Viewpoint-Agnostic Grasp Pipeline using VLM and Partial Observations

Abstract— Robust grasping in cluttered, unstructured environments remains challenging for mobile legged manipulators due to occlusions that lead to partial observations, unreliable depth estimates, and the need for collision-free, execution-feasible approaches. We present an end-to-end pipeline for language-guided grasping that bridges open-vocabulary target selection to safe grasp execution on a real robot. Given a natural-language command, the system grounds the target in RGB using open-vocabulary detection and promptable instance segmentation, extracts an object-centric point cloud from RGB-D, and improves geometric reliability under occlusion via back-projected depth compensation and two-stage point cloud completion. We then generate and collision-filter 6-DoF grasp candidates and select an executable grasp using safety-oriented heuristics that account for reachability, approach feasibility, and clearance. We evaluate the method on a quadruped robot with an arm in two cluttered tabletop scenarios, using paired trials against a view-dependent baseline. The proposed approach achieves a 90% overall success rate (9/10) versus 30% (3/10) for the baseline, demonstrating substantially improved robustness to occlusions and partial observations in clutter.

I. INTRODUCTION

Robust object grasping in cluttered environments remains a fundamental challenge for autonomous robotic manipulation. In real-world deployments, such as inspection, remote intervention, and field operations, robots must interact with partially observed objects under severe occlusions, limited viewpoints, and incomplete depth measurements [1], [2], [3]. In these conditions, successful manipulation requires more than predicting a geometrically valid grasp, it requires that the selected grasp admits a collision-free approach, respects kinematic constraints, and remains stable during execution [4], [5], [6]. Grasps that appear feasible on visible surfaces often become unreliable once hidden geometry, approach trajectories, and physical interaction constraints are considered [7], [8]. Achieving reliable grasp acquisition under such partial observations is, therefore, critical for enabling robotic manipulation in unstructured environments.

Mobile manipulation platforms, including quadruped robots equipped with manipulators and sensors, provide the mobility, dexterity, and perception required to operate in such environments [9], [10], [11], [12]. Traditional grasping algorithms generate candidate grasps from sensory observations and select those that remain stable under occlusion and clutter. Most recent pipelines predict 6-DoF grasps directly from depth or point clouds [13], [14], [15]. For example, a method called GPD operates directly on point clouds, generating and ranking 6-DoF grasp candidates from partial observations [13], [16]. Contact-GraspNet improved grasp quality in clutter by treating the grasping problem as a classification of grasp contact points [14], while Any-

Fig. 1: A legged mobile manipulator performs language-guided object grasping in a cluttered, unstructured scene under partial observations. The proposed pipeline grounds a natural-language target in RGB using open-vocabulary detection and segmentation, estimates object-centric 3D geometry from RGB-D, including completion under occlusion, and selects and executes a reliable 6-DoF grasp on the real robot.

Grasp emphasized robustness and efficiency across spatial and temporal domains by using a dense, point-based grasp prediction formulation evaluated over multiple views and time steps [15]. Other works have explored 3D encodings for manipulation as a viable alternative, including feature splatting for efficient 3D representations [17] and Gaussian-splatting-based simulation pipelines for real2sim2real transfer [18]. While recent grasp perception methods improve robustness in clutter [14], [15], mobile manipulation still requires converting partial, instance-level observations into executable grasps with feasible and secure approach trajectories.

In locomanipulation applications, the central challenge is converting partial, instance-level observations into executable grasps on a mobile platform, as illustrated in Fig. 1. The selected 6-DoF grasp pose must be reliable and achievable while remaining stable during whole-body and base motion. At the same time, in open-world deployments, the target object is often specified semantically rather than pre-segmented. Vision-language models (VLMs) and language-conditioned

perception provide scalable interfaces for task-driven target selection in cluttered scenes [19], [20]. Promptable open-set detectors such as Grounding DINO [21], combined with segmentation models such as SAM 2 [22], enable text-conditioned detection and instance segmentation from RGB observations [23]. However, bridging semantic grounding to reliable 3D grasp execution under partial observation remains challenging. The robot must convert grounded masks into object-centric geometry, infer missing surfaces, and generate grasps that remain feasible under real motion and safety constraints. Existing works typically address perception, grasp prediction, or execution individually rather than as a unified execution-aware pipeline.

