RoboSE: A Simple yet Effective Dual System for Robot Learning

Anonymous CVPR submission

Paper ID 3509

Abstract

Dual-system methods suggest a promising path for hierarchically integrating the high-level Multimodal-Large-Language Model (MLLM) with the low-level action policy model into Vision-Language-Action models (VLAs) to combine both generalization ability and response speed. In this paper, we propose a simple-yet-effective dual-system for robot learning, dubbed RoboSE*, which leverages the strengths of dual-system VLAs, as demonstrated through comprehensive analysis and experimentation. First, we employ* ***pre-alignment*** *of the semantic space to bridge the high-level MLLM with the low-level policy model. Then,* ***prompt tuning*** *is applied to the entire framework without tuning the MLLM parameters, ensuring training efficiency while enhancing generalization and performance. Finally, we introduce an auxiliary task that projects the action token into the action space, compelling the MLLM to engage in* ***multimodal reasoning****. Extensive experimental results demonstrate that* RoboSE *significantly outperforms existing state-of-the-art methods, and solid analysis highlights the critical importance of each component within the model.*

1. Introduction

Vision-Language-Action models (VLAs) demonstrating astounding generalization capabilities on myriad robot tasks [4, 5, 8, 17, 18, 32, 45] fueled by the development of Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs). While smaller action policy models [8, 17, 32, 45] as well has witnessed a rapid advancement in the past few years benefit from rapid response, a crucial requirement for robots operating in dynamic environments. The dual system suggests a promising path for hierarchically integrating the high-level MLLM with the low-level action policy model into VLAs to combine both generalization ability and response speed.

Existing studies [6, 33, 46] focus on pre-training and fine-tuning strategies to integrate policy models with MLLMs as VLAs. However, a simple combination will also make dual-system infeasible due to the largely increased training burdens and significant differences between the latent space of the two types of models, resulting in suboptimal performance outcomes. Thus, the pursuit of effective dual-system framework has been a perennial challenge in robot learning.

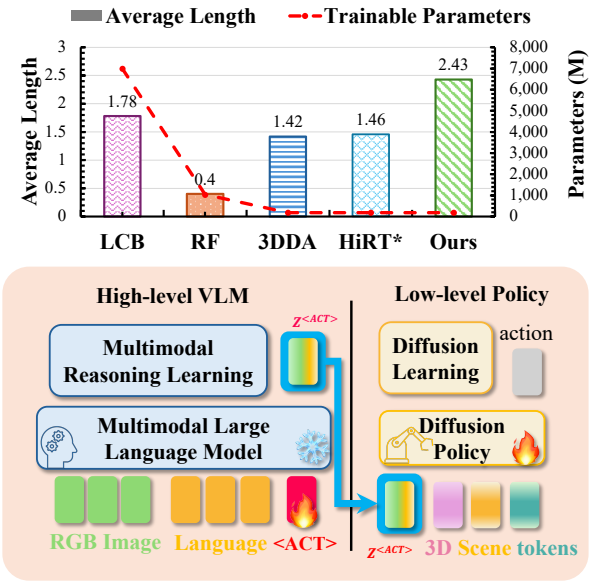

Figure 1. **Overview of RoboSE.** RoboSE is a **simple**-yet-**effective** dual system for robot, which fully leverages the multimodal reasoning capabilities of the high-level MLLM to empower the low-level policy with a minimal training cost. On the top region, we present the performance of RoboSE on the CALVIN-E benchmark. It can be observed that RoboSE achieved the highest average length with the lowest trainable parameters.

