Beyond Visual Grasping: Benchmarking Complex Grasping from Detection to Execution

H. Zhang[1], K. Nguyen[2], C. Munasinghe[3], B. Hela[4], T. Li[1], Z. Luo[1], H. Nguyen[5], H.W van de Venn[3], Y. Zheng[1], R. Prakash[4], T. D. Ta[6,7], A. Nguyen[1], B. Huang[1]

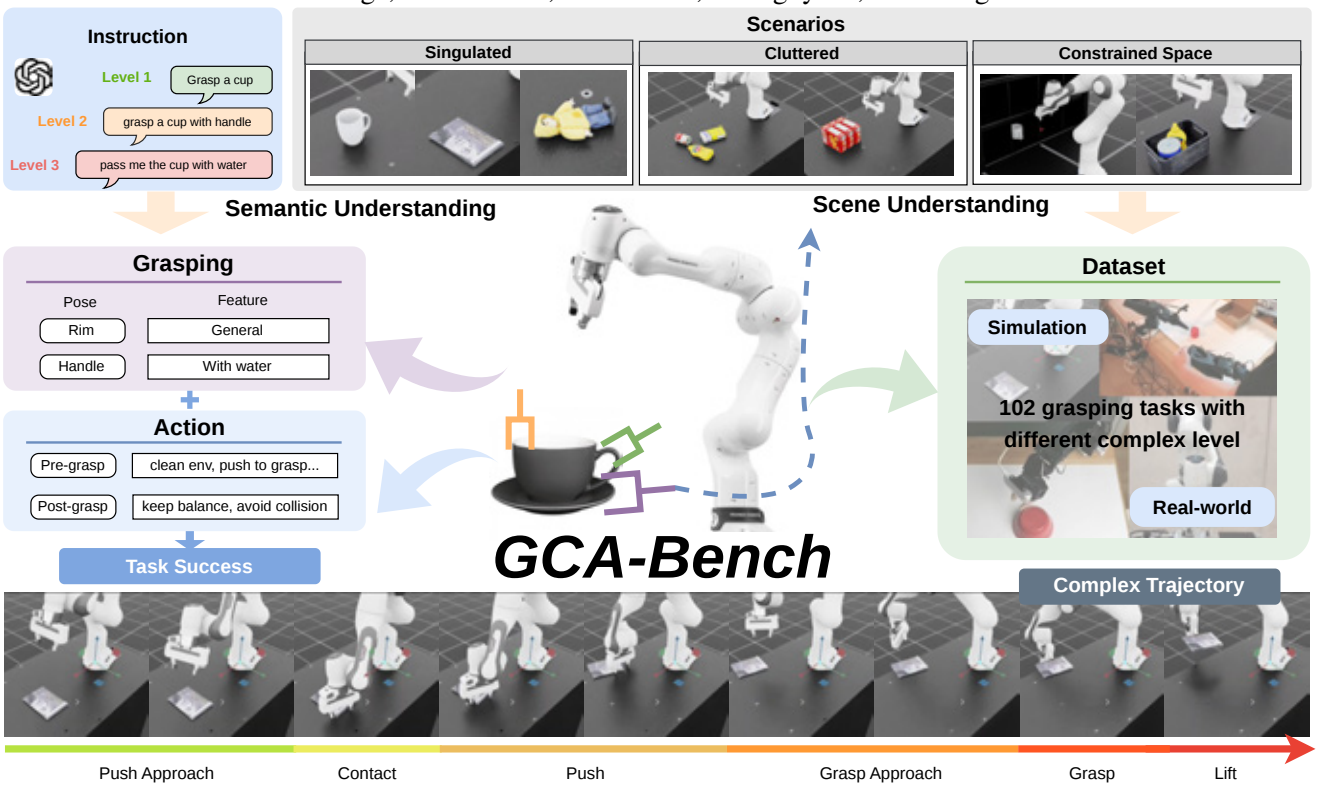

Fig. 1: We propose GCA-Bench, a benchmark for evaluating complex robotic grasping skills as a multi-stage process.

***Abstract*— Robust robotic grasping remains a fundamental challenge for complex real-world applications. Recent advances in large-scale models demonstrate promising capabilities for reasoning in robotic tasks. However, existing benchmarks for grasping primarily focus on isolated, visual-based grasp pose detection, failing to capture the complexity of grasping tasks that require multi-step reasoning and semantic understanding during execution. To address this gap, we propose GCA-Bench, a benchmark featuring challenging *grasping with complex action* scenarios that involve both scene-level reasoning and semantic constraints. GCA-Bench enables the evaluation of recent large foundation models under the same settings. To demonstrate the effectiveness of our new benchmark, we implement a diverse set of baselines, ranging from traditional grasp detection pipelines to end-to-end learning methods. Empirical studies achieve success rates below 70% on complex grasping scenarios, underscoring critical limitations. In addition, we propose new evaluation metrics, analyze critical failure models, and provide insights to guide the development of more robust and generalizable grasping strategies. Our project is available at https://airvlab.github.io/GCA-Bench/**

[1]University of Liverpool, UK
[2]Mohamed bin Zayed University of Artificial Intelligence, UAE
[3]Zurich University of Applied Science, Switzerland
[4]Indian Institute of Science, India
[5]University of Information Technology, HCMC, Vietnam
[6]Keio University, Japan

AN and TT were supported by the Royal Society ISPF International Collaboration Awards (ICA\R1\231067)

I. INTRODUCTION

Grasping constitutes a core capability for autonomous robots with several real-world applications [1], [2]. Recent advances in robotic grasping span multiple directions, including learning-based grasp pose detection and the adoption of foundation models for semantic task understanding [3]. These developments have substantially improved grasp success in standard benchmarks. Despite these advances, grasping performance remains fragile in complex scenarios [4], including manipulating objects with difficult geometries such as flat cards and thin sheets, operating in cluttered environments, retrieving items from confined spaces, or satisfying language-specified task constraints [5], [6].

