DialogVLA: Bridging Reasoning Action in Bimanual Manipulation via Parallel Mixture-of-Experts

Jeonghwan Choi[1] Hyunwoo Kim[1] Ilsung Jin[2] and Donghan Kim[1]

jjjenghi.github.io/dialogvla

***Abstract*— Recently, Vision-Language-Action (VLA) models have shown strong generalization performance in natural language instruction-based robot manipulation. However, in bimanual manipulation environments, misalignment between high-level decision-making and low-level action generation, as well as uncertainty in arm role assignment, remain major challenges. Existing VLA methods implicitly project manipulation stages and arm leadership decisions into a single action space, which does not sufficiently secure structural control stability in complex collaborative situations. In this paper, we propose DialogVLA to address this problem. DialogVLA explicitly reasons about manipulation stages and arm roles through Dialog Reasoning, and binds high-level reasoning with continuous control by conditioning them on MoE(Mixture-of-Experts)-based structural routing and a Flow Matching Action Expert. In addition, we introduce a Two-Stage training strategy that separates reasoning, routing, and action generation to improve training stability and generalization performance. On RoboCasa and RoboTwin benchmarks and in real robot experiments, DialogVLA shows consistent performance improvement compared to existing robot learning policies and VLA models. In particular, it achieves average success rates of 82.8% and 63.9% on RoboTwin Easy and Hard settings, respectively, and 75.4% success rate on real-world bimanual collaborative tasks. These results suggest that structurally connecting high-level reasoning and action generation is a key factor for improving stability and robustness in bimanual manipulation .**

I. INTRODUCTION

Recently, Vision-Language-Action (VLA) models have rapidly developed as general robot policies that generate continuous control actions from image observations and natural language instructions. The possibility of large-scale vision-language pretraining-based general policies is expanding, and training methods that combine web-scale vision-language data with robot demonstration data have shown meaningful progress in zero-shot performance and concept generalization.[1][2][3]

However, in real bimanual manipulation environments, structural limitations exist. Bimanual manipulation is not simply a problem of increasing the action dimension, but requires role assignment between two arms, synchronization, interference avoidance, and stability of phase transition at the same time.[4][5] The model must decide at every time step ”which manipulation stage should be performed now” and ”which arm should take the lead.” Existing VLA policies implicitly project such high-level decisions into a single action vector, and therefore do not explicitly reveal the structural connection between decision and action.[6] As a result, misalignment between reasoning and action occurs, and especially in long-horizon bimanual manipulation, decision errors are likely to accumulate and lead to task failure.[7] In addition, most existing approaches are designed to learn the roles of two arms in a shared representation space, which may cause unstable leadership decisions at certain time steps or motion delay and collision during role switching. In situations with weak spatial bias, such as symmetric layouts or central objects, arm selection ambiguity increases, which leads to initial grasp failure or unnecessary arm switching. These limitations suggest that an explicit design of policy structure is required for bimanual collaborative manipulation.

In this paper, we propose DialogVLA to address this problem. DialogVLA decomposes the manipulation process into two high-level variables: Skill Phase and Arm Role. Skill

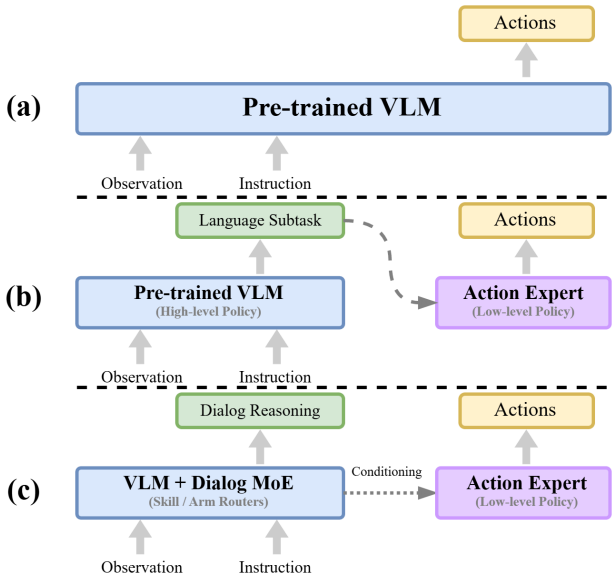

Fig. 1: **Reasoning-to-action paradigms in VLA models.** (a) Standard VLA implicitly maps observation and instruction to actions in a single space. (b) Hierarchical VLA (e.g., Hi Robot) links two separate models via a discrete language subtask. (c) DialogVLA (ours) keeps a single model and couples reasoning to action through continuous MoE conditioning, preserving explicit Skill/Arm decomposition.

[1]Author with Department of Electronic Engineering(AgeTech-Service Convergence Major), college of Electronic information, Yongin republic of Korea jjeonghi@khu.ac.kr, oscar8545@khu.ac.kr, corresponding author donghani@khu.ac.kr

[2]Author with Department of Artificial intelligence, College of Software, KyungHee University, Yongin republic of Korea, ilseong827@khu.ac.kr

