dependencies are identified, the grasps that are currently feasible are validated as follows: The grasps, which result in IK arm configurations in collision with the tabletop, are discarded. Grasps with more than a single object falling between the gripper's fingers are also discarded. Grasps where the gripper's fingers collide with the surface points of the object being grasped are discarded. The fingers are allowed, however, to collide with the object's occlusion volume.

The grasping process ultimately returns both the object reachability dependencies as well as a list of validated grasps and their corresponding scores as evaluated by ContactGraspnet.

**Motion Planning** Planning the robot arm's motion from its rest state to the pre-grasp state, CuRobo (Sundaralingam et al. 2023) is used with the collision volume set by running marching cubes on the combined surface point cloud and occluded voxels. For moving between the pre-grasp pose to the grasp and from the grasp to post-grasp pose, Cartesian motion planning is performed using Pink[2]. No collision checking is performed during these Cartesian motions. Finally from the post-grasp pose to the drop pose, CuRobo is queried again but with the surface point cloud of the grasped object attached to the end-effector.

## Task Planning Solutions for TORC

The objective for TORC task planning is to determine a sequence of pick actions (i.e., picking an object and dropping it into a container outside the workspace) so as to discover and subsequently retrieve a target object. The target need not be directly visible or pickable from the robot's sensor view given the initial view. The corresponding solution should avoid collisions with sensed obstacles and objects as well as occluded regions, in order to minimize scene disturbance. While some scene disturbance is inevitable given the cluttered nature of the scenes, the metric for scene disturbance used in the accompanying evaluation is the number of objects that fall out of the support surface, i.e., a 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 objects to be pushed until they are dropped out of the tabletop.

A random baseline (Approach 0 below), a VLM approach adapted from a previous work (Tziafas and Kasaei 2024) (Approach 1 below), a DG approach adapted from a previous work (Nakhimovich, Miao, and Bekris 2023), 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

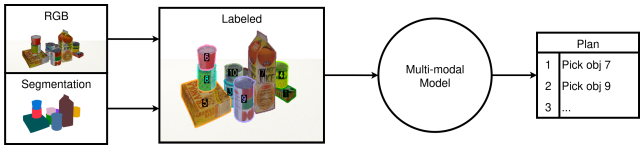


**Figure 4:** Overview of the VLM-SELECT approach.

For the VLM-SELECT selection method, a VLM is prompted with a labeled image of the workspace and asked

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

to come up with a sequence of actions to retrieve the target object, as shown in Fig. 4. The result is parsed and the first object in the sequence is chosen. VLM−SELECT does not have knowledge of which objects have valid grasps. Due to the stochastic output of the VLM, it is called again if it fails to select a graspable object.

This selection strategy is largely inspired from a previous work (Tziafas and Kasaei 2024), which uses VLMs for the entire retrieval process (object grounding, object selection, and grasp selection). In the previous work, however, the target object was always partially visible, whereas in many of the experiments in this work the target object is completely hidden initially. Thus, the object selection procedure from the previous work is replicated as close as possible here with slight changes to the VLM prompt to consider the case of complete occlusion of the target object.

The specific VLM prompts used by this work are available online[3]. An outline of the prompt for VLM−SELECT is the following:

- Input Specification: The model receives an annotated image showing objects highlighted with colors with numeric label IDs displayed in black squares near each object's center. The target object ID is then specified.
- Task Specification: The model is asked to create a sequential plan of actions to make the target object graspable. The VLM is asked to follow a set of rules to reason about blocking objects and occlusions.
- Output Request: A JSON array of actions.

