Right-in-Left Brain VLA: A Closed-loop Perception-in-Reasoning Vision-Language-Action Model for Robotic Manipulation

***Abstract*—Robust robotic manipulation in open and dynamic environments is a fundamental requirement for making robots practical in human daily life. Without a closed loop that integrates perception, reasoning, and execution, robots cannot withstand disturbances and drift away from global goals, hindering practical deployment. Yet existing end-to-end Vision-Language-Action (VLA) models directly map inputs to actions but generalize poorly to unseen scenes in open environments. Hierarchical frameworks rely on lengthy prompt-driven spatial perception that is prone to hallucination, and their planning lacks closed-loop feedback, failing to correct online under disturbances. As a result, error accumulation destabilizes long-horizon task execution. Motivated by these limitations, we propose Right-in-Left Brain VLA, a brain-inspired framework that integrates perception into reasoning through a unified VLM. The "right brain" provides spatial perception by explicitly predicting affordances, poses, and constraints. The "left brain" performs chain-of-thought planning with continuous feedback from the "right brain" to ensure global consistency and correct errors online under disturbances. Experiments show that our method handles disturbances and mitigates error accumulation through closed-loop feedback and online adaptive correction to robustly execute long-horizon tasks and generalize to unseen scenarios. Program and videos are available on https://show-idea.github.io/Right-in-Left-Brain-VLA.**

I. Introduction

Robotic manipulation in open and dynamic environments is a cornerstone capability for service, healthcare, and household robots. A key challenge is robust manipulation for long-horizon tasks [1]-[3]. Despite recent progress, enabling such capability remains an unsolved problem. While end-to-end Vision-Language-Action (VLA) models can directly map perception to control, they often degrade under environmental variations, making them unsuitable for long-horizon tasks in open and dynamic environments [4]-[7]. Consequently, many works adopt hierarchical frameworks based on pretrained Vision-Language Models (VLMs) that explicitly decouple perception, planning, and execution [8]-[12]. Such modularity improves interpretability but introduces error accumulation, where small deviations in early sub-tasks propagate to derail the entire plan [13]. Moreover, existing methods typically rely on prompt engineering with lengthy and complex prompts to infer affordances or spatial constraints. Such implicit reasoning is fragile and inefficient, prone to hallucination in long-horizon tasks. Meanwhile, most perception modules [14] only perform static inference, failing to adapt when objects are moved, which further amplifies errors. In addition, most methods generate the total task sequence upfront without replanning [15], and others plan step by step but easily drift from the global goal [16]-[17].

Overall, these limitations highlight the fundamental gap that current robotic systems lack a unified framework which maintains global task consistency while also enabling online adaptive correction in open environments. Inspired by the functional specialization of the brain [18]-[19], where the left hemisphere excels at logical reasoning and the right hemisphere at spatial perception, we propose Right-in-Left Brain VLA for robotic manipulation in open environments. Whereas most VLAs resemble the "left brain" to emphasize logical reasoning but rely on fragile perception based on lengthy and complex prompts, Right-in-Left Brain VLA integrates perception into reasoning with a unified VLM, as shown in Fig. 1. Through fine-tuning the VLM, the "right brain" not only continuously predicts affordances and manipulation constraints but also performs pose tracking to capture spatial relations among objects during execution. These perceptual outputs are directly leveraged by the same VLM's "left brain" for chain-of-thought (CoT) reasoning to maintain global task consistency and correct sub-tasks from closed-loop feedback, ensuring each sub-task is both physically executable and globally consistent. Following this, the execution layer achieves robust performance by combining analytic control for simple tasks and reinforcement learning (RL) for complex dexterous tasks.

Real-world experiments demonstrate that Right-in-Left Brain VLA not only handles disturbances through continuous perception but also reduces error accumulation via online adaptive correction based on closed-loop feedback to ensure robust execution across both the gripper and dexterous hand in long-horizon tasks. In summary, our contributions are fourfold:

- We propose Right-in-Left Brain VLA, a brain-inspired framework that unifies perception with reasoning for long-horizon tasks in open and dynamic environments.
- We introduce explicit perception by affordance and manipulation constraint prediction, in combination with pose tracking integrated into task-driven spatial reasoning, enabling robust manipulation beyond implicit reasoning.
- A novel chain-of-thought paradigm for long-horizon collaborative task planning enables robots to reason with global consistency and adaptive correction, beyond the limitations of existing hierarchical frameworks.
- Extensive experiments demonstrate our method's robust performance under disturbances and strong generalization to unseen scenes in long-horizon tasks, highlighting its potential for open-world bimanual manipulation.

II. Related Work

Environmental Perception for Robotic Manipulation. For robotic manipulation, the core foundations of environmental perception are affordance and object pose. Prior approaches typically use DINO [20] or SAM [21] for object detection and segmentation, while prompting VLMs with long and complex prompts to implicitly infer affordances and poses [8]-[11]. This paradigm is slow and brittle in long-horizon tasks, and is prone to hallucination. It is also constrained by the detection accuracy of pretrained vision models. Moreover, task-level spatial constraints, such as feasible manipulation angles and distances, are widely used, but are always inferred implicitly

