

ZeroDex: Zero-Shot Long-Horizon Dexterous Manipulation via Multi-View 3D-Grounded VLM Reasoning

Jisoo Kim¹, Sangwon Baik¹, Taeksoo Kim¹, Sungjoo Kim¹, Junyoung Lee¹, Mingi Choi¹,
Kisang Park², Sungjoon Choi², and Hanbyul Joo¹

Abstract—We present ZeroDex, a zero-shot framework for long-horizon dexterous manipulation that grounds language instructions into executable 3D task plans from calibrated multi-view RGB images. Our framework builds on the grounding ability that vision-language models (VLMs) already exhibit in 2D and lifts their predictions into 3D through multi-view fusion, which extends VLM grounding beyond image-space keypoints to the metric 3D quantities that manipulation actually requires: contact points, placement targets, and tool-motion anchors. Because execution is driven by these 3D groundings rather than a learned policy, the system preserves the VLM’s open-world generalization, without collecting demonstrations or post-training for a specific task or embodiment. The VLM decomposes a complex instruction into a sequence of atomic tasks, each executed by a primitive from a small reusable library that covers manipulation tasks including pick-and-place and tool-use: for tool-use, we retrieve an object-centric atomic action corresponding to the inferred skill category and align its stored 6D tool trajectory to the grounded keypoints; for dexterous execution, we expand the lifted grasp keypoint into a task-conditioned grasp affordance region and generate feasible grasp-motion pairs with an arm-hand motion generator. Real-world experiments show improved 3D grounding accuracy and higher task success than zero-shot visual-prompting and constraint-based baselines, both of which ground from a single RGB-D view, and our demonstration-free system outperforms VLA policies fine-tuned on 100 task demonstrations. We further demonstrate long-horizon manipulation, where execution verification and replanning recover from intermediate failures, and show that the system generalizes to unseen objects and tool-use tasks in novel scenes.

Index Terms—Zero-Shot Manipulation, Dexterous Grasping, Multi-view Grounding

I. INTRODUCTION

A long-standing goal in robotics is to build general-purpose systems that perform long-horizon manipulation from high-level language instructions. Beyond recognizing objects, such systems must ground instructions in task-relevant 3D geometry: where to place an object, which part to contact, and how to orient and move a tool during execution. This requirement is especially stringent for dexterous hands, where small 3D grounding errors cause unstable grasps, collisions, inverse-kinematics failures, or contact on the wrong functional region of a tool.

The dominant approach learns this behavior end-to-end. Recent vision-language-action (VLA) models make strong progress toward general-purpose manipulation by predicting actions directly from large-scale robot data [1], [2], [3], [4], [5]. However, achieving reliable performance on diverse

manipulation tasks requires extensive data collection, task-specific adaptation, or environment and embodiment specific fine-tuning, which makes them hard to scale across the diversity of objects, tools, and spatial configurations found in open-ended settings. Human demonstration retargeting offers another route, but the embodiment gap between human and robotic hands produces physically infeasible contacts or unstable grasps, and often demands additional refinement or reinforcement learning before deployment. In this paper, we explore an alternative strategy that builds a modular interface between vision-language reasoning and motion control. Instead of learning a single end-to-end policy, we decouple semantic reasoning, handled by a vision-language model (VLM), from physical execution, handled by motion primitives or controllers. The key observation is that modern VLMs already answer, zero-shot, most of the sub-questions that manipulation requires: what to grasp, where to grasp it as a functional affordance, how to move it, and in what order to carry out the steps of a task. Given this competence, we do not need to train a new policy. Once the VLM produces a plan, we decompose a complex task into a sequence of simple atomic tasks such as pick and move, and we execute each atomic motion with a reliable primitive controller. Because most manipulation tasks can be expressed as compositions of such atomic tasks, a small library of primitives driven by VLM reasoning covers a broad range of tasks. We find that VLMs produce accurate plans and decompositions across a wide range of tasks, as demonstrated in our evaluations.

This modular design, however, hinges on a geometric requirement that 2D reasoning cannot satisfy on its own, as the relevant quantities are inherently three-dimensional. Where to grasp must be specified in 3D rather than 2D, and more importantly, how to move the end effector and where to move an object are 3D trajectory quantities. A single view rarely carries enough geometric information to reason about these 3D trajectories reliably, so attaching a VLM to a single image is limited for this purpose. Our central idea is to fuse VLM grounding across multiple views. Multi-view fusion lifts view-dependent 2D predictions into consistent 3D quantities, and it extends VLM-based grounding to a far wider set of problems than single-view reasoning can reach, including 3D contact points, placement targets, and tool-motion anchors. Within this framework we triangulate VLM groundings across views to recover 3D positions and directions. Since VLM 2D groundings are semantic and view-dependent, we take the 3D candidate with the widest

¹Seoul National University, Seoul, Republic of Korea.

²Korea University, Seoul, Republic of Korea.

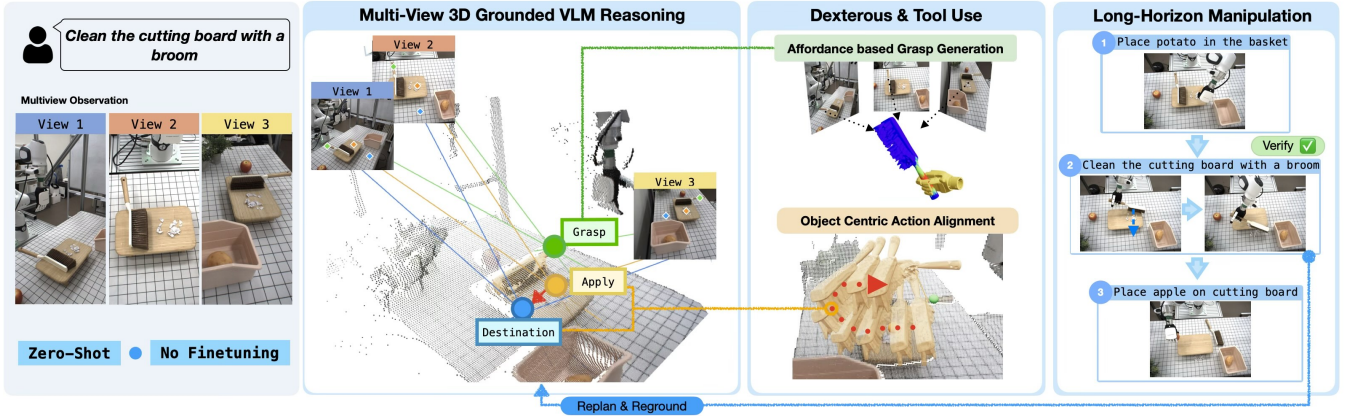


Fig. 1. **ZeroDex**. Given a language instruction and calibrated multi-view observations, our framework uses multi-view VLM grounding with robust triangulation to infer task-relevant 3D groundings, generates affordance-aware dexterous grasps, and executes pick-and-place or tool-use plans through reusable action primitives.

cross-view agreement rather than averaging, which keeps grounding reliable under occlusion and partial visibility.

