Evo-1: Lightweight Vision-Language-Action Model with Preserved Semantic Alignment

Supplementary Material

1. Implementation Details

1.1. Training Details

All experiments in this work are conducted with distributed training on 8× NVIDIA A100 GPUs. Across all benchmarks, Evo-1 is optimized following the two-stage training paradigm introduced in the main paper, designed to ensure stable perception–action alignment while preserving the semantic structure of the pretrained VLM.

Two-stage optimization scheme. The first stage focuses on stabilizing the integration layers and the action head under a fixed visual-language embedding space. During this stage, the backbone is entirely frozen (`finetune_vlm = False`), while the integration module and the action expert remain trainable. This phase emphasizes learning a consistent alignment between vision-language representations and low-level control signals.

The second stage performs joint finetuning of all components. Starting from the final checkpoint of Stage 1, the backbone is unfrozen and optimized together with the integration layers and the action head. This stage enables Evo-1 to refine the semantic grounding of the frozen backbone and adapt it to the downstream control tasks.

Training pipeline. All datasets follow the LeRobot v2.1 data format, ensuring a unified structure for images, states, and actions across different embodiments. Image observations are uniformly resized to 448×448. State and action vectors are padded to a fixed 24-dimensional representation to accommodate embodiment differences while maintaining a consistent model interface. For action prediction, we adopt an action trunk size of $H = 50$, which serves as the horizon for the action generation process. Data augmentation remains active throughout both stages to enhance robustness.

In our Meta-World experiments, Stage 1 is trained for 10k steps with the VLM frozen, and Stage 2 proceeds for 65k steps with full-model finetuning enabled. Other benchmarks adopt the same overall scheme, with minor adjustments to training steps depending on dataset scale.

1.2. Hyperparameter Details

Tables 1 and 2 summarize the key optimization and model hyperparameters used in the Meta-World experiments, which reflect the default configuration of Evo-1 across most benchmarks.

Hyperparameter	Stage 1	Stage 2
Learning rate	1×10^{-5}	1×10^{-5}
Batch size	16	16
Max steps	10k	65k
Warmup steps	1k	1k
Gradient clipping	1.0	1.0
Weight decay	0.001	0.001
Log interval	10	10
Resume from Stage 1	No	Yes

Table 1. Optimization hyperparameters used for Meta-World training.

Setting	Value
Model Configuration	
Backbone	InternVL3-1B
Action head	FlowMatching
Transformer layers	8
Dropout	0.2
Input Configuration	
Image size	448
Use augmentation	True
State dimension	24 (padded)
Action dimension	24 (padded)
Horizon	50

Table 2. Model and input configurations for Meta-World experiments.

2. Details of Simulation Experiments

2.1. Details of Meta-World Benchmark

Task Setup. The Meta-World benchmark consists of 50 distinct robotic manipulation tasks designed for evaluating multi-task learning algorithms. For each task, the benchmark provides multiple task variations, such as different initial object positions or goal configurations, enabling the study of generalization across a broad task distribution rather than narrow parametric changes. The goal of the benchmark is to support the development of algorithms capable of acquiring new tasks more efficiently by leveraging prior experience. To this end, the collection of 50 manipulation tasks forms a sufficiently diverse task distribution in-