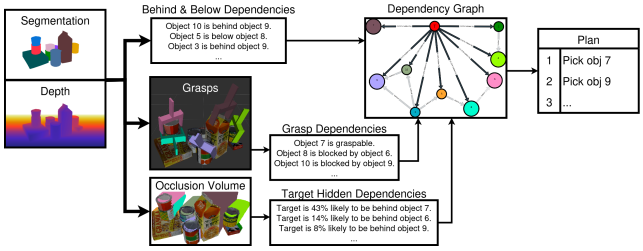


**Figure 5:** Overview of the DG−SELECT approach.

## Approach 2. DG−SELECT

An engineered selection method (DG−SELECT) is adapted from previous work (Nakhimovich, Miao, and Bekris 2023), which uses dependency graphs that express object relationships. In the previous work, these dependencies could be computed explicitly given ground truth knowledge of the object's models and poses. Here, these dependencies are computed from perceptual information without access to object models.

This method involves creating a directed graph $G(V, E)$, the Dependency Graph, where each node $v \in V$ represents a visible object and each directed edge $e = (v_1 \rightarrow v2) \in E$ represents a dependency in pick order between objects $v_1$ and $v_2$, i.e., the object corresponding to the source node $v_1$ cannot be picked unless the object corresponding to the target node $v_2$ is picked first.

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

Specifically, an edge $v_1 \rightarrow v_2$ in the Dependency Graph can arise due to the following relationships: object $v_1$ is 'behind' object $v_2$, object $v_1$ is 'below' object $v_2$, or object $v_2$ is 'grasp blocking' object $v_1$. These relationships are identified as follows.

1. Given a segmented image of object masks, each object mask $m_i$ is dilated by a few pixels and the intersection of each pair of dilated masks $(\text{DILATE}(m_i), \text{DILATE}(m_j))$ is used to find a set of adjacent object masks $M$.
2. For each mask pair $(m_a, m_b) \in M$ the average depth of the boundary of $m_b$, $\delta(m_b) = \text{DILATE}(m_a) \cap m_b$ is compared against the average depth of the boundary of $m_a$, $\delta(m_a) = \text{DILATE}(m_b) \cap m_a$. The mask with the larger depth value is defined to be 'behind' the other.
3. In reality some of these pairs do not accurately represent a 'behind' relationship but rather a 'below' relationship. Correcting 'behind' to 'below' relationships is performed by computing the surface normals of points in $\delta(m_a)$ and $\delta(m_b)$. For object $a$ to be marked 'below' $b$ all of the following need to hold for a sufficient number of boundary points ($> 50$) on $a$:
   - The points are in close proximity to $b$ in 3D space.
   - The points have normal vectors pointing within $45°$ of the positive z axis.
   - The points are below the closest point on object $b$.

Given the above steps, define the identified set of 'behind' object pairs as $H$ and 'below' object pairs as $O$. The 'grasp blocking' dependency pairs $B$ are computed based on the grasp dependency lists (generated by the grasping planner and IK solutions as outlined in the previous section). A 'grasp blocking' edge $v_1 \rightarrow v_2$ has a weight equal to $\frac{\text{number of grasps blocked by object } v_2}{\text{total number of (blocked) grasps for } v_1}$. Then, the dependency graph $G(V, E)$ is constructed as follows:

1. All segmented objects are added as nodes $v \in V$.
2. All 'below' edges in $O$ are added in the set $E$. The weight of each edge going out of node $v_1$ is $1/\text{DEG}(v_1)$ indicating no preference to remove a specific object above $v_1$.
3. For objects, which are not already 'below' another object, outgoing 'grasp blocking' edges in $B$ are added in the set $E$ as long as all of their grasps are blocked by other objects. The edge weights are set as described above.
4. For objects lacking outgoing edges up to this point, outgoing 'behind' edges in $H$ are added in $E$. The weight of each edge going out of node $v_1$ is $1/\text{DEG}(v_1)$ indicating no preference to remove a particular object that is in front of $v_1$.
5. Lastly, if the target object is not visible, a 'hidden target node' is added in $V$. Additional 'hidden by' edges are added from the 'hidden target node' to every visible object with normalized weight proportional to the volume of the occlusion region cast by that object. This prioritizes removing first objects with large occlusion shadow.