The lifted keypoints are by themselves sufficient for most pick-and-place tasks, where a grasp point, a transport waypoint, and a placement target fully specify the motion. Tasks such as pouring or sweeping instead depend on how the tool moves relative to the target over time, a 6D trajectory that a few keypoints cannot express; we obtain it by reusing short, environment-independent object trajectories and aligning them to the grounded geometry. We represent these tool-use behaviors as a *Bag of Atomic Actions*, a library of short 6D object trajectories indexed by interaction type. For a new scene, the appropriate primitive is retrieved and aligned to the grounded task geometry. To support dexterous-hand execution, we apply the same multi-view grounding to estimate functional contact regions, generate candidate grasps on those regions, and filter them by inverse-kinematics and collision feasibility over the full tool-use trajectory. For long-horizon tasks, the VLM verifies the outcome of each executed subtask and maintains a textual memory of the overall scenario and of what has been executed so far; together these let the system re-ground, retry, or replan the remaining subtasks after execution failures or scene changes.

We instantiate these ideas in ZeroDex, a single zero-shot system, and evaluate it on a diverse set of real-world dexterous manipulation tasks. Our main contributions are as follows. First, we present ZeroDex, a framework for long-horizon dexterous manipulation that couples VLM-based task planning and multi-view 3D grounding with a library of reusable atomic action primitives, enabling zero-shot execution in unseen scenes. Second, we propose a multi-view VLM grounding approach that fuses view-dependent 2D predictions into reliable 3D quantities through consensus-based cross-view triangulation, remaining robust under occlusion and partial visibility. Third, we show that the same multi-view grounding serves both halves of execution: the lifted interaction and terminal keypoints provide the correspondences that align a stored 6D tool trajectory to the current scene, while the lifted grasp keypoint is expanded into a functional

3D affordance region for grasp generation, with candidates filtered by trajectory-level inverse-kinematics and collision feasibility. Finally, we validate the complete system on a diverse set of real-world manipulation tasks, where multi-view fusion yields over $3\times$ more repeatable 3D grounding than the single-view RGB-D lifting of MOKA [6], and higher success than zero-shot visual-prompting and constraint-based baselines, while our demonstration-free system outperforms VLA policies fine-tuned on 100 task-specific demonstrations.

II. RELATED WORK

Vision-Language-Action Models for Manipulation Vision-language-action (VLA) models map images and instructions to robot actions, often by adapting vision-language backbones [7], [8], [9] with large robot datasets [10], [11], [12], [13], [14]. Beyond foundational robot policies [1], [2], [4], [3], [5], recent work improves action modeling [5], [15], fine-tuning efficiency [16], spatial grounding [17], [18], open-world generalization [19], and dexterous or humanoid embodiments [20], [15], [21], [22], [23]. Despite strong in-distribution performance, such policies, including recent variants that add reasoning or 3D structure inside the policy [24], [25], rely on large robot data and still require additional data or adaptation to cover new objects, tools, and embodiments. Retargeting human demonstrations [26], [27], [28] reduces collection cost, but the embodiment gap can yield infeasible contacts that require further refinement. We instead act zero-shot in unseen scenes, without task-specific demonstrations or weight updates.

Zero-Shot Manipulation with Foundation Models Another line drives manipulation directly with pretrained foundation models, without task-specific training, and differs mainly in the intermediate representation it extracts. Code-generation methods prompt large language models to produce high-level plans or executable control code over perception and motion APIs [29], [30], [31]. Visual-prompting methods query vision-language models to mark the task on images as keypoints, visual marks, or affordances [6], [32], [33], [34], [35]. Zero-shot orchestrators [36], [37] use a VLM

to sequence and verify skills, yet delegate execution to learned policies and typically ground from a single view. These methods share our zero-shot, training-free motivation, but many still rely on image-space or sparse intermediate representations that are brittle for the task-relevant 3D geometry required for dexterous manipulation, including contact points, placement targets, and tool trajectories. This gap motivates grounding tasks directly in 3D.

3D Grounding and Dexterous Execution Lifting 2D observations to 3D is classically studied in multi-view stereo [38]; for manipulation, recent work gives VLMs spatial reasoning [39], [40], aggregates views into 3D groundings [41], or casts VLM reasoning as 3D constraints [31], [32], [42], often aided by depth or pose foundation models [43], [44]. For execution, prior work transfers object-centric skills or functional correspondences [45], [46], synthesizes dexterous grasps via samplers, generative models, or affordance conditioning [47], [48], [49], [50], and uses language or VLM feedback for verification and replanning [51]. Unlike prior systems that treat 3D grounding, tool motion, and dexterous grasping as separate components, we fuse multi-view VLM groundings into explicit metric 3D quantities, align object-centric 6D tool trajectories to them, and verify collision and kinematic feasibility before execution [52], [53], all without training.

III. METHOD

Given calibrated multi-view RGB images $\mathcal{I} = \{\mathbf{I}_v\}_{v=1}^M$ and a subtask $\mathbf{l}$ (e.g., “Pour from a kettle”), ZeroDex returns an arm–hand execution trajectory $\mathcal{T}_{\text{robot}}$ and its physical feasibility $\eta \in \{0, 1\}$:

$$(\mathcal{T}_{\text{robot}}, \eta) = \mathcal{Z}(\mathcal{I}, \mathbf{l}). \quad (1)$$

Our central representation is a desired object trajectory $\mathcal{T}_{\text{obj}}$ specifying what the manipulated object should do (Section III-C). Its spatial constraints come from reference-view primitives $\mathcal{Q}_r$, which associate action types with relevant image locations (Section III-A), and their metric world-frame keypoints $\mathcal{X}$ obtained through multi-view lifting (Section III-B). Grasp and motion generation determines how the arm and hand can perform this object trajectory (Section III-D); execution proceeds only if $\eta = 1$. The same executor supports long-horizon tasks through planning and execution verification (Section III-E). Figure 1 shows our overall pipeline.