We define complex grasping as grasping that requires scene and semantic reasoning capabilities beyond isolated grasp pose prediction to overcome geometric, spatial, or semantic difficulty. Existing efforts to address these challenges generally follow two complementary paradigms. One

TABLE I: Comparison of grasping benchmarks and datasets.

Benchmark	Focus	Language[1]	Trajectory[2]	Grasp Evaluation[3]	Complex Grasping[4]	Sim Data[5]	Real Data[6]
RLBench [7]	Manipulation	✗	✓	✗	✗	✓	✗
Robocasa [8]	Manipulation	✗	✓	✗	✗	✓	✗
LIBERO [9]	Manipulation	✓	✓	✗	✗	✓	✗
VLAbench [10]	Generality	✓	✓	✗	✗	✓	✗
COLOSSEUM [11]	Perturbation	✗	✓	✗	✗	✓	✗
GraspNet-1Billion [12]	Grasp pose detection	✗	✗	✓	✗	✗	✓
GraspAnything [13]	Grasp pose detection	✓	✗	✓	✗	✓	✗
FMB [14]	Assembly problems	✗	✓	✓	✓	✗	✓
Fetchbench [4]	Fetching	✗	✓	✓	✗	✗	✗
SynGrasp-1B [15]	Grasp pose detection	✗	✓	✓	✗	✓	✗
GCA-Bench (ours)	Complex grasping	✓	✓	✓	✓	✓	✓

[1] Language instructions. [2] Task execution trajectory is considered. [3] Grasp ability evaluation. [4] Complexity categorization of grasping tasks. [5] Synthetic or simulated data collection. [6] Real-world data collection.

emphasizes hardware specialization, introducing suction-based [16], multi-fingered [17], or adaptive end-effectors [18] to expand physical capabilities. The other seeks to enhance algorithmic intelligence for parallel-jaw grippers, which continue to dominate practical robotic deployments. Algorithmic strategies include data-driven grasp pose prediction integrated with motion planning [12], [13], [19], leveraging large-scale foundation models for semantic reasoning [20]–[22], and end-to-end architectures that unify perception and control [23], [24]. While these techniques show promise, it remains insufficiently understood how these approaches perform across diverse and complex grasping scenarios. This gap motivates the need for systematic evaluation frameworks capable of exposing failure cases and informing the next generation of algorithmic designs for grasping systems.

Current benchmarking efforts in robotic grasping remain largely focused on visual grasp pose detection [13], [25], [26], where success is evaluated in isolation rather than across the full manipulation sequence. Recent research [4] has demonstrated that even state-of-the-art methods exhibit substantial performance degradation when evaluated across the entire grasping pipeline. Furthermore, recent large models such as π_0 [27], which bypass explicit grasp detection in favor of direct action prediction, are making traditional detection-oriented benchmarks increasingly inadequate. This gap underscores the need for a unified benchmarking framework that enables fair and comprehensive evaluation of modern grasping algorithms. To address this gap, we introduce GCA-Bench, a benchmark designed to evaluate grasping with complex actions across multiple levels of difficulty. Unlike conventional benchmarks that focus solely on grasp pose detection, GCA-Bench requires integrated reasoning over trajectory planning, including spatial reasoning and multi-step coordination, as well as semantic task understanding. Our contributions include:

- We introduce GCA-Bench, a complex benchmark to evaluate robotic grasping with complex action scenarios, encompassing both scene-level reasoning and semantic constraints throughout the full execution pipeline.
- We conduct intensive experiments and analysis to reveal critical limitations of existing methods and provide insights to develop more robust and generalizable robotic grasping systems for real-world applications.

II. Related Work

Complex Grasping. Conventional grasping systems separate the problem into grasp pose detection [20], [28], [29] and collision-free motion planning [4], [30]. While effective in detection benchmarks, such systems assume static execution and lack online adaptation or holistic reasoning over the entire manipulation process [31], [32]. In more complex settings, grasping must handle occlusions, clutter, ungraspable features, and spatial constraints, while aligning grasp strategies with task intent [4], [6], [33]. End-to-end policy learning attempts to close this gap by mapping perception and language inputs directly to actions [34]. Reinforcement learning enables adaptive strategies such as push–grasp sequences and recovery behaviors [35]–[37], while imitation learning improves data-efficient acquisition of contact-rich skills [38], [39]. More recently, foundation models have enhanced semantic grounding and generalization for language-conditioned grasping [15], [40]–[42]. Nevertheless, robust adaptation under strict geometric and semantic constraints remains an open challenge [4], [6].

Grasping Benchmarks. Over the years, many benchmarking frameworks have been developed for manipulation and grasping tasks. As shown in Table I, manipulation benchmarking works such as LIBERO [9] and VLABench [10] provide comprehensive task-level evaluation but focus primarily on high-level manipulation policies and long-horizon tasks rather than low-level grasping execution. However, current benchmarks on grasping, such as GraspNet-1Billion [12] and Grasp-anything [13], mainly focus on grasp pose detection without considering the execution. FetchBench [4] represents a notable advance in addressing the integration of grasping and motion planning within fetching tasks. However, it only focused on the difficulty of collision avoidance rather than on the complexity of the tasks themselves. In contrast, GCA-Bench is the first benchmark designed for complex grasping tasks from detection to execution.

We propose GCA-Bench, a robotic benchmark for challenging grasping scenarios that require understanding how humans perform the task. Unlike benchmarks such as LIBERO [9], which mainly target common everyday manipulation skills (e.g., opening a microwave), GCA-Bench emphasizes human-like task understanding for difficult and realistic real-world robotic grasps.