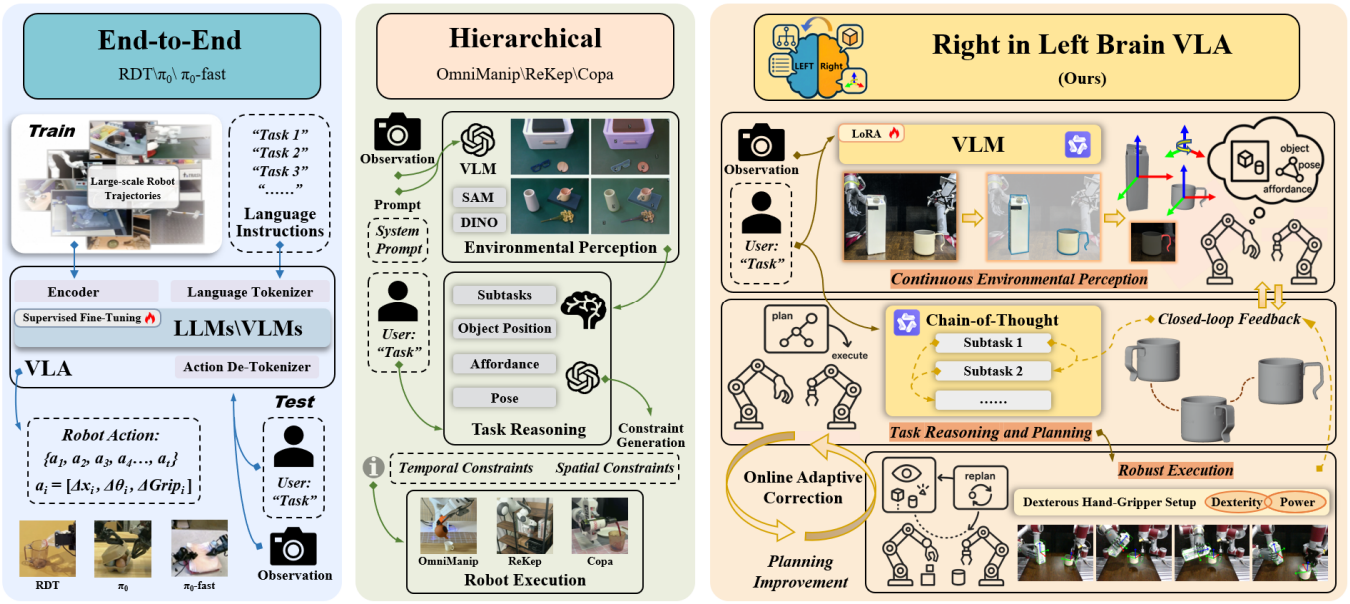


Fig. 1 **Comparison of Right-in-Left Brain VLA and previous methods**. End-to-end VLAs map inputs to actions but are brittle under distribution shifts, limiting long-horizon robustness in open environments. Previous hierarchical frameworks rely on lengthy prompt-driven spatial perception and lack feedback-driven online replanning, causing error accumulation under disturbances. Right-in-Left Brain VLA (ours) integrates explicit spatial perception into CoT-based planning with a unified VLM, enabling online adaptive correction via closed-loop feedback and robust execution in long-horizon collaborative tasks.

from static pose estimates in planning modules [14]. As a result, they cannot adapt to pose changes under disturbances [15]. These gaps motivate the "right brain" of our framework, which explicitly predicts affordances and manipulation constraints to improve manipulation robustness through fine-tuning the same VLM used for task reasoning. In addition, it performs pose tracking to handle disturbances and employs task-driven spatial constraints to support reliable interactions between objects.

Long-Horizon Task Planning with VLMs. VLMs are increasingly used for robotic task reasoning and planning [9]-[11], [22]-[25]. End-to-end VLA models are trained on internet-scale multimodal data to directly map perception to execution [4]-[7], [26]-[28], but remain brittle under lighting and viewpoint changes, as well as distribution shifts [29], which limits long-horizon robustness in open-world settings. Alternatively, hierarchical frameworks use the VLM as the high-level planner on top of motion-planning and execution modules [9], [12], [14], [30], improving generalization and interpretability at a lower annotation cost. Representative directions include leveraging task semantics to generate relational keypoint constraints for manipulation planning [8], [9], adopting object-centric representations for commonsense reasoning and 3D scene understanding [11], [14], as well as decomposing tasks into invariant and actionable intermediate representations to improve transferability [12], [24]. Despite significant progress in single-arm manipulation, robust adaptation to disturbances in bimanual collaboration for long-horizon tasks remains underexplored in open-world settings. Recent studies leverage CoT prompting to generate subtasks for long-horizon tasks [15]-[17], reducing error propagation and improving accuracy. However, these approaches still lack feedback-driven online replanning to handle disturbances. In this work, we propose a CoT-based planner (the "left brain" of our framework) for long-horizon collaborative tasks that operates stepwise and performs online replanning during execution with closed-loop feedback from the continuous spatial perception of the "right brain" to maintain global task consistency and adapt to disturbances.

I. Method

To address the challenge of robust robotic manipulation for long-horizon tasks in open and dynamic environments, we propose the Right-in-Left Brain VLA. The framework consists of four modules: 1) affordance and manipulation constraint prediction (Section III-A), 2) pose tracking and spatial constraints reasoning (Section III-B), 3) CoT planning with online adaptive correction (Section III-C), and 4) hybrid execution with analytic control and RL (Section III-D). The system provides affordances, manipulation constraints, poses, and spatial constraints for CoT planning, enabling robust execution with both a gripper and a dexterous hand.

A. Affordance and Manipulation Constraint Prediction

We fine-tune the Qwen2-VL-7B [31] to predict affordances and manipulation constraints, since this task can be formulated as a multimodal reasoning problem. In this setting, the model must localize manipulable regions and output structured action constraints that directly guide object manipulation, particularly for analytic control. Fine-tuning mitigates hallucination and enables fast and reliable prediction. The implementation is illustrated in Fig. 2.

Label Construction. To provide supervised labels for affordance and constraint prediction, we extend the HANDAL [32] and YCB [33] datasets to construct structured tuples:

$$\tau = \left(B_{obj}, B_{mani}, C_{mani}\right) \quad (1)$$

where B_{obj} is the object bounding box (BBox), B_{mani} is the manipulation box denoting the manipulable region, and C_{mani} encodes manipulation constraints for both the gripper and the dexterous hand. These constraints are instantiated from commonsense manipulation priors, such as grasp orientation, gripper opening and finger flexion, which are critical for analytic control.

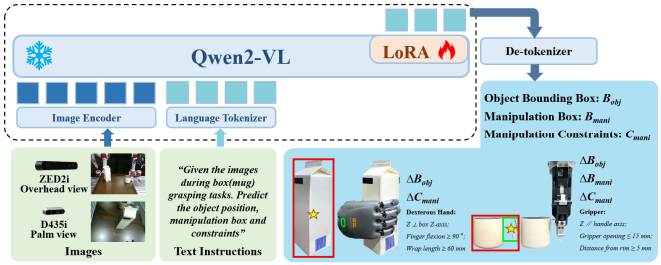
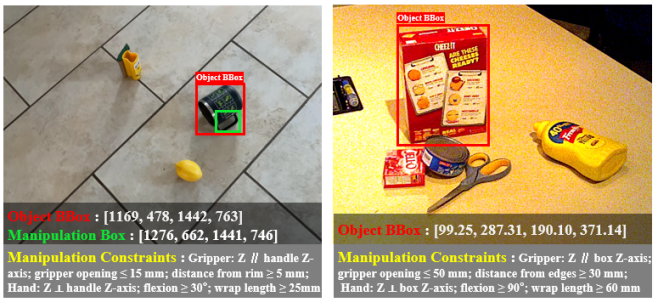


Fig. 2. Qwen2-VL fine-tuned with LoRA. Given multi-view images and task instructions, the model outputs structured tuples $(B_{obj}, B_{mani}, C_{mani})$.