A. Reference-View Semantic Grounding

We represent a subtask as an ordered sequence of action primitives, each specifying an action type and the image locations needed to perform it. Semantic descriptions identify what these locations refer to, providing correspondence cues for subsequent 3D lifting. In a selected reference view $\mathbf{I}_r$, this representation is

$$\begin{aligned} \mathcal{Q}_r &= \{(m^t, \mathcal{P}_r^t)\}_{t=1}^T, \\ \mathcal{P}_r^t &= \{(\mathbf{p}_r^{t,j}, \mathbf{d}^{t,j})\}_{j=1}^{N_t}. \end{aligned} \quad (2)$$

Here, T is the number of primitives, $m^t \in \{\textit{grasp}, \textit{apply_action}, \textit{waypoint}, \textit{release}, \textit{hold}\}$ is the primitive type, and $\mathcal{P}_r^t$ contains N_t 2D keypoints $\mathbf{p}_r^{t,j} \in \mathbb{R}^2$ paired with text descriptions $\mathbf{d}^{t,j}$.

Pick-and-place uses *grasp*, *waypoint*, and *release* to specify a grasp point, a transport waypoint, and a placement location. Tool use employs *grasp*, *apply_action*, and a terminal *release* or *hold*. Its grasp step identifies both a grasp point and a functional tip on the tool (e.g., a broom head or kettle spout); the action step identifies an interaction point on the target; and the terminal step specifies the final location and whether to release or maintain the grasp.

To construct this representation, the VLM Φ selects the reference view and identifies the task entities:

$$(r, z, g_z) = \Phi(\mathcal{I}, \mathbf{l}), \quad (3)$$

where $r \in \{1, \dots, M\}$ is the view index and $z \in \{\textit{pick}, \textit{tool}\}$ selects the manipulation mode. The grounding tuple is $g_{\text{pick}} = (O_{\text{tar}}, \mathbf{p}_{\text{dst}})$ for pick-and-place and $g_{\text{tool}} = (O_{\text{tool}}, c, O_{\text{tar}}, \mathbf{p}_{\text{dst}})$ for tool use. Here, O_{tar} is the target object, $\mathbf{p}_{\text{dst}}$ is its destination pixel in $\mathbf{I}_r$, and O_{tool} and c specify the tool object and skill category. If the target remains stationary, $\mathbf{p}_{\text{dst}} = \text{NULL}$. A planning prompt $\mathbf{l}'$ combines $\mathbf{l}$ and g_z to produce

$$\mathcal{Q}_r = \Phi(\mathbf{I}_r, \mathbf{l}'). \quad (4)$$

We denote the hand-manipulated object by $O_{\text{man}} \in \{O_{\text{tar}}, O_{\text{tool}}\}$: it is O_{tar} for pick-and-place and O_{tool} for tool use.

B. Multi-View Fusion-Based 3D Lifting

The semantic plan identifies relevant image locations but does not determine their metric depth. We recover world-frame 3D keypoints $\mathcal{X} = \{\mathbf{X}_{*}^{t,j}\}_{t,j}$ by seeking geometric agreement across calibrated views: a candidate 3D point should reproject near the independently grounded 2D locations in multiple images. We omit the per-keypoint indices (t, j) below and let π_v denote calibrated world-to-image projection.

A grounding prompt $\mathbf{l}''$ formed from $\mathbf{l}$, m^t , and $\mathbf{d}^{t,j}$ yields predictions $\bar{\mathbf{p}}_v = \Phi(\mathbf{I}_v, \mathbf{l}'')$ in each view. Triangulating each view pair $a < b$ gives a world-frame 3D keypoint candidate $\mathbf{X}_{a,b} = \text{Triangulate}(\bar{\mathbf{p}}_a, \bar{\mathbf{p}}_b)$. Its RANSAC [54]-style consensus score counts the views whose reprojection errors fall within a pixel threshold ϵ_{tri} :

$$S_{\text{tri}}(\mathbf{X}) = \sum_{v=1}^M \mathbf{1}(\|\pi_v(\mathbf{X}) - \bar{\mathbf{p}}_v\|_2 \leq \epsilon_{\text{tri}}), \quad (5)$$

where $\mathbf{1}(\cdot)$ is an indicator function. The candidate with the largest support provides the lifted keypoint:

$$(a^*, b^*) = \arg \max_{a < b} S_{\text{tri}}(\mathbf{X}_{a,b}), \quad (6)$$

$$\mathbf{X}_{*} = \mathbf{X}_{a^*, b^*}.$$

Calibrated camera extrinsics are incorporated into triangulation, so all lifted keypoints share the world frame.

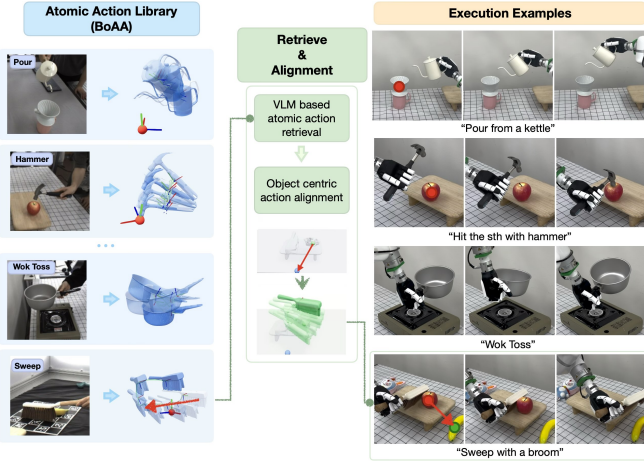


Fig. 2. **Bag of Atomic Actions.** Examples of object-centric atomic actions executed on the real robot.

