AdaClearGrasp: Adaptive Clearing for Robust Dexterous Grasping in Densely Cluttered Environments

Zixuan Chen[1,*], Wenquan Zhang[1,*], Jing Fang[2], Ruiming Zeng[1],
Zhixuan Xu[3], Yiwen Hou[3], Xinke Wang[1], Jieqi Shi[1], Jing Huo[1], Yang Gao[1]
[1]Nanjing University, [2]Nanjing University of Posts and Telecommunications, [3]National University of Singapore
*Equal contribution.

***Abstract*—In densely cluttered environments, physical interference, visual occlusions, and unstable contacts often cause direct dexterous grasping to fail, while aggressive singulation may compromise safety. Enabling a robot to adaptively decide whether to clear surrounding objects or directly grasp the target is therefore crucial for robust manipulation. We present AdaClearGrasp, a closed-loop decision-execution framework for adaptive clearing and dexterous grasping in densely cluttered scenes. A pretrained vision-language model (VLM) interprets visual observations and language instructions to reason about grasp interference and to generate a high-level plan, which invokes structured atomic skills through a Model Context Protocol (MCP) interface. For target acquisition, we instantiate a geometry-aware reinforcement learning grasping skill, GeoGrasp, built on a relative hand-object distance representation that generalizes to unseen object geometries in simulation. During execution, visual feedback monitors outcomes and triggers replanning upon failure. To evaluate language-conditioned dexterous grasping under graded clutter, we introduce `Clutter-Bench`, a ManiSkill3 benchmark with seven target objects and three clutter levels (210 scenarios). On a physical xArm7-XHand platform, AdaClearGrasp transfers from simulation without fine-tuning, reaching 70% success across 90 real trials. Project website: https://chenzixuan99.github.io/adaclear-grasp.github.io/.**

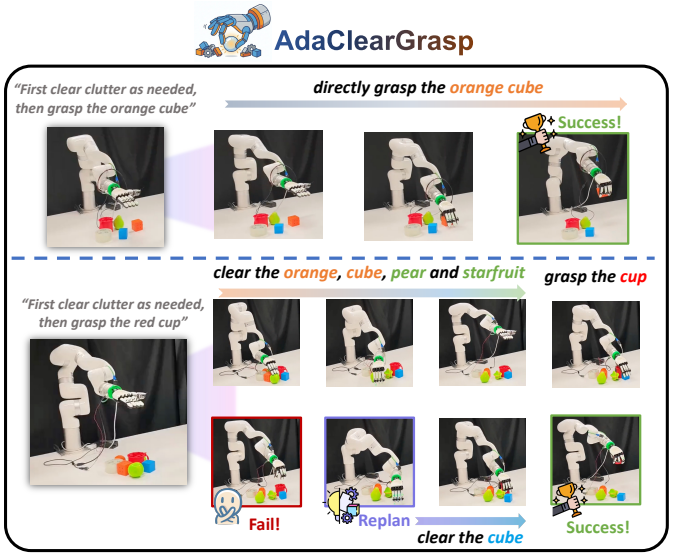

Fig. 1. **AdaClearGrasp** uses VLM-based reasoning to perform adaptive clearing for robust dexterous grasping of a language-specified target in densely cluttered scenes.