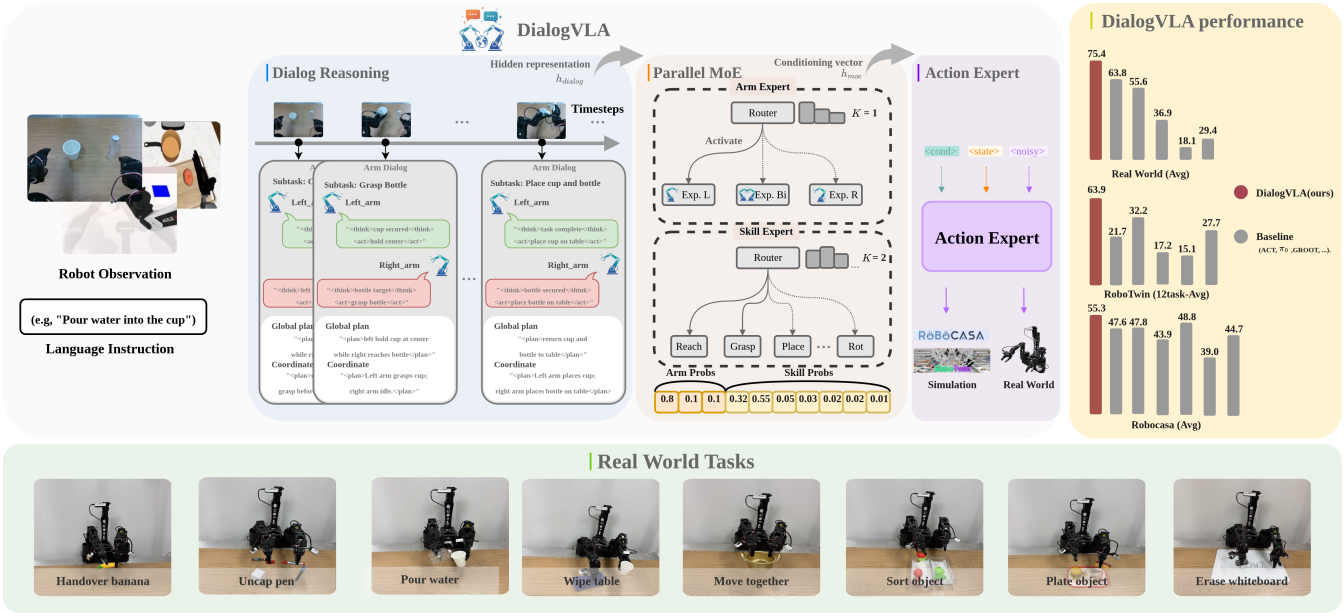

Fig. 2: **DialogVLA Overall.** We present DialogVLA, a dialog reasoning-based vision-language-action model for bimanual robot coordination. The model generates structured dialog reasoning to decompose tasks into per-arm subtasks, routes through Parallel MoE to select skill and arm experts, and produces coordinated bimanual actions via Flow Matching. DialogVLA achieves state-of-the-art performance across RoboTwin simulation, RoboCasa simulation, and real-world bimanual manipulation tasks

Phase represents the flow of manipulation such as approach, grasp, and lift, and Arm Role defines the leadership among the left arm, right arm, and collaborative state. To achieve this, we replace the FFN inside the VLM backbone with a parallel Mixture-of-Experts (MoE)[8] structure and independently construct a Skill Router and an Arm Router. Each router selects experts based on the hidden representation formed through Dialog Reasoning, and the routing results are injected as conditioning signals into the action generation stage to strengthen the structural binding between reasoning and action. In addition, to improve training stability, we introduce a Two-Stage Training strategy. In the first stage, the reasoning and routing structures are learned, and in the second stage, the Flow Matching-based Action Expert is learned while fixing them. This separated training reduces gradient interference between high-level reasoning and low-level continuous control distribution learning.

Experimental results on RoboCasa[9] and RoboTwin benchmarks[10], as well as on the real SO-ARM 101 platform, show that DialogVLA achieves consistent performance improvement compared to existing robot learning policies and recent VLA-based models. In particular, it achieves an average success rate of 63.9% in the RoboTwin Hard setting and records an average success rate of 75.4% in real-world bimanual collaborative tasks. In addition, through an ablation study, we quantitatively verify the contribution of MoE-based action generation and the Two-Stage Training strategy.

The main contributions of this paper are as follows:

- We propose DialogVLA, a structural VLA framework that explicitly decomposes Skill Phase and Arm Role for bimanual manipulation.
- We present a structure that models collaborative states in the internal representation space by configuring Skill MoE and Arm MoE in parallel and directly delivers them as conditioning signals to action generation.
- We achieve stable bimanual continuous control through a Two-Stage Training strategy that separates reasoning and routing learning from action generation learning, and experimentally validate it in both simulation and real robot environments.

II. RELATED WORK

A. *Mixture-of-Experts-based VLA Extension*

MoE[8] is a structure that efficiently expands model capacity by selecting experts according to the input condition. In large-scale language and vision-language models, MoE has been used as a method to achieve high representation power with limited active parameters, and recently there have been attempts to apply it to robot control policies.[11][12] Some studies convert pretrained VLA into an MoE structure to improve parameter efficiency and real-time performance, and propose strategies that branch experts according to task conditions.[13] However, these studies mainly focus on model scale expansion or domain branching, and exploration of structures that separate Skill Phase and Arm Role as independent variables and directly bind them to action generation is limited. This study is different in that it configures Skill MoE and Arm MoE in parallel to explicitly model collaborative states and uses them as conditional inputs for action generation.