The method computes the sink node most likely to make progress in retrieval. Treating each normalized edge weight as a a 'psuedo probability', the probability of a sink node

making progress towards retrieving the target object is estimated as the sum of probabilities of each simple path to the target node, where the probability of each path is the product of the weight of each edge along the path. The next object to be picked is then determined by the sink node corresponding to the maximum likelihood of progress in the retrieval task. More formally, given $\mathcal{P}(t, v)$ to be the set of all simple paths from $t$ to sink $v$, the probability $p(v) \propto \sum_{P \in \mathcal{P}(t,v)} \prod_{e \in P} \frac{1}{w_e}$ where $w_e$ is the weight associated with edge $e \in E$. Given the set of sink nodes $S$ with their computed probabilities $p(v)$ for each $v \in S$, the next object to pick is selected by the sink corresponding towith maximum probability:

$$v^* = _{v \in S}\ p(v)$$

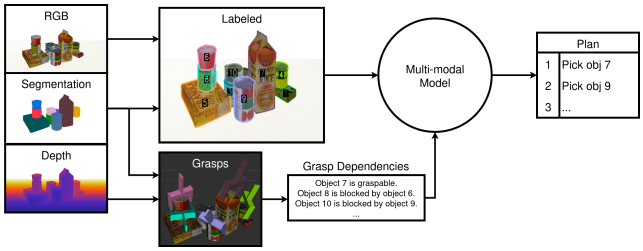


**Figure 6:** Overview of the VLM-FIXES-DG approach.

## Approach 3. VLM-FIXES-DG

The failure modes of the VLM-SELECT and DG-SELECT approaches motivate a hybrid approach using both a VLM and dependency planning (VLM-FIXES-DG). Fig. 6 shows the corresponding pipeline. In this approach a VLM is not prompted to directly return the object to be picked. Instead, it is prompted with candidate dependency relations between pairs of objects generated by the same heuristic reasoning employed by the DG-SELECT approach. The approach relies on the 'visual understanding' of the VLM to fix up incorrect dependencies and ultimately construct a more accurate dependency graph. Specifically, the fixed list of dependencies are used to construct the graph as enumerated at the end of DG-SELECT approach description and the next object selected for picking is chosen from that new dependency graph accordingly.

In the VLM-FIXES-DG approach the identification between 'below' and 'behind' relationships is performed by prompting a VLM.

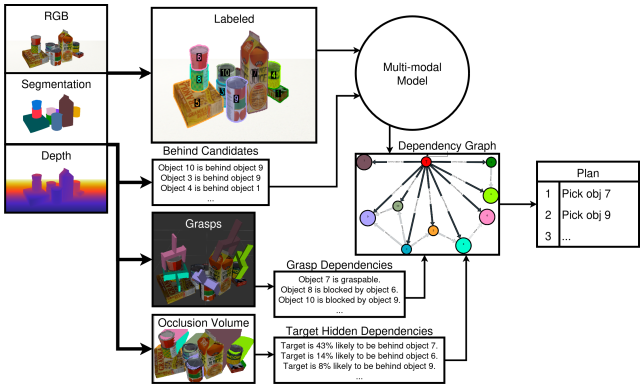


**Figure 7:** Overview of the VLM+GRASPS approach.


**Figure 8:** *Structured* and *unstructured* scenes in simulated experiments. **Left:** The robot's initial view. **Right:** Same scene visualized from the back. The target object is highlighted by a red segment and 0 label.

## Approach 4. VLM+GRASPS

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 information. If the VLM selects an object that is not directly pickable, then the approach is attempted again.

For the VLM+GRASPS method, the grasp dependencies are provided as a list of sentences of the form "Object A is blocked by object B" or of the form "Object C is graspable" if object $C$ has valid grasps available. As explained in the subsection on grasping above, valid grasps are those whose IK solutions were found not to be in collision with other objects or the static environment, in which the gripper was found to not be in collision with the object to be grasped, and in which the object to be grasped is the only one between the gripper fingers at the grasp pose.