C. Object Trajectory Construction

The lifted keypoints specify where manipulation should occur; the desired 6D trajectory $\mathcal{T}_{\text{obj}}$ specifies how O_{man} moves between these locations. For pick-and-place, it connects the current object pose to the placement location through the transport waypoint while preserving the object's initial orientation. Tool use additionally requires a motion pattern describing how the tool acts on the target. Both use placement or terminal locations refined before trajectory construction.

a) Tool use via atomic action alignment.: Our *Bag of Atomic Actions* supplies reusable object-centric motion patterns for skills such as pouring and sweeping. Each entry $\mathcal{A} = (c, \mathcal{T}, \mathbf{X}^s, \mathbf{X}^e)$ stores a skill category c , a sequence $\mathcal{T} = (T_i)_{i=0}^{N_a}$ of tool poses $T_i \in SE(3)$, and 3D start/end anchors $\mathbf{X}^s, \mathbf{X}^e \in \mathbb{R}^3$. We build the library offline from recorded demonstrations and trajectories generated using VLMPose [45] (Figure 2).

At test time, we retrieve an entry matching c . The stored start anchor $\mathbf{X}^s$ corresponds to the lifted *apply* action point $\mathbf{X}^{\text{app}}$, and the end anchor $\mathbf{X}^e$ to the refined *release* or *hold* location $\mathbf{X}^{\text{term}}$. A trajectory alignment operator f_{align} adapts the stored motion using these correspondences, with the alignment rotation restricted to a predefined axis:

$$\mathcal{T}_{\text{obj}} = f_{\text{align}}(\mathcal{T}; (\mathbf{X}^s, \mathbf{X}^e), (\mathbf{X}^{\text{app}}, \mathbf{X}^{\text{term}})). \quad (7)$$

b) Collision-aware endpoint refinement.: Placement and terminal keypoints must leave room for the manipulated object at its final pose. For a lifted location $\mathbf{X}^{\text{loc}} \in \mathbb{R}^3$, we form a local grid of candidate positions $\mathcal{G}(\mathbf{X}^{\text{loc}})$. With orientation fixed, we evaluate the object's environment penetration depth $\phi(\mathbf{X}') \in \mathbb{R}$ at each candidate $\mathbf{X}' \in \mathcal{G}(\mathbf{X}^{\text{loc}})$ and select the nearest position with zero penetration:

$$\mathbf{X}_*^{\text{loc}} = \arg \min_{\substack{\mathbf{X}' \in \mathcal{G}(\mathbf{X}^{\text{loc}}) \\ \phi(\mathbf{X}') = 0}} \|\mathbf{X}' - \mathbf{X}^{\text{loc}}\|_2. \quad (8)$$

If no valid position exists, grounding fails. This refinement checks the object at the endpoint; feasibility along the robot's

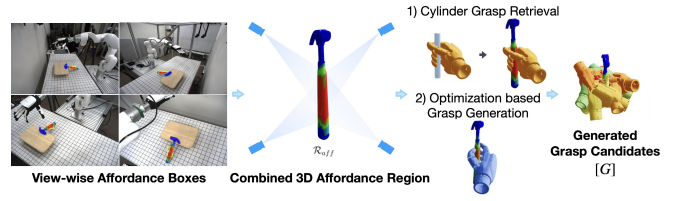


Fig. 3. **Affordance grounding and grasp generation.** Multi-view affordance bounding boxes are combined to identify a 3D affordance region.

motion is addressed in Section III-D.

D. Dexterous Grasp and Motion Generation

Executing $\mathcal{T}_{\text{obj}}$ requires a stable grasp and a feasible arm-hand trajectory. The lifted grasp keypoint $\mathbf{X}^{\text{grasp}} \in \mathcal{X}$ identifies a task-relevant object part; we use it to locate a surface region on which the hand can establish contact, then test whether grasps from this region permit the desired motion.

In each view, an affordance prompt $\mathbf{l}'''$ combines $\mathbf{l}$, the projection $\pi_v(\mathbf{X}^{\text{grasp}})$, and its semantic description. The VLM predicts a graspable box $\mathbf{B}_v = \Phi(\mathbf{l}_v, \mathbf{l}''') \in \mathbb{R}^{2 \times 2}$ on O_{man} . For each world-frame vertex $\mathbf{q}_i \in \mathbb{R}^3$ of the object mesh of O_{man} , we compute the fraction of views whose boxes contain its projection and retain vertices above a threshold τ_{aff} :

$$s(\mathbf{q}_i) = \frac{1}{M} \sum_{v=1}^M \mathbf{1}(\pi_v(\mathbf{q}_i) \in \mathbf{B}_v), \quad (9)$$

$$\mathcal{R}_{\text{aff}} = \{\mathbf{q}_i \mid s(\mathbf{q}_i) \geq \tau_{\text{aff}}\}.$$

We generate grasps in $\mathcal{R}_{\text{aff}}$ using cylindrical templates for handle-like affordances and an optimization-based generator [48] for general geometries. Physics simulation tests their stability under external forces along, and torques about, each coordinate axis. Stable candidates form the grasp candidate set G .

For each grasp $\gamma \in G$, an arm-hand motion generator [52], [53] f_{motion} seeks a robot trajectory that carries O_{man} along $\mathcal{T}_{\text{obj}}$ while satisfying kinematic and collision constraints throughout the motion:

$$(\mathcal{T}_{\text{robot}}, \eta) = f_{\text{motion}}(\mathcal{T}_{\text{obj}}, \gamma). \quad (10)$$

We execute a trajectory with $\eta = 1$, following the planned grasp and terminal release/hold behavior. If no grasp yields a feasible trajectory, the module reports failure. Tool use follows the same procedure, with affordances computed on the tool mesh.

E. Extension to Long-Horizon Tasks

We extend ZeroDex to long-horizon tasks through a planning and verification loop. The VLM Φ manages the remaining subtasks using current observations and a textual memory of prior attempts. Given a high-level instruction $\mathbf{l}_{\text{task}}$ and initial observations $\mathcal{I}_0$, Φ initializes an ordered subtask queue $\mathcal{L}^0$ and memory h_0 . At attempt k , we select a pending subtask $\mathbf{l}^k$ from $\mathcal{L}^k$ and invoke $(\mathcal{T}_{\text{robot}}^k, \eta_k) = \mathcal{Z}(\mathcal{I}_k, \mathbf{l}^k)$.