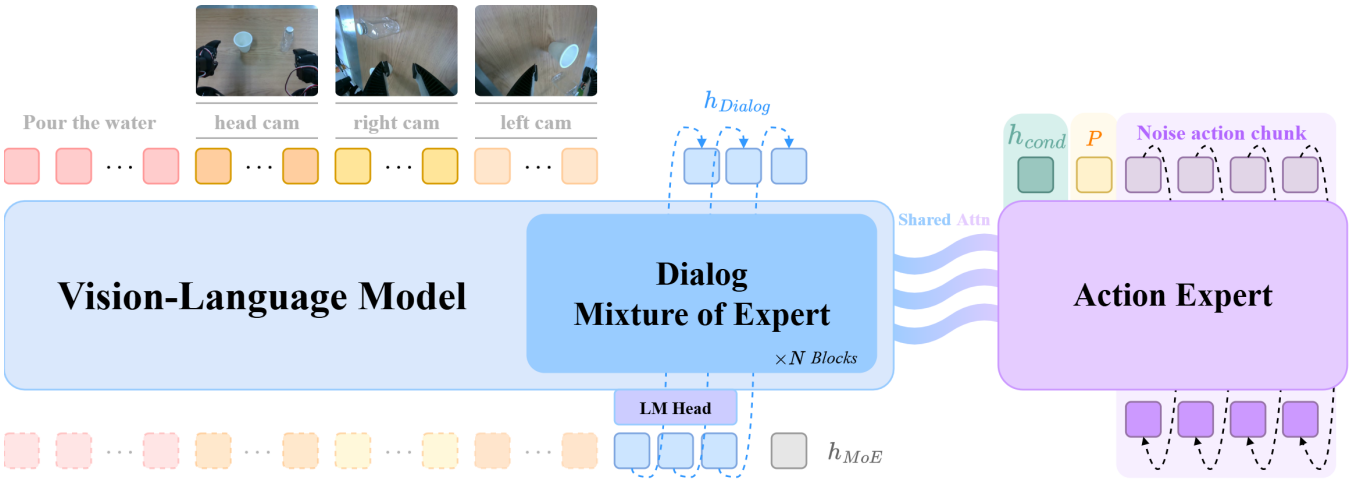

Fig. 3: **DialogVLA Architecture.** Multi-view images from three cameras (head, right wrist, left wrist) and a language instruction are encoded by the Vision-Language Model. The Dialog Mixture of Expert module generates dialog reasoning tokens (h_{Dialog}) and MoE conditioning vectors (h_{MoE}) through N transformer blocks with parallel skill and arm routers. The Action Expert receives the conditioning vector (h_{cond}), proprioceptive state ($\mathbf{p}$), and a noisy action chunk, then denoises it into coordinated bimanual actions via shared attention with the VLM's hidden representations.

B. Reasoning-Based Bimanual Policies

Approaches that combine high-level reasoning and low-level control have been studied to improve policy generalization and interpretability. Recently, studies have been conducted to predict future states through Visual Chain-of-Thought or to model the manipulation process using language-based intermediate representations, but in many cases these reasoning results are not extended to the stage of structurally guiding the actual control signal generation process.[14][15] Bimanual manipulation requires precise role assignment and synchronization strategies due to high degrees of freedom and interference between arms. Attempts have been made to learn bimanual coordination through generative model-based policies, but the coordination structure is generally implicitly embedded in the internal representation. This study is distinguished from previous approaches in that it explicitly predicts Skill Phase and Arm assignment at each time step and injects them as conditioning signals into the action generation module to structurally couple reasoning and action.

III. METHOD

A. Problem Formulation

This study aims to design a VLA policy that generates continuous control actions for a bimanual robot from multi-view visual observations and natural language instructions. At time step t, the input consists of multi-camera observation o_t and a natural language task instruction l, and the output is a continuous action chunk of length H, denoted as $a_{t:t+H} \in \mathbb{R}^{H\times D}$ (D is the degree of freedom of the robot). A general VLA policy directly derives control actions from vision-language representations as follows:

$$a_{t:t+H} = \pi_\theta(o_t, l). \tag{1}$$

However, this Direct mapping method does not explicitly model the structural connection between high-level decision-making and low-level action generation. In particular, the key variables of bimanual manipulation, Skill Phase and Arm Role information, are implicitly projected into the action vector, which may reduce control stability in complex collaborative situations. To address this, we define two discrete high-level variables at each time step: Skill Phase $s_t \in \mathcal{S}$ and Arm assignment $r_t \in \mathcal{A}$. Here, $\mathcal{S} = \{\text{reach, grasp, release, move, lift, place, rotate}\}$ and $\mathcal{A} = \{\text{left, right, bimanual}\}$. These variables are inferred from observation o_t and language instruction l through the Dialog Reasoning module:

$$(s_t, r_t) = f_\phi(o_t, l). \tag{2}$$

The continuous control action is generated by a policy conditioned on the inferred high-level variables:

$$a_{t:t+H} = \pi_\theta(o_t, l, s_t, r_t). \tag{3}$$

DialogVLA forms a hierarchical policy structure that explicitly models the Structural Coupling between high-level reasoning and low-level control $a_{t:t+H}$. The DialogVLA structure is shown in Fig. 2.

