ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Hierarchical LLM-VLA-Controller Integration for Task Generalization

***Abstract*— Vision-Language-Action (VLA) models often struggle with generalization due to their tendency to memorize training data rather than understanding task semantics. This paper proposes a hierarchical framework that integrates Large Language Models (LLMs) with VLA models to overcome these limitations. By leveraging GPT-4o as a high-level planner, our system decomposes complex instructions into atomic sub-tasks executable by a low-level VLA. We introduce a "Home Pose Controller" between sub-tasks to ensure physical stability. Experimental results on the LIBERO-10 benchmark demonstrate that our approach achieves a 90% success rate on decomposable tasks, significantly outperforming the 9% baseline of the standalone VLA model.**

I. OVERVIEW

Current VLA models, such as OpenVLA, map visual and textual inputs directly to actions. However, they lack the reasoning capabilities required for complex, multi-step instructions or abstract goals. This leads to a sharp performance drop when encountering task combinations not seen during training. In this work, we argue that a hierarchical structure—using an LLM for reasoning and a VLA for execution—can bridge this gap. Our framework interprets high-level intent, plans a sequence of atomic actions, and maintains stability through a dedicated pose controller, resulting in robust generalization across diverse manipulation tasks.

II. METHODOLOGY

Our system employs a two-tier hierarchy: a high-level LLM agent (GPT-4o) and a low-level VLA executor (OpenVLA-oft).

A. LLM Agent (GPT-4o)

The LLM acts as a **semantic bridge** with two key roles: (1) decomposing complex instructions into **atomic sub-tasks**, and (2) interpreting abstract instructions into **structured task plans**. As shown in Fig. 1, the agent reasons through complex or ambiguous goals to generate a sequential list of executable primitive tasks.

B. Prompting Strategy

To effectively leverage GPT-4o as a high-level planner, we designed a structured system prompt that defines the agent's role and constraints. The LLM is instructed to act as a robot task planner that coordinates a low-level VLA controller. A key constraint in the prompt is the mandatory insertion of a "Move to Home Pose" command between any two manipulation sub-tasks to ensure transition stability. An example of the prompt structure and the resulting decomposition is shown in Table I.

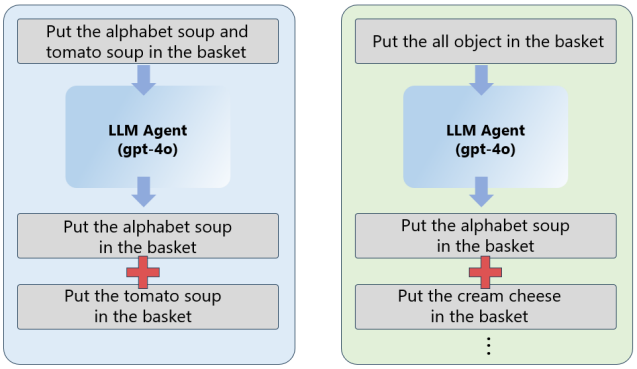

Fig. 1. LLM Agent reasoning: Decomposing high-level instructions into atomic sub-tasks (Case 1: multi-object, Case 2: abstract goals).

TABLE I
EXAMPLE OF LLM PROMPT AND SUB-TASK DECOMPOSITION

Component	Content
System Role	You are an intelligent robot task planner. Your goal is to decompose complex user instructions into atomic sub-tasks.
Tool Access	You have access to a VLA controller capable of executing primitive tasks: [Put X in basket, Close X, Pick up X, etc.].
Constraint	Between every physical manipulation task, you MUST insert a "Move to Home Pose" command to reset the robot state.
User Input	"Put both the alphabet soup and the tomato sauce in the basket."
Output List	(1) Put the alphabet soup in the basket (2) Move to Home Pose (3) Put the tomato soup in the basket (4) Move to Home Pose

C. Vision-Language-Action (OpenVLA-oft)

The VLA executes the sub-tasks planned by the LLM Agent, fine-tuned on **15 primitive tasks** from the **LIBERO-90** dataset. Each sub-task is **sequentially executed** in the order specified by the LLM.

D. Home Pose Controller

The Home Pose Controller maintains the robot's **initial joint configuration** and returns to it between consecutive sub-tasks. Without this module, physical discontinuities between sub-task transitions cause execution failures. The LLM Agent explicitly incorporates **home pose commands** into the task plan to ensure smooth and stable sequential execution (Fig. 2).