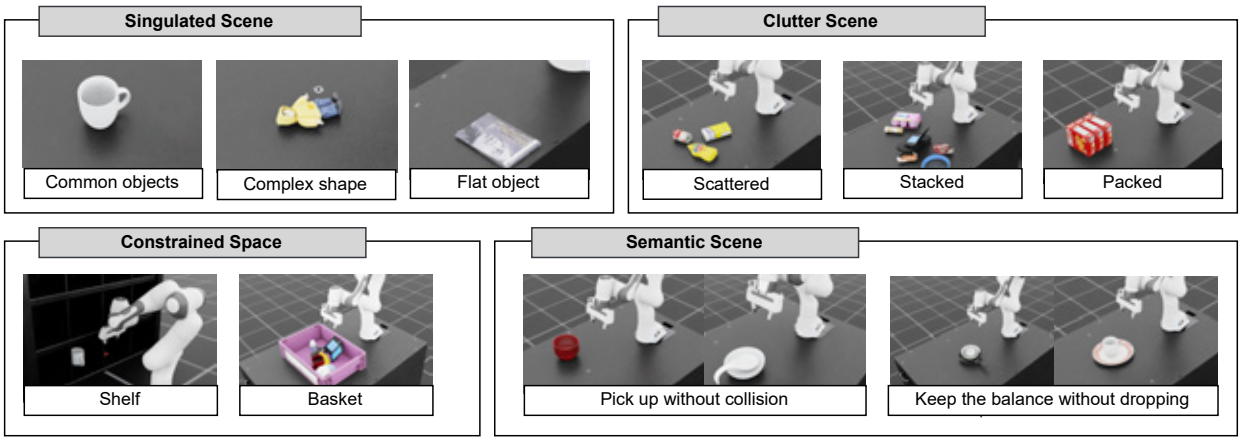

Fig. 2: Overview of scenario design in GCA-Bench. The benchmark comprises four categories of manipulation tasks: (*i*) *Singulated Scene*, involving isolated objects of varying shapes and poses; (*ii*) *Cluttered Scene*, where objects are scattered, stacked, or densely packed; (*iii*) *Constrained Space*, requiring precise manipulation in constrained environments such as shelves or baskets; and (*iv*) *Semantic Scene*, requiring the robot to strictly follow the language instructions.

III. The GCA-Bench

A. Scenario Design

To enable comprehensive evaluation of robotic grasping and manipulation, we design 102 task scenarios with varying levels of difficulty, ranging from singulated objects to cluttered environments, constrained narrow spaces, and semantic tasks requiring contextual reasoning (Fig. 2) with 102 tasks in total.

Singulated Scene. This basic scenario involves grasping and lifting single objects placed on a flat surface. The objects are selected to span a variety of properties, such as common objects which are mostly from YCB datasets [43], and objects with complex shape such as some irregularly shaped toys and flat objects that are inherently challenging to grasp. To ensure robust evaluation, we vary object pose and placement. For example, a bowl may be positioned upright or upside down, significantly affecting grasp difficulty.

Cluttered Scene. Cluttered environments simulate real-world scenarios where objects are overlapping or stacked. We use the same object set to construct clutter scenes with increasing complexity. These include:

- *Scattered Scene*: Objects are randomly distributed across the workspace with sufficient spacing.
- *Stacked Scene*: Objects are piled tightly, resulting in severe occlusions and increased risk of collision.
- *Packed Scene*: Objects are densely arranged along axes, leaving minimal clearance for gripper insertion.

These settings challenge the robot to detect objects under occlusion and plan access paths.

Constrained Space. Constrained environments, such as shelves or baskets, test the robot's ability to operate within limited spatial bounds. These scenarios demand precise motion planning, adaptive collision avoidance, and safe navigation. The difficulty is determined by object placement and surrounding obstructions. For example, shelf-mounted objects may be occluded by frames or adjacent items, while baskets often contain tightly packed objects requiring careful maneuvering to avoid damage. Objects placed in corners or against walls further restrict feasible grasp trajectories.

Semantic Scene. We design challenging tasks specifically targeting semantic understanding. Semantic scenarios evaluate the robot's ability to reason about task-level context beyond geometric constraints, requiring high-level decision making in both grasp pose selection and trajectory planning. For instance, in a cup set task, the robot must grasp without disturbing balance to prevent spillage, which demands timely feedback and online adjustment. Similarly, tasks such as retrieving a fragile object from a cluttered scene require precise motion planning and careful collision avoidance.

B. Instruction Generation

To enable language-driven human–robot interaction, we use a large language model (ChatGPT-4o mini [44]) to generate task instructions at three levels of complexity. This choice is motivated by how humans naturally communicate with robots: instructions are often expressed in flexible, high-level language, may omit details, and vary widely in specificity. ChatGPT-4o mini [44] is well-suited for this because it can produce diverse, fluent, and context-aware instructions that resemble natural user prompts, rather than templated command scripts. Generated instructions were manually reviewed to remove ambiguous object references, mismatched complexity levels, and ungrammatical phrasing.

Level 1 - Basic Instructions. Simple commands that specify the target object, e.g., "Grasp the cup." These instructions require object recognition and basic grasp execution without further context.

Level 2 – Specific Instructions with Constraints. These instructions include explicit requirements, such as "Grasp the cup by the handle" or "Pick up the book from the top edge." Robots must interpret object affordances and plan grasp strategies that satisfy constraints.

Level 3 – Complex Task Instructions. High-level semantic tasks require reasoning and multi-step planning. Examples include "Pass me a cup with water" or "Give me a knife." These tasks demand context-aware grasping, e.g., avoiding