Mug (HANDAL) *Box (YCB)*

Fig. 3 Representative examples from HANDAL and YCB.

Specifically, HANDAL includes 17 household categories and provides object BBoxes and handle masks, from which B_{mani} is derived. YCB contributes diverse everyday objects with object BBoxes but lacks handle annotations, so its tuples omit B_{mani}. From each dataset, we select 10 object categories for our experiments. Representative examples are shown in Fig. 3, with additional details provided in Appendix A.

Parameter-Efficient Fine-Tuning with LoRA. We adopt LoRA [34] for parameter-efficient finetuning of Qwen2-VL-7B. LoRA decomposes each pretrained weight matrix into a frozen base W_0 and a low-rank adaptation term $AB^\top$:

$$W = W_0 + AB^\top, r \ll min(d,k) \quad (2)$$

where d and k are the input and output dimensions, and r is the low rank. In this way, only a small fraction of parameters are updated to acquire manipulation-specific knowledge, while the backbone's general multimodal capacity is preserved. This enables reliable affordance and constraint prediction with low training cost.

B. Pose tracking and spatial constraints reasoning

In open and dynamic environments, instance-level pose estimation methods often fail to initialize due to the lack of available CAD models, making long-horizon tracking fragile from the start. To address this issue, we employ an offline 3D diffusion model [35] that jointly leverages overhead view and palm view observations to generate an explicit CAD prior. This prior not only serves as a geometric constraint during initialization but also acts as a reusable geometric template for re-initialization and refinement, thereby eliminating the dependence on real CAD models. We follow the mainstream paradigm of single-frame initialization and track-by-refine. Unlike FoundationPose [36], our method addresses pose estimation in CAD-free scenarios by leveraging explicit priors for reliable initialization. In addition, it suppresses error accumulation and enhances recovery in long-horizon tasks.

During the first-frame initialization $(t = 0)$, the system takes as input the cropped target image patch I_t from the detected target object BBox B_{obj}, the corresponding mask M_t processed by Qwen2-VL, depth image D_t, and camera intrinsics K. An initial pose $\widehat{P_0}$ is obtained by solving a sparse correspondence-based PnP problem:

$$\widehat{P_0} = PnP(\mathcal{C}(I_0, M_0, D_0), K) \quad (3)$$

where $\mathcal{C}(\cdot)$ denotes the 2D–3D correspondences extracted from the input. For each subsequent frame $(t > 0)$, the pose estimation inherits the previous result as initialization and undergoes fast iterative refinement. If the reprojection error e_{reproj} exceeds the threshold ($e_{reproj}(\hat{P}_t) > \delta_r$) during tracking, the system falls back to the explicit CAD prior and triggers re-initialization, recovering from drift and preventing long-horizon error accumulation.

Beyond initialization and drift recovery, we incorporate two constraints during optimization to handle shape variations across different object instances.

Symmetry Constraint. For objects with rotational or mirror symmetry, multiple equivalent pose solutions may exist. The estimation may oscillate among equivalent solutions, causing instability. We compute the minimum Chamfer distance between the predicted point cloud and the explicit CAD template under the action of the symmetry group, and select the optimal alignment to ensure consistency in pose estimation. We define the Chamfer distance under symmetry constraints as:

$$d_{sym}(P,Q) = \min_{g \in G} \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q_g} \| p - q \|_2 \quad (4)$$

where G denotes the symmetry group, and Q_g is the CAD prior transformed by symmetry operation g.

Scale Normalization. For objects from different categories, we first normalize their diameters to unit scale so that the estimation is carried out under a unified normalized scale. For scale-sensitive categories, we further introduce a lightweight scale regularization term:

$$\mathcal{L}_{scale} = \| s - \hat{s} \|_2^2 + \lambda \| P - \hat{P} \|_F^2 \quad (5)$$

where s and $\hat{s}$ denote the ground-truth and estimated scales, P and $\hat{P}$ represent the ground-truth and predicted poses, and λ is a weighting factor. This ensures cross-category consistency and stability. During inference, candidate poses are re-weighted by semantic constraints, suppressing implausible hypotheses and promoting feasible ones. As a result, the final estimate is both geometrically consistent with the CAD template and functionally plausible for manipulation.

We further impose task-driven spatial constraints to restrict feasible poses. The Qwen2-VL interprets the task instruction and scene context to infer interpretable geometric requirements with confidence scores, such as feasible directions and ranges. For example, in the pouring task, the system predicts a reasonable tilt range of the box while excluding physically meaningless poses. This mechanism ensures that pose estimation satisfies both geometric consistency and functional plausibility. Through this approach, spatial constraints are integrated with semantic reasoning, enabling adaptive alignment across different tasks. These constraints are qualitatively visualized in Fig. 6(a).

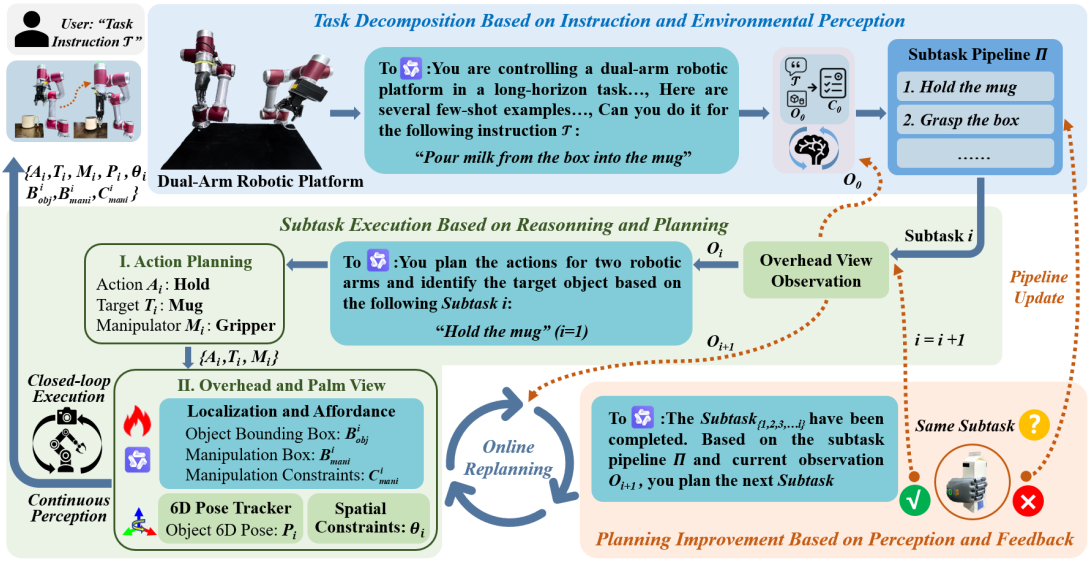


Fig. 4. Workflow of CoT planning with online adaptive correction. The planner first decomposes the task into an initial subtask pipeline, then alternates between subtask execution and feedback-driven replanning. Continuous perception from the "right brain" supports visual servoing for closed-loop execution, while feedback-based online replanning maintains global consistency under disturbances.