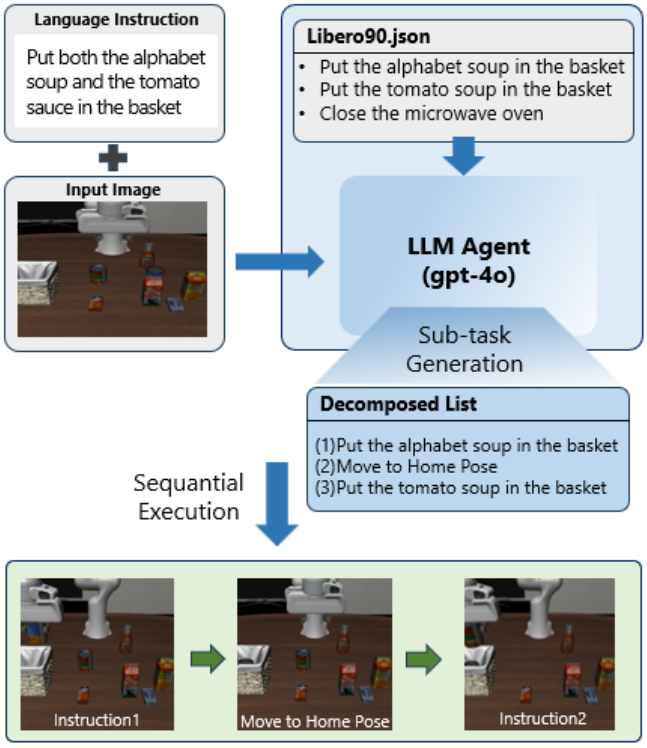

Fig. 2. Proposed hierarchical LLM-VLA-Controller framework architecture.

III. SIMULATION AND RESULTS

We evaluate our framework in the **MuJoCo** simulation environment using the **LIBERO-10** benchmark. The LLM Agent is powered by **GPT-4o**, and the VLA executor is **OpenVLA-oft**, initialized from weights pre-trained on the **LIBERO-90** dataset. We conducted 10 trials for each of the 10 tasks (Total 100 episodes).

TABLE II
DETAILED SUCCESS RATES ON LIBERO-10 TASKS

No.	Task Name	Succ. Rate
0	Put alphabet soup & tomato sauce in basket	100% (10/10)
1	Put cream cheese & butter in basket	100% (10/10)
2	Turn on stove and put moka pot on it	0% (0/10)
3	Put black bowl in bottom drawer & close it	70% (7/10)
4	Put mugs on left and right plates	100% (10/10)
5	Pick up book and place it in back of caddy	90% (9/10)
6	Put white mug on plate & pudding to right	90% (9/10)
7	Put alphabet soup & cream cheese in basket	80% (8/10)
8	Put both moka pots on the stove	0% (0/10)
9	Put mug in microwave and close it	0% (0/10)

A. Analysis

While the baseline OpenVLA-oft-7B model achieved only 9%, our hierarchical framework reached 63%. We observed that Tasks 2, 8, and 9 resulted in 0% success because they were structurally non-decomposable into the LIBERO-90 primitive commands. For instance, Task 2 (turning on a stove) requires a specific rotary motion that was not present in the 15 primitive tasks of the training set. Since our framework relies on mapping high-level intent to known primitives, the absence of a corresponding low-level policy for 'rotating a knob' led to execution failures.

TABLE III
SUCCESS RATE COMPARISON SUMMARY

Model Name	Success Rate
Libero90 pretrained model (Baseline)	9% (9/100)
Libero90 + LLM (Ours)	63% (63/100)
Libero90 + LLM (Excl. Task 2, 8, 9)	90% (63/70)

A critical observation is the role of the Home Pose Controller. The significant drop in performance without resetting the pose indicates that current VLA models largely rely on memorized trajectories from specific starting configurations. By enforcing a Home Pose, we effectively align the environmental state with the VLA's learned starting conditions, providing strong evidence that standalone VLAs still struggle with semantic understanding and favor visual-motor memorization.

IV. CONCLUSION

This research demonstrates that integrating LLMs as planners with a Home Pose Controller significantly enhances generalization in robot manipulation. This approach achieved a 90% success rate on decomposable tasks, a substantial improvement over the 9% baseline. Future work will focus on transitioning to a closed-loop system where the LLM can detect execution failures in real-time and dynamically adjust plans through a self-feedback mechanism, leading to a more resilient and autonomous embodied AI system.

REFERENCES

[1] M. J. Kim, C. Finn, and P. Liang, "Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success," in *Proc. CVPR*, 2025.
[2] J. Liang et al., "Code as Policies: Language Model Programs for Embodied Control," in *Proc. ICRA*, 2023.
[3] Y. Li et al., "Hamster: Hierarchical Action Models for Open-World Robot Manipulation," in *Proc. ICLR*, 2025.
[4] L. X. Shi et al., "Hi Robot: Open-Ended Instruction Following with Hierarchical VLA Models," in *Proc. ICML*, 2025.