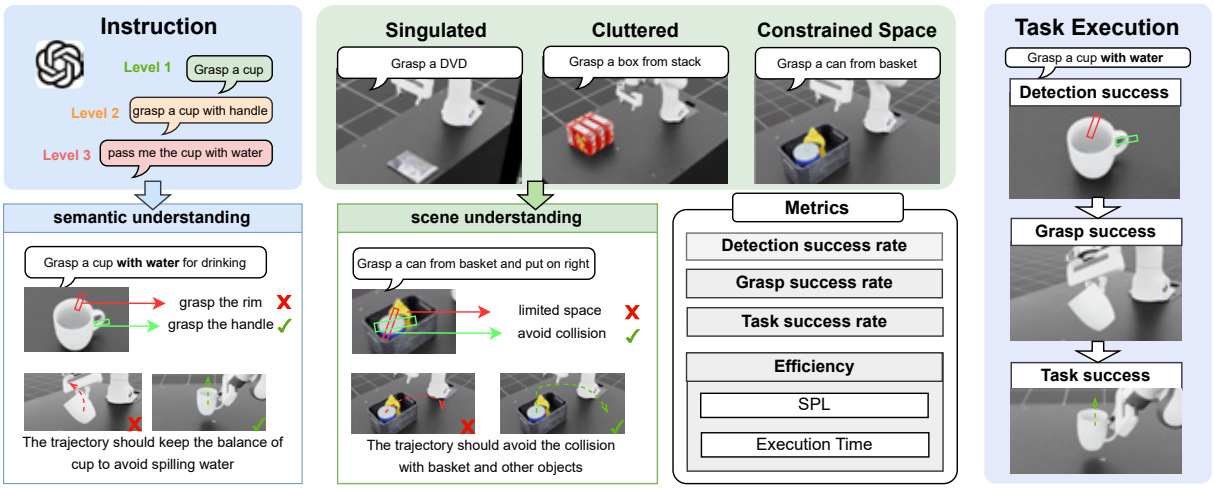

Fig. 3: Tasks are designed to evaluate both semantic and scene understanding. Execution is assessed along a pipeline of detection, grasping, and task-level success. Complex tasks require not only correct grasping but also proper trajectory execution, as illustrated by the examples. Performance is measured using detection success rate, grasp success rate, task success rate, and efficiency metrics including SPL and execution time.

tilting a full cup or grasping a knife safely by the handle, requiring semantic understanding of function, safety, and trajectory planning.

C. Environment Setup

We utilize NVIDIA Isaac Lab [45] as the primary simulation platform for robot learning and policy training. Isaac Lab is a GPU-accelerated, open-source framework designed to unify and simplify robotics research workflows. Our benchmark is designed to flexibly integrate different robotic arms. In this work, we focus on the 7-DOF Franka Emika Panda with a Franka gripper. To provide comprehensive visual coverage of the workspace and meet the requirements of different algorithms, the robot is set up with four RGB-D cameras as shown in Fig. 4: a wrist camera mounted on the gripper, a third-view camera positioned in front of the arm, a side-view camera, and an over-shoulder camera. The end-effector pose is represented using 3D coordinates for position and quaternions for orientation. For standardising the evaluation, we inherited annotated objects from the MultiGripperGrasp dataset [46] with 30.4 million grasps over 345 objects using 11 different grippers. To construct diverse evaluation scenes, we additionally leverage various assets from Nvidia Omniverse, including containers and furniture such as tables, cabinets, and shelves.

D. Evaluation Metrics

As shown in Fig. 3, our benchmark evaluates grasping methods across varying task difficulties, considering both semantic understanding and scene reasoning. Beyond grasping success, we also assess task execution, providing insights into the limitations of current grasping pipelines and the gap between perception and execution. To evaluate full grasp execution, we design evaluation metrics at several graduali-ties, from detection, grasping, to task achievement. Success is defined not only by grasp completion but also by grasp quality and subsequent actions, such as maintaining object balance or avoiding collisions. In this setting, the robot must

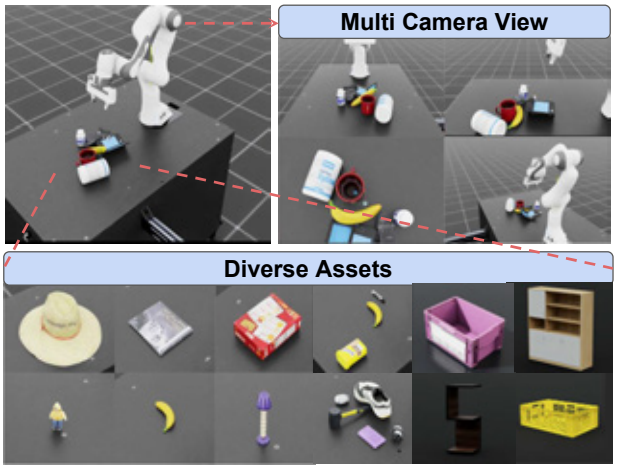

Fig. 4: Simulation setup in IsaacLab with four cameras: front, wrist-mounted, side, and over-the-shoulder. Task environments are constructed using diverse assets with varying physical properties from NVIDIA Omniverse.

execute grasps that satisfy semantic requirements rather than merely secure the object. Comparing performance across instruction levels allows us to isolate semantic reasoning from low-level manipulation, providing a more nuanced assessment of an algorithm's ability to interpret and act on task semantics in real-world contexts.

Metrics. We evaluate performance using several complementary metrics. All success-based metrics share a common formulation:

$$\text{Success}(M) = \frac{1}{N}\sum_{i=1}^{N} S_i^{(M)}, \tag{1}$$

where N is the total #trials, $S_i^{(M)} \in \{0,1\}$ is a binary indicator of whether trial i is successful under metric M. Different metrics correspond to different success conditions:

- **Detection Success Rate (DSR):** $S_i^{(\text{DSR})} = 1$ if the target object is correctly detected from the initial camera

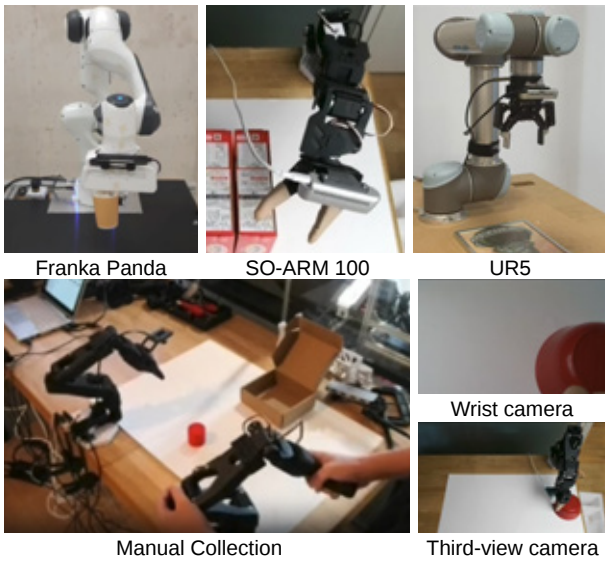

Fig. 5: Real-world data collection setup with three robotic arms: Franka Emika Panda, SO-ARM 100, and UR5. We employ one wrist-mounted camera and one third-view camera, capturing both RGB and depth images.

observation.