C. CoT Planning with Online Adaptive Correction

To maintain global task consistency and robustness under disturbances, we design a CoT planner with online adaptive correction. The overall workflow is illustrated in Fig. 4. The planner first decomposes the task into an initial subtask pipeline, and then alternates between sub-task execution and feedback-based replanning to dynamically improve the plan under disturbances.

Task Decomposition. Given a task instruction $\mathcal{T}$, the planner captures an initial observation O_0 and grounds the $\mathcal{T}$ in O_0 using Qwen2-VL to obtain a structured semantic context C_0, which resolves ambiguous or abstract instructions. Constrained by $\mathcal{T}$, the planner decomposes the task into an initial long-horizon subtask pipeline Π

$$\Pi = \tau_1, \tau_2, \dots, \tau_N \tag{6}$$

ensuring that each subtask is globally consistent with $\mathcal{T}$.

Subtask Execution. For subtask τ_i, the planner first predicts the action intention, including the action type A_i, the target object T_i, and the manipulator M_i (gripper or dexterous hand). Conditioned on M_i, the current observation O_i is continuously processed by the "right brain" (Sec. III-A and Sec. III-B) into a structured scene state S_i:

$$S_i = \left(B_{obj}, B_{mani}, C_{mani}, P_i, \theta_i\right) \tag{7}$$

These are combined into an execution tuple

$$\varepsilon_i = \left(A_i, T_i, M_i, B_{obj}, B_{mani}, C_{mani}, P_i, \theta_i\right) \tag{8}$$

which is then passed to the execution layer.

Feedback and Online Correction. After executing ε_i, the planner captures a new observation O_{i+1}. Qwen2-VL then infers the next sub-task τ'_{i+1}, conditioned on O_{i+1}, the completed sub-tasks $\{\tau_1, \dots, \tau_i\}$ and the subtask pipeline Π. If τ'_{i+1} remains consistent with the τ_{i+1}, the planner continues with the original sub-task τ_{i+1}. Otherwise, τ_{i+1} is replaced by τ'_{i+1} and the pipeline is updated. This closed loop continues until the global instruction $\mathcal{T}$ is satisfied.

This process constitutes an adaptive correction mechanism. During execution, continuous perception from the "right brain" enables visual servoing to detect disturbances, while at the task level, feedback-based replanning improves future sub-tasks to align with the global instruction. As a result, these two layers reduce error accumulation and preserve global consistency.

D. Hybrid Execution with Analytic Control and RL

We adopt a heterogeneous bimanual setup with a gripper and a dexterous hand, using their complementary strengths of power and dexterity. However, this configuration introduces an execution challenge. Analytic control, based on scene state S_i from the "right brain", is effective for low-complexity tasks, where classical methods such as IK-based motion planning can be directly applied [37]. However, many high-complexity tasks inherently require dexterous hand to execute, where analytic control is computationally expensive and fragile. To address this limitation, execution layer adopts a hybrid strategy, combining analytic control for simple tasks with RL for complex dexterous tasks. As analytic control is mature, we primarily detail the RL component, demonstrated on two representative tasks, namely drawer and cabinet opening.

RL Algorithm. We use Proximal Policy Optimization (PPO) to train dexterous manipulation policies. PPO [38] stabilizes training by clipping the policy ratio in the surrogate objective, which is formally defined as

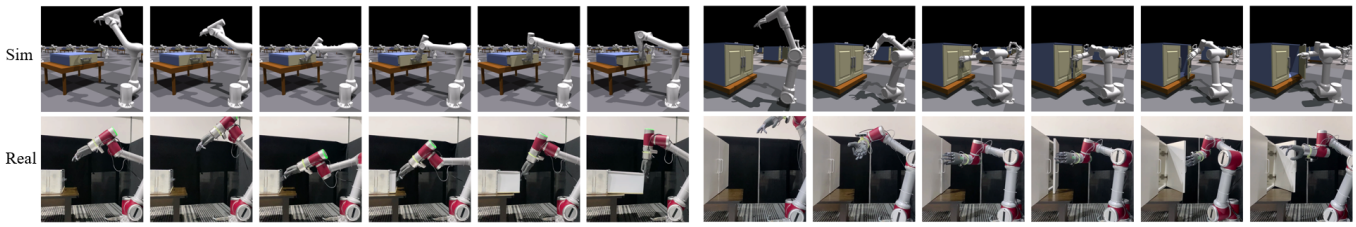
$$\mathcal{L}^{CLIP}(\theta) = E_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{9}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between the new and old policy, $\hat{A}_t$ is the estimated advantage, and ϵ is a clipping parameter that controls the update magnitude. Policies are trained in IsaacGym using robot proprioception and object states as observations to output low-level joint actions for both the 6-DoF arm and the dexterous hand [39].

Reward Design. We design task-specific staged rewards for drawer and cabinet opening to provide dense guidance and stabilize learning across different manipulation phases. For both tasks, the total reward at timestep t is defined as

$$r_{task}(t) = r_{reach}(t) + r_{wrap}(t) + r_{open}(t) \tag{10}$$

Task-specific designs for drawer and cabinet opening are detailed below.



(a) Drawer opening (b) Cabinet opening

Fig. 5. RL policies execution on drawer and cabinet opening in simulation and the real world. (a) Drawer opening. (b) Cabinet opening. To bridge the sim-to-real gap, domain randomization and observation normalization are applied following the practices of Isaac Gym and rl_games.