Difficulty	Task	Prompt
Easy	button-press-topdown	Press a button from the top
	button-press-topdown-wall	Bypass a wall and press a button from the top
	button-press	Press a button
	button-press-wall	Bypass a wall and press a button
	coffee-button	Push a button on the coffee machine
	dial-turn	Rotate a dial 180 degrees
	door-close	Close a door with a revolving joint
	door-lock	Lock the door by rotating the lock clockwise
	door-unlock	Unlock the door by rotating the lock counter-clockwise
	door	Open a door with a revolving joint
	drawer-close	Push and close a drawer
	drawer-open	Open a drawer
	faucet-close	Rotate the faucet clockwise
	faucet-open	Rotate the faucet counter-clockwise
	handle-press-side	Press a handle down sideways
	handle-press	Press a handle down
	handle-pull-side	Pull a handle up sideways
	handle-pull	Pull a handle up
	lever-pull	Pull a lever down 90 degrees
	peg-unplug-side	Unplug a peg sideways
	plate-slide-back-side	Get a plate from the cabinet sideways
	plate-slide-back	Get a plate from the cabinet
	plate-slide-side	Slide a plate into a cabinet sideways
	plate-slide	Slide a plate into a cabinet
	reach	Reach a goal position
	reach-wall	Bypass a wall and reach a goal
	window-close	Push and close a window
	window-open	Push and open a window
Medium	basketball	Dunk the basketball into the basket
	bin-picking	Grasp the puck from one bin and place it into another bin
	box-close	Grasp the cover and close the box with it
	coffee-pull	Pull a mug from a coffee machine
	coffee-push	Push a mug under a coffee machine
	hammer	Hammer a screw on the wall
	peg-insertion-side	Insert a peg sideways
	push-wall	Bypass a wall and push a puck to a goal
	soccer	Kick a soccer into the goal
	sweep-into-goal	Sweep a puck into a hole
	sweep	Sweep a puck off the table
Hard	hand-insert	Insert the gripper into a hole
	nut-assembly	Pick up a nut and place it onto a peg
	pick-out-of-hole	Pick up a puck from a hole
	pick-place	Pick and place a puck to a goal
	push-back	Push the puck back to a goal
	push	Push the puck to a goal
Very Hard	nut-disassemble	Pick a nut out of a peg
	pick-place-wall	Pick a puck, bypass a wall and place the puck
	shelf-place	Pick and place a puck onto a shelf
	stick-pull	Grasp a stick and pull a box with the stick
	stick-push	Grasp a stick and push a box using the stick

Table 3. Meta-World Tasks Grouped by Difficulty

tended to encourage policies to generalize to entirely new, held-out tasks.

For each of the 50 Meta-World manipulation tasks, we collect 50 high-quality demonstration trajectories. All demonstrations are obtained under the same observation and action interface described in the benchmark, with randomized initial object and goal configurations for every rollout to ensure sufficient intra-task variability. Across all 50 tasks, this results in a total of 2,500 demonstrations. The complete list of task names and their corresponding descrip-

tions is provided in Table 3, which enumerates all 50 tasks used in our simulation experiments.

Meta-World Execution Examples. Figure 3 presents several representative execution trajectories produced by our policy on different Meta-World tasks. Each example highlights the overall manipulation process, showing how the agent observes the scene, generates appropriate actions, and completes the task under varying object positions and environmental conditions.

2.2. Details of LIBERO Benchmark

Task Setup. LIBERO is a benchmark designed to evaluate lifelong learning in robot manipulation, focusing on the transfer of both declarative and procedural knowledge. The tasks in LIBERO are categorized into four primary task suites as shown in Table 4: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Each suite aims to test different aspects of robot learning, including how well the robot can generalize learned knowledge to new situations, transfer spatial and object-specific information, and adapt to diverse task goals.

1. **LIBERO-Spatial.** This suite tests the robot's ability to understand and manipulate spatial relationships between objects. In tasks from this suite, the robot is tasked with manipulating objects (such as placing a bowl on a plate), where the challenge lies in learning how objects relate spatially in the environment.
2. **LIBERO-Object.** Tasks in this suite test the robot's ability to recognize and manipulate different objects. The primary challenge is in learning to handle various objects, each with different shapes, sizes, and properties.
3. **LIBERO-Goal.** These tasks focus on goal-directed manipulation where the robot must learn to achieve specific task goals, such as placing objects in predefined locations. The goal is usually fixed, but the actions to reach the goal may vary.
4. **LIBERO-Long.** This suite combines elements from the previous three, introducing more complex tasks that require a combination of declarative and procedural knowledge. The LIBERO-100 suite is more diverse, with a larger set of tasks, enabling a thorough evaluation of the robot's ability to perform a wide range of tasks and generalize across various environments. The suite includes both short-horizon tasks and long-horizon tasks, with 90 tasks requiring quick decision-making and 10 tasks designed to test long-term planning and execution.

The goal of these tasks is not only to test the robot's ability to perform specific tasks but also to understand how well the robot can transfer learned skills across different task settings, handling new object types, new goals, and different spatial arrangements.

LIBERO Execution Examples. Figure 4 presents qualitative examples of our policy executing several representative tasks from the LIBERO benchmark. The rollouts illustrate how the agent interprets the language instruction, identifies the relevant objects, and performs the required manipulations under different spatial arrangements and initial states. These examples highlight the model's ability to produce reliable and consistent behaviors across a variety of LIBERO task settings.