I. Introduction

Deploying robotic manipulation in real-world environments requires robust dexterous grasping in densely cluttered scenes [1]–[3]. In such settings, target objects are often surrounded by other items, causing physical interference, visual occlusions, and unstable contacts that make direct grasping unreliable. Prior work shows that effective manipulation in clutter often requires interacting with surrounding objects, such as pushing, singulating, or rearranging them, to expose the target [4]–[6]. However, excessive clearing introduces unnecessary interactions and risks damaging nearby items. Robust grasping in clutter therefore requires not only precise low-level control but also high-level decision-making about *when* and *how* to clear before grasping. Most existing dexterous grasping methods assume relatively isolated targets or rely on end-to-end RL policies that map perception directly to control [7]–[9]. Recent work extends RL to cluttered scenes, e.g., ClutterDexGrasp [10] and DexSinGrasp [6], producing emergent behaviors such as obstacle nudging. These methods largely treat manipulation as a low-level control problem and rely on emergent strategies that struggle in dense clutter, where long-horizon reasoning and adaptive interactions are needed. In parallel, vision-language models (VLMs) have shown strong semantic reasoning for manipulation [11]–[14]. Yet methods such as VLM Scaffolding [14] handle a single object with open-loop execution, and others provide limited feedback and are not designed for dense clutter. Deciding whether and how to clear is a multi-constraint problem (visibility, reachability, interaction safety) that must be tightly coupled with low-level control, which open-loop reasoning often fails to satisfy.

We propose **AdaClearGrasp**, a closed-loop framework that integrates hierarchical VLM reasoning with dexterous skills for adaptive clearing. A VLM planner jointly analyzes images and instructions to decide whether surrounding objects obstruct the target and whether clearing is required; when interference is detected, it emits parameterized clearing skills, and once the target is accessible it invokes an RL grasping skill, **GeoGrasp**. Rather than adopting an off-the-shelf grasping policy, we build GeoGrasp on the geometry-aware relative hand-object distance representation of RobustDexGrasp [15] and adapt it to our clutter-clearing setting: we re-implement the geometry-aware observation, design a task-specific dense re-

ward for the reach–grasp–lift objective, and wrap the resulting policy as a reusable atomic skill that the planner can invoke on demand through the MCP interface. This makes the grasping policy robust to diverse object geometries while remaining directly composable with high-level clearing decisions. A closed loop then monitors progress through visual feedback and triggers replanning upon failure. We further introduce **`Clutter-Bench`**, a graded ManiSkill3 [16] benchmark, and validate sim-to-real transfer on physical hardware. Our contributions are: (i) a closed-loop framework that models clutter clearing as adaptive high-level planning, integrating VLM reasoning with structured MCP skill invocation and typed failure feedback; (ii) `Clutter-Bench`, a standardized graded benchmark for language-conditioned target grasping in clutter, together with **GeoGrasp**, a geometry-aware RL grasping skill that we adapt from RobustDexGrasp [15] with a task-specific reward and integrate as a reusable atomic skill for clutter clearing; and (iii) an evaluation in simulation and sim-to-real transfer on real hardware.

II. Method

AdaClearGrasp operates in a closed loop (Fig. 2): (1) a VLM planner analyzes the scene and generates a high-level plan; (2) the plan is translated into parameterized atomic skills through the Model Context Protocol (MCP); (3) a skill library of clearing primitives, recovery primitives, and the RL-based GeoGrasp policy executes the interactions; and (4) visual feedback triggers replanning on failure.

A. Problem Formulation

We formulate target grasping in clutter as a hierarchical POMDP (S, A, T, R, O, γ), where S is the scene state, $A = A_{\text{high}} \cup A_{\text{low}}$ splits into high-level semantic actions and low-level control, T the dynamics, R the grasping reward, O the observation space (an RGB image and a list of detected objects), and γ the discount factor. At step t, the planner receives $o_t \in O$ with the image, language goal L, and feedback f_{t-1}, and outputs $a_{\text{high}} \in A_{\text{high}} = A_{\text{clear}} \cup A_{\text{grasp}}$, where $A_{\text{clear}} = \{\textit{push}, \textit{pull}, \textit{move\_to}, \textit{lift}, \textit{lower}\}$ and $A_{\text{grasp}} = \{\textit{grasp}\}$. Clearing actions invoke a geometric motion planner; *grasp* invokes the learned GeoGrasp policy.