1) Drawer Opening. For this task, r_{reach} encourages all fingers to approach the drawer handle, with larger weights on the index finger, thumb, and middle finger. r_{wrap} encourages a proper grasp configuration by rewarding the index and middle fingers positioned above the drawer handle while the thumb is below. Finally, r_{open} encourages the agent to open the drawer by rewarding the displacement of the drawer joint. 2) Cabinet Opening. For this task, r_{reach} encourages the palm center to approach the cabinet handle. r_{wrap} encourages both orientation alignment and stable manipulation around the handle. Unlike drawer opening, which mainly involves translational motion, cabinet opening additionally requires rotation. Therefore, an additional orientation alignment is set to ensure that the fingers align with the handle. Stable manipulation is further encouraged by rewarding the index finger on the left side of the handle and the thumb on the right. Finally, r_{open} encourages the agent to open the cabinet by rewarding the rotation of the cabinet joint. Full formulations of the two tasks are provided in Appendix B.

Visualization of Task Executions. Qualitative results in both simulation and the real world demonstrate effective policy execution on the two tasks, as shown in Fig. 5. The corresponding video is provided in the supplementary video.

I. Experiments

In this section, we aim to answer the following questions: 1) How effectively does Right-in-Left Brain VLA perform diverse long-horizon tasks in real-world settings (Sec. IV-B)? 2) Can it preserve global task consistency under exogenous disturbances through online replanning (Sec. IV-C)? 3) To what extent does it generalize to unseen scenes (Sec. IV-D)? 4) What are the contributions of its key modules as revealed by ablation studies (Sec. IV-E)?

A. Experimental Setup

Hardware. We set up a real-world tabletop workspace. We use two JAKA ZU5 robots (6-DoF), equipped with a BrainCo Hand (11-DoF) and a DH AG95 parallel gripper (1-DoF). For analytic control, the system is controlled via MoveIt1, using RRT-Connect for motion planning and ROS controllers for trajectory execution. For perception, an Intel RealSense D435i on the dexterous hand provides a palm view and a ZED 2i camera provides an overhead view. All experiments are conducted on a workstation with 4× NVIDIA RTX 4090 GPUs. All VLM components are implemented with Qwen2-VL-7B.

Tasks and Evaluation. We evaluate our framework on two sets of tasks to compare against hierarchical frameworks and end-to-end VLAs. For hierarchical frameworks, we design four long-horizon collaborative tasks: 1) Pour milk from the box into the mug; 2) Put marker pens into the pen holder; 3) Take out the bottle from the drawer; 4) Take out the bottle from the cabinet. For end-to-end VLAs, we design four primitive tasks: 1) Pick up the target object; 2) Place the object at the target location; 3) Open the drawer by at least 10 cm; 4) Open the cabinet by at least 45°. Performance is evaluated by the average success rate (ASR) over 20 trials.

Baselines. Existing hierarchical frameworks are not directly compatible with our dexterous hand setup and cannot be deployed on our platform. Therefore, we construct comparable variants for ablation: 1) Right-in-Left Brain VLA (Ours): the full proposed framework; 2) Ours (DINOv2): replacing object detection with DINOv2 [40]; 3) Ours (FoundationPose): replacing pose estimation with FoundationPose; 4) Ours (GraspNet): replacing affordance analysis with GraspNet [41]; 5) Ours (GPT-4o): replacing the planner with GPT-4o via the OpenAI API. For end-to-end VLAs, we evaluate RDT [5], π_0 [4], and π_0-fast [4] on primitive tasks, fine-tuned using 20 demonstrations per task.

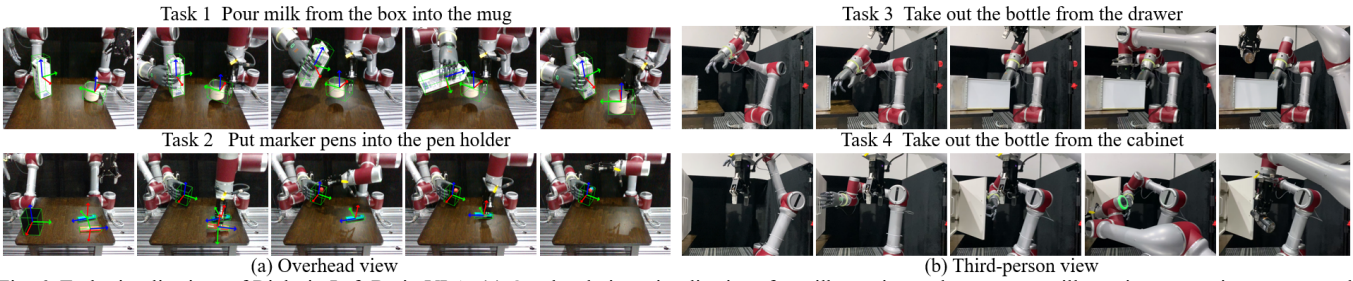
B. Results Analysis

Results on Long-horizon Collaborative Tasks. The results are shown in Table I. Right-in-Left Brain VLA achieves the best performance across all four tasks with an average ASR of 87.5%, outperforming the best baseline by 13.75%, which demonstrates robust execution in long-horizon collaborative tasks. Replacing perception modules lowers the average ASR by 13.75%-38.75%, while replacing the planner decreases the average ASR by 15%, highlighting the role of explicit perception and chain-of-thought planning in long-horizon tasks. Task visualizations of Right-in-Left Brain VLA on the four long-horizon collaborative tasks are shown in Fig. 6, and the corresponding video is provided in the supplementary video.

Results on Primitive Tasks. The results are reported in Table II Right-in-Left Brain VLA consistently outperforms all baselines on the four primitive tasks. In particular, the ASR exceeds 95% across both pick and place tasks and reaches above 90% on cabinet and drawer opening, validating that our framework is capable of solving low-complexity tasks guided by affordance, pose, and constraints, while integrating RL-trained policies to handle complex and dexterous tasks. In contrast, although end-to-end VLAs achieve an average ASR of around 33.33% on pick and 15% on place, they record no successful trials on cabinet and drawer opening, which indicates the difficulty of generalizing complex manipulations with limited demonstrations.

TABLE I. Results on Long-Horizon Collaborative Tasks (%)

Model	Task1	Task2	Task3	Task4	Mean
Ours (DINOv2)	80	55	85	70	72.50
Ours (FoundationPose)	20	25	85	65	48.75
Ours (GraspNet)	45	70	90	90	73.75
Ours (GPT-4o)	75	70	70	75	72.50
Ours (full framework)	**85**	**80**	**95**	**90**	**87.50**



(a) Overhead view (b) Third-person view

Fig. 6. Task visualizations of Right-in-Left Brain VLA. (a) Overhead view visualizations for milk pouring and pen storage, illustrating perception outputs and task-driven spatial constraints. In pouring, the box's z-axis is constrained to be nearly perpendicular to the mug's z-axis, while in pen storage the pen's z-axis is constrained to be nearly parallel to the holder's z-axis. (b) Third-person view visualizations for drawer and cabinet opening, highlighting the execution process.