A high level diagram of the approach is shown in Fig. 7.

# Experiments

The implementation of grasping and motion planning in the evaluation software is similar to that of the CGN-CuRobo pipeline evaluated on fetching problems (Han et al. 2024). While the grasp success rates for the baseline to fetch target objects are relatively low, it is classically representative of industry systems for object manipulation. Specifically, in simulation, the software pipeline uses ground truth object instance segmentation from MuJoCo, TSDF-fusion for scene

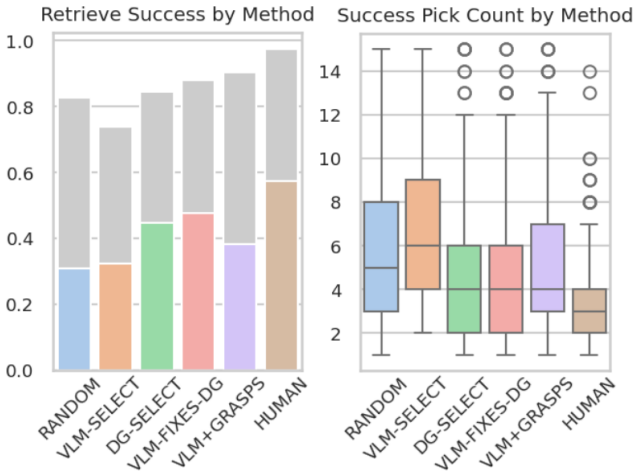


**Figure 9:** Simulation results. (Left) Success rate, where lower colored bars represent target object retrieval without any drops while the top gray bars represent object retrieval with drops of occluding objects. (Right) Distribution of number of picks required.

reconstruction and carving out occlusion volumes, ContactGraspNet for grasp planning, and CuRobo for motion planning.

## Simulated Dataset

Randomly generating retrieval scenes leads to frequent failures of grasping. These failure modes are not only due to the poor grasp predictions, but also often attributed to object-object interaction during retraction of an object from the workspace, which cannot be trivially resolved by leveraging privileged simulator information. This prevents to benchmark the object retrieval sequence planning, the key focus of this work. Therefore, in order to reduce the impact by the grasping process alone, the process of generating the scene dataset consisted of manual labeling and tuning, as well as filtering using hand-specified rules. The result of this process is a dataset of 40 `TORC` problems, each defined by a tabletop scene and a target object to be retrieved. See Figure 8 for visualization of a few simulated scenes.

As a starting point, scenes were taken from GraspClutter6d (Back et al. 2025). Each scene from GraspClutter6d consisted of a cluttered arrangement of objects as well as ground-truth grasps for each object. To produce our retrieval-focused scene dataset, we performed the following steps:

1. For each seen in the GraspClutter6d data set, an automated script placed these objects at the middle of a tabletop in a MuJoCo scene.
2. To approximately categorize scenes by difficulty, one of the authors manually labeled each scene as *structured* or *unstructured* based on whether the majority of objects are oriented in a qualitatively "natural" resting pose or not.
3. Another author then manually labeled the candidate target objects in each scene, with preference for partially and fully occluded objects.
4. Candidate targets that were irretrievable were then pruned as follows. For each target, all objects (1) directly in front of it, (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 both collision-free (with the static scene) and IK-feasible, the corresponding `TORC` problem was excluded from the dataset.
5. To ensure each problem is both non-trivial and solvable, scenes were iteratively refined based on feedback from a human expert. Specifically, the human stood in for the object selection component of the pipeline (see Fig. 3). If the expert selected the target immediately without removing any other objects, the `TORC` problem was 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 (e.g., by modifying object poses or adding/removing objects) until it admitted a solution with a task sequence of at least two steps.