- **Grasp Success Rate (GSR):** $S_i^{(\text{GSR})} = 1$ if the robot successfully grasps and lifts the target object to a predefined height.
- **Task Success Rate (TSR):** $S_i^{(\text{TSR})} = 1$ if the grasp both succeeds and satisfies task-specific constraints.

We further evaluate execution efficiency using the success weighted by path length (SPL) metric:

$$\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \times \frac{l_i^*}{\max(l_i^*, l_i)}, \tag{2}$$

where S_i is a binary success indicator, l_i^* is the shortest path length, and l_i is the actual executed path length.

To compare run-time efficiency across methods, we report the Execution Time (ET) in seconds for each approach. All experiments were conducted on the same hardware, an Intel Xeon 4215R (3.2 GHz) CPU and an NVIDIA GeForce RTX 4080 GPU, to ensure a consistent and fair comparison.

IV. Experiments

Since our goal is to benchmark the grasping effectiveness of recent methods, especially VLA models, we first collect a dataset to enable fine-tuning of these models. In our preliminary experiments, zero-shot use of these approaches shows significantly low accuracy on the complex tasks required by our design, so fine-tuning is necessary to obtain reliable results for a fair comparison. This is also widely observed by several recent works [4], [47].

A. Data Collection

Since many of our tasks involve complex object interactions that remain unsolved by current autonomous grasping systems, we collect trajectories manually. For each task, we vary trajectories and grasp poses to ensure diversity and robustness. In simulation, we use a 3Dconnexion SpaceMouse, which allows smooth trajectory demonstrations with fine-grained control. For real-world experiments, we employ the robot's teaching mode and SpaceMouse, enabling natural teleoperation and precise execution. We collect both successful and failed trials. Failure cases, such as drops, missed grasps, or collisions, provide informative negative supervision for learning-based methods and allow fine-grained analysis of task difficulty. Each trajectory includes synchronized observations from both a wrist-mounted and a third-person camera, recording RGB and depth images. In addition, we log robot state information, including end-effector pose, joint positions, and executed actions. The action space is represented as the delta of the end-effector position and orientation $[x, y, z, r, p, y]$ with the binary gripper state. To this end, we collected a dataset with 5000 simulation trajectories and 800 trajectories from real robots to support fine-tuning. Fig. 5 shows our real-world data collection setup, and Fig. 6 shows examples of collected trajectories.

B. Baselines

To evaluate the performance of existing grasping systems on GCA-Bench, we consider representative baselines from two major paradigms: (*i*) grasp detection with motion planning and (*ii*) end-to-end VLA policies. For the grasp detection pipeline, we include AnyGrasp [28], a state-of-the-art 6-DoF grasp detection model, and GraspMAS [48], a multi-agent framework that leverages large language models for language-conditioned grasp generation. To assess the impact of motion feasibility and collision avoidance, we evaluate these predicted grasps both independently and in combination with cuRobo [30], which performs collision-aware trajectory optimization and control. For VLA models, we first evaluate GraspVLA [15] in a zero-shot setting, which leverages large dataset for strong zero-shot generalization. We further consider general VLA models finetuned on our dataset, including OpenVLA [49] and its variant OpenVLA-OFT [50], as well as models from the π-series: π_0 [27] and $\pi_{0.5}$ [51]. These models are fine-tuned using LoRA on collected dataset, for 60k steps on 4 NVIDIA A100 GPUs.

For each task category, we sample 5 scenarios. Within each scenario, we randomize object poses and arrangements across 10 trials, yielding 50 trials per task category. Based on the design of GCA-Bench, we conduct experiments to address the following research questions:

Q1: How do existing baselines perform on our benchmark? Can they support complex grasping tasks?

Q2: What is the gap between visual-based grasp detection and real-world grasp execution? Can current motion planning or end-to-end algorithms bridge this gap?

Q3: Can GCA-Bench improve the performance on real-world tasks?

C. Results

Q1: How do existing baselines perform? Table II shows that the overall success rate across all four major task categories remains below 70% for each baseline, indicating that existing methods struggle to support complex grasping tasks.

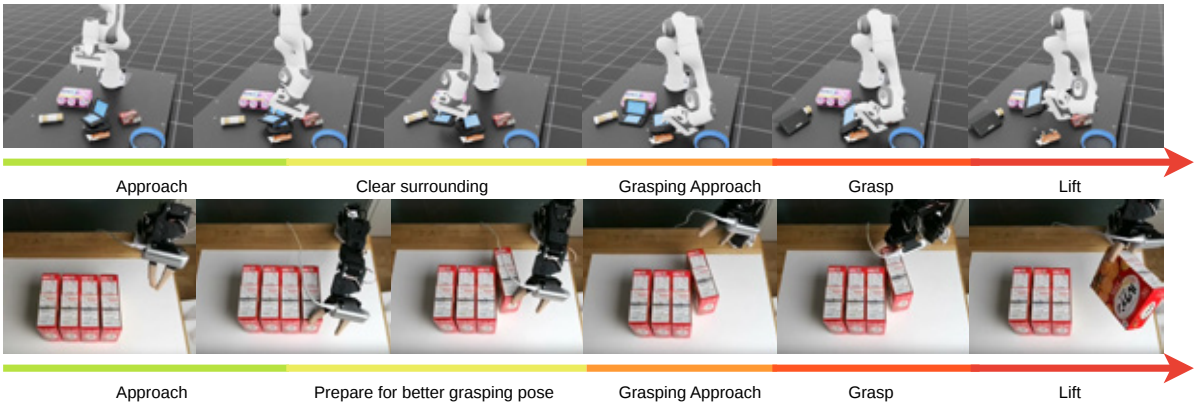

Fig. 6: Action complexity from our collected data. Complex cluttered scenes cannot be solved by visual detection alone and require sufficient trajectory planning. For example, in stacked object scenarios, surrounding objects need to be cleared to fetch the target safely, and additional actions such as pushing are required to prepare for a better grasping pose.