In this paper, we address end-to-end robust grasping of objects in clutter for mobile legged manipulation, from natural language-driven target selection to safe execution on a real robot under partial observations. The key objective is reliability in unstructured scenes by developing a framework capable of identifying the intended object in clutter, estimating graspable geometry from incomplete views, and executing a collision-free, stable grasp while enforcing safety constraints throughout. To achieve this, we integrate open-vocabulary, language-guided target selection with object-centric 3D estimation and spatial-aware grasping in an end-to-end fashion. Target selection is grounded in RGB observations via VLM-guided querying and open-vocabulary detection and segmentation, enabling scalable specification of novel objects without task-specific retraining. Geometry is then derived from RGB-D, extracting and aggregating object-centric point clouds from masked observations, with back-projected depth compensation to improve robustness to missing returns. We explicitly address occlusion by using point cloud completion models, resulting in a grasping-ready representation even when significant portions of the object are unobserved. Finally, grasps are selected for safe execution by incorporating approach feasibility, clearance, and collision risk, and are executed through motion planning on the real robot. Overall, our approach enables safe, repeatable grasp acquisition in clutter at runtime from partial observations, providing a scalable pipeline from target selection to execution in real-world environments.

The main contributions of this paper are summarized as follows:

- **Unified End-to-End Framework:** An integrated pipeline that bridges natural language-driven target specification and execution-feasible grasping for mobile legged robots operating in cluttered environments.
- **Execution-Aware Grasp Selection:** A grasping strategy that incorporates approach feasibility, clearance, collision constraints, and whole-body kinematic limits to ensure reliable real-world execution.
- **Occlusion-Resilient Geometry Estimation:** An object-centric 3D reconstruction process from masked RGB-D observations with depth back-projection and MGPC-based shape completion to handle severe partial observations.
- **Mobile Locomanipulation:** Coordinated base reposi-

tioning and arm execution driven by execution-feasible grasp selection to improve accessibility and reliability in clutter.
- **Real-World Validation:** Experimental validation demonstrating safe and repeatable grasp acquisition in cluttered environments on a quadruped mobile manipulation platform.

The rest of this paper is organized as follows: Section II describes the proposed view-agnostic grasping pipeline, and Section III presents the experiments conducted to validate the framework. Section IV presents the results of this study, and finally, Section V concludes the paper.

II. METHODS

The proposed pipeline for viewpoint-agnostic grasping is based on three main modules: A) object detection and segmentation, B) point cloud generation and estimation, C) grasp pose generation, selection, and D) execution. The framework, shown in Fig. 2, takes RGB-D images from the robot's cameras as input and is able to perform grasping and locomanipulation in cluttered environments. The entire framework is implemented using Robot Operating System (ROS) 2 [27]. The framework's code is available in our GitHub[1] repository.

A. Detection and Segmentation

The perception stage takes RGB images from the robot's front cameras as input and outputs an object-centric instance mask that is later used to extract 3D geometry from the volumetric map (Sec. II-B). While the robot provides stereo RGB-D data at 15 Hz, detection and segmentation are performed using RGB only. Depth is used to generate object geometry (Sec. II-B) via nvblox-based depth integration and point cloud extraction [24].

a) Target specification and open-vocabulary detection, segmentation, and tracking: The operator specifies the target object via a natural-language command (e.g., "blue bottle") that should be within the robot's view range. Given the text query, the target is initially localized using the open-vocabulary detector GroundingDINO [21], which returns candidate bounding boxes with confidence scores. We select the highest-scoring box associated with the query, denoted $B^\star$, and use it to initialize SAM 2 for instance segmentation and *video tracking* [22]. During execution, SAM 2 maintains the target mask across subsequent frames. GroundingDINO is invoked again only if tracking fails (i.e., SAM 2 outputs no valid mask), at which point detection is re-initialized and tracking resumes. If no valid hypothesis is produced after re-initialization, the system does not proceed to grasp planning and continues acquiring observations.