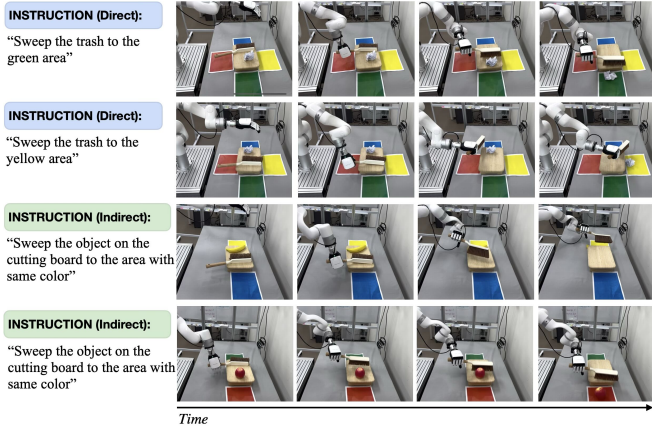


Fig. 4. **Qualitative Results.** We evaluate both direct and indirect styles of instructions and to successful 3D grounding across diverse environments.

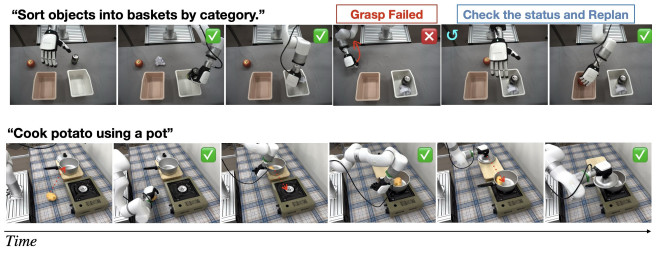


Fig. 5. **Qualitative Results.** Long-horizon manipulation examples. The shown scenarios consist of multiple subtasks. In the example above, the grasp fails, and the VLM detects the failure state and replans the next action.

The generated trajectory is executed only if $\eta_k = 1$. After the attempt, we acquire new observations $\mathcal{I}_{k+1}$, which allow Φ to assess the resulting scene and update the memory and remaining plan:

$$(\alpha_k, h_{k+1}, \mathcal{L}^{k+1}) = \Phi(\mathcal{I}_{k+1}, \mathbf{l}_{\text{task}}, \mathcal{L}^k, h_k). \quad (11)$$

The decision α_k selects *continue*, *retry*, *replan*, or *done*. The updated queue guides subsequent calls to $\mathcal{Z}$, which grounds retried or revised subtasks again in the current scene. This loop supports recovery from intermediate failures, object displacement, and newly observed scene changes (Figure 5 and supplementary video).

IV. EXPERIMENTS

We evaluate our framework on zero-shot robot manipulation in a real-world tabletop setting, assessing its scalability from simple tasks to long-horizon scenarios. Our evaluation covers four key capabilities: (1) target grounding amidst distractors and collision robustness (e.g., placing inferred object into a basket), (2) spatial-relation reasoning (e.g., placing a pot on a stove), (3) affordance-aware tool use (e.g., sweeping objects with a broom), and (4) long-horizon sequencing (e.g., cooking and organizing objects).

A. Implementation Details

Our framework is robot-agnostic apart from the final arm-hand motion generation, and we use two platforms: a Franka arm and an xArm, each equipped with an Inspire

dexterous hand. All comparisons against zero-shot grounding baselines (Table I, Table II) are conducted on the Franka. The comparison against fine-tuned VLA policies is conducted on the xArm, because our teleoperation rig for demonstration collection is built on that arm; our method and both VLA baselines are evaluated on the same xArm platform, scene list, and success criterion, so the comparison remains matched. Qualitative results use both arms. The tabletop environment is observed by four calibrated RGB cameras, including a stereo pair, and all experiments use this four-view configuration. We use FoundationStereo [43] for stereo depth estimation and FoundationPose [44] for multi-object 6D pose estimation. All VLM queries, including reference-frame grounding, view-wise grounding, affordance prediction, and execution verification, use Gemini Robotics-ER 1.6 with medium thinking level. Arm-hand motions are produced by an arm-hand motion generator [52], [53]. The VLM prompts are provided in the supplementary video.

B. Baselines

We compare our zero-shot framework against three families of baselines. (i) **MOKA** [6]. A zero-shot visual-prompting method whose keypoint-and-motion-primitive representation most closely matches ours. We follow its formulation, deprojecting object-surface keypoints from a single-view RGB-D depth map and letting the VLM select the height of the free-space waypoints. MOKA uses two options here (the same height as the target point, or above it) and leaves the actual offset unspecified; we instead provide a small set of discrete offsets so that taller objects and container rims can be cleared, which favors the baseline. Its grasp keypoint is expanded into a fixed-radius contact region and fed to the same affordance-based grasp generator, so the two methods differ only in grounding. (ii) **ReKep** [32]. A zero-shot method that expresses a task as relational constraints over keypoints deprojected from the same single RGB-D view, and solves for end-effector poses that satisfy them. We use the authors’ released implementation for keypoint proposal and constraint generation, and route the resulting target poses through the same grasp and arm-hand execution stack as the other methods, so that ReKep and ours differ only in how the task is grounded in 3D. For other Zero-Shot Method, We exclude baselines that cannot be placed on the same execution stack or that require additional policy training. (iii) **VLA models**. GR00T [20] and Being-H0 [22]. We collect 100 teleoperated pick-and-place demonstrations (about one hour of trajectories) in which the target object identity is fixed while its position, the surrounding objects, and the placement vary. We post-train both models on this same set using their official configurations ($H=16$, batch 64, LR 10^{-4} , 30k steps, $4 \times A6000$; relative end-effector arm actions, 6-DoF absolute hand actions, two input views for GR00T, one for Being-H0, synchronous inference at 30 Hz). Our method uses no robot demonstrations or weight updates.

TABLE I

REAL-ROBOT SUCCESS RATES. N IS THE NUMBER OF TASK-SPECIFIC TELEOPERATION DEMONSTRATIONS FOR FINE-TUNING; FOR THE FINE-TUNED VLA BASELINES WE ADDITIONALLY REPORT STAGE-WISE SUCCESS.

Method	N	Standard ($n=10$)	Cluttered & Precise ($n=10$)	All ($n=20$)
<i>Zero-shot grounding</i>				
MOKA [6]	0	0.60	0.20	0.40
ReKep [32]	0	0.50	0.60	0.55
Ours	0	0.90	0.80	0.85
<i>Fine-tuned VLA</i>				
		Throw Away Trash ($n=20$)		
		<i>Approach</i>	<i>Grasp</i>	<i>Success</i>
GR00T [20]	100	0.95	0.35	0.30
Being-H0 [22]	100	1.00	0.05	0.00
Ours	0	1.00	0.95	0.95

C. Metrics

Success Rate. A trial is successful if the outcome required by the language instruction is achieved.