TABLE II: Comparison of baseline methods on GCA-Bench.

	Singulated			Clutter			Constrained Space			Semantic		
Method	TSR↑	SPL↑	ET↓	TSR↑	SPL↑	ET↓	TSR↑	SPL↑	ET↓	TSR↑	SPL↑	ET↓
AnyGrasp [28]	0.37	0.15	13.07	0.27	0.11	13.07	0.13	0.19	**9.05**	–	–	–
GraspMAS [48]	0.23	0.18	30.47	0.07	0.07	46.14	0.14	0.28	31.88	0.15	0.73	45.24
AnyGrasp [28] + cuRobo [30]	0.41	0.68	12.41	0.27	0.58	**12.74**	0.32	0.69	9.65	–	–	–
GraspMAS [48] + cuRobo [30]	0.28	0.72	29.78	0.08	0.67	43.72	0.11	**0.72**	30.46	0.16	**0.84**	45.02
GraspVLA [15]	0.07	0.43	223.15	0.08	0.35	513.61	0.05	0.51	194.50	0.08	0.75	153.84
OpenVLA [49]	0.58	0.54	97.64	0.42	0.68	203.60	0.41	0.65	139.04	0.43	0.68	174.17
OpenVLA-OFT [50]	0.73	0.68	**8.32**	0.57	0.71	24.51	**0.51**	0.52	15.6	0.42	0.70	18.36
π_0 [27]	**0.77**	**0.77**	9.37	0.52	0.64	22.65	**0.51**	0.71	14.4	0.45	0.61	17.7
$\pi_{0.5}$ [51]	**0.77**	0.73	13.35	**0.61**	**0.81**	16.77	0.46	0.16	13.35	**0.53**	0.65	**14.85**

Detection-based pipelines achieve low task success rates on complex grasping. Although incorporating motion planning slightly improves performance by reducing collisions, it does not address the fundamental limitations of grasp detection. In particular, traditional pipelines lack adaptive replanning and task-level reasoning, which are critical for multi-step or constraint-heavy grasping. Despite its claimed zero-shot generalization ability, GraspVLA fails on most tasks, likely because its training data is biased toward standard grasping settings. Fine-tuned VLA models demonstrate substantially better performance, with $\pi_{0.5}$ achieving the best overall results. However, performance still degrades as task complexity increases. Moreover, the relatively low success rate on semantic tasks suggests that current VLA models still have limited high-level reasoning capabilities.

Table III breaks down the performance of different methods across scenes of varying difficulty. This table shows that detection-based two-stage methods are efficient and robust in simple tasks such as grasping common objects and scattered scenes, but fail in complex scenarios due to the lack of real-time feedback. Adding collision-free motion planning improves performance in constrained environments but can sometimes reduce success, for example, in cluttered scenes, where incidental contact can facilitate stable grasps. These results show that current pipelines are brittle and far from robust across diverse complex tasks.

Table IV further evaluates semantic instruction-following performance under increasing levels of complexity. Grasp-

TABLE III: Evaluation results across different task scenarios.

	Singulated			Clutter			Constrained	
Method	simple	complex	flat	scattered	stacked	packed	basket	shelf
AnyGrasp [28]	0.62	0.50	0.0	0.44	**0.26**	0.12	0.22	0.04
GraspMAS [48]	0.32	0.36	0.0	0.08	0.12	0.02	0.28	0.0
AnyGrasp [28]+cR [30]	0.76	0.48	0.0	0.48	0.22	0.10	0.54	0.10
GraspMAS [48]+cR [30]	0.44	0.40	0.0	0.08	0.12	0.04	0.22	0.0
GraspVLA [15]	0.18	0.02	0.0	0.14	0.06	0.04	0.10	0.0
OpenVLA [49]	0.88	0.62	0.24	0.53	0.08	0.64	0.60	0.22
OpenVLA-OFT [50]	**0.98**	**0.82**	0.40	0.78	0.20	0.74	0.66	0.36
π_0 [27]	0.96	0.74	**0.58**	0.72	0.12	0.72	0.50	0.34
$\pi_{0.5}$ [51]	**0.98**	0.76	0.56	**0.82**	0.24	**0.76**	**0.68**	**0.42**

MAS maintains reasonable performance on basic and simple-constraint tasks by leveraging LLM-based reasoning, but its reasoning is confined to localizing the object or part specified by the instruction and generating a grasp pose on it. In contrast, VLA models rely on action patterns learned from demonstrations rather than explicit language understanding. They perform well when instructions closely match the training distribution but struggle with novel semantic constraints, revealing that their instruction-following capability is largely a form of pattern matching rather than true task reasoning.

TABLE IV: Instruction following results

	Basic		Simple Constraint		High-level	
Method	GSR↑	TSR↑	GSR↑	TSR↑	GSR↑	TSR↑
GraspMAS [48]	0.46	0.42	0.42	0.16	0.12	0.04
GraspMAS [48]+cR [30]	0.46	0.42	0.44	0.16	0.12	0.04
GraspVLA [15]	0.2	0.16	0.12	0.08	0.06	0.0
OpenVLA [49]	0.76	0.76	0.78	0.32	0.68	0.20
OpenVLA-OFT [50]	0.82	0.80	0.80	0.30	0.82	0.16
π_0 [27]	0.86	0.82	0.84	0.28	0.78	0.24
$\pi_{0.5}$ [51]	**0.92**	**0.92**	**0.90**	**0.42**	**0.88**	**0.30**

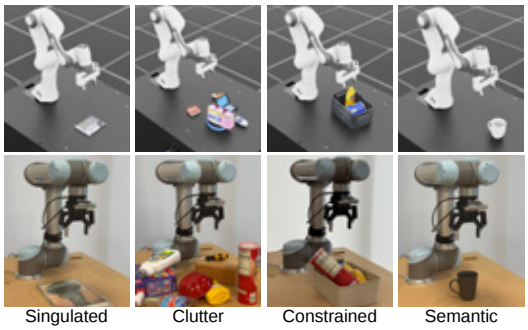

Fig. 7: Paired task scenarios for real-world validation.