B. VLM Planning and MCP Skill Invocation

The planner uses a pretrained VLM (Qwen3-VL-32B-Instruct [17]) and is constrained for reliability by three structured safety constraints: (i) a *closed observation space* (only an RGB image and a fixed object list, preventing hallucinated targets); (ii) *bounded action parameters* (sides in {left,center,right}, push/pull distances clamped, and target poses verified by the IK solver before execution, with infeasible actions rejected and returned as feedback); and (iii) *verifiable execution feedback* (each skill returns a typed status, and the planner cannot advance without one). The VLM emits a structured JSON object with the action name, arguments, and a reason field. To bridge high-level reasoning and low-level control, manipulation primitives are exposed as model-agnostic tools through an MCP server [18]: when the VLM emits a tool call (e.g., *push(side="left")*), the server dispatches it to the corresponding skill executor, decoupling reasoning from hardware and enabling modular skill expansion.

C. Atomic Skills and GeoGrasp

Clearing and recovery. *Push*/*pull* create space by approaching an object from a specified side, with the end-effector aligned to the motion direction; *move*/*lift*/*lower* reposition the hand; recovery primitives reset the arm and hand to safe configurations when singularities or persistent collisions are detected.

GeoGrasp. For target acquisition we use GeoGrasp, a dexterous RL grasping skill that we build on the geometry-aware relative hand-object distance representation of RobustDexGrasp [15] and adapt to our clutter-clearing setting. Rather than adopting an off-the-shelf policy, we re-implement the geometry-aware observation, design a task-specific dense reward for the reach–grasp–lift objective, and wrap the resulting policy as a reusable atomic skill that the planner invokes on demand through the MCP interface. The observation is a 59-dimensional vector that captures local hand-object geometry: unit-normalized nearest-neighbor vectors from 18 hand keypoints to the object point cloud ($\mathbb{R}^{54}$), the target height ($\mathbb{R}^1$), and the end-effector height and orientation ($\mathbb{R}^4$). Grounding the policy in local directional geometry rather than appearance reduces scene overfitting and the sim-to-real gap. The policy is trained with PPO [19] using a task-specific dense reward

$$R_t = R_{\text{lift}} + R_{\text{success}} + R_{\text{contact}} + R_{\text{nn}} - C_{\text{action}}, \qquad (1)$$

which combines a lifting term, a sparse success bonus for raising the object above $0.15\,\text{m}$, a contact-stability term, a nearest-neighbor shaping term that rewards decreasing the hand-object distance, and an action penalty. The policy (a 3-layer MLP) is trained on three objects (Cube, Cup, Apple) in a clutter-free environment; despite this small training set, it generalizes to unseen geometries and transfers to real hardware without fine-tuning (Sec. III).

D. Closed-loop Execution and Failure Detection

The VLM monitors the state via visual feedback: if the target is obstructed it selects a clearing skill, and once the path is clear it invokes *grasp*. Each skill reports a typed status so replanning is grounded in measurable conditions rather than qualitative descriptions. *Grasp slip* is detected from loss of finger-object contact or a drop in the tracked object height after lift-off; *stuck* from end-effector displacement staying below a threshold over consecutive control steps despite a non-zero command; *IK/pathfinding failure* when no valid solution exists for a requested pose; and *success* when the target is lifted above $0.15\,\text{m}$ and held. In simulation these quantities come from the simulator state; on the real platform, object poses come from FoundationPose [20] and proprioception from the arm and hand encoders. On failure, the typed status is returned to the planner, which generates a new strategy (e.g., another clearing direction or a reset) until the target is grasped or a step limit is reached.