Grounding Quality. We assess groundings along two axes: reproducibility and physical validity. For reproducibility, we repeat the pipeline on identical images and compute the RMS deviation σ of the lifted 3D grasp keypoint about its centroid, reporting σ averaged over scenes together with the worst value $\delta_{\max}$ (mm). For physical validity, we pose the manipulated object at each predicted (unrefined) placement or tool waypoints and evaluate the environment penetration depth ϕ_m of Equation (8). For each scene, we compute the worst-case ($\mathcal{P}_{\max}^o$) and mean ($\mathcal{P}_{\text{avg}}^o$) penetration depth (mm) over its waypoints, together with the collision rate Coll%, the fraction of those waypoints exceeding 1 mm penetration. Table entries report these per-scene values averaged over scenes.

Grasp Feasibility. For grasp candidates generated from a grounding, IK% is the fraction for which collision-aware inverse kinematics succeeds both at the grasp pose and at the grounded waypoint with the object attached, and the clear rate is the fraction that is reachable and collision-free, i.e., with hand-scene penetration below the 5 mm tolerance of the sphere-based hand collision model used by the motion generator.

Long-Horizon Success Rate. For sequential tasks, a trial is considered successful only if all required steps are completed in the correct order. We report both the number of trials and the success rate. Unlike the single-step success rate, which is counted on the first attempt, we report long-horizon success both without retry and with a budget of two attempts per subtask, i.e., a single retry.

D. Results

Real-robot success. Table I compares our method with the zero-shot grounding baselines on real-robot tasks. The standard condition uses uncluttered scenes with well-separated objects and no tall obstacles; the cluttered-and-precise condition adds surrounding distractors and raised, walled targets such as a basket. Our method reaches 0.85 overall, against 0.40 for MOKA and 0.55 for ReKep, and leads on both

TABLE II

WHERE ZERO-SHOT METHODS FAIL. FAILURE COUNTS BY STAGE; SUCCESS REPEATS TABLE I. SOLVER IS REKEP’S CONSTRAINT-SOLVING STAGE, WHICH THE OTHERS LACK.

Method	Success ($n=20$)	Ground	Solver	Grasp	Exec.
MOKA [6]	0.40	6	–	3	3
ReKep [32]	0.55	7	2	0	0
Ours	0.85	0	–	3	0

TABLE III

EFFECT OF AFFORDANCE CONDITIONING. SUCC. IS REAL-ROBOT SUCCESS OVER 10 TRIALS. #G: MEAN GRASP CANDIDATES PER SCENE; CLEAR% AND IK% (SEC. IV-C) ARE POOLED OVER ALL CANDIDATES IN OFFLINE.

Affordance	Tool-use(5 scenes)				Pick-and-place(10 scenes)			
	#G	Clear%	IK%	Succ.	#G	Clear%	IK%	Succ.
off	22	16.7	16.7	2/10	69	23.4	5.6	9/10
on	16	96.2	35.0	7/10	27	96.0	37.2	9/10

conditions: 0.90 on the standard condition, where MOKA and ReKep reach 0.60 and 0.50. The advantage is most pronounced in cluttered scenes and tasks requiring precise placement, where multi-view 3D grounding localizes more reliably under occlusion and resolves the depth ambiguity of lifting 2D predictions to 3D: on the cluttered-and-precise condition our method achieves 0.80, compared to 0.20 for MOKA and 0.60 for ReKep. Table II localizes the failures under the shared execution stack: six of MOKA’s twelve failures and seven of ReKep’s nine occur at grounding, whereas none of ours do. Grounding is the largest failure mode for both baselines, and more so for the stronger one: ReKep succeeds more often overall than MOKA yet attributes a larger share of its failures to grounding. Single-view depth lifting is brittle here: depth is discontinuous at object boundaries, so a few pixels of 2D error at the rim of a raised container drop the lifted point to the table behind it, placing the target in collision or out of reach. More fundamentally, lifting 2D groundings into 3D by deprojecting them and leaving the residual depth ambiguity to VLM reasoning is imperfect by nature. Table I further shows that both VLA baselines, fine-tuned on 100 task-specific demonstrations, remain far behind: Being-H0 fails in all trials and GR00T reaches 0.30, against 0.95 for our zero-shot system. The stage-wise breakdown locates the gap at contact rather than perception: both policies approach the target in nearly every trial (0.95 and 1.00) but secure a stable grasp in only 0.35 and 0.05 of them, underscoring how slowly VLAs adapt to a new dexterous embodiment and the value of a demonstration-free approach.

3D grounding quality. Table IV evaluates grounding repeatability and physical validity against MOKA.¹ Our multi-view fusion is about $3\times$ more repeatable than MOKA’s single-view RGB-D lifting (σ of 2.5 vs. 8.1 mm; worst case

¹ReKep is excluded from these metrics: its target comes from its constraint solver and is often specified as a range (e.g., 5–10 cm above the keypoint), so these metrics would score clearance rather than grounding; see Tables I and II.

TABLE IV

3D GROUNDING QUALITY VS. MOKA [6] ON THE SAME 15 SCENES.
ALL LENGTHS IN MM; METRICS ARE DEFINED IN SEC. IV-C.

Grasp repeatability (10 scenes $\times$ 8 runs)			Method	$\mathcal{P}_{\max}^o \downarrow$	$\mathcal{P}_{\text{avg}}^o \downarrow$	Coll% $\downarrow$	IK% $\uparrow$
			<i>Tool</i> (5 scenes)				
			MOKA [6]	20.4	20.4	20	30.3
			Ours	20.3	4.7	5	35.0
			<i>P&P</i> (10 scenes)				
			MOKA [6]	37.2	36.3	100	24.2
			Ours	32.6	16.3	17	37.2
Method	$\sigma \downarrow$	$\delta_{\max} \downarrow$					
MOKA [6]	8.1	64.9					
Ours	2.5	3.9					

TABLE V

LONG-HORIZON SUCCESS PER SUB-STEP AND END-TO-END. MAIN
NUMBERS ALLOW ONE RETRY PER SUBTASK; NUMBERS IN
PARENTHESES ARE WITHOUT RETRY.