2.3. Details of RoboTwin Benchmark

Task Setup. To evaluate the manipulation performance of Evo-1 on dual-arm robot, we conduct experiments on a subset of the RoboTwin 2.0 benchmark, a diverse manipulation suite designed to assess precision control, multi-stage manipulation, and goal-directed object placement. Among the full set of RoboTwin tasks, we select four representative manipulation scenarios:

1. **Click Alarmclock.** The robot is required to press the top button of an alarm clock. This task evaluates fine-grained end-effector positioning and controlled vertical actuation necessary for successful clicking.
2. **Dump Bin Bigbin.** The robot grasps a small bin, lifts it, and dumps its contents into a larger bin. This task requires stable grasping, lifting, rotation, and accurate release sequencing.
3. **Place Bread Basket.** The robot must pick up a piece of bread and place it into a basket. This involves handling lightweight objects and performing precise insertion motions into a confined receptacle.
4. **Place Can Basket.** The robot grasps a cylindrical can and places it inside a designated basket. The task assesses grasp stability on rigid objects and accurate placement.

For each of the four tasks, we collect 50 high-quality demonstration trajectories following the RoboTwin data collection protocol. These tasks jointly cover precision contact behaviors, multi-step motion sequences, and constrained-object placement, enabling a comprehensive assessment of Evo-1's manipulation generalization capabilities.

RoboTwin Execution Examples. Figure 5 illustrates several execution examples of our policy on the RobotWin benchmark. The visualized trajectories show how the agent responds to task instructions, detects the target items in cluttered scenes, and carries out the required placement or relocation actions across different object and bin configurations. These examples demonstrate that the model can maintain stable and purposeful behaviors even when facing diverse layouts and visually complex environments.

Task Category	Task Instruction
LIBERO-Spatial	Pick up the black bowl between the plate and the ramekin and place it on the plate. Pick up the black bowl next to the ramekin and place it on the plate. Pick up the black bowl from table center and place it on the plate. Pick up the black bowl on the cookie box and place it on the plate. Pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate. Pick up the black bowl next to the ramekin and place it on the plate. Pick up the black bowl next to the cookie box and place it on the plate. Pick up the black bowl on the stove and place it on the plate. Pick up the black bowl next to the plate and place it on the plate. Pick up the black bowl on the wooden cabinet and place it on the plate.
LIBERO-Object	Pick up the orange juice and place it in the basket. Pick up the cream cheese and place it in the basket. Pick up the salad dressing and place it in the basket. Pick up the BBQ sauce and place it in the basket. Pick up the ketchup and place it in the basket. Pick up the tomato sauce and place it in the basket. Pick up the butter and place it in the basket. Pick up the milk and place it in the basket. Pick up the chocolate pudding and place it in the basket. Pick up the orange juice and place it in the basket.
LIBERO-Goal	Open the middle drawer of the cabinet. Put the bowl on the stove. Put the wine bottle on top of the cabinet. Open the top drawer and put the bowl inside. Put the bowl on top of the cabinet. Push the plate to the front of the stove. Put the cream cheese in the bowl. Turn on the stove. Put the black bowl on the top drawer of the cabinet. Put the wine bottle on the rack.
LIBERO-Long	Put both the alphabet soup and the tomato sauce in the basket. Put both the cream cheese box and the butter in the basket. Turn on the stove and put the moka pot on it. Put the black bowl in the bottom drawer of the cabinet and close it. Put the white mug on the left plate and put the yellow and white mug on the right plate. Pick up the book and place it in the back compartment of the caddy. Put the white mug on the plate and put the chocolate pudding to the right of the plate. Put both the alphabet soup and the cream cheese box in the basket. Put both moka pots on the stove. Put the yellow and white mug in the microwave and close it.

Table 4. LIBERO Task Instructions

3. Details of Real World Experiments

3.1. Robot Setup

As shown in Figure 1, our real-world system is built on a 6-DoF xArm6 manipulator equipped with two RGB cameras: a wrist-mounted camera that provides close-range, egocentric observations of object–gripper interactions, and a fixed environment camera that captures global scene context across the entire tabletop workspace. The robot is controlled through the official xArmAPI.