B. DialogVLA Architecture

*1) **Dialog Reasoning Procedure**:* Dialog Reasoning is performed based on the Head-view image among the multi-view observations. The training dataset in this study consists of simulation data D_s and real robot data D_r, and each episode is a video sequence that includes continuous control actions. For high-level reasoning supervision, structured dialog labels at the frame level are manually constructed for 5% of D_s and 10% of D_r in the entire dataset. These labels are defined in JSON format and include Skill Phase, global plan, roles and reasons of the left and right arms, and coordination

strategy. The backbone of this study, PaliGemma 3B[16], is a lightweight VLM and has limited representation capacity to directly learn complex bimanual collaborative reasoning. To address this, we apply Knowledge distillation using a large-scale VLM, Qwen3-VL-32B-Instruct[17], as a teacher model. Based on the manually constructed labels, the teacher model automatically generates structured reasoning for the remaining data, and the student model, PaliGemma 3B, uses this as supervised learning signals. At time step t, given the head-view observation o_t^{head} and natural language instruction l, the VLM backbone integrates them to form a hidden representation h_{Dialog}. Dialog Reasoning autoregressively generates a structured dialog d_t that explains the current Skill Phase and bimanual roles based on h_{Dialog}. The LM Head projects h_{Dialog} into the vocabulary space to explicitly reveal high-level decisions for Skill Phase and Arm Role. This decision information is embedded in h_{Dialog}, and the representation is passed to the MoE Router and used for expert selection.

*2) **Dialog Mixture of Experts**:* Existing VLA models implicitly process all manipulation skills and coordination through a single FFN, and different Skill Phase and Arm Role information are mixed in the same parameter space. Especially in bimanual collaboration, shared representations without structural decomposition may limit the model's representation power. To address this, DialogVLA expands all FFN layers of the VLM backbone into a parallel MoE structure. Each MoE layer consists of a Skill Expert group that handles Skill Phase and an Arm Expert group that handles Arm Role, and the two groups select experts through independent routers. The h_{Dialog} formed in Dialog Reasoning is used as a shared input to both routers.

The Skill Router computes gating scores for the 7 Skill Phases ($\mathcal{S}$), and activates the top two experts to allow smooth transition between phases:

$$\mathbf{g}_{\text{skill}} = R_{\text{skill}}(h_{Dialog}) \in \mathbb{R}^7, \quad k_s = \text{Top2}(\mathbf{g}_{\text{skill}}) \tag{4}$$

The Arm Router computes gating scores for the 3 Arm Roles ($\mathcal{A}$), and selects a single expert to ensure clear role assignment:

$$\mathbf{g}_{\text{arm}} = R_{\text{arm}}(h_{Dialog}) \in \mathbb{R}^3, \quad k_a = \arg\max_i \mathbf{g}_{\text{arm}} \tag{5}$$

The selected Skill Expert and Arm Expert perform independent FFN transformations on the input h_t, and reflect Skill Phase-specific and Arm Role-specific information in the internal representation. The output of the Skill Expert is combined by a weighted sum using gating weights π_k, and is merged with the output of the Arm Expert to form the final representation of the layer. The probability distributions $(\mathbf{p}_s, \mathbf{p}_a)$ computed from the routers at each layer are collected as conditional signals for the Action Expert. During MoE conversion, all experts are initialized as copies of the original FFN weights, and immediately after conversion, the output is guaranteed to be identical to the original Transformer. After that, a LoRA adapter is inserted into each expert, and only low-rank parameters are trained. Detailed

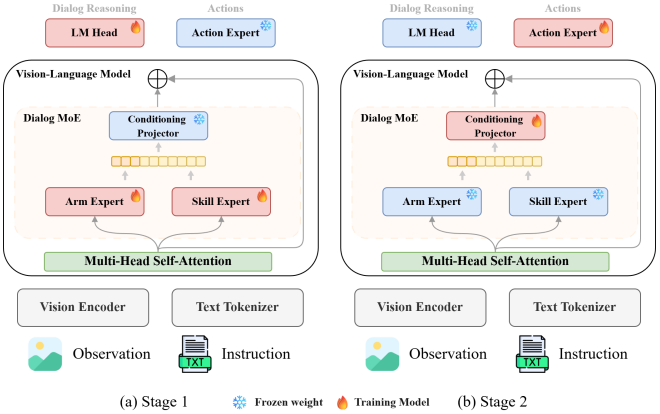

Fig. 4: **Two-stage training pipeline of DialogVLA.** Stage 1 trains the Dialog MoE (Skill/Arm routers and LM Head) for dialog reasoning and MoE conditioning, with the Conditioning Projector and Action Expert frozen. Stage 2 freezes the VLM and Dialog MoE, training only the Conditioning Projector and Action Expert to produce bimanual actions from the learned routing signals.

router supervision and training methods are described in (Section C).

*3) **MoE-Conditioned Action Generation**:* To structurally control action generation of the Action Expert, the Skill Phase probability $\mathbf{p}_s \in \mathbb{R}^7$ and Arm Role probability $\mathbf{p}_a \in \mathbb{R}^3$, which are the results of MoE routing, are used as conditioning signals. The two probability vectors are concatenated to form a 10-dimensional MoE conditioning vector $\mathbf{c}_{\text{moe}}$, which is expanded into a high-dimensional representation through a Conditioning Projector:

$$\mathbf{h}_{\text{moe}} = f_{\text{proj}}(\mathbf{c}_{\text{moe}}) \in \mathbb{R}^{1024}. \tag{6}$$

The expanded conditioning vector is additively combined with the time embedding $\mathbf{h}_{\text{time}}$. In this study, Adaptive RMS Normalization (AdaRMS) is applied to adjust the feature scale according to the time step and high-level state during the denoising process, and the final conditioning signal $\mathbf{h}_{\text{cond}}$ is defined as follows:

$$\mathbf{h}_{\text{cond}} = \mathbf{h}_{\text{time}} + \mathbf{h}_{\text{moe}}. \tag{7}$$