To this end, we propose a simple-yet-effective dual-system for robot learning, dubbed RoboSE, which sufficiently leverages the strengths of dual-system VLAs, as shown in Figure 1. In general, incorporating a projector implemented as a simple linear layer can effectively bridge the semantic spaces of different pre-trained models. However, experimental results indicate that training the whole system from scratch may lead to model collapse. Hence,

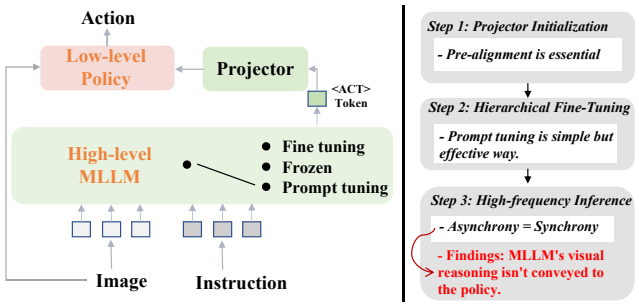

Figure 2. **Comparison of different design paradigms of Dual system**. A dual system usually contains two components: a large high-level MLLM and a small low-level policy. They are bridged with a special token <ACT>. We discussed the design of the dual system from three aspects: Projector initialization, Hierarchical Fine-Tuning, and High-frequency Inference.

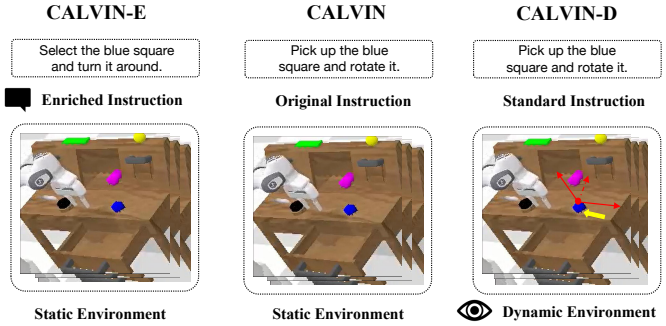

Figure 3. **Three types of evaluation setting.** CALVIN-E (Left) replaces the original instruction with GPT-4 enriched language instruction. CALVIN (middle) is the original setting without any change. CALVIN-D (right) contains five dynamic scenarios with the original language instruction.

we employ **pre-alignment** of the semantic space to bridge the high-level MLLM with the low-level policy model. For hierarchical training, we compared various strategies, including fine-tuning, freezing, and prompt tuning. Ultimately, **prompt tuning** is applied to the entire framework instead of tuning the MLLM parameters, thereby ensuring training efficiency while enhancing generalization and performance. Besides, the existing dual-system is inadequate for fully leveraging the visual reasoning capabilities of the MLLM. To account for this, we mapped the latent embeddings of action tokens into semantic space to analyze what these action tokens from MLLM convey. The poor results indicate that the visual guidance information from the MLLM is not effectively transferred to the downstream action policy. Therefore, we introduce an auxiliary task that projects the action token into the action space, compelling the MLLM to engage in **multimodal reasoning**.

Subsequently, we construct a new benchmark, CALVIN-D, because there is no ready-made benchmark with dynamic tasks to evaluate decision-making in real-time environments for VLAs. Combined with previous CALVIN-E [24] (enriched instruction) and origin CALVIN benchmark, we build comprehensive evaluation metrics for the current dual system to test generalization ability for text instruction and visual dynamic environment. Notably, RoboSE outperforms the SOTA methods by noticeable margins in all benchmarks without bells and whistles. Overall, our main contributions are as follows:

- We introduce the CALVIN-D benchmark, which provides a diverse set of dynamic tasks for evaluating multimodal models in real-world scenarios.
- We propose a simple-yet-effective dual-system for robot learning, dubbed RoboSE, which sufficiently leverages the strengths of dual-system VLAs.
- Extensive experimental results and analysis demonstrate the superiority of RoboSE.

2. Background

Model architecture. There are three main type of existing Visual-Language-Action models (VLAs): large model-based, small model-based, and dual system that combines both. The large model-based approach leverages the inherent ability to generalize across language and visual modalities, enhancing the generalization of action generation. In contrast, small models with fewer parameters can adapt more rapidly to environmental changes, demonstrating exceptional dynamic responsiveness. In this paper, we focus on the dual system, as we aim to combine the strengths of generalization capabilities while also ensuring real-time responsiveness to environmental feedback.