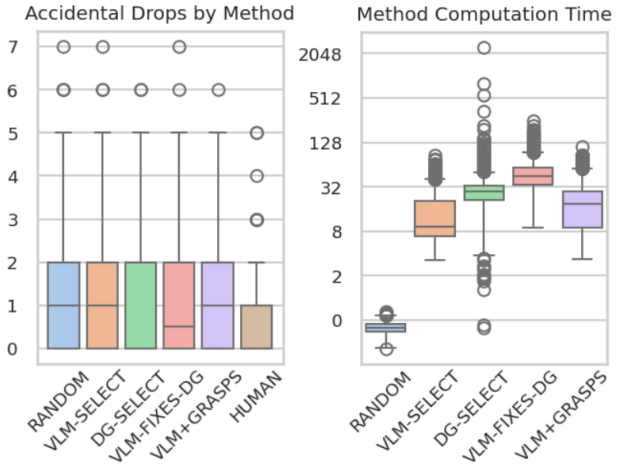


**Figure 10:** Simulation results. (Left) Distribution of number of drops. (Right) Distribution of computation times in seconds.

## Simulated Evaluation and Analysis

The `TORC` problems were first validated with human experts, who acted as task planners to ensure that scenes were solvable given the right object sequence. 5 different human experts performed object selection on each of 40 problems, for a total of 200 human trials. Once scenes were validated, each selection strategy (`RANDOM`, `VLM-SELECT`, `DG-SELECT`, `VLM+GRASPS`, `VLM-FIXES-DG`) was evaluated 10 times in each problem for a total of 400 experiments. The VLM used in these experiments was Google's Gemini Robotics-ER 1.5 (Team et al. 2025) since it is supposed to be pretrained for robotics tasks and is easily available for public use.

In Fig. 9 and 10 the success rate, computation times, number of object selections, and number of dropped objects are reported. The results indicate using VLM for object selection out of the box (despite being pretrained for robotics tasks) is only marginally safer than making random selections while taking more total picks than any other method. The engineered dependency graph approach, `DG-SELECT`, is safer than `RANDOM` and `VLM-SELECT` but involves more manual efforts and expert knowledge to design. Meanwhile, feeding the grasp information into the VLM as in the `VLM+GRASPS` approach was trivial to implement and matched the safety

of DG-SELECT. The number of pick actions, however, to solve TORC did increase slightly. The proposed approach VLM-FIXES-DG, which used the VLM to adjust the dependency graph was the safest approach and matched the pick action optimality of the DG-SELECT approach.

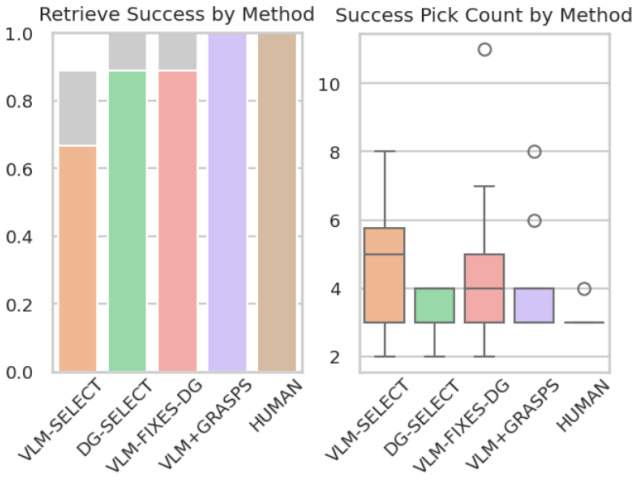


**Figure 11:** Real-world results. (Left) Success rate, where lower colored bars represent target object retrieval without any drops while the top gray bars represent object retrieval with drops of occluding objects. (Right) Distribution of number of picks required.

## Real-World Evaluation

The real-world experiments are performed with the Yaskawa Motoman sda10f, with a Robotiq 85 gripper attached on the right arm, as shown in Fig. 2.