b) Promptable instance segmentation and tracking: To obtain a pixel-accurate object mask in clutter, we refine $B^\star$ using SAM 2 [22] as the segmentation model. The selected box $B^\star$ is passed as a box prompt to SAM 2, producing a binary instance mask M for the target. We apply
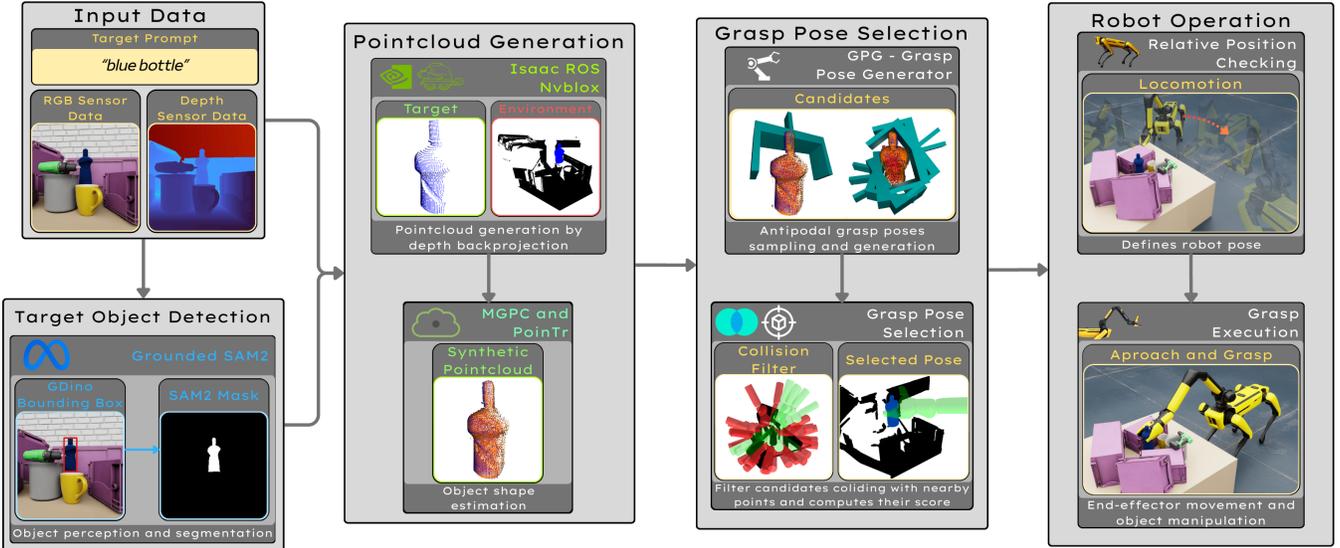

Fig. 2: System overview of the proposed viewpoint-agnostic grasping pipeline. The system receives a natural-language target prompt (e.g., "blue bottle") together with synchronized RGB and depth observations. The prompt is grounded in RGB using Grounding DINO [21] to obtain a target bounding box and SAM 2 [22] to produce an instance mask. The mask is then used to extract an object-centric partial point cloud from depth via Isaac ROS Nvblox [24] using depth backprojection. To mitigate occlusions and sparse depth, the object geometry is completed in two stages: MGPC [25] generates synthetic points conditioned on the prompt, RGB, and the partial point cloud, and PoinTr [26] further densifies the geometry by completing fixed-size local patches. Given the densified point cloud, Grasp Pose Generator (GPG) [16] samples antipodal 6-DoF grasp candidates, which are collision-filtered against nearby scene points and ranked to select an execution-feasible grasp. Finally, the robot executes a state-machine locomanipulation routine that (when needed) repositions the base for reachability and clearance, followed by end-effector approach, grasp closure, and object lift.

a lightweight morphological erosion to M using OpenCV to suppress boundary leakage into nearby clutter and improve mask robustness for 3D extraction. The output of this module is the mask M, which is then used to extract an object-centric point cloud from RGB-D observations (Sec. II-B).

B. Point Cloud Generation and Estimation

This stage converts the RGB-only instance mask into object-centric 3D geometry suitable for grasp synthesis under partial observations. We use the robot's RGB-D cameras to obtain depth images and leverage Isaac ROS nvblox for efficient GPU-accelerated depth processing and *point cloud extraction* [24]. In contrast to volumetric TSDF/ESDF mapping, our pipeline operates directly on point clouds and uses back-projected depth compensation to reduce sparsity and missing returns.

a) Object-centric point cloud extraction: For each RGB-D frame, nvblox is employed to back-project the depth image and produce a registered scene point cloud in the robot frame. We then apply the instance mask M (Sec. II-A) to retain only points that project inside the segmented target region, resulting in an object-centric partial point cloud P_{partial}. To improve robustness under partial views, masked point clouds are aggregated over time in a common reference frame using the robot state estimation. This multi-frame accumulation increases surface coverage while remaining lightweight and scalable for real-time operation.

b) Back-projected depth compensation: Depth in clutter often contains holes, flying pixels near discontinuities, and missing returns on thin or specular structures. To mitigate this, we apply a back-projected depth-compensation step in the nvblox point cloud generation pipeline before extracting P_{partial}. Small depth holes are filled, and outliers are attenuated using local neighborhood consistency in the image plane, producing a denser and more stable object-centric point cloud without requiring a volumetric distance map.

c) Completion from partial observations (MGPC): Even after multi-frame accumulation and depth compensation, P_{partial} remains incomplete due to self-occlusions and backside surfaces. We apply MGPC [25] to estimate missing geometry by leveraging multimodal context (prompt, RGB, and the partial point cloud). MGPC requires a fixed-size point input. Therefore, we subsample P_{partial} to 2048 points and infer a synthetic point set P_{mgpc} with 8192 points, the default model output. We then merge synthetic and observed points to obtain an intermediate cloud, as given by Eq. 1.

$$P_{\text{mid}} = P_{\text{partial}} \cup P_{\text{mgpc}}, \tag{1}$$

This increases surface coverage while remaining consistent with the visible geometry. In practice, for an input of N observed points, this step results in $N + 8192$ points prior to PoinTr [26] refinement.