As illustrated in Figure 2, the vanilla dual system commonly consists of three components: Large Langauge Model (LLM), action projector that bridges the action token from the perception to action modalities, and downstream action policy. The projector was normally implemented as a linear layer. The model takes visual and text input and generates action outputs. According to the different methods of obtaining action tokens in common practice [33, 46], the dual system has the following variants: (1) *Fine-tuning*: train the whole MLLM parameters to generate the specific action token; (2) *Frozen*: extracting the output of MLLM and generate action token by average pooling; (3) *Prompt-tuning*: add additional learnable token to the end of input sequence while freezing parameters of MLLM.

Integration strategies. Following common practice [33, 46], we also need to investigate better-bridging strategies to integrate a high-level MLLM with a low-level action policy model, employing the following three key strategies: (1) *Projector initialization*: the MLLM and policy are separately pre-trained, while the projector is usually initialized from

Table 1. **Comparison of different training strategies for the dual system.** We tried different combinations of training methods, and all the models were trained on the ABC split of CALVIN with the original language instruction data. We tested them on split D with the original instruction (CALVIN) and GPT-4 enriched instruction (CALVIN-E), respectively, following the RoboFlamingo [23] evaluation setting.

Benchmark	Projector Pre-alignment	Hierarchical Fine-tuning	Task completed in a row (%) ↑					Avg. Len ↑
			1	2	3	4	5	
CALVIN	×	Fine-tuning	0	0	0	0	0	0
	×	Frozen	0	0	0	0	0	0
	✓	Fine-tuning	96	83	68	58	48	3.53
	✓	Frozen	90	74	61	54	40	3.33
	✓	Prompt-tuning	94	77	67	60	47	3.45
CALVIN-E	✓	Fine-tuning	76	49	30	15	4	1.74
	✓	Frozen	72	37	21	11	5	1.46
	✓	Prompt-tuning	72	55	40	26	20	2.13

random weights. Therefore, following existing literature, we first pre-train the projector to bridge the MLLM and policy while freezing policy on image-text-action pairs; (2) *Hierarchical fine-tuning*: we then tested various high-level bridging methods—finetuning, frozen, and prompt tuning—to determine which approach best preserves the model's language generalization capabilities; (3) *Hierarchical inference*: finally, due to the inference gap between MLLM and the policy, existing literature tends to adopt asynchronous inference to enhance the inference frequency of this framework. However, asynchronous visual inputs may lead to erroneous guidance from MLLM, causing the policy to generate incorrect actions, especially in dynamic scenarios. Thus, we compared asynchronous and synchronous inference to evaluate the impact of this configuration.

Evaluation. Moreover, we also evaluate our model in enriched language contexts (CALVIN-E) and dynamic visual scenarios (CALVIN-D) apart from standard scenarios, which allows us to assess further the linguistic and visual generalization capabilities of the VLA model. Detailed descriptions of the specific scenario setups can be found in Figure 3 and Section 5.2.

3. Revisiting the key components of dual system

In this section, we analyze the key components essential for the design of the dual system.

3.1. Training phase 1: projector initialization

Since the MLLM and the policy are trained independently, semantic differences exist between the upstream and downstream components. Therefore, a projector is needed to connect the two models. Based on previous experiences with LLaVA-like models [18], pre-training a projector before the collaborative training between the final two models can help achieve better alignment of semantics across different layers. As shown in Table 1, the model fails to train successfully without pre-alignment due to the excessive semantic gap between layers while bringing in optimization conflicts between policy training and semantic alignment. **Thus, the pre-alignment of the projector is crucial.**

3.2. Training phase 2: hierarchical fine-tuning

Language generalization is the most crucial characteristic of a dual system. Different hierarchical fine-tuning methods can produce varying effects on the overall system.