3.2. Data Collection

We collect real-world demonstrations following the LeRobot 2.1 data specification. Each episode consists of synchronized observations recorded at 30Hz, including two RGB image streams: `image_1` from the fixed environment camera and `image_2` from the wrist-mounted camera. The proprioceptive state is represented by the robot's absolute joint angles, and actions are stored using the same absolute joint angle to ensure consistent replay and supervision. All demonstrations are gathered using this unified multi-view setup, which provides both global scene context and fine-grained manipulation details that are essential for learning robust visuomotor policies.

3.3. Task Setup and Success Criteria

We evaluate our system on four real-world manipulation tasks. For each task, we define a clear and objective suc-

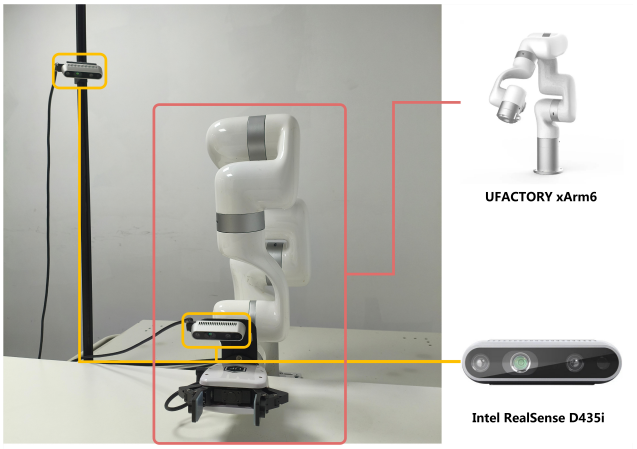

Figure 1. **Robot setup of Real-World Experiments.**

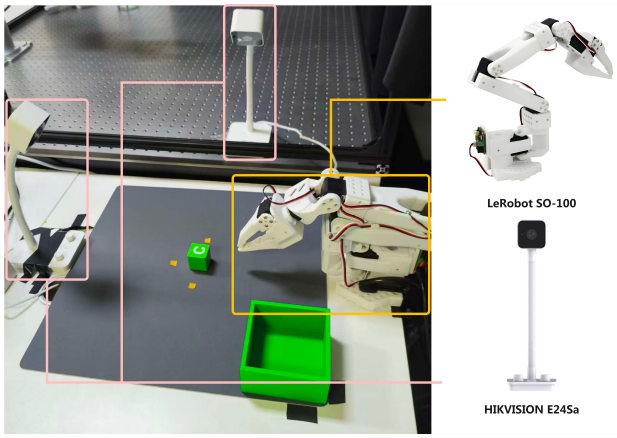

Figure 2. **Robot setup of LeRobot SO-100 Experiments.**

cess criterion:

1. **Pick and Place Can.** The robot must grasp a beverage can from varying initial positions and place it into a designated white box. A trial is considered successful if the can is fully inside the box at the end of the episode and remains stably settled without rolling out.
2. **Pour Foam from Cup.** The robot lifts a foam-filled cup and rotates it to pour the foam into the white box. Success is achieved if the majority of the foam is correctly poured into the box such that a visible accumulation of foam is present inside, with minimal spillage outside the target area.
3. **Hand Delivery.** The robot must grasp a beverage can and hand it over to a human operator. A trial is considered successful if the can is placed securely into the human hand, with the operator able to hold it without needing to adjust or chase the object.
4. **Can Stacking.** The robot grasps a beverage can and stacks it onto another identical can placed on the table. Success is defined as the top can remaining stably stacked for at least two seconds without sliding or toppling.

3.4. Real Wolrd Execution Examples

Figure 6 provides qualitative examples of our policy performing the four real-world tasks described above. The visual sequences illustrate how the robot perceives the scene through live camera inputs, identifies the relevant objects, and executes the required manipulation behaviors under natural variations in lighting, object placement, and human interaction. Across the different task types—including container placement, pouring, handover, and stacking—the system exhibits stable motion generation and consistent task completion, demonstrating reliable transfer of the learned policy to the physical xArm6 platform.