TABLE II. Results on Primitive Tasks (%)

Model	Task1	Task2	Task3	Task4	Mean
RDT	25	5	0	0	7.50
π_0	45	25	0	0	17.50
π_0-fast	30	15	0	0	11.25
Ours (full framework)	**100**	**95**	**90**	**90**	**93.75**

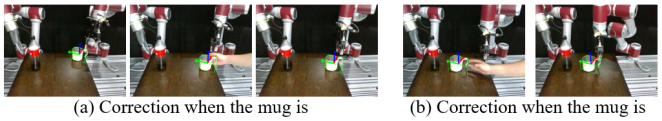
C. Robustness Evaluation under Exogenous Disturbances

In this section, we evaluate the robustness of Right-in-Left Brain VLA against exogenous disturbances. To simulate real-world disturbances, we manually change the pose of the target object immediately before contact. This disturbance protocol is applied to all long-horizon collaborative tasks defined in Section IV-A. This evaluation tests whether continuous perception from the "right brain" and feedback-driven online adaptive correction in the "left brain" can preserve global task consistency under disturbances, in contrast to hierarchical baselines that rely on static observations.

As shown in Table III, Right-in-Left Brain VLA maintains the ASR above 75% on all tasks under disturbances. In contrast, baselines with replaced perception modules fail to adapt to disturbances due to static observations. Baseline with GPT-4o lacks feedback-driven adaptive correction and collapses under disturbances. This demonstrates that our framework not only sustains reliable long-horizon execution but also handles exogenous disturbances, which conventional hierarchical and end-to-end approaches cannot handle. Fig. 7 illustrates the adaptive correction in the coke pouring, and the corresponding video is provided in the supplementary video.

TABLE III. Results on Long-Horizon Collaborative Tasks under Exogenous Disturbances (%)

Model	Task1	Task2	Task3	Task4	Mean
Ours (DINOv2)	0	5	0	0	1.25
Ours (FoundationPose)	5	10	0	0	3.75
Ours (GraspNet)	5	5	0	0	2.50
Ours (GPT-4o)	0	0	0	0	0
Ours (full framework)	**80**	**75**	**75**	**75**	**76.25**



(a) Correction when the mug is perturbed longitudinally (b) Correction when the mug is perturbed laterally

Fig. 7. Online adaptive correction in the coke pouring task under exogenous disturbances. (a) Correction when the mug is perturbed longitudinally. (b) Correction when the mug is perturbed laterally.

D. Generalization Evaluation under Unseen Scenes

We evaluate generalization to unseen lighting conditions, with the lighting setups illustrated in Fig. 8. This evaluation is performed on the *milk pouring* task and the primitive *pick* task. Since this setting primarily tests perception performance, we compare Right-in-Left Brain VLA against perception-replacement baselines, as well as end-to-end VLA models.

As shown in Table IV and Table V, Right-in-Left Brain VLA achieves much higher ASR than all replacement baselines under unseen lighting. Replacing explicit detection with DINOv2 reduces average ASR by 28.33%, since the generic detection model suffers a significant accuracy drop under low-contrast edges and weak textures. FoundationPose lowers average ASR by 60%, since highlights and shadows cause correspondence loss and pose drift. GraspNet decreases average ASR by 58.33%, as the handle of mug is often missed in dim light, yielding unstable grasps. End-to-end VLAs suffer severe degradation even on simple pick tasks, underscoring their inability to generalize under environmental changes.

TABLE IV. Results on Milk Pouring task (%)

Model	Scene1	Scene2	Scene3	Mean
Ours (DINOv2)	40	65	20	41.67
Ours (FoundationPose)	10	15	5	10
Ours (GraspNet)	15	20	0	11.67
Ours (full framework)	**70**	**75**	**65**	**70**

TABLE V. Results on Pick Task (%)

Model	Scene1	Scene2	Scene3	Mean
RDT	5	5	0	3.33
π_0	15	10	0	8.33
π_0-fast	10	5	0	5.00
Ours (full framework)	**85**	**75**	**70**	**76.67**

(a) Front-Dim Illumination (b) Overhead Spotlight (c) Slit Illumination

Fig. 8. Unseen lighting conditions. Three settings are designed. (a) Front-Dim Illumination, where dim frontal illumination causes weak textures and shallow shadows. (b) Overhead Spotlight, where a single source produces specular highlights and deep cast shadows. (c) Slit Illumination, where a narrow horizontal light band causes partial object visibility.

E. Ablation Studies

To highlight the role of each component in handling disturbances, we conduct ablation studies on the four long-horizon collaborative tasks under exogenous disturbances. Specifically, 1) we remove the prediction of affordances and manipulation constraints, manipulating only at object centers or random positions; 2) we replace pose tracking and spatial constraint reasoning with fixed vertical poses; 3) we omit the COT planner, letting Qwen2-VL directly output actions. As illustrated in Table VI, removing any of these components

results in a significant reduction in performance. Without affordance and manipulation constraints, the average ASR decreases by 27.5%. In this case, not only will the system attempt infeasible grasps, but the closed-loop feedback is also ineffective because adaptive correction cannot compensate for invalid grasp choices. Fixed vertical poses lower average ASR by 26.25%, since the system cannot adapt to object pose shifts. Omitting the COT planner yields no successful trials, as the lack of online adaptive correction prevents the model from replanning under disturbances.

TABLE VI. Results on Ablation Studies (%)

Model	Task1	Task2	Task3	Task4	Mean
Ours w/o affordance and manipulation constraint	10	70	85	75	60.00
Ours w/o pose tracking and spatial reasoning	40	75	65	65	61.25
Ours w/o COT planner	0	0	0	0	0
Ours (full framework)	**85**	**80**	**95**	**90**	**87.50**

I. Conclusion and Future Work

Inspired by the functional specialization of the human brain, we propose the Right-in-Left Brain VLA, a robust bimanual manipulation framework for open and dynamic environments. This framework integrates perception into reasoning with a unified VLM, enabling continuous spatial perception from the "right brain" and providing closed-loop feedback to the "left brain" for reasoning and step-wise planning. This structure allows the system to perform online adaptive corrections and replanning to ensure global consistency during execution, overcoming limitations such as error accumulation and poor robustness to disturbances and generalization to unseen scenes, which provides a promising foundation for the practical deployment of robots in real-world environments filled with unknown variations and sudden disturbances.