The input of the Action Expert consists of the conditioning signal $\mathbf{h}_{\text{cond}}$, robot state information (Proprioception) $\mathbf{p}$, and a noise-corrupted action chunk $\mathbf{x}_t$. $\mathbf{p}$ is discretized into 256 bins as input. To separate routing probability extraction and action generation during inference, the following **2-Pass forward** structure is introduced:

1) **Pass 1 (Routing Extraction):** Only the VLM backbone is executed to compute routing probabilities at each Transformer layer. The probabilities of all layers are averaged to obtain the final $\mathbf{c}_{\text{moe}}$. In this stage, the Action Expert is not executed and gradients are blocked.
2) **Pass 2 (Conditioned Action Generation):** This is performed with a **Prefix-Suffix** structure. The Prefix, composed of vision, language, and robot state, is

TABLE I: RoboCasa GR1 Tabletop benchmark results for 24 PnP tasks (success rate %). Verb-grouped averages over 50 rollouts per task.

Task (Avg Only)	Isaac-GR00T N1.6	QwenGR00T +Qwen3VL	QwenPI +Qwen3VL	QwenOFT +Qwen3VL	QwenFAST +Qwen3VL	VisionOnly QwenGR00T	**DialogVLA (Ours)**
PnP * to * Close (Avg)	24.2	50.3	42.3	43.7	35.0	**54.0**	44.6
PnP Novel From Cuttingboard To * (Avg)	56.9	52.8	46.0	50.4	50.4	45.2	**57.6**
PnP Novel From Placemat To * (Avg)	51.9	38.0	43.5	41.5	33.5	32.5	**53.5**
PnP Novel From Tray To * (Avg)	55.1	39.2	44.0	49.2	32.0	44.8	**64.0**
PnP Novel From Plate To * (Avg)	57.6	58.5	44.0	**61.0**	45.0	42.0	56.8
Overall Average	47.6	47.8	43.9	48.8	39.0	44.7	**55.3**

TABLE II: RoboTwin 2.0 benchmark results for 12 specific tasks (success rate %).

	ACT		π_0/		**RDT**		**DialogVLA(Ours)**	
Task Name	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Dump Bin Bigbin	68.0	1.0	83.0	24.0	64.0	32.0	**94.0**	**55.0**
Grab Roller	94.0	25.0	96.0	80.0	74.0	43.0	**98.0**	**83.0**
Handover Block	42.0	0.0	45.0	8.0	45.0	14.0	**88.0**	**67.0**
Lift Pot	88.0	0.0	84.0	36.0	72.0	9.0	**92.0**	**59.0**
Pick Dual Bottles	31.0	0.0	57.0	12.0	42.0	13.0	**82.0**	**63.0**
Place Bread Skillet	7.0	0.0	23.0	1.0	5.0	1.0	**91.0**	**77.0**
Place Cans Plasticbox	16.0	0.0	34.0	2.0	6.0	5.0	**57.0**	**18.0**
Place Dual Shoes	9.0	0.0	15.0	0.0	4.0	4.0	**77.0**	**63.0**
Put Object Cabinet	15.0	0.0	68.0	18.0	33.0	18.0	**66.0**	**67.0**
Scan Object	2.0	0.0	18.0	1.0	4.0	1.0	**72.0**	**54.0**
Stack Blocks Three	0.0	0.0	17.0	0.0	2.0	0.0	**86.0**	**82.0**
Stack Bowls Three	48.0	0.0	66.0	24.0	51.0	17.0	**91.0**	**79.0**
Overall Average	**34.5**	**2.2**	**50.5**	**17.2**	**33.5**	**13.1**	**82.8**	**63.9**

processed by the VLM, and the Suffix, composed of action tokens, is processed by the Action Expert. The two modules share the same Self-attention space so that contextual information from the Prefix is delivered to the Suffix.

Here, $\mathbf{h}_{\text{cond}}$ does not affect the VLM backbone computation, and is injected only into the AdaRMS layers inside the Action Expert to adjust the feature scale of each layer. Through this, the Action Expert can generate a velocity field that matches the current Skill Phase and Arm Role even under the same visual observation.

C. Two-Stage Training Strategy

Dialog Reasoning and Action Generation have different training objectives. The former aims at high-level semantic reasoning, while the latter aims at generating a continuous control distribution. If these two objectives are optimized at the same time with a single loss function, performance degradation may occur due to gradient interference. To prevent this, we apply a Two-Stage Training strategy that separates reasoning and routing learning from action generation learning. The overall Two-Stage Training structure is shown in Fig. 3.

1) ***Stage 1:*** *Dialog Reasoning and Router Learning:* The purpose of Stage 1 is to stably learn Dialog Reasoning and the MoE router so that the hidden representation reflects Skill Phase and Arm Role information. In this stage, only the reasoning and routing modules are trained while the Action Expert is fixed. Since Stage 1 learns high-level decision ability that is independent of specific robot hardware, it is performed in an embodiment-agnostic manner. The simulation dataset D_s and the real robot dataset D_r are combined and trained as a single model, and the inputs are the head-view image o_t^{head} and the natural language instruction l. The structured reasoning JSON data labeled by the teacher model is used as supervised learning signals. Dialog generation is trained with auto-regressive cross-entropy, and the router is supervised with ground truth for Skill Phase and Arm Role labels. Each expert is initialized as a copy of the original FFN weights to preserve pretrained representations, and a rank-16 LoRA adapter is inserted so that only low-rank parameters are trained. Through this, about 114 times parameter reduction is achieved compared to the total expert parameters.