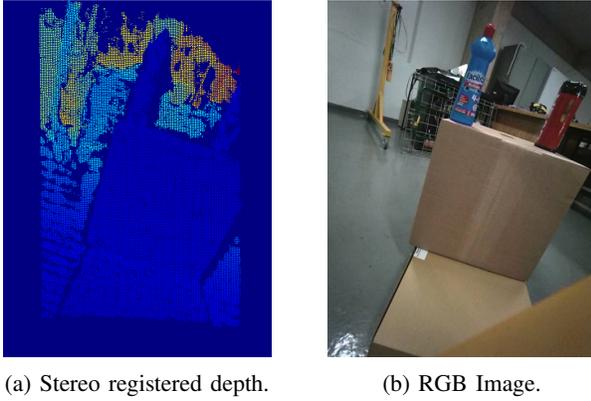(a) Stereo registered depth.   (b) RGB Image.

Fig. 3: Spot front-left registered stereo and RGB images example taken with the robot still. The images showcase the noise and limited resolution of the available sensors.

d) Point cloud refinement and densification (PoinTr): Since the Grasp Pose Generator (GPG) is sensitive to normal estimation quality, we further densify the object geometry using PoinTr [26], a point-cloud-only completion model. Since PoinTr requires a fixed input size, we decompose P_{mid} into overlapping local patches of 2048 points using a KD-tree neighborhood query. Each patch is completed independently by PoinTr, generating additional points that refine the local surface structure. The union of completed patches is then merged with P_{mid} to obtain the final densified cloud P_{complete}, which typically increases the point count substantially (e.g., from ∼2k to ∼10k after MGPC, and further after PoinTr refinement). The final point cloud P_{complete} is passed to the grasp module for 6-DoF grasp proposal generation, selection, and safe execution.

C. Grasp Pose Generation and Selection

Given the completed object point cloud P_{complete} (Sec. II-B), we sample 1000 candidate 6-DoF grasps and select a single grasp $g^\star$ for execution under clutter and reachability constraints. Since GPG relies on surface normal estimates, densifying P_{complete} improves normal stability and increases the diversity of feasible grasp hypotheses. We use the Grasp Pose Generator (GPG) to sample a candidate set $\mathcal{G} = \{g_i\}$ on P_{complete}, where each candidate $g_i = (\mathbf{p}_i, \mathbf{R}_i)$ defines a gripper pose with position $\mathbf{p}_i$ and orientation $\mathbf{R}_i$ in the robot frame. We configure GPG's antipodal sampling parameters (jaw width and contact constraints) to match the Spot jaw gripper geometry, ensuring that sampled grasps are physically realizable on the hardware.

a) Collision filtering: To enforce feasibility in cluttered environments, each candidate is validated by collision checking against the local scene geometry. We evaluate gripper-environment intersections using a parallelized kernel that tests the gripper mesh (at pose g_i) against the environment point cloud in a neighborhood around the target. Candidates that result in collisions with the scene are rejected, resulting in a filtered set $\mathcal{G}_{\text{free}} \subseteq \mathcal{G}$.

b) Heuristic ranking: From $\mathcal{G}_{\text{free}}$, we select the grasp that best trades off approach feasibility and grasp stability using a weighted cost function, given by Eq. 2.

$$\mathcal{C}(g_i) = w_\theta \, |\Delta\theta_i| \, + \, w_\phi \, \phi_i \, + \, w_c \, \|\mathbf{p}_i - \mathbf{c}\| \, + \, \mathcal{P}(r_i), \quad (2)$$

where $\mathbf{c}$ is the centroid of P_{complete} and $r_i = \|\mathbf{p}_i - \mathbf{p}_{\text{base}}\|$ is the distance from the robot base to the candidate grasp position.

- **Alignment ($|\Delta\theta_i|$):** angular deviation (rad) between the base-to-target direction and the grasp approach direction. This biases grasps toward the robot's nominal approach and typically requires less base repositioning.
- **Approach bias (ϕ_i):** binary penalty that discourages unfavorable approach directions (e.g., approaching from below), which are more likely to be kinematically constrained or blocked in clutter.
- **Centrality ($\|\mathbf{p}_i - \mathbf{c}\|$):** distance to the object centroid, which favors more centered grasps that are less sensitive to partial geometry and contact uncertainty.
- **Reach constraint ($\mathcal{P}(r_i)$):** hard penalty enforcing a maximum reach radius $r_{\max}$, as given by Eq. 3.

$$\mathcal{P}(r_i) = \begin{cases} 0, & r_i \leq r_{\max}, \\ M, & r_i > r_{\max}, \end{cases} \quad (3)$$

where M is a large constant.