Fine-tuning vs. Frozen. As shown in Table 1, results indicate that the frozen setting performs worse than the fine-tuned large model setting. The significant distribution gap between the embeddings generated by MLLM and those produced by the low-level policy's original text encoder presents a challenge. Fine-tuning the pre-trained policy while keeping the large model frozen makes it difficult for the pre-trained policy to adapt to the out-of-distribution embeddings. This situation requires substantial adjustments to the pre-trained policy to align with the new distribution.

Since aligning semantics does not necessarily require updating the model parameters—since this introduces additional training costs and may lead to catastrophic forgetting—we also tested prompt tuning of MLLM.

Prompt-tuning. In the standard setting, prompt tuning the MLLM yields results similar to finetuning, while the enriched instruction setting achieves the best performance. In contrast to the frozen setting, the prompt tuning still enables adjustments to the action token embeddings, allowing for better alignment with the downstream policy, thereby maintaining semantic coherence between the upper and lower levels. Compared to finetuning, prompt tuning does not modify the model parameters, allowing it to preserve the pre-trained knowledge of the large model and avoid catastrophic forgetting. As a result, it achieves better performance in language generalization experiments. In addition to enhancing model performance, prompt tuning offers a more lightweight approach. With the large model's parameters frozen, only the prompt embeddings need to be trained, significantly improving training efficiency. This allows us to

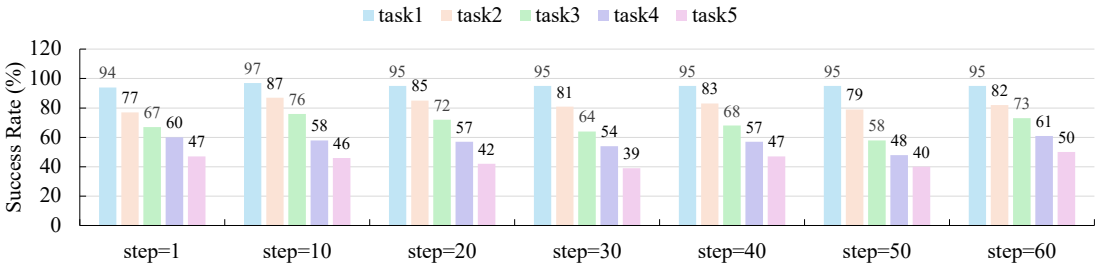

Figure 4. **Evaluations on hierarchical inference.** We evaluate the performance of the dual system on the CALVIN benchmark, with inference steps set to 1 and 60, respectively."Steps" refers to the inference steps of action policy during a single MLLM inference step. The longest environmental steps of the action policy [17] are 60, which means MLLM only inference once and represents the most typical asynchronous scenarios.

use a simpler and more effective method to adjust the output distribution of the large model, better supporting the training of low-level policies. **Thus, prompt tuning is a simple but effective way.**

3.3. Testing phase: hierarchical inference

Visual generalization in dynamic scenes is another critical question to analyze for a dual system. Asynchronous inference may lead to erroneous guidance from MLLM, causing the policy to generate incorrect actions, especially in dynamic scenarios. To explore the extent of the negative effect, we investigate the performance differences between asynchronous and synchronous inference. Notably, all experiments in this section were conducted under the paradigm of prompt tuning.

3.3.1. Asynchronous vs. Synchronous

We evaluated different asynchronous steps on CALVIN's standard and dynamic environment. In a dynamic environment, we test objects' five different dynamic settings: stationary, moving left, moving up, moving diagonally, and moving in a circle. We observe a surprising conclusion in Figure 4: regardless of the number of steps between the large model's inferences, the performance changes are quite similar. Moreover, even in dynamic scenarios, the experimental results are consistent. **Thus, we can conclude that asynchronous won't damage the overall performance.**

3.3.2. Findings of existing dual systems design flaw.

The results in section 3.3.1 indicate that the current MLLM is not sensitive to changes in the current environment, which is counterintuitive. To explore the underlying reasons, we mapped the latent embeddings of action tokens into semantic space and calculated the similarity of different words to analyze what these action tokens from MLLM convey. The experiment involves dynamic scenarios where a blue block consistently moves to the left.