The overall loss function of Stage 1 is defined as follows:

$$L_{\text{stage1}} = L_{\text{dialog}} + \lambda_r L_{\text{router}} + \lambda_{\text{aux}} L_{\text{aux}}, \tag{8}$$

where L_{dialog} is the structured dialog loss, and L_{router} is the classification loss for Skill Phase and Arm Role labels. L_{aux} is an auxiliary loss to prevent load concentration on specific experts.

2) ***Stage 2:*** *MoE-Conditioned Action Learning:* The purpose of Stage 2 is to train the Flow Matching-based Action Expert to generate continuous control action chunks based on the MoE conditioning built in Stage 1. In this stage, the trained VLM backbone and MoE router are frozen, and only the Action Expert and related projector layers are trained. Since Stage 2 depends on the robot's action space, it is performed in an Embodiment-specific manner. Robots with different degrees of freedom and dynamics form different control distributions, so separate optimization

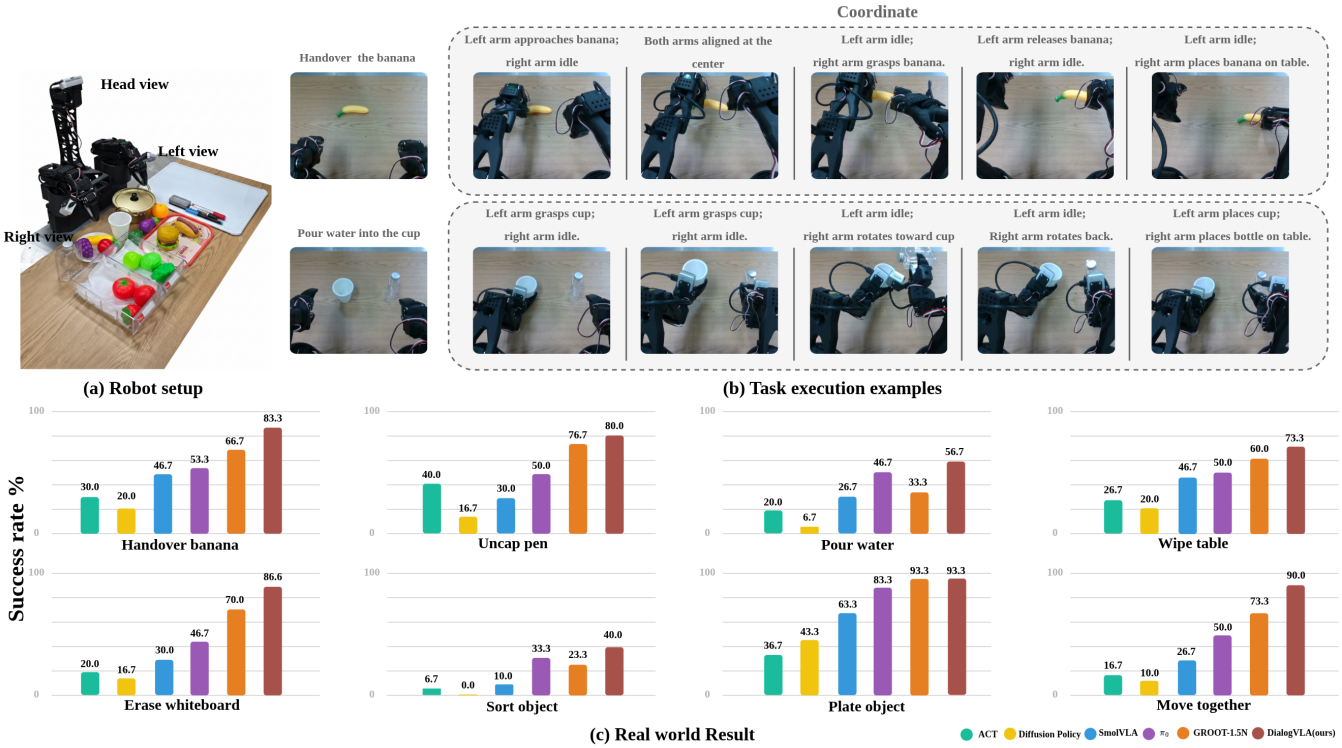

Fig. 5: **Real-world experimental setup and results.** (a) Bimanual robot setup with SO-ARM 101 equipped with three cameras (head, left wrist, right wrist). (b) Task execution examples showing dialog-guided coordination for "Handover the banana" and "Pour water into the cup" tasks, where each frame is annotated with the model's per-arm reasoning and coordination strategy. (c) Success rates across 8 real-world bimanual tasks, where DialogVLA consistently outperforms all baselines.

is performed for each platform. Action data is normalized based on the lower 1% and upper 99% quantiles, and robot state information (Proprioception) is discretized into 256 bins as input. The Flow Matching framework is applied to define a linear interpolation path between the clean action a and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and the Action Expert is trained to predict the velocity field of this path. During training, the intermediate state $\mathbf{x}_t$ generated by time step $t \sim \text{Beta}(1.5, 1.0)$, the robot state $\mathbf{p}$, and the conditioning vector $\mathbf{h}_{\text{cond}}$ that combines time embedding and MoE routing information are used as inputs. The final loss function is as follows:

$$L_{\text{stage2}} = \left\| v_\theta(\mathbf{x}_t, t, \mathbf{h}_{\text{cond}}, \mathbf{p}) - \mathbf{u}_t \right\|^2, \tag{9}$$

where $\mathbf{u}_t = a - \epsilon$ is the ideal velocity vector from the noise state to the clean action. Through this, the Action Expert reflects Skill Phase and Arm Role information and performs consistent continuous control even in bimanual collaborative situations.