The final grasp is chosen as given by Eq. 4.

$$g^\star = \arg \min_{g_i \in \mathcal{G}_{\text{free}}} \mathcal{C}(g_i). \quad (4)$$

In hazardous environments, the weights (w_θ, w_ϕ, w_c) and limit $(r_{\max})$ provide explicit parameters to tune the grasp selection toward safer, more reliable grasps in cluttered scenes.

D. Grasp Execution and Motion Control

Execution of the selected grasp $g^\star$ is managed by a finite-state machine that coordinates base repositioning and arm motion to ensure reachability. If the grasp is not reachable from the current stance, the robot commands a base waypoint along the grasp approach direction at a stand-off distance that improves manipulability while keeping the target within the arm workspace. This repositioning step is used to satisfy reachability and clearance constraints prior to arm motion.

After base alignment, the manipulation sequence proceeds in two stages:

a) Pre-grasp approach: We define a pre-grasp pose g_{pre} by offsetting $g^\star$ along the gripper approach axis by a safety distance δ:

$$g_{\text{pre}} = g^\star \oplus (\delta \, \hat{\mathbf{x}}), \quad (5)$$

where $\hat{\mathbf{x}}$ is the local approach axis of the gripper frame and $\oplus$ denotes pose composition.

b) Final insertion and closure: From g_{pre}, the end-effector executes a short Cartesian insertion of length ℓ (in our implementation, $\ell = 5\,\text{cm}$) along the approach axis to reach $g^\star$. Once the final pose is reached, the robot gripper service commands the gripper to close to secure the object.
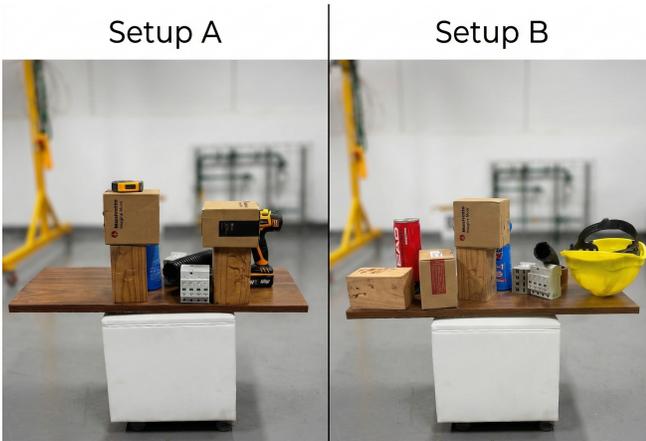
Fig. 4: Experimental setups for evaluating the viewpoint-agnostic grasp pipeline. The environments consist of cluttered industrial and household objects. Setup A (left) was used for experiments to identify and grasp a power drill partially obscured by boxes and electrical components. Setup B (right) requires the pipeline to target a blue bottle situated behind different boxes. These configurations test the model's ability to generate grasps despite the challenging scenarios.

III. EXPERIMENTS

We evaluate the proposed viewpoint-agnostic grasping pipeline on a Boston Dynamics Spot equipped with an arm, in cluttered scenes representative of unstructured deployments (See Fig. 1). The goal of the experiments is to quantify the benefits of our geometry estimation and grasping generation pipeline under partial observations, in particular, point cloud completion and spatial-aware grasp selection, compared with a view-dependent baseline that fixes the robot position upon detection.

A. Hardware and Software

All experiments were conducted on a Boston Dynamics Spot quadruped equipped with the Spot Arm (6-DoF) and a jaw gripper, using the robot's onboard RGB-D sensing. Computation was split between the robot and an external workstation connected via Ethernet. The workstation was used to run the perception models and geometry/grasp optimization, while Spot executed locomotion and manipulation through its onboard controllers.

a) Sensing: Spot provides five RGB-D sensors. In the experiments, the front-left and front-right cameras were used. RGB images were streamed at $15\,\text{Hz}$ with VGA resolution (640×480), and stereo depth images were streamed at 424×240 pixels. Examples of those images are shown in Fig. 3.

b) Compute: The external workstation is equipped with two NVIDIA RTX A2000 GPUs (12 GB each), 125 GB system memory, and an Intel® Xeon® w3-2435 CPU ($\times 16$ cores). All learning-based components (GroundingDINO, SAM 2, and MGPC) and the associated point-cloud processing were processed on this workstation.

c) Software and communication: Robot communication and command execution are handled through the Boston Dynamics Spot SDK. The overall system is implemented in ROS 2 and deployed using Docker containers (ROS 2 Humble and Jazzy), with the workstation acting as the primary compute node and Spot as the execution platform. The pipeline runs end-to-end in real time, synchronized with the RGB-D camera stream and the state-machine execution described in Sec. II-D.

d) Implementation details: For reproducibility, the full set of experimental parameters (perception thresholds, point-cloud completion settings, and grasp-selection weights) is documented in the code repository.