For real experiments, comparison methods are evaluated on three difference scenes, where each method is repeated for three times to account for randomness. These scenes were constructed by placing household objects on a large grid on a tabletop and their positions are recorded for replication to be evaluated by different methods. The scenes were carefully designed such that it requires at least two picks to retrieve the target object. Figures 1 and 13 show the images taken from one human-in-the-loop trial of each scene. Results are reported in Figure 11 and 12.

The experiments performed on the real system validate the results seen in simulation. Furthermore, while the real system consumes more noisy perception results than the privileged information in simulation - such as incorrect detection of which object is the target - success was actually quite high on the real scenes overall. This can be attributed to the possibly smaller real-to-sim gap where the Grasp Planner and VLM chosen are more familiar with real point clouds or images respectively than those from simulation.

## Conclusion

This work evaluates the efficacy of VLMs as task planners for robotic object retrieval in cluttered environments, comparing against and integrating them with engineered geometric solutions. Given the results, it appears that the best role for VLMs in robotic manipulation tasks is not for high level decision making but rather for extracting relevant information

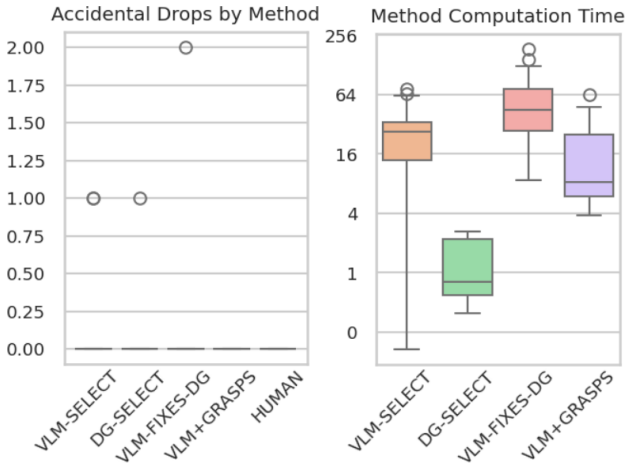


**Figure 12:** Real-world results. (Left) Distribution of number of drops. (Right) Distribution of computation times in seconds.

from the world via images and feeding that information into classical task planning processes.

While the hybrid strategies proposed in this study demonstrated improved performance over individual baselines a primary bottleneck is the performance of the grasp planner. Fortunately, the modular pipeline used in this work means that further improvements to grasp planning in the literature can be used to increase the overall success rate of the experiments presented here.

Despite the grasping limitation the overarching conclusion remains that while VLMs provide valuable spatial reasoning capabilities, they are often insufficient on their own for high-precision manipulation in clutter. Consequently, they benefit significantly from the integration of additional depth and geometric reasoning to improve retrieval accuracy. Moreover, VLMs are useful in the other direction to generate data structures that can be used within a classical task planning framework.

Beyond runtime performance gains, our framework can also be used to effectively generate successful manipulation sequences. These can serve as valuable training data for finetuning future models to intrinsically make the correct object selection choices for complex manipulation tasks. Future work will focus on leveraging this generated data for such finetuning efforts. Additionally, the aim is to move beyond the current open-loop pipeline by replacing the modular choices of sensing, planning, and execution with Vision-Language-Action (VLA) approaches to enable more reactive and robust execution in dynamic real-world settings.

## References

Back, S.; Lee, J.; Kim, K.; Rho, H.; Lee, G.; Kang, R.; Lee, S.; Noh, S.; Lee, Y.; Lee, T.; et al. 2025. GraspClutter6D: A Large-scale Real-world Dataset for Robust Perception and Grasping in Cluttered Scenes. *arXiv preprint arXiv:2504.06866*.