Task	Step 1	Step 2	Step 3	End-to-End
Organize Objects	10/10	10/10 (7/10)	8/10 (6/7)	8/10 (6/10)
Cooking (with tool-usage)	9/10 (8/10)	9/9 (7/8)	9/9 (6/7)	9/10 (6/10)

3.9 vs. 64.9 mm) and yields lower object penetration and collision rates in both tool-use and pick-and-place (Coll% of 5 vs. 20 and 17 vs. 100; $\mathcal{P}_{\text{avg}}^o$ of 4.7 vs. 20.4 and 16.3 vs. 36.3 mm). It also leaves more executable grasp candidates under the same collision-aware IK check: IK% rises from 30.3 to 35.0 on tool-use and from 24.2 to 37.2 on pick-and-place. Collision-aware refinement (Equation (8)) further raises motion-planning success from 7/10 to 10/10 over ten scenes by rescuing placements on raised surfaces (e.g., a cutting board), where the unrefined grounding puts the object inside the board instead of on top of it.

Affordance conditioning. Table III ablates the multi-view affordance region of Section III-D. Conditioning grasp generation on the affordance region raises the clear rate from 16.7% to 96.2% (tool-use) and from 23.4% to 96.0% (pick-and-place), and IK feasibility from 16.7% to 35.0% and from 5.6% to 37.2%, respectively. On the tool-use tasks this translates into task success (2/10 to 7/10), whereas pick-and-place is already saturated (9/10 in both settings). Multi-view grounding and affordance conditioning thus act on two independent axes: the former determines where objects are placed, while the latter determines whether the resulting grasps are reachable and collision-free.

Long-horizon manipulation. Table V reports per-step and end-to-end success rates for long-horizon tasks. With a single retry per subtask, end-to-end success is 8/10 on Organize Objects and 9/10 on Cooking, against 6/10 for both without retry. Task Grounding errors and transient grasp failures are recovered by execution verification followed by re-grounding, retry, or replanning from the plan memory (Section III-E), whereas arm joint limits and environment collisions remain the main unrecovered failures.

Runtime. Multi-view grounding takes 19.4 s per pass: 6.7 s for reference-view grounding, 9.6 s for the four per-view queries, and 3.1 s for triangulation and the associated overhead. Arm-hand motion generation then spends 4.8 s

on scene construction and planner initialization, followed by 9.4 s of planning that covers grasp batch IK, collision checking, collision-aware endpoint refinement, and trajectory computation. For dexterous grasp generation, we validate every candidate in physics simulation before execution to select more reliable grasps, so its cost grows with the number of candidates requested rather than with scene complexity, and we therefore do not report it as a single number.

Extension to Non-Prehensile Manipulation. The same pipeline also covers non-prehensile behaviors such as pressing a button or pushing a drawer, which involve no grasp: we convert the grounded 3D primitive into a hand trajectory rather than an object trajectory. The corresponding atomic action stores a short sequence of hand poses instead of object poses and is aligned to the lifted primitives exactly as in the tool-use case, while execution bypasses grasp generation entirely and sends the aligned hand trajectory directly to the arm-hand motion generator. Qualitative examples are provided in the supplementary video.

V. DISCUSSION

We present ZeroDex, a zero-shot long-horizon manipulation framework that bridges VLM reasoning with physical execution via multi-view 3D grounding. By decomposing language instructions into sequences of 3D-grounded manipulation primitives, our system seamlessly supports both standard pick-and-place and complex tool-use tasks by spatially aligning object-centric atomic actions to the target scene. Experimental results demonstrate that our multi-view fusion strategy outperforms visual-prompting and constraint-based baselines in grounding repeatability, spatial accuracy, and robustness against occlusion, and that our demonstration-free system exceeds VLA policies fine-tuned on task demonstrations. Furthermore, the primitive-level formulation naturally supports execution verification with re-grounding and replanning from an explicit plan memory, allowing the system to verify task progress and recover from intermediate failures during long-horizon tasks.

Limitations and Future Work. Despite its capabilities, our framework exhibits a few limitations. First, the system’s performance is inherently bounded by the reasoning reliability of the underlying 2D VLM. Although multi-view lifting and local collision-aware refinement improve the geometric consistency of grounded predictions, errors in task decomposition, affordance selection, or semantic grounding in 2D can still propagate to downstream execution failures. Second, the physical execution relies on external arm-hand motion generators, so kinematic singularities, collision-checker timeouts, or unstable grasp executions can still induce failures. Third, tool-use motions that are absent from the Bag of Atomic Actions cannot be executed directly. The library is, however, an expandable cache rather than a fixed vocabulary: new skills are added by recording object-centric trajectories from human demonstrations or by synthesizing them with VLMPose [45], and the same mechanism supplies the hand-level primitives used for non-prehensile behaviors, whereas deformable objects, whose motion is not representable as

a rigid 6D trajectory, remain future work. Fourth, our current formulation focuses on object-centric manipulation and tool use, and does not explicitly support dexterous in-hand manipulation, such as rotating an object within the hand or operating a held tool such as scissors. Future work will explore integrating native 3D vision-language models and reactive low-level policies to further enhance the speed and robustness of zero-shot dexterous manipulation.

