CoA-VLA: Improving Vision-Language-Action Models via Visual-Textual Chain-of-Affordance

Supplementary Material

6.1. Video Demo

We provide a video recording in the supplementary material.

Table 3. Summarization for the number of demonstrations and average trajectory length for our real-world tasks.

#	Task	# of Demonstrations	Average Trajectory Length
1	PlaceCar	89	301.8
2	PlaceBread	102	113.2
3	NailHammer	80	182.8
4	PourTea	91	429.4
5	CleanTrash	185	114.3
6	WipeWater	62	199.9
7	HangCup	83	131.4

6.2. Evaluation Tasks

In this section, we give a detailed description of the evaluated tasks that we discussed in Table 1. We provide the number of demonstrations for each task and the average trajectory length in Table 3.

- **PlaceCar.** We randomly place the toy car on the right side of the drawer. The model is asked to pick up the toy car, put it into the drawer, and eventually close the drawer. This is a long-horizon task that requires multiple steps of action.
- **PlaceBread.** The model needs to pick up the bread and place it on an empty spot on the plate, avoiding placing it on the fruit. The bread is randomly placed on the table. The model needs to pick up the bread and place it on the empty spot on the plate.
- **NailHammer.** We evaluate the model's proficiency in utilizing tools effectively by assessing its ability to perform a sequence of precise actions with a hammer. The model must first identify the correct grasp point on the hammer, ensuring a stable and ergonomic grip suitable for controlled operation. It must then carefully pick up the hammer without causing it to topple or disturb its surroundings. Once the hammer is securely held, the model is tasked with driving a nail into a designated spot with precision.
- **PourTea.** In this task, the robot is required to perform a sequence of actions involving a tea cup and a teapot. First, the robot must place the tea cup onto the tea tray. Next, it needs to pick up the teapot and pour tea into the teacup. Both the tea cup and the teapot are randomly positioned within a defined range on the table. A key aspect of the task is the robot's ability to accurately grasp the teapot by its stem. To ensure consistency during data collection, the tea pot's stem is always oriented facing the robot, simplifying the grasping process while still challenging the model's precision and manipulation skills.
- **CleanTrash.** In this task, the robot is required to perform a sequence of actions to clean up trash on a table. The task has two distinct scenarios. In the first scenario, with no obstacles, the robot must identify and pick up the randomly placed trash, then deposit it into the trash bin. The trash items are distributed across the table in a random manner. In the second scenario, a flower pot is placed on the table as an obstacle. The robot must avoid colliding with the flower pot while picking up the trash and placing it into the trash bin. The trash's location remains random, and the robot must navigate carefully to avoid knocking over the flower pot during the cleanup process. A key aspect of this task is the robot's ability to accurately avoid the flower pot while maintaining efficiency in picking up and discarding the trash.
- **WiperWater** In this task, the robot is required to clean up water from a table by using a sponge. The sponge is placed on the right side of the table, and the robot must pick it up and use it to wipe the water from the surface, moving from right to left. During this process, the robot must avoid any objects placed on the table, such as vases, cups, boxes, and other items. A key challenge in this task is the robot's ability to manipulate the sponge effectively while navigating around the obstacles without causing any collisions, ensuring that the entire table is cleaned efficiently. The robot's precision in both grasping the sponge and avoiding the table items is critical for completing the task successfully.
- **HangCup** In this task, the robot is required to pick up cups that are randomly scattered on the table and hang them on a cup rack. The robot must handle the cups carefully to avoid damaging them and ensure that the rack is not disturbed or knocked over during the process. The task challenges the robot's precision in both grasping the cups and placing them securely on the rack while maintaining stability in the environment. Successful completion relies on careful manipulation and accurate placement.

Setup for visual generalization. In this scenario, we evaluate the model's robustness and its ability to generalize visual perception across diverse and challenging environmental conditions. The robot is tasked with performing ma-