Although extensive experiments demonstrate that it can efficiently handle disturbances, it is still limited by the capabilities of foundational VLMs, which restrict the speed of replanning. The framework still has promising progress with the development of the foundational models. In addition, we continue to use RL strategies to enrich the ability of dexterous hand to cope with more complex tasks in future work.

Appendix

A. Dataset and Label Construction

We construct unified supervised labels for affordance and manipulation constraint prediction based on the HANDAL and YCB datasets, selecting 10 representative categories from each dataset, as shown in Fig. 9 and Fig. 10. All bounding boxes are parameterized as $[x_{min}, y_{min}, x_{max}, y_{max}]$.

Fig. 9 Ten representative household categories from HANDAL

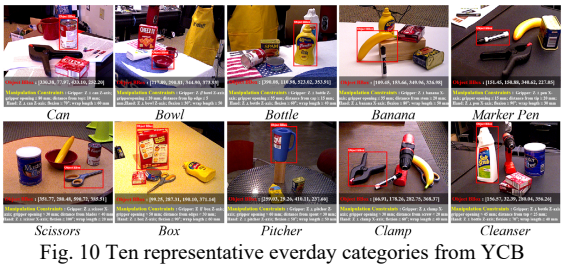


Fig. 10 Ten representative everday categories from YCB

B. Reward Formulation

Drawer Opening. The total reward at timestep t is defined as: $r_{drawer}(t) = r_{reach}(t) + r_{wrap}(t) + r_{open}(t)$.

The r_{reach} encourages all fingers to approach the drawer handle, with larger weights assigned to the index finger, thumb, and middle finger, which is defined as :$r_{reach}(t) = w_1 \left[(r_0 - d_{index}(t)) + (r_0 - d_{thumb}(t)) + (r_0 - d_{middle}(t))\right] + w_2 \left[(r_0 - d_{ring}(t)) + (r_0 - d_{little}(t))\right]$. $d_{index}(t)$, $d_{thumb}(t)$, $d_{middle}(t)$, $d_{ring}(t)$, $d_{little}(t)$ are the Euclidean distances from each fingertip to the handle. r_0 is the reference distance for fingers to manipulate the handle for opening. w_1 is the weight for the main fingers. w_2 is the weight for the auxiliary fingers. The r_{wrap} encourages a proper grasp configuration, as it rewards the index finger and middle finger being positioned above the drawer handle while the thumb is positioned below it, which is defined as:$r_{wrap}(t) = \lambda_{wrap} \cdot 1\{z_{index}(t) > z_{handle}\} \cdot 1\{z_{middle}(t) > z_{handle}\} \cdot 1\{z_{thumb}(t) < z_{handle}\}$. $z_{index}(t)$, $z_{middle}(t)$, $z_{thumb}(t)$ denote the vertical positions of the index finger, middle finger and thumb, respectively. λ_{wrap} is the scale factor to control the strength of the reward. The r_{open} encourages the agent to open the drawer, as it rewards the opening of drawers based on joint displacement, which is defined as: $r_{open}(t) = \lambda_{open} \cdot d_{open}(t)$. $d_{open}(t)$ denotes the displacement of the drawer joint. λ_{open} is the scale factor to control the strength of the reward.

Cabinet Opening. The total reward at timestep t is defined as: $r_{cabinet}(t) = r_{reach}(t) + r_{wrap}(t) + r_{open}(t)$.

The r_{reach} encourages the palm center to approach the cabinet handle. It is defined as: $r_{reach}(t) = \lambda_{\text{reach}} \cdot 1/(1 + d(t)^2)$. Here, $d(t)$ denote the Euclidean distance between the gripper grasp position and the cabinet handle, λ_{reach} is the scale factor to control the strength of the reward. The r_{wrap} encourages orientation alignment and stable manipulation around the handle. Unlike drawer opening, which is mostly translational, cabinet opening requires rotation. Therefore, an orientation alignment is set to ensure that the fingers align with the handle. A stable manipulation is further encouraged by rewarding the index finger on the left side of the handle and the thumb on the right. The r_{wrap} is defined as follows.

$$r_{wrap}(t) = \lambda_{rot} \cdot align(t) + \lambda_{finger} \cdot surround(t) \quad (11)$$

$$align(t) = \frac{1}{2}\left(c_f(t)\left|c_f(t)\right| + c_u(t)\left|c_u(t)\right|\right) \quad (12)$$

$$c_f(t) = \mathbf{a}^f_{fingers}(t) \cdot \mathbf{a}^f_{cabinet}(t) \quad (13)$$

$$c_u(t) == \mathbf{a}^u_{fingers}(t) \cdot \mathbf{a}^u_{cabinet}(t) \quad (14)$$

$$surround(t) = \left(r_0 - d_{index}(t)\right) + \left(r_0 - d_{thumb}(t)\right) \quad (15)$$

Here, $align(t)$ measures the directional consistency between the fingers and the handle, and $surround(t)$ rewards proper placement, giving higher reward when the index finger is on the left side of the handle and the thumb is on the right. $c_f(t)$ and $c_u(t)$ denote cosine similarities between the fingers and

handle axes. $\mathbf{a}^{f}_{fingers}(t)$ and $\mathbf{a}^{u}_{fingers}(t)$ denote the forward and upward axes of the fingers, $\mathbf{a}^{f}_{cabinet}(t)$ and $\mathbf{a}^{u}_{cabinet}(t)$ are the corresponding axes of the handle. $d_{index}(t)$ and $d_{thumb}(t)$ are the distances from the fingertips to the handle, and r_0 is the reference distance for stable opening.

The r_{open} encourages the agent to open the cabinet, as it rewards the rotation of the cabinet joint based on angular displacement, which is defined as: $r_{open}(t) = \lambda_{open} \cdot d_{open}(t)$. $d_{open}(t)$. $d_{open}(t)$ denotes the angular displacement of the cabinet door joint. λ_{open} is the scale factor to control the strength of the reward.

REFERENCES

[1] J. Zhang, L. Tang, Y. Song, Q. Meng, H. Qian, J. Shao. "FLTRNN: Faithful Long-Horizon Task Planning for Robotics with Large Language Models," *in 2024 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2024, pp. 6680-6686.