Similarity with spatial words at different time steps. We observe that regardless of whether the robotic arm moves left or right, the probability of "right" is consistently higher than that of "left," while the probabilities of different spatial prepositions remain almost unchanged over time. This indicates that the action token has learned a semantic feature that remains constant and is unrelated to changes in the environment. The higher probability of "right" compared to "left" may be due to "right" carrying more semantic information; for example, "right" can also imply correctness, contributing to its consistently high probability.

Top 10 similar words at different time steps. We observe that the latent embedding primarily encodes the target object, spatial relations, and action semantics from the instruction, along with some noise. It means that the latent embedding mainly summarizes the textual instruction and is largely insensitive to changes in visual information. In other words, the current training method does not effectively leverage the visual reasoning capabilities of the MLLM. Instead, the MLLM merely transmits the semantics of the instructions to the low-level policy. If this is the case, we could achieve the current dual system's performance using a single LLM instead of an MLLM.

Therefore, existing dual systems have a design flaw: the visual guidance information from the MLLM is not transferred to the downstream action policy.

4. Methodology

Based on the above analysis, four conclusions are summarized: (1) Pre-alignment of the projector is crucial. (2) Prompt tuning is a simple but effective training strategy. (3) Existing approaches insufficiently leverage the visual reasoning capabilities of MLLMs. Therefore, we propose a new dual-system framework based on the above experience. Specifically, we employ prompt-tuning to adapt the output of the large model rather than directly fine-tuning the MLLM itself. Additionally, we introduce an auxiliary task to exploit MLLM's visual reasoning capabilities fully. This approach results in a more robust latent embedding that effectively integrates visual and textual information.