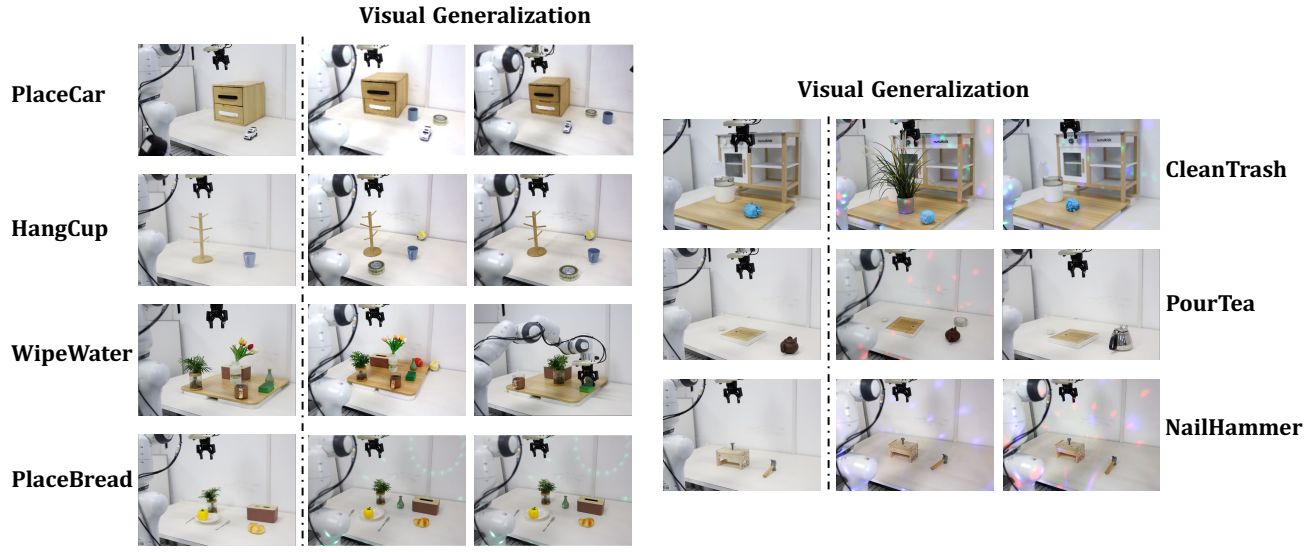

Figure 6. **Visual Generalization.** We evaluate each method on multi-task learning and visual generalization, which includes handling additional distractors and interference from colored light. We also test the ability to grasp objects of the same type but with varying shapes, such as different teapots, as well as teapots in different orientations.

nipulation tasks while navigating visual complexities such as randomly placed distractors, varying lighting conditions, and a visually cluttered, colorful background. These challenges are designed to test the model's capability to stay focused on the primary task, effectively filter out irrelevant visual distractions, and adapt to dynamic and unpredictable visual environments. The objective is to ensure the robot can consistently and accurately identify and interact with target objects, even under significant deviations from typical operational settings.

6.3. Details for Real Robot Experiments

We train our method in a multi-task setting without relying on pre-trained weights from DiffusionVLA. Instead, we leverage our constructed dataset for pre-training. Specifically, we initialize the learning rate at 2e-5 and maintain a fixed learning rate throughout the pre-training phase, which spans 5 epochs. During this stage, the parameters of the pre-trained Vision-Language Model (VLM) are frozen, and LoRA is employed to fine-tune the model. For fine-tuning, we adopt a similar approach, starting with an initial learning rate of 2e-6. However, in this phase, we apply a cosine learning rate decay schedule and train the model for an additional 5 epochs. This training strategy ensures both effective adaptation and stability across pre-training and fine-tuning stages, optimizing the model for multi-task performance.

For the baselines, we generally adopt a consistent training strategy. In the case of OpenVLA, the vanilla implementation utilizes only a single camera view. To extend this, we incorporate all three camera views, feeding each view into the same visual encoder and concatenating their outputs for processing. We leverage OpenVLA's pre-trained weights and trains for 20 epochs, as we observe that it typically requires a longer training time to achieve convergence. For the Diffusion Policy, we utilize DistilBERT to process language instructions, following an approach similar to YAY [41]. As for DiffusionVLA, we employ their pre-trained weights and construct a reasoning dataset using their data construction pipeline to maintain consistency with their methodology. To ensure fair evaluation, we use the final checkpoints of all models, including ours, avoiding any form of cherry-picking. This approach allows for a robust comparison and highlights the performance differences across the various models.

6.4. Details for LIBERO Simulation

LIBERO is a robot learning benchmark comprising over 130 language-conditioned manipulation tasks. We follow the setting as in OpenVLA [22] open-sourced code and test on four task suites: LIBERO-Spatial, LIBERO-Goal, LIBERO-Object, and LIBERO-Long.

Each suite includes 10 distinct tasks with 50 demonstrations per task. Each task suite emphasizes unique challenges in imitation learning: LIBERO-Goal features tasks with similar object categories but different goals. LIBERO-Spatial requires policies to adapt to varying spatial arrangements of the same objects. LIBERO-Object keeps the layout consistent while changing the objects. During experimentation, our method uses a static camera, and a wrist-mounted camera all methods are evaluated across 1500 trials in total. We filter out the failure data and increase the image resolution to 224×224. The affordance data is gen-