REFERENCES

- [1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [2] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *CoRL*, 2023.
- [3] 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 preprint arXiv:2406.09246*, 2024.
- [4] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.
- [5] 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 preprint arXiv:2410.24164*, 2024.
- [6] F. Liu, K. Fang, P. Abbeel, and S. Levine, “Moka: Open-world robotic manipulation through mark-based visual prompting,” *arXiv preprint arXiv:2403.03174*, 2024.
- [7] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschanen, E. Bugliarello *et al.*, “Paligemma: A versatile 3b vlm for transfer,” *arXiv preprint arXiv:2407.07726*, 2024.
- [8] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu *et al.*, “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” in *CVPR*, 2024.
- [9] 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.
- [10] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, “Bridge data: Boosting generalization of robotic skills with cross-domain datasets,” *arXiv preprint arXiv:2109.13396*, 2021.
- [11] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, “Bridgedata v2: A dataset for robot learning at scale,” in *CoRL*, 2023.
- [12] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandekar, A. Jain *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *ICRA*, 2024.
- [13] 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.
- [14] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang *et al.*, “Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems,” *arXiv preprint arXiv:2503.06669*, 2025.
- [15] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng, “Dexvla: Vision-language model with plug-in diffusion expert for general robot control,” *arXiv preprint arXiv:2502.05855*, 2025.
- [16] M. J. Kim, C. Finn, and P. Liang, “Fine-tuning vision-language-action models: Optimizing speed and success,” *arXiv preprint arXiv:2502.19645*, 2025.
- [17] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, “Spatialvla: Exploring spatial representations for visual-language-action model,” *arXiv preprint arXiv:2501.15830*, 2025.
- [18] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang *et al.*, “Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation,” *arXiv preprint arXiv:2411.19650*, 2024.
- [19] G. R. Team, A. Abdolmaleki, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, A. Balakrishna, N. Batchelor, A. Bewley, J. Bingham *et al.*, “Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer,” *arXiv preprint arXiv:2510.03342*, 2025.
- [20] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, “Gr00t n1: An open foundation model for generalist humanoid robots,” *arXiv preprint arXiv:2503.14734*, 2025.
- [21] Y. Zhong, X. Huang, R. Li, C. Zhang, Z. Chen, T. Guan, F. Zeng, K. N. Lui, Y. Ye, Y. Liang *et al.*, “Dexgraspvla: A vision-language-action framework towards general dexterous grasping,” in *AAAI*, 2026.
- [22] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu, “Being-h0: vision-language-action pretraining from large-scale human videos,” *arXiv preprint arXiv:2507.15597*, 2025.
- [23] D. Kim, H. Jang, M. Koo, S. Jang, T. Kim, B. Kim, B. Yoon, C. Jang, D. Choi, D. Han *et al.*, “Rldx-1 technical report,” *arXiv preprint arXiv:2605.03269*, 2026.
- [24] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, “Robotic control via embodied chain-of-thought reasoning,” *arXiv preprint arXiv:2407.08693*, 2024.
- [25] Q. Fan, Z. Li, J. Hu, Y. Lu, W. Chen, Q. Shen, X.-x. Long, Y. Cai, T. Lu, S. Wang *et al.*, “Peafowl: Perception-enhanced multi-view vision-language-action for bimanual manipulation,” *IEEE Robotics and Automation Letters*, 2026.
- [26] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, “Dexmv: Imitation learning for dexterous manipulation from human videos,” in *ECCV*, 2022.
- [27] A. Sivakumar, K. Shaw, and D. Pathak, “Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube,” *arXiv preprint arXiv:2202.10448*, 2022.
- [28] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, “Dexcap: Scalable and portable mocap data collection system for dexterous manipulation,” *arXiv preprint arXiv:2403.07788*, 2024.
- [29] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” in *ICRA*, 2023.
- [30] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “Progprompt: Program generation for situated robot task planning using large language models,” *AMR*, vol. 47, no. 8, pp. 999–1012, 2023.
- [31] 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.
- [32] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, “Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation,” *arXiv preprint arXiv:2409.01652*, 2024.
- [33] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu *et al.*, “Pivot: Iterative visual prompting elicits actionable knowledge for vlms,” *arXiv preprint arXiv:2402.07872*, 2024.
- [34] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, “Robopoint: A vision-language model for spatial affordance prediction for robotics,” *arXiv preprint arXiv:2406.10721*, 2024.
- [35] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, “Manipulate-anything: Automating real-world robots using vision-language models,” *arXiv preprint arXiv:2406.18915*, 2024.
- [36] J. Shi, R. Yang, K. Chao, B. S. Wan, Y. S. Shao, J. Lei, J. Qian, L. Le, P. Chaudhari, K. Daniilidis *et al.*, “Maestro: Orchestrating robotics modules with vision-language models for zero-shot generalist robots,” in *NeurIPS 2025 Workshop on Space in Vision, Language, and Embodied AI*, 2025.
- [37] L. Galanti, D. Shah, and T. Dao, “Addressing the orchestration gap in generalist robots via physical agency,” *arXiv preprint arXiv:2607.21725*, 2026.
- [38] Y. Furukawa and C. Hernández, “Multi-view stereo: A tutorial,” *Fnt CGV*, vol. 9, no. 1-2, pp. 1–148, 2015.
- [39] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, “Spatialvlm: Endowing vision-language models with spatial reasoning capabilities,” in *CVPR*, 2024.

- [40] C. H. Song, V. Blukis, J. Tremblay, S. Tyree, Y. Su, and S. Birchfield, "Robospacial: Teaching spatial understanding to 2d and 3d vision-language models for robotics," in *CVPR*, 2025.
- [41] R. Xu, Z. Huang, T. Wang, Y. Chen, J. Pang, and D. Lin, "Vlm-grounder: A vlm agent for zero-shot 3d visual grounding," *arXiv preprint arXiv:2410.13860*, 2024.
- [42] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong, "Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints," in *CVPR*, 2025.
- [43] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *CVPR*, 2025.
- [44] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024.
- [45] S. Baik, G. Kim, M. Choi, and H. Joo, "Text-guided 6d object pose rearrangement via closed-loop vlm agents," *arXiv preprint arXiv:2604.09781*, 2026.
- [46] C. Tang, A. Xiao, Y. Deng, T. Hu, W. Dong, H. Zhang, D. Hsu, and H. Zhang, "Mimicfunc: Imitating tool manipulation from a single human video via functional correspondence," *arXiv preprint arXiv:2508.13534*, 2025.
- [47] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgrasynet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," *arXiv preprint arXiv:2210.02697*, 2022.
- [48] J. Chen, Y. Ke, and H. Wang, "Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization," in *ICRA*, 2025.
- [49] J. Chen, Y. Ke, L. Peng, and H. Wang, "Dexonomy: Synthesizing all dexterous grasp types in a grasp taxonomy," *arXiv preprint arXiv:2504.18829*, 2025.
- [50] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao, "Rt-affordance: Affordances are versatile intermediate representations for robot manipulation," 2025.
- [51] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.
- [52] B. Sundaralingam, A. Murali, and S. Birchfield, "curobo2: Dynamics-aware motion generation with depth-fused distance fields for high-dof robots," *arXiv preprint arXiv:2603.05493*, 2026.
- [53] K. Park, C. Kim, K. Lee, and S. Choi, "Natural functional gradients for smooth trajectory optimization," *arXiv preprint arXiv:2605.28202*, 2026.
- [54] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, 1981.