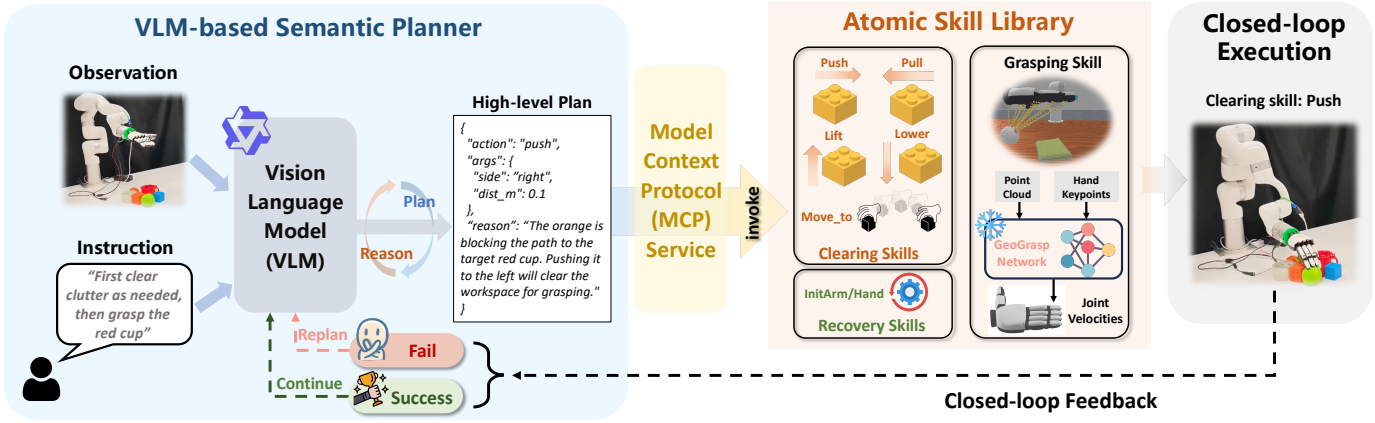

Fig. 2. **Overview of AdaClearGrasp.** A VLM-based semantic planner (left) reasons about the scene and generates high-level plans, which are translated into atomic skills through the MCP service (middle). The skill library contains clearing primitives, recovery mechanisms, and the geometry-aware RL grasping skill GeoGrasp. The system runs in a closed loop (right), monitoring typed feedback to trigger replanning or recovery.

TABLE I
SUCCESS RATES ON CLUTTER-BENCH, ADACLEARGRASP VS. THE VLM SCAFFOLDING BASELINE ACROSS THREE DIFFICULTY TIERS. AVERAGED OVER 10 INDEPENDENT TRIALS PER (OBJECT, CONFIGURATION).

Target Object	Level-1		Level-2		Level-3	
	VLM Scaf.	**Ours**	VLM Scaf.	**Ours**	VLM Scaf.	**Ours**
Cube	0.2	**0.8**	0.0	**0.9**	0.0	**1.0**
Can	0.1	**0.9**	0.0	**1.0**	0.0	**0.9**
Pear	0.0	**1.0**	0.0	**1.0**	0.0	**0.7**
Apple	0.1	**1.0**	0.0	**0.8**	0.0	**0.9**
Mug	0.0	**0.8**	0.0	**0.8**	0.0	**0.9**
Lego	0.0	**1.0**	0.0	**0.7**	0.0	**0.5**
Ball	0.0	**0.7**	0.0	**0.7**	0.0	**0.4**
Average	0.06	**0.89**	0.00	**0.84**	0.00	**0.76**

E. Clutter-Bench

`Clutter-Bench`, built on ManiSkill3, provides a graded, reproducible protocol for goal-conditioned dexterous grasping in clutter. Unlike grasp datasets such as DexGraspNet [8] and UniDexGrasp [7], which evaluate grasp synthesis for largely isolated objects, `Clutter-Bench` evaluates the integrated loop of reasoning, clearing, and grasping under language-specified targets. It includes seven YCB target objects (cube, can, pear, apple, mug, lego, ball) and three difficulty levels defined by obstacle count (2/4/6), with ten pre-generated configurations per (object, level) pair stored as JSON for identical layouts across methods. The level index controls only obstacle count; spatial density and occlusion vary across configurations within a level due to randomized placement.

III. EXPERIMENTS

We evaluate AdaClearGrasp in simulation and the real world, asking: (RQ1) does it outperform a VLM baseline across clutter levels? (RQ2) how do adaptive clearing and closed-loop feedback each contribute? (RQ3) how well does our GeoGrasp skill generalize to unseen object geometries? and (RQ4) does the full system transfer from simulation to real hardware?