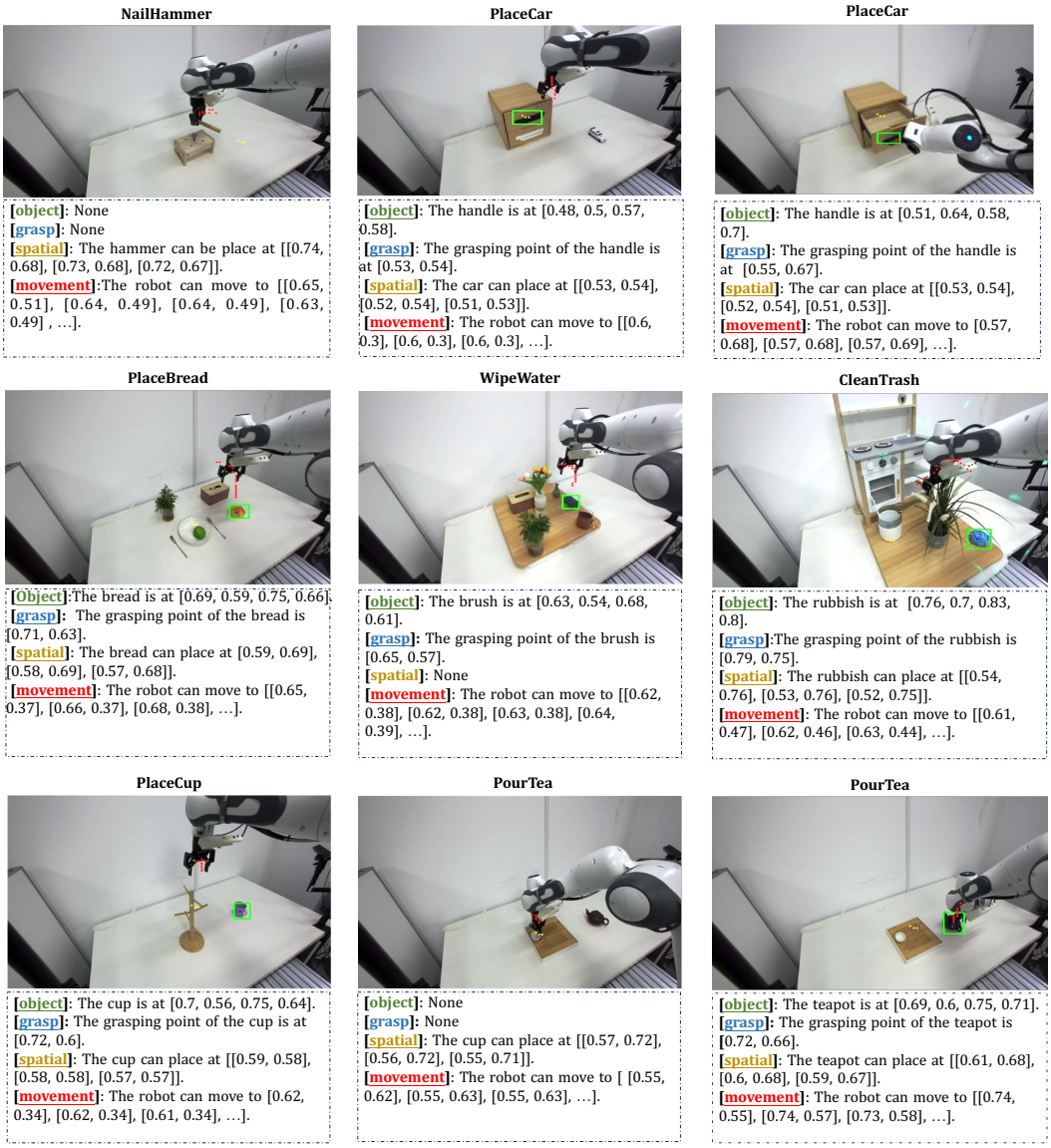

Figure 7. More detailed examples of successful Chain-of-Affordance.

Table 4. **Ablation study on visual affordance and textual affordance.** Our experiments demonstrate that both affordance are important for VLA.

Method / Task	LIBERO-Spatial Success Rate (↑)	LIBERO-Object Success Rate (↑)	LIBERO-Goal Success Rate (↑)	LIBERO-Long Success Rate (↑)	Average Success Rate (↑)
CoA-VLA	$85.3 \pm 0.9\%$	$93.1 \pm 1.0\%$	$85.8 \pm 0.9\%$	$55.0 \pm 1.2\%$	$79.8 \pm 0.5\%$
w/o visual affordance	$84.3 \pm 0.5\%$	$91.5 \pm 0.7\%$	$83.9 \pm 1.0\%$	$54.6 \pm 1.2\%$	$78.6 \pm 0.9\%$
w/o textual affordance	$81.6 \pm 0.7\%$	$89.8 \pm 0.9\%$	$80.1 \pm 1.0\%$	$52.5 \pm 0.9\%$	$76.0 \pm 0.9\%$

erated using our proposed pipeline for data in LIBERO. In Table 2, we directly cite the results of Diffusion Policy, Octo, and OpenVLA from OpenVLA's paper. Therefore, to ensure all methods are evaluated fairly, we evaluated our methods across 500 trials for each task suite, and the reported performance is the average success rate over three random seeds. We use the same test data as in OpenVLA. For the baseline ScaleDP, except for using all camera views,

Table 5. **Ablation study on dynamic affordance selection.** Removing dynamic affordance selection causes introduction of redundant affordance into the learning process, which cause the model to perform even worse than the baseline without it.