B. Experimental Setups

Two cluttered tabletop setups, shown in Fig. 4 were used, differing mainly in the grasp target:

- **Setup 1 (Drill):** the target object is a handheld drill placed among different objects and occluders.
- **Setup 2 (Blue bottle):** the target object is a blue bottle placed among different objects and occluders.

In both setups, the table height was comparable to the robot base height, allowing consistent visibility from the front cameras. The table surface contained a variety of objects and obstacles intentionally arranged to occlude the target and induce partial observations. The robot always faced the table at the start of each trial.

C. Methods Compared

For each setup, we compare:

- **Ours (viewpoint-agnostic):** full pipeline including mask-conditioned point cloud extraction, multi-frame accumulation, MGPC and PoinTr point cloud completion, and mobile grasp selection/execution (see Section II).
- **Baseline (view-dependent):** A conventional grasping pipeline that uses the same perception front-end and GPG-based grasp candidate generation, followed by identical collision filtering and heuristic ranking. However, grasp planning is performed directly on the single-view partial point cloud without multi-frame accumulation or point cloud completion, and the robot executes grasps from its initial stance without base repositioning.

This baseline isolates the impact of the proposed viewpoint-agnostic geometry estimation and mobile grasp execution strategy. In contrast to our method, the robot must commit to a grasp using only the partial object geometry visible from the initial viewpoint, which reflects a common deployment setting in many existing grasp pipelines.

D. Trial Protocol

Each setup consists of 10 trials. For each setup, 5 trials are performed with our method and 5 with the baseline. To ensure a fair comparison, we use a paired protocol: for a given initial robot pose, we execute one trial with our method and one trial with the baseline, starting from the same position. Across pairs, the initial robot position is

Scenario	Run	Object	Success or Failure (alongside failure reason)	
			Our Method	Baseline (View-Dependent)
A	1	drill	✓	✗ (reachability failure)
	2	drill	✓	✗ (approach collision (clutter))
	3	drill	✓	✗ (approach collision (clutter))
	4	drill	✗ (reachability failure)	✗ (reachability failure)
	5	drill	✓	✗ (approach collision (clutter))
B	1	blue bottle	✓	✗ (approach collision (target))
	2	blue bottle	✓	✓
	3	blue bottle	✓	✗ (approach collision (clutter))
	4	blue bottle	✓	✓
	5	blue bottle	✓	✓
Total success rate			**(9/10)**	**(3/10)**

TABLE I: Success/failure outcomes from the experiments performed across scenarios A and B. The failure reason is presented whenever the robot fails to either reach or securely grasp the target object.

changed (while maintaining the same scene configuration and the robot facing the setup) to test robustness to viewpoint-dependent occlusion and depth sparsity. After each grasp attempt, the robot returns to its initial pose before starting the next trial.

Each trial follows the same sequence:

1) **Perception:** the operator provides a natural-language target query ("drill" or "blue bottle"). The system detects and segments the target and extracts an object-centric point cloud.
2) **Geometry estimation:** our method applies depth compensation and point cloud completion using MGPC and PoinTr; the baseline uses only the partial point cloud from the initial view.
3) **Grasp selection:** grasp candidates are generated and ranked, producing a selected grasp pose $g^\star$.
4) **Execution:** the robot repositions its base if needed to satisfy reachability, executes a pre-grasp approach, performs a final Cartesian insertion, and closes the gripper.

E. Success Criteria

A trial is labeled as **successful** if the robot (i) closes the gripper on the target object, (ii) lifts it from the table, and (iii) maintains a stable grasp for a short verification period without dropping or slipping. We report success rates per setup and method, and analyze failure modes qualitatively in Sec. IV.

IV. RESULTS AND DISCUSSION

Table I summarizes the grasping outcomes across the two experimental scenarios, while Fig. 5 illustrates the experiments performed in this work. Our method achieved a total success rate of **9/10** trials, compared to **3/10** for the view-dependent baseline. In Scenario A (drill), our pipeline succeeded in **4/5** trials, whereas the baseline failed in all **5/5** trials. In Scenario B (blue bottle), our method succeeded in **5/5** trials, while the baseline succeeded in **3/5** trials. Overall, incorporating partial-observation geometry estimation and completion substantially improved robustness to clutter and occlusions in both setups.