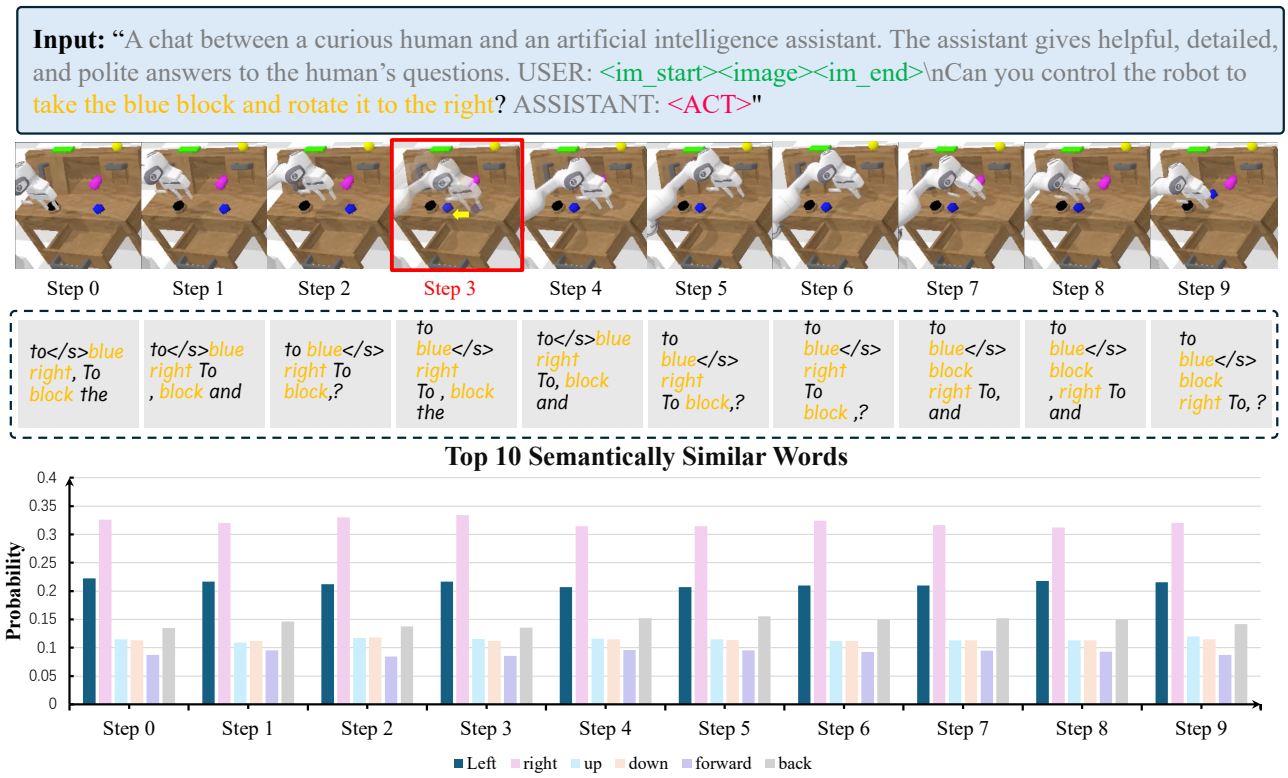

Figure 5. **Evaluation on the shortcoming of existing dual systems.** From top to bottom, the first row displays the input to the MLLM. The second row visualizes a special scenario where, at environment step 3, the blue block is manually shifted to the left. In the third row, we present the top 10 words that are semantically closest to the latent embedding. The bottom row illustrates the probability distribution of spatial words associated with the latent embedding.

4.1. Architecture

Network. Our system comprises two main components: a pre-trained VLM f_ϕ and a pre-trained policy π_θ, with parameters ϕ and θ, respectively. The VLM includes a text-only large language model and a vision encoder, which projects images into the embedding space of the language model, allowing for a multi-modal understanding of textual and visual inputs. The pre-trained policy consists of a vision encoder and transformer-based diffusion model. Using multiple cross-attention layers, the diffusion model incorporates a lot of conditioning information, such as 3D scene representations, proprioception information, and condition/instruction tokens from the high-level model. In this work, we leverage LLaVA [27] as the high-level VLM and 3D Diffuser Actor as the low-level pre-trained diffusion policy. Notably, we use a linear layer to replace the 3D Diffuser Actor's text encoder, aligning the dimension of the latent embedding output by the large model with the input dimension of the low-level policy.

Input and Output. The whole system is designed to mimic demonstration trajectories in the format $\{l, (o_1, a_1), (o_2, a_2), ...\}$, where $l = \{w_i \in \mathrm{R}^d\}_{i=1}^N$ represents a task-specific language instruction of length N with an input dimension d, and o_t and a_t denote the visual observation and corresponding robot action at each timestep t. The input observation o_t consists of two RGB-D images from different viewpoints. The output action a_t defines the end-effector's pose, which is decomposed into 3D location, rotation, and gripper state (open/close): $a_t = \{a_t^{\mathrm{l}} \in \mathbb{R}^3, a_t^{\mathrm{r}} \in \mathbb{R}^6, a_t^{\mathrm{g}} \in \{0, 1\}\}$. The VLM f_ϕ processes language instruction l and the third-view RGB image o_t', outputting the latent embedding z_t for low-level policy. The low-level pre-trained policy π_θ takes as input the noisy trajectory τ_t^i, diffusion step i, and the conditioning information from the environment observation o_t, the latent embedding z_t, and proprioception c_t of timestep t, predicting the action trajectory $\tau_t = (a_{t:t+T}^{\mathrm{l}}, a_{t:t+T}^{\mathrm{r}})$ and binary states $a_{t:t+T}^{g}$ at each timestep t, over a temporal horizon T.