IV. EXPERIMENTS

In this section, we quantitatively analyze how the structural design of DialogVLA affects bimanual manipulation performance. We aim to answer the following three questions (1) Does MoE-conditioned action generation provide more stable control performance compared to existing VLA models? (2) Do structured Dialog Reasoning and parallel MoE routing contribute to performance improvement in bimanual manipulation? (3) Is the Two-Stage Training strategy effective for training stability and generalization performance?

A. Experimental Setup

We evaluate DialogVLA on two simulation benchmarks (RoboCasa[9], RoboTwin 2.0[10]) and a real robot platform. All experiments use task success rate as the main metric and compare with recent VLA and generative-based policies under the same evaluation settings.

Datasets: RoboCasa GR-1 Tabletop provides 1,000 episodes per task for 24 manipulation tasks, resulting in a total of 24,000 episodes. RoboTwin 2.0 includes 50 bimanual manipulation tasks and is divided into Easy and Hard settings. The Easy setting consists of 2,500 episodes, and the Hard setting includes 25,000 episodes with lighting changes and random layout, forming a total of 27,500 episodes. For real robot experiments, we use the SO-ARM 101 platform and select 8 bimanual collaborative tasks. For each task, 200 episodes are collected, resulting in a total of 1,600 demonstration episodes used for training and evaluation.

Training Details: All training is conducted on a single NVIDIA RTX PRO 6000 GPU. In Stage 1, the model is trained for 10 epochs over about 10 days using the AdamW

optimizer with a learning rate of 3×10^{-4} and batch size 4. In Stage 2, the action generation model is trained for a total of 100,000 steps with a learning rate of 2.5×10^{-5} and batch size 8. The action chunk size is fixed to $H = 50$, and the best weights are selected based on validation dataset performance.

B. Evaluation Results

RoboCasa GR1 Tabletop Benchmark: RoboCasa GR-1 Tabletop is a bimanual manipulation environment based on the GR-1 humanoid, and consists of 24 Pick-and-Place (PnP) tasks that include different object and target position combinations. This environment uses a 44-DoF action space, and to support this, the action output dimension of the model is expanded to 44 and Stage 2 is trained separately. Since Stage 1 is trained in an embodiment-agnostic manner, the same checkpoint is used. As shown in Table I, DialogVLA achieves the best performance with an overall average success rate of 55.3% over 50 rollouts per task. In particular, its strength is clear in situations where the object is placed at the center of the workspace and Arm selection ambiguity increases. This is interpreted as a result of the Dialog Reasoning-based Skill/Arm decomposition structure that encourages a stable control strategy regardless of object position changes. In some failure cases, insufficient precision in Gripper orientation control is observed, and due to the limitations of the simulation physics engine, abnormal object bouncing immediately after contact or dynamic instability caused by small collisions is also observed. This simulator-induced dynamics instability is identified as an environmental limitation that commonly appears across all compared models.

RoboTwin 2.0 Benchmark: In the RoboTwin 2.0 benchmark, among 50 tasks, 12 bimanual manipulation tasks that require bimanual collaboration are selected for evaluation. The selected tasks include simultaneous manipulation of two arms or role assignment and synchronization. The evaluation is conducted based on the 14-DoF bimanual robot embodiment of the ALOHA-AgileX platform, and each task is executed with 100 rollouts. Performance is measured by task success rate, and the overall performance is reported as the average over 12 tasks. As shown in Table II, DialogVLA achieves the highest performance compared to existing models on all 12 tasks. It achieves an average of 82.8% in the Easy setting and 63.9% in the Hard setting. In particular, large performance improvements are observed in tasks such as Handover, Pick Dual Bottles, and Stack Blocks Three, where role switching and collaboration are critical. In the Hard setting, the performance of all models decreases, but DialogVLA maintains relatively stable success rates.

Real-World Evaluation: Real robot evaluation was conducted on the SO-ARM 101 platform with 8 bimanual manipulation tasks. Each task is designed to verify the phase decomposition and Arm Role decision ability of DialogVLA, and focuses on long-horizon manipulation that requires bimanual coordination or includes complex phase transitions. For performance measurement, each task is executed 30 times, and ACT[18], Diffusion Policy[19], SmolVLA[20], π_o[2], and GROOT-1.5N[21] are evaluated under the same protocol for comparison. As shown in Fig. 4(c), DialogVLA achieves the highest success rate on all tasks. In particular, DialogVLA shows significantly better performance than other models on the Move task, which requires coordinated manipulation between two arms. This result indicates that the structural design that explicitly separates Skill Phase and Arm Role enables stable role assignment and phase transitions during complex bimanual collaboration.

TABLE III: **Ablation study on real-world Handover and Move tasks.** Success rates (%) over 20 trials per task are reported.

Method	Success Rate (%)		
	Handover	Move	Avg
w/o MoE Conditioning	65.0 (13/20)	45.0 (9/20)	55.0
w/o Arm Expert	70.0 (14/20)	55.0 (11/20)	62.5
w/o Skill Expert	70.0 (14/20)	65.0 (13/20)	67.5
w/o Stage 1	55.0 (11/20)	35.0 (7/20)	45.0
DialogVLA	**85.0 (17/20)**	**90.0 (18/20)**	**87.5**

C. Ablation Study

In this section, we conduct the following ablation experiments to verify the contribution of key design components of DialogVLA.