A. Failure mode analysis

From Table I, the observed failures modes (FM) that occurred in the experiments were divided into three categories:

- **FM-1: Reachability failure**: the robot did not achieve a feasible pre-grasp configuration or could not reach the desired pose within the arm workspace.
- **FM-2: Approach collision (target)**: collision occurred while approaching the target, typically due to limited clearance along the approach direction.
- **FM-3: Approach collision (clutter)**: collision occurred with surrounding objects/occluders during approach.

The baseline fails predominantly due to approach collisions (FM-2/FM-3) in both scenarios, indicating that relying on the initial, view-dependent partial point cloud produces grasp candidates that are locally plausible but not executable once approach clearance is considered. In contrast, our method mitigates collision-driven failures, with only a single failure across all trials (Scenario A, run 4) categorized as reachability failure (FM-1). This supports the hypothesis that improving object-centric geometry under partial observations (depth compensation + completion) increases the number of grasps that remain feasible under collision and reachability constraints.

B. Limitations

While the proposed pipeline improves grasp robustness in clutter, the current evaluation highlights practical limitations:

a) Target visibility and VLM grounding: The target object must be sufficiently within the robot's field of view for the VLM-driven query, open-vocabulary detection, and segmentation to reliably identify the intended instance. If the prompt is ambiguous or the object is heavily occluded/out of view, grounding may fail or select the wrong instance, preventing the pipeline from proceeding.

b) Limited by VLM semantics: Object specification is constrained by what the VLM and the open-vocabulary detector can reliably interpret from the scene. For domain-specific objects, unusual tools, or safety-critical part-level intent, fine-tuning (or task-specific prompt engineering and verification) may be required to maintain reliability.
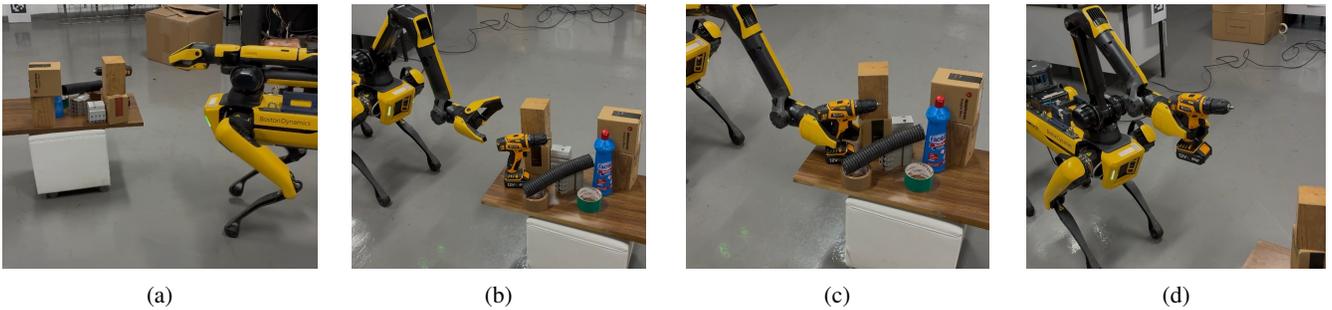
(a)　　　　　　(b)　　　　　　(c)　　　　　　(d)

Fig. 5: Sequence demonstrating the grasp execution experiments using the proposed end-to-end pipeline on the real robot. (a) After language-guided target selection and instance segmentation, the system estimates object-centric 3D geometry from partial RGB-D observations (including completion), selects an execution-feasible 6-DoF grasp under collision and reachability constraints, and repositions the base to satisfy reachability and clearance for the planned approach. (b) The robot aligns to the target and commands the arm to a collision-free pre-grasp pose with a safety offset. (c) The end-effector executes a short Cartesian insertion along the grasp approach direction and closes the gripper to secure the object. (d) The object is lifted to confirm grasp success and stability under post-grasp interaction.

c) Depth quality remains a challenge: Even with depth compensation and point cloud completion, severe depth noise and limited sensor resolution can still degrade geometry estimation and collision checking. In our hardware setup, the stereo depth stream has limited resolution and exhibits noise, which can lead to missing surfaces and outliers in clutter. In such cases, completion may not sufficiently recover the correct shape for safe grasp selection, especially for thin or reflective objects.