4. Deployment on LeRobot SO-100 robot

Task Setup. We deploy our policy on the LeRobot SO-100 platform, a compact desktop manipulator designed for fine-grained tabletop manipulation. In our real-world task as shown in Figure 2, a small cube is manually placed at varying positions on the table. The robot must visually locate the cube, approach it with a precise grasp, lift it, and place it into a fixed container positioned on the same tabletop. This setup evaluates the system's ability to perform reliable pick-and-place manipulation on small objects under mild variation in object pose and scene configuration.

SO-100 Execution Examples. Figure 7 showcases representative executions of our policy on the SO-100 desktop manipulation setup. In these examples, the robot observes the tabletop scene, identifies the target cube, and performs a smooth pick-and-place motion to deposit the object into the designated container. The rollouts illustrate that the system can handle natural variations in cube position and orientation while maintaining stable grasps and accurate placements. Overall, the results confirm that the learned policy transfers effectively to the SO-100 robot and produces reliable performance in real-world tabletop manipulation tasks.

Figure 3. **Examples of Meta-World Execution.**

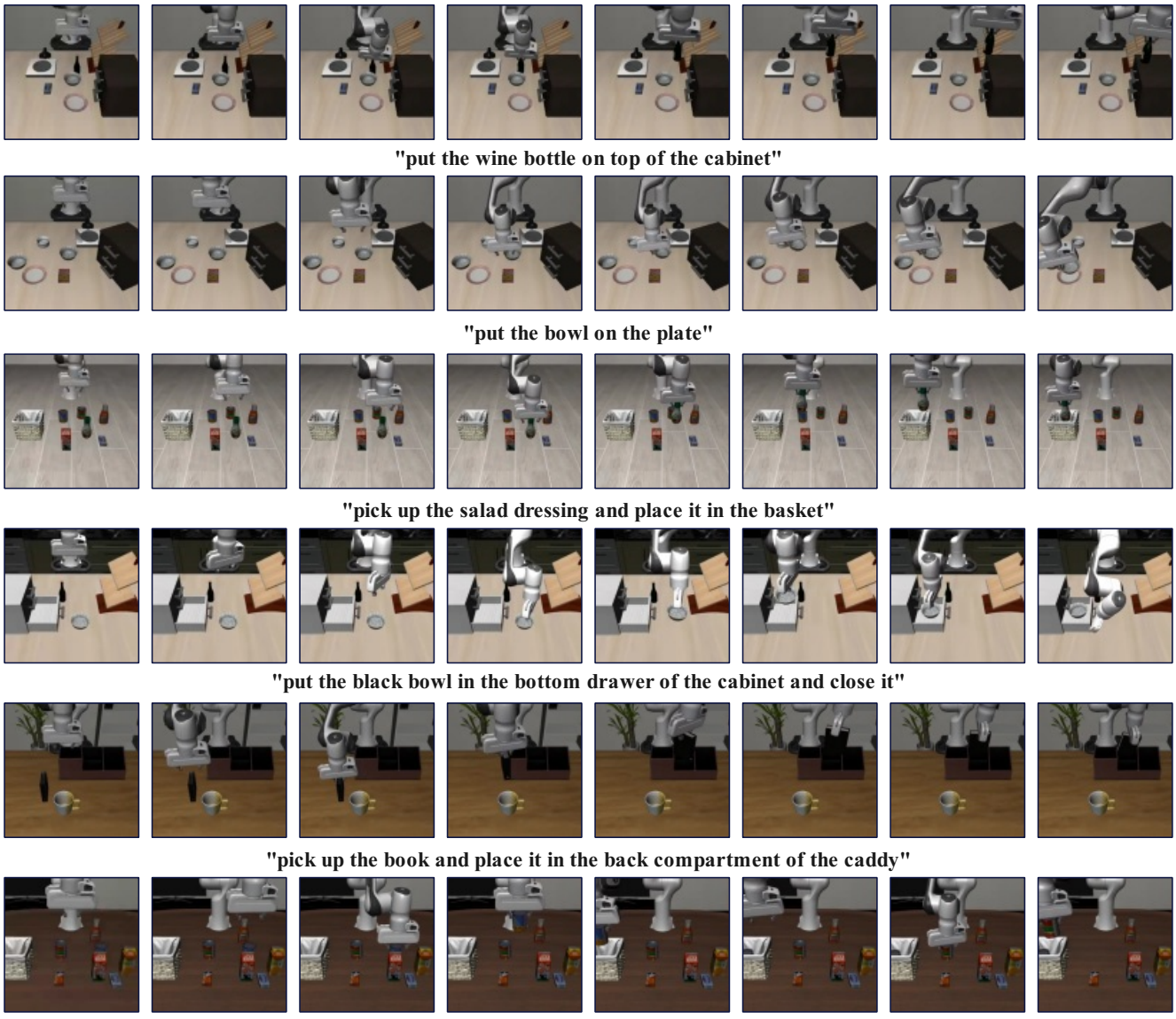

Figure 4. **Examples of LIBERO Execution.**

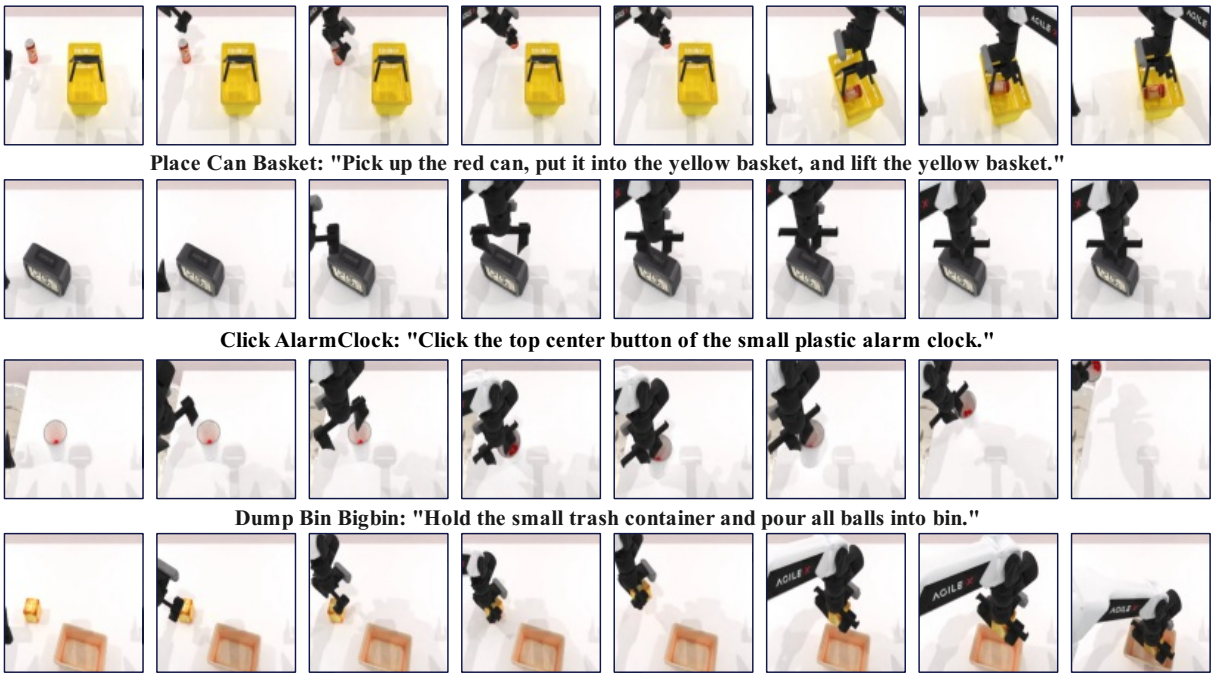

Figure 5. **Examples of RoboTwin Execution.**

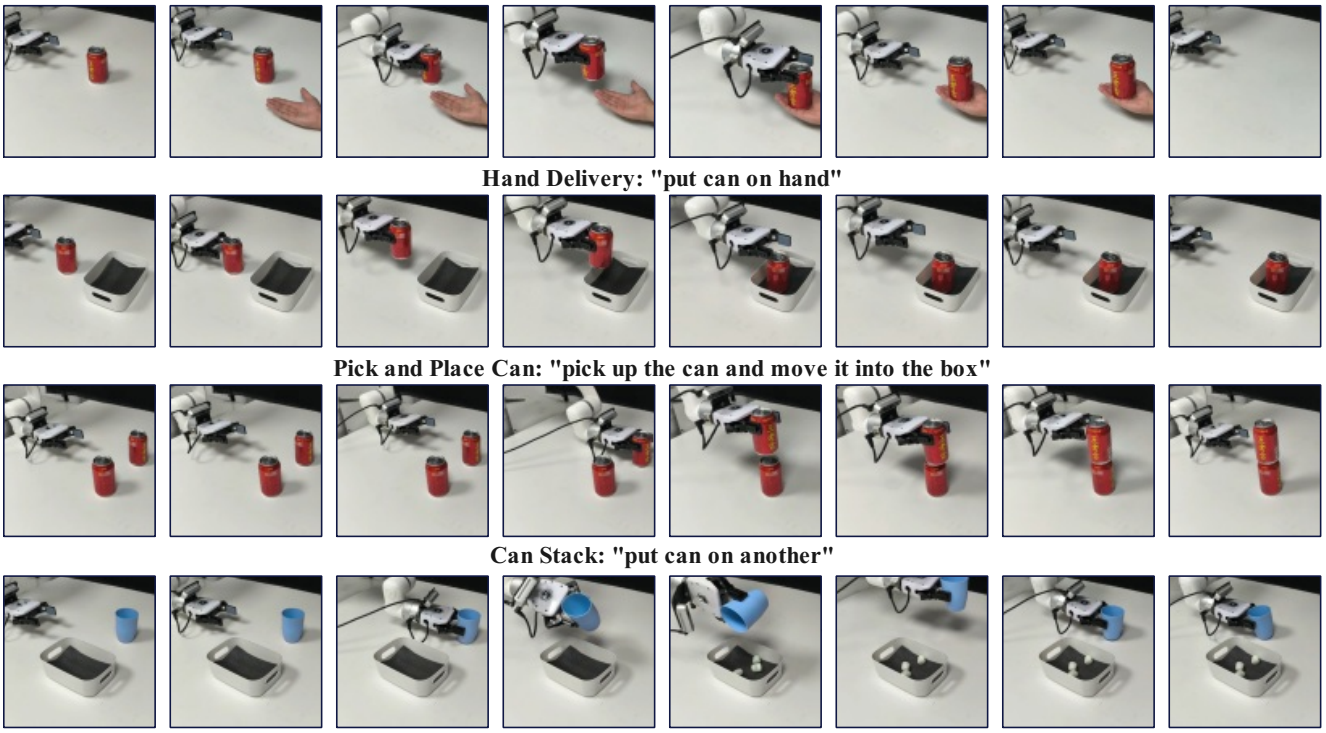

Figure 6. **Examples of Real-World Execution.**

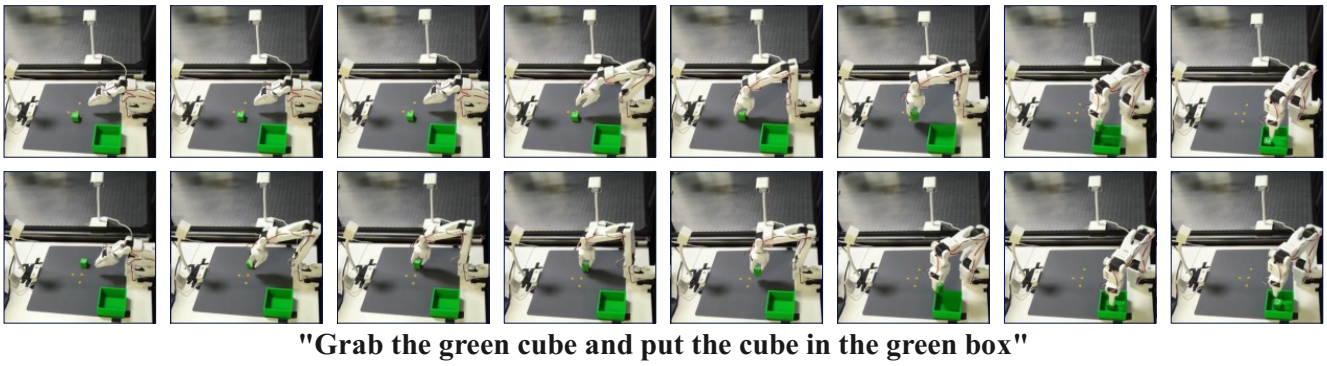

Figure 7. **Examples of SO-100 Execution.**