Danielczuk, M.; Kurenkov, A.; Balakrishna, A.; Matl, M.; Wang, D.; Martín-Martín, R.; Garg, A.; Savarese, S.; and Goldberg, K. 2019. Mechanical search: Multi-step retrieval

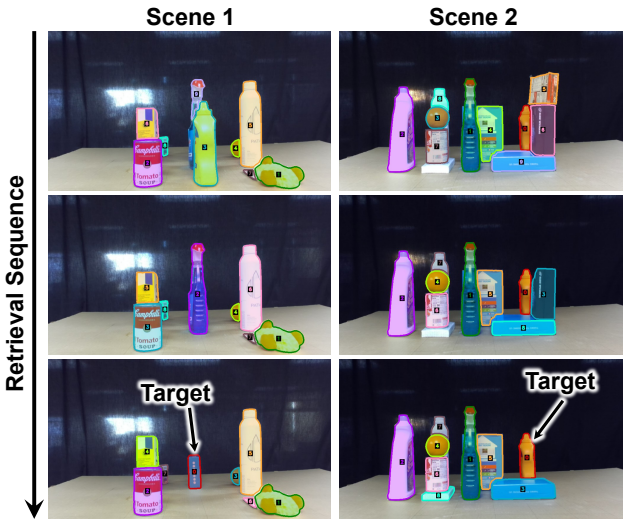


**Figure 13:** Example retrieval process on two different scenes in real-world human-in-the-loop experiments. Image observations are taken before each retrieval action and object instance segmentation is overlaid for visualization. The final identified target object is labeled as 0 with red silhouette.

of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, 1614–1621. IEEE.

Dogar, M. R.; Koval, M. C.; Tallavajhula, A.; and Srinivasa, S. S. 2014. Object search by manipulation. *Autonomous Robots*, 36(1): 153–167.

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

Grinvald, M.; Tombari, F.; Siegwart, R.; and Nieto, J. 2021. TSDF++: A multi-object formulation for dynamic object tracking and reconstruction. In *2021 IEEE international conference on robotics and automation (ICRA)*, 14192–14198. IEEE.

Han, B.; Parakh, M.; Geng, D.; Defay, J. A.; Luyang, G.; and Deng, J. 2024. Fetchbench: A simulation benchmark for robot fetching. *arXiv preprint arXiv:2406.11793*.

Huang, H.; Danielczuk, M.; Kim, C. M.; Fu, L.; Tam, Z.; Ichnowski, J.; Angelova, A.; Ichter, B.; and Goldberg, K. 2022a. Mechanical Search on Shelves using a Novel "Bluction" Tool. *IEEE International Conference on Robotics and Automation (ICRA)*.

Huang, H.; Dominguez-Kuhne, M.; Satish, V.; Danielczuk, M.; Sanders, K.; Ichnowski, J.; Lee, A.; Angelova, A.; Vanhoucke, V.; and Goldberg, K. 2021. Mechanical search on shelves using lateral access x-ray. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2045–2052. IEEE.

Huang, H.; Fu, L.; Danielczuk, M.; Kim, C. M.; Tam, Z.; Ichnowski, J.; Angelova, A.; Ichter, B.; and Goldberg, K. 2022b. Mechanical Search on Shelves with Efficient Stacking and Destacking of Objects. *arXiv preprint arXiv:2207.02347*.

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

Kumar, K. N.; Essa, I.; and Ha, S. 2022. Graph-based cluttered scene generation and interactive exploration using deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, 7521–7527. IEEE.

Lee, T.; Kang, G.; Wen, B.; Kim, Y.; Back, S.; Kweon, I. S.; Shim, D. H.; and Yoon, K.-J. 2025. DeLTa: Demonstration and Language-Guided Novel Transparent Object Manipulation. *arXiv preprint arXiv:2510.05662*.

Li, J.; Meger, D.; and Dudek, G. 2016. Learning to generalize 3d spatial relationships. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 5744–5749. IEEE.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. 2023. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Ma, W.; Chou, Y.-C.; Liu, Q.; Wang, X.; de Melo, C.; Xie, J.; and Yuille, A. 2025. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *arXiv preprint arXiv:2504.20024*.

Mitash, C.; Wen, B.; Bekris, K.; and Boularias, A. 2020. Scene-level pose estimation for multiple instances of densely packed objects. In *Conference on Robot Learning*, 1133–1145. PMLR.