C. Key Observations

The results highlight three important observations about the proposed pipeline. First, incorporating object-centric geometry estimation with partial-observation completion increases the reliability of grasp candidate generation under occlusions, enabling successful grasps even when only limited surface information is initially visible. Second, execution-aware grasp selection combined with collision-aware filtering produces grasps that remain feasible during real robot execution, reducing approach collisions that commonly occur when grasp planning relies solely on view-dependent geometry. Third, integrating mobile base repositioning with grasp planning improves accessibility in cluttered environments by allowing the robot to satisfy reachability and approach constraints before arm execution. Together, these components enable the robot to identify and execute grasps that remain feasible under real-world constraints, resulting in a substantial improvement in the success rate across both experimental scenarios.

V. CONCLUSION

This paper presented an end-to-end pipeline for language-guided, viewpoint-agnostic grasping in clutter with a legged mobile manipulator. The proposed system grounds a natural-language target in RGB using open-vocabulary detection and promptable segmentation, extracts object-centric geometry from RGB-D, and mitigates partial observations through depth compensation and point cloud completion. Based on the resulting geometry, the system generates and ranks 6-DoF grasp candidates using execution-aware, safety-oriented heuristics and executes the selected grasp via motion planning and a state-machine controller on a real Spot platform.

Experiments on two cluttered tabletop scenarios demonstrated that incorporating partial-observation geometry estimation and completion substantially improves grasp reliability compared to a view-dependent baseline, particularly in the presence of occlusions and limited clearance. These results support the main hypothesis that robust grasping in unstructured environments benefits from explicitly bridging semantic target grounding to object-centric 3D estimation and to execution-feasible grasp selection. Future work will extend the evaluation to a broader range of objects and cluttered environments and will investigate a tighter integration between geometry estimation and planning under uncertainty, including online verification of target grounding, improved depth robustness, and reduced reliance on external compute for fully onboard deployment.

REFERENCES

[1] S. Karlsson, B. Lindqvist, and G. Nikolakopulos, "Ensuring robot-human safety for the bd spot using active visual tracking and nmpc with velocity obstacles," *IEEE Access*, vol. 10, 2022.

[2] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, "Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study," *Journal of Building Engineering*, vol. 65, p. 105814, 2023.

[3] R. Galin and R. Meshcheryakov, "Automation and robotics in the context of industry 4.0: the shift to collaborative robots," in *IOP Conference Series: Materials Science and Engineering*, 2019.

[4] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[5] R. Zurbrügg, Y. Liu, F. Engelmann, S. Kumar, M. Hutter, V. Patil, and F. Yu, "Icgnet: A unified approach for instance-centric grasping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 4140–4146.

[6] Z. Liu, Z. Wang, S. Huang, J. Zhou, and J. Lu, "Ge-grasp: Efficient target-oriented grasping in dense clutter," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[7] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, 2013.

[8] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *arXiv preprint arXiv:2104.01542*, 2021.

[9] Z. He, K. Lei, Y. Ze, K. Sreenath, Z. Li, and H. Xu, "Learning visual quadrupedal loco-manipulation from demonstrations," in *2024 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2024, pp. 9102–9109.

[10] C. Wang, O. K. Adak, and R. Fuentes, "Whole-body control loco-manipulation strategy for quadruped robots on deformable terrains," in *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2024, pp. 886–891.

[11] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar, "Roloma: Robust loco-manipulation for quadruped robots with arms," *Autonomous Robots*, vol. 47, no. 8, pp. 1463–1481, 2023.

[12] A. Rigo, M. Hu, S. K. Gupta, and Q. Nguyen, "Hierarchical optimization-based control for whole-body loco-manipulation of heavy objects," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 322–15 328.

[13] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[14] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[15] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.

[16] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[17] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, "Graspsplats: Efficient manipulation with 3d feature splatting," *arXiv preprint arXiv:2409.02084*, 2024.

[18] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang, "Robogsim: A real2sim2real robotic gaussian splatting simulator," *arXiv preprint arXiv:2411.11839*, 2024.

[19] T. van Oort, D. Miller, W. N. Browne, N. Marticorena, J. Haviland, and N. Suenderhauf, "Open-vocabulary part-based grasping," *arXiv preprint arXiv:2406.05951*, 2024.

[20] Y. Tang, S. Zhang, X. Hao, P. Wang, J. Wu, Z. Wang, and S. Zhang, "Affordgrasp: In-context affordance reasoning for open-vocabulary task-oriented grasping in clutter," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[21] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*. Springer, 2024, pp. 38–55.

[22] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," in *The Thirteenth International Conference on Learning Representations*, 2024.

[23] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[24] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[25] J. Liu, Y. Zhao, H. Ma, Z. Liu, J. Wang, and W. Zou, "Mgpc: Multi-modal network for generalizable point cloud completion with modality dropout and progressive decoding," *arXiv preprint arXiv:2601.03660*, 2026.

[26] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 498–12 507.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.