Setup. The simulation is built on ManiSkill3 with an xArm7 arm and a dexterous XHand, observed by a front-facing RGB-D camera (128×128). GeoGrasp is trained with PPO for 6M steps. The high-level planner is Qwen3-VL-32B-Instruct. We report Success Rate (SR): a trial succeeds when the target is grasped, lifted above 15 cm, and held for 2 s, all within a budget of 40 planning steps. The real platform replicates this setup; FoundationPose [20] estimates the 6D object poses passed to the planner, and GeoGrasp is transferred directly without any fine-tuning.

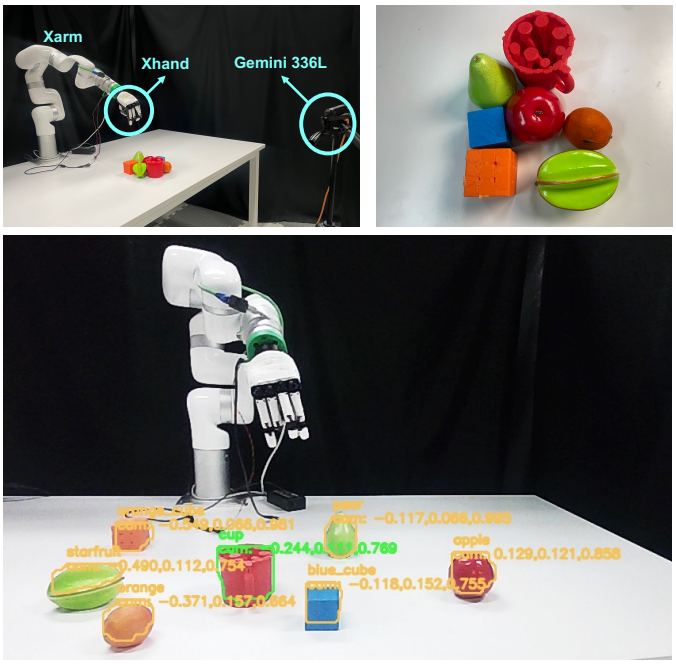

Fig. 3. **Real-world implementation.** (Top) xArm7 with XHand and an external Gemini 336L camera. (Bottom) 6D pose estimation via FoundationPose, which feeds the VLM planner.

Performance on `Clutter-Bench` (RQ1). We compare against *VLM Scaffolding* [14], which predicts keypoints and 3D trajectories with a VLM and is designed for single-object interaction without explicit obstacle handling (Table I). As clutter increases, the baseline reaches only 6% at the sparse Level-1 and 0% at Levels 2–3, because it models manipulation as collision-free waypoints that rarely exist under occlusion. In

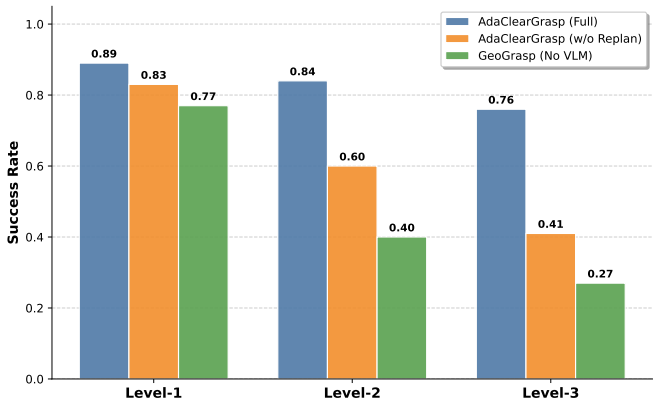

Fig. 4. Ablation on adaptive clearing and closed-loop feedback across the three clutter levels.