Mota, T.; and Sridharan, M. 2018. 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*.

Nakhimovich, D.; Miao, Y.; and Bekris, K. E. 2023. Resolution Complete In-Place Object Retrieval given Known Object Models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3714–3720. IEEE.

Nam, C.; Cheong, S. H.; Lee, J.; Kim, D. H.; and Kim, C. 2021. Fast and resilient manipulation planning for object retrieval in cluttered and confined environments. *IEEE Transactions on Robotics*, 37(5): 1539–1552.

Neau, M.; Santos, P. E.; Bosser, A.-G.; and Buche, C. 2024. REACT: Real-time Efficiency and Accuracy Compromise for Tradeoffs in Scene Graph Generation. arXiv:2405.16116.

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

Poon, J.; Cui, Y.; Ooga, J.; Ogawa, A.; and Matsubara, T. 2019. Probabilistic active filtering for object search in clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, 7256–7261. IEEE.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; Mintun, E.; Pan, J.; Alwala, K. V.; Carion, N.; Wu, C.-Y.; Girshick, R.; Dollár, P.; and Feichtenhofer, C. 2024. SAM 2: Segment Anything in Images and Videos. arXiv:2408.00714.

Song, C. H.; Blukis, V.; Tremblay, J.; Tyree, S.; Su, Y.; and Birchfield, S. 2025. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15768–15780.

Sundaralingam, B.; Hari, S. K. S.; Fishman, A.; Garrett, C.; Wyk, K. V.; Blukis, V.; Millane, A.; Oleynikova, H.; Handa, A.; Ramos, F.; Ratliff, N.; and Fox, D. 2023. cuRobo: Parallelized Collision-Free Minimum-Jerk Robot Motion Generation. arXiv:2310.17274.

Sundermeyer, M.; Mousavian, A.; Triebel, R.; and Fox, D. 2021. Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. *2021 IEEE International Conference on Robotics and Automation (ICRA)*.

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

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Tziafas, G.; and Kasaei, H. 2024. Towards open-world grasping with large vision-language models. *arXiv preprint arXiv:2406.18722*.

Wang, R.; Mitash, C.; Lu, S.; Boehm, D.; and Bekris, K. E. 2020. Safe and effective picking paths in clutter given discrete distributions of object poses. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5715–5721. IEEE.

Wen, B.; Lian, W.; Bekris, K.; and Schaal, S. 2022. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, 6401–6408. IEEE.

Wen, B.; Trepte, M.; Aribido, J.; Kautz, J.; Gallo, O.; and Birchfield, S. 2025. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5249–5260.

Wen, B.; Yang, W.; Kautz, J.; and Birchfield, S. 2024. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17868–17879.

Xiao, Y.; Katt, S.; ten Pas, A.; Chen, S.; and Amato, C. 2019. Online planning for target object search in clutter under partial observability. In *2019 International Conference on Robotics and Automation (ICRA)*, 8241–8247. IEEE.

Yang, Z.; Garrett, C.; Fox, D.; Lozano-Pérez, T.; and Kaelbling, L. P. 2025. Guiding long-horizon task and motion planning with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 16847–16853. IEEE.

Zantout, N.; Zhang, H.; Kachana, P.; Qiu, J.; Chen, G.; Zhang, J.; and Wang, W. 2025. SORT3D: Spatial Object-centric Reasoning Toolbox for Zero-Shot 3D Grounding Using Large Language Models. *arXiv preprint arXiv:2504.18684*.

Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; and Funkhouser, T. 2017. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *CVPR*.

Zhao, W.; and Chen, W. 2021. Hierarchical POMDP planning for object manipulation in clutter. *Robotics and Autonomous Systems*, 139: 103736.

Zhu, Y.; Tremblay, J.; Birchfield, S.; and Zhu, Y. 2021. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 6541–6548. IEEE.