Method / Task	LIBERO-Spatial Success Rate (↑)	LIBERO-Object Success Rate (↑)	LIBERO-Goal Success Rate (↑)	LIBERO-Long Success Rate (↑)	Average Success Rate (↑)	Inference Speed
CoA-VLA	$85.3 \pm 0.9\%$	$93.1 \pm 1.0\%$	$85.8 \pm 0.9\%$	$55.0 \pm 1.2\%$	$79.8 \pm 0.5\%$	6Hz
- dynamic affordance selection	$85.1 \pm 0.9\%$	$92.4 \pm 1.0\%$	$85.2 \pm 1.0\%$	$55.2 \pm 1.1\%$	$79.5 \pm 1.0\%$	1Hz

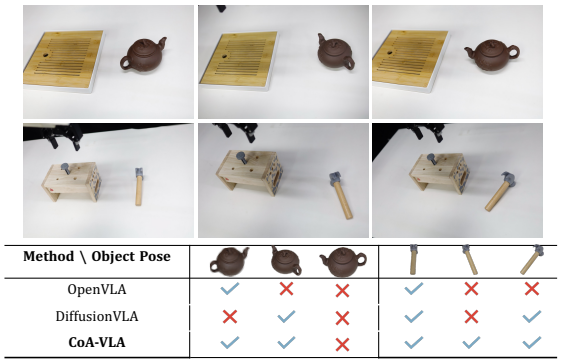

Figure 8. **Generalization on object pose.** CoA can pick up objects with unseen poses, benefiting from grasp affordance.

all other implementations kept the same.

7. More Experiments

7.1. Ablation Study on Visual-Textual Affordance

Our primary contribution lies in the introduction of textual affordances and visual affordances, paired with a novel visual-textual co-injection module designed to synergistically integrate these modalities into policy learning. To validate their individual and combined efficacy, we conduct a systematic ablation study (Table 4) on the LIBERO robotic task benchmark. Our key finding is that both textual and visual affordances are critical to model performance. Removing either modality leads to significant degradation in task success rates. While both modalities contribute uniquely, textual affordances exhibit stronger influence on policy optimization. We hypothesize that this stems from language's inherent capacity to encode task-specific semantics (e.g., "pour-able" or "graspable"), which provides clearer optimization signals compared to visual features that require implicit spatial grounding. These results underscore the importance of our co-injection module, which dynamically balances and fuses multimodal affordances to maximize policy robustness in diverse environments.

7.2. Ablation Study on Dynamic Affordance Selection

Utilizing all affordances can be computationally expensive and time-consuming. Therefore, we introduce a dynamic affordance selection mechanism. This approach focuses on selectively utilizing only the most relevant affordances at each time step. As demonstrated in Table 5, our method outperforms a baseline model that employs all affordances indiscriminately. Surprisingly, using all affordances results in a lower average success rate compared to our dynamic selection approach. We hypothesize that the irrelevant affordances introduce noise during the optimization process, hindering the model's learning ability. To further analyze the impact of dynamic selection, we measured inference speed on an Nvidia 3090 GPU. We averaged the running time over all tasks, with each task measured across 5 trials. Our results show that utilizing all affordances significantly impacts inference speed, causing the model to run 6 times slower than our proposed method. This highlights the substantial efficiency gains achieved through dynamic affordance selection

7.3. Generalization to Unseen Object Pose

We assessed CoA-VLA's ability to generalize to previously unseen object orientations, as illustrated in Figure 8. Our evaluation focused on two objects: a hammer and a teapot. In the training phase, both objects were consistently presented with their handles oriented vertically relative to the robot. To test the model's generalization capabilities, we introduced novel poses that were absent from the training data, challenging CoA-VLA to grasp these objects in unfamiliar orientations. We observed that CoA-VLA successfully managed most scenarios, demonstrating a remarkable ability to adapt to new object poses even without explicit training on these orientations. In contrast, OpenVLA succeeded only in the simplest cases, struggling with more complex orientations. However, when the objects were positioned horizontally relative to the robot, all models, including CoA-VLA, were unsuccessful in achieving a stable grasp. Despite this limitation, our grasp affordance approach shows promising results, enabling CoA-VLA to handle a wide range of novel object poses.