[2] 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.

[3] Y. Yang, L. Zhao, M. Ding, G. Bertasius, D. Szafir. "BOSS: Benchmark for Observation Space Shift in Long-Horizon Task," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 8882-8889, 2025.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom et al. "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024.

[5] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, J. Zhu. "Rdt-1b: a diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv: 2410.07864,* 2024.

[6] A. Brohan, N. Brown, J. Carbajal et al. "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*. 2022.

[7] D. Ghosh, H. Walke, K. Pertsch et al. "Octo: An Open-Source Generalist Robot Policy," *arXiv preprint arXiv:2405.12213*. 2024.

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

[9] W. Huang, C. Wang, Y. Li, R. Zhang, L, Fei-Fei. "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[10] F. Liu, K. Fang, P. Abbeel, S. Levine. "Open-world robotic manipulation through markbased visual prompting" *arXiv preprint arXiv:2403.03174*, 2024.

[11] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, H. Dong. "Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints" *in 2025 IEEE Computer Vision and Pattern Recognition Conference (CVPR)*. IEEE, 2025.

[12] K. Zhang, R. Xu, P. R, J. Lin, H. Wu, L, Lin. X. Liang. "RoBridge: A Hierarchical Architecture Bridging Cognition and Execution for General Robotic Manipulation," *arXiv preprint arXiv:2505.01709*,2025.

[13] Y. Fan, P. Ding, S. Bai, X. Tong, Y. Zhu et al. "Long-VLA: Unleashing Long-Horizon Capability of Vision Language Action Model for Robot Manipulation," *arXiv preprint arXiv:2508.19958*, 2025.

[14] H. Hang, F. Lin, Y Hu, S. Wang, Y, Gao. "CoPa: General Robotic Manipulation through Spatial Constraints of Parts with Foundation Models," *in 2024 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2024, pp. 9488-9495.

[15] H. Hang, M. Can, K. Tan et al. "GraphCoT-VLA: A 3D Spatial-Aware Reasoning Vision-Language-Action Modelfor Robotic Manipulation with Ambiguous Instructions". *arXiv preprint arXiv:2508.07650*, 2025.

[16] W. Zhang, T. Hu, Y. Qiao, H. Zhang, Y. Qin et al. "Chain-of-Action: Trajectory Autoregressive Modeling for Robotic Manipulation," *arXiv preprint arXiv:2506.09990*, 2025.

[17] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, S. Levine. "Robotic Control via Embodied Chain-of-Thought Reasoning," *arXiv preprint arXiv:2407.08693*, 2024.

[18] V. Karolis, M. Corbetta, M. Schotten. "The architecture of functional lateralisation and its relationship to callosal connectivity in the human brain," *Nature Communications*, vol. 10, 1417, 2019.

[19] X. Liang, J. Luo, Q. Bi et al. "Functional divergence between the two cerebral hemispheres contributes to human fluid intelligence," *Communications Biology*, vol. 8, 764, 2025.

[20] S. Liu, Z. Zeng, T. Ren et al. "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," *arXiv preprint arXiv:2304.02643*, 2023.

[21] A. Kirillov, E. Mintun, N. Ravi et al. "Segment Anything," *arXiv preprint arXiv:2303.05499*, 2023

[22] Y. Wei, M. Lin, Y. Lin et al. "AffordDexGrasp: Open-set Language-guided Dexterous Grasp with Generalizable-Instructive Affordance," *arXiv preprint arXiv:2503.07360*. 2025.

[23] N. Wake, A. Kanehira, K. Sasabuchi et al. "GPT-4V(ision) for Robotics: Multimodal Task Planning from Human Demonstration," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, 10567-10574, 2024.

[24] S. Nasiriany, S. Kirmani, T. Ding et al. "RT-Affordance: Affordances are Versatile Intermediate Representations for Robot Manipulation," *in 2025 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2025, pp. 8249-8257.

[25] Y. Shuai, R. Yu, Y. Chen, Z. Jiang, X. Song, N. Wang et al. "PUGS: Zero-shot Physical Understanding with Gaussian Splatting," *in 2025 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2025, pp. 4478-4485.

[26] M. Kim, P. Karl, T. Xiao et al. "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv preprint arXiv:2406.09246*. 2024.

[27] J. Wen, Y. Zhu, J. Li et al. "TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation," *arXiv preprint arXiv:2409.12514.* 2024.

[28] R. Sapkota, Y. Cao, I. Roumeliotis, M. Karkee. "Vision-Language-Action Models: Concepts, Progress, Applications and Challenges," *arXiv preprint arXiv:2505.04769.* 2025.

[29] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, D. Fox. "THE COLOSSEUM: A Benchmark for Evaluating Generalization for Robotic Manipulation," *arXiv preprint arXiv:2402.08191.* 2024.

[30] A. Wu et al. "In the Wild Ungraspable Object Picking with Bimanual Nonprehensile Manipulation," *in 2025 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2025, pp. 5669-5676.

[31] A.Yang, B. Yang, B. Hui et al. "Qwen2 Technical Report," *arXiv preprint arXiv:2407.10671.* 2024.

[32] A. Guo, B. Wen, J. Yuan et al. "HANDAL: A Dataset of Real-World Manipulable Object Categories with Pose Annotations, Affordances, and Reconstructions," *arXiv preprint arXiv:2308.01477.* 2023.

[33] B. Calli et al. "The YCB object and model set: Towards common benchmarks for manipulation research," *in International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.

[34] E. J.Hu, Y. Shen et al. "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685.* 2021.

[35] M. Liu, R. Shi, L. Chen et al. "One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion," *in 2024 IEEE Computer Vision and Pattern Recognition Conference (CVPR)*. IEEE, 2024, pp. 10072-10083.

[36] B. Wen, W. Yang et al. "Foundationpose: Unified 6d pose estimation and tracking of novel objects," *in 2024 IEEE Computer Vision and Pattern Recognition Conference (CVPR)*. 2024, pp. 17868-17879.

[37] J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Education, Inc, 2005.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford. "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[39] M. Viktor et al. "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[40] M. Oquab et al. "DINOv2: Learning Robust Visual Features without Supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[41] H. Fang, C. Wang, M. Gou, C. Lu. "GraspNet: A Large-Scale Clustered and Densely Annotated Dataset for Object Grasping," *arXiv preprint arXiv:1912.13470*, 2019.