Q2: What is the gap between detection and execution? Table V reveals the gap between detection capability and execution success by normalizing grasp success rates against detection performance. The results show that the ratio is consistently below 0.5, indicating that even when objects are successfully detected, executing stable grasps in complex scenarios remains challenging. To analyze this gap more systematically, we break down the failures into three levels: detection, grasp, and task execution, each revealing different limitations of the pipeline. (*i*) Detection Failures: Common objects are reliably detected, whereas irregular shapes and occluded items remain challenging. Robustness could be improved through multi-view or temporal fusion, and further enhanced with self-supervised refinement or online adaptation in cluttered or partially observed environments. *(ii)* Grasp Failures: Small, thin, or slippery objects frequently slip during execution, even when a valid grasp pose is detected. Cluttered and constrained spaces further exacerbate failures by restricting maneuverability. These cases often require preparatory actions or adaptive strategies. *(iii)* Task Failures: As shown in Table IV, for tasks with semantic or physical constraints, such as grasping a filled cup, open-loop pipelines fail due to the absence of feedback. We observe that current methods cannot detect or compensate when objects tilt or when collisions occur, leading to frequent task failures. This underscores the need for closed-loop control and adaptive planning in complex manipulation tasks.

TABLE V: Detection-normalized grasp success rates.

Method	Singulated	Clutter	Constrained
AnyGrasp [28]	0.37	0.27	0.14
GraspMAS [48]	0.28	0.12	0.23
AnyGrasp [28] + cuRobo [30]	**0.41**	**0.28**	**0.34**
GraspMAS [48] + cuRobo [30]	0.34	0.12	0.18

Q3: Can GCA-Bench improve the performance on real-world tasks? To validate that GCA-Bench provides a meaningful evaluation platform and supports deployment on real-world grasping tasks, we conduct experiments using a UR5 robot with a Robotiq 2F-85 gripper. We evaluate four representative tasks for each category as shown in Fig. 7. For each task, the fine-tuned $\pi_{0.5}$ policy is tested over 20 trials and compared with simulation results on our benchmark. As shown in Fig. 8, the results reveal consistent performance. Singulated and Clutter remain challenging due to precise multi-step manipulation requirements, while

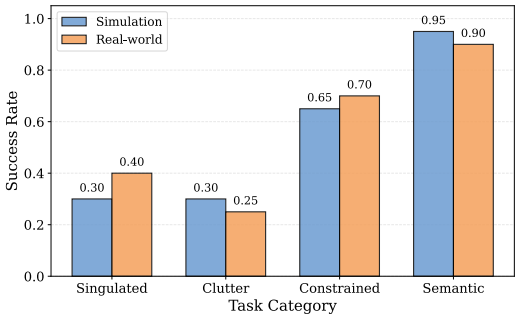

Fig. 8: Real-world performance with simulation baseline.

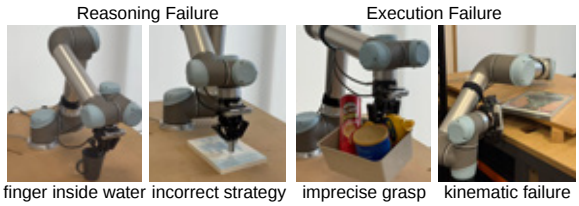

Fig. 9: Failure cases.

Constrained tasks benefit from collision-aware behaviors learned in simulation. Semantic tasks achieve the highest success rate, indicating that diverse language instructions in our dataset improve instruction-following in real-world settings. Notably, real-world performance closely aligns with simulation results, suggesting that GCA-Bench provides a reliable and practical benchmark suite for developing grasping methods capable of handling complex real-world grasping.

Fig. 9 shows representative failure cases. For unseen instruction, the policy struggles with high-level semantic reasoning and object affordance understanding (e.g., "pass me a cup of water"). Similarly, for unseen flat objects, the policy fails to associate them with similar tasks present in the training dataset. For seen tasks, failures mainly arise from imprecise grasps, leading to incorrect object misgrasp or unstable grasps. Additionally, the VLA policy occasionally predicts actions that lead to kinematically infeasible configurations without recovery, highlighting a gap between action prediction and low-level motion feasibility.

V. Conclusions

We introduce GCA-Bench, a benchmark for evaluating robotic grasping across the full pipeline under both scene-level and semantic complexity. Our experiments show that existing methods perform well in simple settings, but success rates drop sharply in cluttered, constrained, or instruction-driven tasks, exposing a persistent gap between perception and reliable execution. GCA-Bench is additionally well-suited for testing VLA models because it couples language-conditioned, semantically grounded grasp goals with the full demands of execution, enabling a direct assessment of whether VLAs truly ground instructions into robust actions rather than relying on perception or unconstrained motions. A current limitation is that GCA-Bench focuses on parallel-jaw grippers; future work will expand to more diverse gripper types and settings. GCA-Bench will be made publicly available to support the benchmarking and future study.

REFERENCES

[1] Z. Xie, X. Liang, and C. Roberto, "Learning-based robotic grasping: A review," *Frontiers in Robotics and AI*, 2023.

[2] M. Dong and J. Zhang, "A review of robotic grasp detection technology," *Robotica*, 2023.

[3] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," *Artificial Intelligence Review*, 2021.

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

[5] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, *et al.*, "Deep learning approaches to grasp synthesis: A review," *IEEE T-RO*, 2023.

[6] A. Efendi, Y.-H. Shao, and C.-Y. Huang, "Technological development and optimization of pushing and grasping functions in robot arms: A review," *Measurement*, 2025.

[7] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE RA-L*, 2020.

[8] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv:2406.02523*, 2024.

[9] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *NeurIPS*, 2023.

[10] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang, *et al.*, "Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks," *arXiv:2412.18194*, 2024.