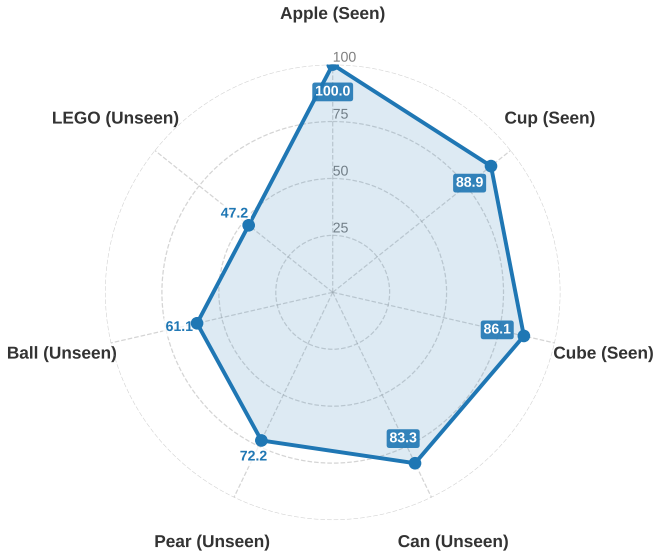

Fig. 5. Per-object grasp success of GeoGrasp on 7 categories (3 seen, 4 unseen) in clutter-free environments.

contrast, AdaClearGrasp sustains **89%**, **84%**, and **76%** across the three levels, and its advantage widens as the scene becomes more constrained, showing that adaptive clearing, rather than a single direct-grasp attempt, is what keeps success high under dense clutter.

Ablation (RQ2). We isolate the two key components (Fig. 4). Removing VLM-guided clearing (*GeoGrasp, No VLM*, i.e., direct grasping) drops SR from 77% at Level-1 to 40% (Level-2) and 27% (Level-3), confirming that in dense scenes the target is often physically inaccessible without first rearranging its surroundings. Removing closed-loop replanning (*w/o Replan*) degrades performance from 83% at Level-1 to 41% at Level-3; re-enabling dynamic replanning (up to five attempts) recovers much of this gap (e.g., +35% at Level-3) by correcting failed grasps or incomplete clearing. The two components are complementary: clearing makes the target reachable, while closed-loop feedback makes execution robust to the errors that clearing itself can introduce.

Generalization of GeoGrasp (RQ3). To probe the grasping skill in isolation, we run clutter-free grasping on 7 categories, 3 seen and 4 unseen (Fig. 5). On the seen objects the policy reaches 100.0% (Apple), 88.9% (Cup), and 86.1% (Cube). Trained on only these three objects, it still generalizes to the unseen *Can* (83.3%) and *Pear* (72.2%), and remains usable on the harder rolling *Ball* (61.1%) and non-convex *LEGO* (47.2%) without fine-tuning. This indicates that grounding the policy in the adapted geometry-aware representation, rather than object appearance, is what drives transfer across shapes.

TABLE II
REAL-WORLD SIM-TO-REAL TRANSFER (SEEN CATEGORIES). APPLE/CUBE/CUP ARE THE GEOGRASP TRAINING CATEGORIES; RESULTS ARE REPORTED AS SIM-TO-REAL TRANSFER.

Target Object	Clutter Level (Obstacles)			Avg.
	2	4	6	
Apple	9/10	7/10	5/10	**70%**
Orange Cube	10/10	8/10	6/10	**80%**
Cup	8/10	6/10	4/10	**60%**
Total Success	27/30	21/30	15/30	**70%**
Success Rate	90%	70%	50%	

Sim-to-real transfer (RQ4). We deploy the full system on the physical platform for 90 trials across 3 target objects, each at 3 clutter levels (Table II, Fig. 3). The three real targets coincide with the GeoGrasp training categories, so this experiment measures sim-to-real transfer on seen categories rather than real-world generalization. Despite the reality gap, the system reaches **70.0%** overall (63/90) without any fine-tuning. SR is **90.0%**, **70.0%**, and **50.0%** at the three clutter levels, mirroring the simulation trend: denser scenes leave smaller error margins, where pose-estimation jitter or slippage during clearing can cause collisions. Accordingly, most failures in dense scenes stemmed from such real-world uncertainties, e.g., the cup being knocked over while clearing an adjacent obstacle.