Ablation study setup

1) **w/o MoE Conditioning:** To verify the effect of MoE routing results, we remove the MoE Conditioning Projector output and train the Action Expert using only $\mathbf{h}_{\text{cond}} = \mathbf{h}_{\text{time}}$. In this setting, only time embedding and proprioception information are used as input, and Phase and Arm routing information are not used.
2) **Skill-only / Arm-only Conditioning:** To analyze the contribution of the two routing axes of Parallel MoE, one routing information is disabled. In Skill-only, the arm routing probability is fixed to a uniform distribution so that only skill information is used for conditioning. In Arm-only, the skill routing probability is fixed to a uniform distribution so that only arm information is reflected.
3) **w/o Stage 1 (Stage 2 only):** To verify the effect of the Two-Stage strategy, Stage 2 is directly performed on a pretrained π_0 base model without Stage 1. In this case, the MoE structure exists, but the router and expert LoRA are not pretrained, so conditioning is applied without semantic decomposition.

In all ablation settings, the Stage 2 training data, hyperparameters, and evaluation protocol are kept the same.

Ablation Study Results

In Table III, DialogVLA achieves the highest performance with an average of 87.5% compared to all ablation settings. When MoE conditioning is removed, the average performance drops to 55.0%, and the decrease is especially large in the Move task. Skill-only(62.5%) and Arm-only(67.5%) settings maintain a certain level of performance but do not

reach the Full model. In Handover, Skill information tends to be relatively important, while in Move, Arm information tends to be relatively important. When Stage 1 is removed, the average performance drops to 45.0%, which is the lowest, suggesting that routing without pretraining causes unstable conditioning. These results support the complementary effect of the Two-Stage Training and parallel MoE structure.

V. CONCLUSION

This study proposes DialogVLA, which explicitly decomposes Skill Phase and Arm Role and directly conditions them into action generation to address the reasoning-action misalignment problem in bimanual manipulation. By introducing a parallel MoE structure, we strengthen the structural coupling between high-level reasoning and low-level continuous control, and achieve stable control even in complex collaborative situations. Through RoboCasa, RoboTwin, and real robot experiments, we demonstrate better generalization performance and coordination ability compared to existing VLA models, and verify the effectiveness of the structural design through ablation experiments. However, there are limitations due to dependence on a fixed skill set and sensitivity to reasoning accuracy, and future work will extend to more complex environments through automatic skill expansion and uncertainty-aware routing methods.

ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program(IITP-2023-RS-2023-00266615) supervised by the IITP(Institute for Information Communications Technology Planning Evaluation), This work was supported by the BK21 plus program "AgeTech-Service Convergence Major" through the National Research Foundation (NRF) funded by the Ministry of Education of Korea[5120200313836], Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022- 00155911, Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University)), This work was supported by the Technology Innovation Program(RS-2024-00469147, RS-2023-KT225094, RS-2025-02702977, RS-2024-00444344, RS-2024-00507746, RS-2026-25548755) funded By the Ministry of Trade Industry Energy(MOTIE, Korea), This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) and Korea Smart Farm RD Foundation(KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration(RDA)(2540000882)

REFERENCES

[1] 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.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550," *arXiv preprint ARXIV.2410.24164*.

[3] 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.

[4] M. Grotz, M. Shridhar, Y.-W. Chao, T. Asfour, and D. Fox, "Peract2: Benchmarking and learning for robotic bimanual manipulation tasks," in *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2024.

[5] I. Liu, C. Arthur, S. He, D. Seita, and G. Sukhatme, "Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation," *arXiv preprint arXiv:2407.04152*, 2024.

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

[7] H. Deng, W. Guo, Q. Wang, Z. Wu, and Z. Wang, "Safebimanual: Diffusion-based trajectory optimization for safe bimanual manipulation," *arXiv preprint arXiv:2508.18268*, 2025.

[8] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[9] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[10] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu *et al.*, "Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation," *arXiv preprint arXiv:2506.18088*, 2025.

[11] Z. Du, B. Liu, Y. Liang, Y. Shen, H. Cao, X. Zheng, Z. Feng, Z. Wu, J. Yang, and Y.-G. Jiang, "Himoe-vla: Hierarchical mixture-of-experts for generalist vision-language-action policies," *arXiv preprint arXiv:2512.05693*, 2025.

[12] D. Kuzmenko and N. Shvai, "Moira: Modular instruction routing architecture for multi-task robotics," *Neurocomputing*, p. 132962, 2026.

[13] W. Shen, Y. Liu, Y. Wu, Z. Liang, S. Gu, D. Wang, T. Nian, L. Xu, Y. Qin, J. Pang *et al.*, "Expertise need not monopolize: Action-specialized mixture of experts for vision-language-action learning," *arXiv preprint arXiv:2510.14300*, 2025.

[14] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1702–1713.

[15] J. Wen, M. Zhu, J. Liu, Z. Liu, Y. Yang, L. Zhang, S. Zhang, Y. Zhu, and Y. Xu, "dvla: Diffusion vision-language-action model with multimodal chain-of-thought," *arXiv preprint arXiv:2509.25681*, 2025.

[16] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello *et al.*, "Paligemma: A versatile 3b vlm for transfer," *arXiv preprint arXiv:2407.07726*, 2024.

[17] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[18] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[19] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[20] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[21] 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.