[11] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, "The colosseum: A benchmark for evaluating generalization for robotic manipulation," *arXiv:2402.08191*, 2024.

[12] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *CVPR*, 2020.

[13] A. D. Vuong, M. N. Vu, H. Le, B. Huang, H. T. T. Binh, T. Vo, A. Kugi, and A. Nguyen, "Grasp-anything: Large-scale grasp dataset from foundation models," in *ICRA*, 2024.

[14] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "Fmb: a functional manipulation benchmark for generalizable robotic learning," *IJRR*, 2025.

[15] S. Deng, M. Yan, S. Wei, H. Ma, *et al.*, "Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data," *arXiv:2505.03233*, 2025.

[16] H. Cao, H.-S. Fang, W. Liu, and C. Lu, "Suctionnet-1billion: A large-scale benchmark for suction grasping," *IEEE Robotics and Automation Letters*, 2021.

[17] J. Lundell, E. Corona, T. N. Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyrki, "Multi-fingan: Generative coarse-to-fine sampling of multi-finger grasps," in *ICRA*, 2021.

[18] Z. Xu, B. Qi, S. Agrawal, and S. Song, "Adagrasp: Learning an adaptive gripper-aware grasping policy," in *ICRA*, 2021.

[19] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *IROS*, 2018.

[20] A. D. Vuong, M. N. Vu, B. Huang, N. Nguyen, H. Le, T. Vo, and A. Nguyen, "Language-driven grasp detection," in *CVPR*, 2024.

[21] T. Ngyen, M. N. Vu, A. Vuong, D. Nguyen, T. Vo, N. Le, and A. Nguyen, "Open-vocabulary affordance detection in 3d point clouds," in *IROS*, 2023.

[22] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, *et al.*, "Open-world object manipulation using pre-trained vision-language models," *arXiv:2303.00905*, 2023.

[23] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, "Review of deep reinforcement learning-based object grasping: Techniques, open challenges, and recommendations," *IEEE Access*, 2020.

[24] J. Hua, L. Zeng, G. Li, and Z. Ju, "Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning," *Sensors*, 2021.

[25] Z. Liu, W. Liu, Y. Qin, F. Xiang, *et al.*, "Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation," *IEEE RA-L*, 2021.

[26] N. Nguyen, M. N. Vu, B. Huang, A. Vuong, N. Le, T. Vo, and A. Nguyen, "Lightweight language-driven grasp detection using conditional consistency model," in *IROS*, 2024.

[27] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A vision-language-action flow model for general robot control." *arXiv.2410.24164*.

[28] 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 T-RO*, 2023.

[29] T. Nguyen, M. N. Vu, B. Huang, A. Vuong, Q. Vuong, N. Le, T. Vo, and A. Nguyen, "Language-driven 6-dof grasp detection using negative prompt guidance," in *ECCV*, 2024.

[30] B. Sundaralingam *et al.*, "Curobo: Parallelized collision-free robot motion generation," in *ICRA*, 2023.

[31] H. Geng, S. Wei, C. Deng, B. Shen, H. Wang, and L. Guibas, "Sage: Bridging semantic and actionable parts for generalizable manipulation of articulated objects," *arXiv:2312.01307*, 2023.

[32] Y. Liu, A. Qualmann, Z. Yu, M. Gabriel, P. Schillinger, M. Spies, N. A. Vien, and A. Geiger, "Efficient end-to-end detection of 6-dof grasps for robotic bin picking," in *ICRA*, 2024.

[33] K. Li, J. Wang, L. Yang, C. Lu, and B. Dai, "Semgrasp: Semantic grasp generation via language aligned discretization," in *ECCV*, 2024.

[34] H. Gao, J. Zhao, Y. Yang, and C. Sun, "An end-to-end multi-dimensional perception network architecture for robotic grasp detection with target edge collision-aware strategy," *IEEE T-ASE*, 2025.

[35] A. Lobbezoo and H.-J. Kwon, "Simulated and real robotic reach, grasp, and pick-and-place using combined reinforcement learning and traditional controls," *Robotics*, 2023.

[36] H. Sekkat, O. Moutik, L. Ourabah, B. ElKari, Y. Chaibi, and T. Ait Tchakoucht, "Review of reinforcement learning for robotic grasping: Analysis and recommendations," *SOIC*, 2024.

[37] Y. Huang, D. Liu, Z. Liu, K. Wang, Q. Wang, and J. Tan, "A novel robotic grasping method for moving objects based on multi-agent deep reinforcement learning," *Robotics and Computer-Integrated Manufacturing*, 2024.

[38] K. Wang, Y. Fan, and I. Sakuma, "Robot grasp planning: A learning from demonstration-based approach," *Sensors*, 2024.

[39] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE RA-L*, 2020.

[40] Y. Zhong, X. Huang, R. Li, C. Zhang, Y. Liang, Y. Yang, and Y. Chen, "Dexgraspvla: A vision-language-action framework towards general dexterous grasping," *arXiv:2502.20900*, 2025.

[41] A. Murali, B. Sundaralingam, Y.-W. Chao, W. Yuan, J. Yamada, M. Carlson, F. Ramos, S. Birchfield, D. Fox, and C. Eppner, "Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training," *arXiv:2507.13097*, 2025.

[42] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, *et al.*, "Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping," *IEEE R-AL*, 2024.

[43] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

[44] OpenAI, "Introducing ChatGPT," Software, accessed: July 6th 2023. [Online]. Available: https://openai.com/blog/chatgpt/

[45] M. Mittal, C. Yu, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE RA-L*, 2023.

[46] L. F. Casas, N. Khargonkar, B. Prabhakaran, and Y. Xiang, "Multi-grippergrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands," in *IROS*, 2024.

[47] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[48] Q. Nguyen, T. Le, H. Nguyen, T. Vo, T. D. Ta, B. Huang, M. N. Vu, and A. Nguyen, "Graspmas: Zero-shot language-driven grasp detection with multi-agent system," in *IROS*, 2025.

[49] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv:2406.09246*, 2024.

[50] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[51] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.