IV. CONCLUSION AND LIMITATIONS

We presented AdaClearGrasp, a closed-loop hierarchical framework for adaptive clearing and robust dexterous grasping in densely cluttered environments. A VLM reasons over scene semantics and instructions to select among clearing, direct grasping, and recovery skills, while a geometry-aware RL skill (GeoGrasp) handles target acquisition and a closed-loop feedback mechanism triggers replanning on failure. On `Clutter-Bench`, AdaClearGrasp substantially outperforms a VLM-guided baseline, and it transfers to real hardware at 70% success.

Limitations. The high-level decision quality depends on the chosen VLM, which we have not exhaustively compared; the real-world planner relies on FoundationPose, so low-accuracy perception can degrade decisions; `Clutter-Bench` currently spans 7 objects and up to 6 obstacles; and the real targets coincide with the training categories, so real-world evidence is limited to sim-to-real transfer. Strengthening external baselines, analyzing VLM decision quality quantitatively, and testing unseen real objects are natural next steps. We also plan to extend the skill library (e.g., bimanual or tool-use skills) through the same MCP interface.

REFERENCES

[1] A. I. Weinberg, A. Shirizly, O. Azulay, and A. Sintov, "Survey of learning approaches for robotic in-hand manipulation," *arXiv preprint arXiv:2401.07915*, 2024.

[2] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on robot learning*. PMLR, 2022, pp. 297–307.

[3] S. Yuan, L. Shao, C. L. Yako, A. Gruebele, and J. K. Salisbury, "Design and control of roller grasper v2 for in-hand manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9151–9158.

[4] M. R. Dogar and S. S. Srinivasa, "A planning framework for non-prehensile manipulation under clutter and uncertainty," in *Robotics: Science and Systems (RSS)*, 2012.

[5] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[6] L. Xu, Z. Liu, Z. Gui, J. Guo, Z. Jiang, T. Zhang, Z. Xu, C. Gao, and L. Shao, "Dexsingrasp: Learning a unified policy for dexterous object singulation and grasping in densely cluttered environments," *IEEE Robotics and Automation Letters*, vol. 11, no. 2, pp. 1346–1353, 2025.

[7] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, *et al.*, "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4737–4746.

[8] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," *arXiv preprint arXiv:2210.02697*, 2022.

[9] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, "$\mathcal{D}(R,O)$-Grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," *arXiv preprint arXiv:2410.01702*, 2024.

[10] Z. Chen, Q. Yan, Y. Chen, T. Wu, J. Zhang, Z. Ding, J. Li, Y. Yang, and H. Dong, "Clutterdexgrasp: A sim-to-real system for general dexterous grasping in cluttered scenes," *arXiv preprint arXiv:2506.14317*, 2025.

[11] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[12] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[13] H. Liu, S. Guo, P. Mai, J. Cao, H. Li, and J. Ma, "Robodexvlm: Visual language model-enabled task planning and motion control for dexterous robot manipulation," *arXiv preprint arXiv:2503.01616*, 2025.

[14] V. de Bakker, J. Hejna, T. G. W. Lum, O. Celik, A. Taranovic, D. Blessing, G. Neumann, J. Bohg, and D. Sadigh, "Scaffolding dexterous manipulation with vision-language models," *arXiv preprint arXiv:2506.19212*, 2025.

[15] H. Zhang, L. Huang, S. Christen, J. Song, *et al.*, "Robust dexterous grasping of general objects," in *Conference on Robot Learning*. PMLR, 2025, pp. 3035–3050.

[16] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," *arXiv preprint arXiv:2107.14483*, 2021.

[17] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[18] X. Hou, Y. Zhao, S. Wang, and H. Wang, "Model context protocol (mcp): Landscape, security threats, and future research directions," *ACM Transactions on Software Engineering and Methodology*, 2025.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 17 868–17 879.