4.2. Training

Prompt Tuning. In order to avoid the degradation of MLLM, we introduce one learnable token $\texttt{<ACT>} \in \mathbb{R}^d$ at end of language instruction l. The new instruction l' is defined as

CVPR
#3509

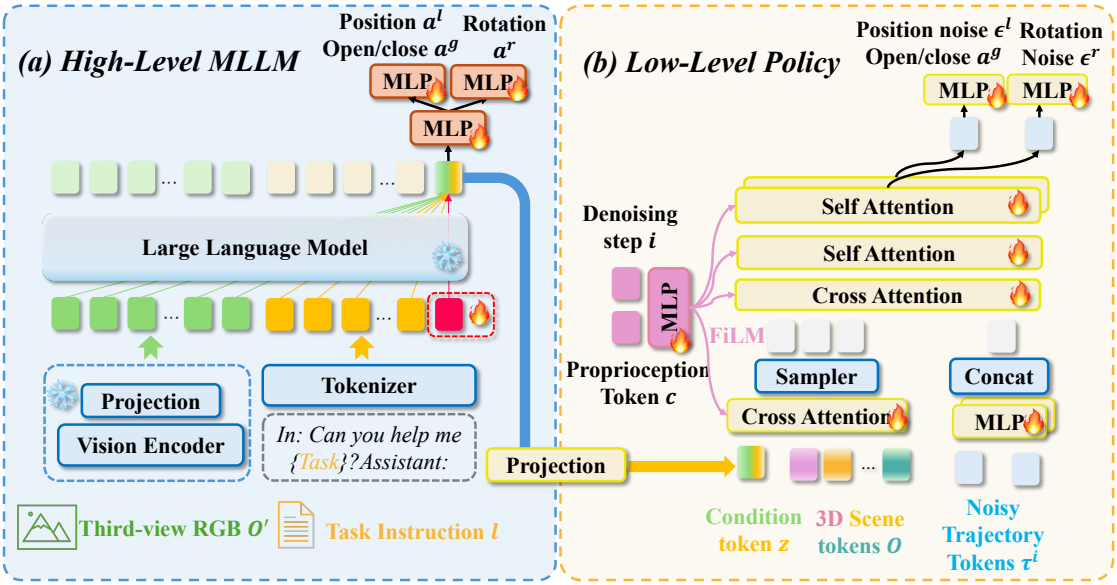

Figure 6. **Detailed framework of the RoboSE.** (a) The high-level MLLM (left) takes third-view RGB o', task instruction l, and a learnable token <ACT> as input. After processing through the Large Language Model (LLM), we extract the feature embedding from the final layer of the <ACT> token as the latent goal for the low-level policy. To fully leverage the MLLM's multimodal reasoning capability, we propose an auxiliary task, using MLPs to predict the action (location a^l, rotation a^r, open/close a^g) based on this feature embedding $z^{\texttt{<ACT>}}$, ensuring it encapsulates both visual and textual information. (b) The low-level policy (right) receives the latent goal from the high-level MLLM, combines it with 3D scene tokens o and proprioception token c, and iteratively predicts action noise ϵ to produce an accurate action trajectory τ and gripper state a^g. Notably, our approach keep all parameters of the MLLM frozen and fine-tune the learnable prompt to adjust the MLLM's output, significantly reducing training costs compared to previous methods.

$l' = \{l, \texttt{<ACT>}\}$. During training, all parameters of VLM are frozen; we only update the embedding of learnable token <ACT>.

Multimodal Reasoning Learning. As we discussed in section 3.3, we know that these previous methods do not fully utilize VLM's visual reasoning capability. Specifically, they align the output of the large VLM model with the output from the text encoder of CLIP. Using purely textual information to supervise the fine-tuning of the VLM can lead to the degradation of multimodal reasoning capability. Therefore, we designed an auxiliary task to leverage the multimodal reasoning capability of the VLM fully. This task is very simple and requires no additional data preparation process. The output embedding $z_t^{\texttt{<ACT>}} = f_\phi(o'_t, l')$ from the learnable prompt token is passed through linear layers to predict the action trajectories τ_t and gripper actions a_t^g. Through supervised training on this task, we ensure that the large model has to utilize visual input information and that the latent embedding contains a blend of multimodal information. The loss function is defined as follows:

$$\begin{aligned}\mathcal{L}_{lm}(\texttt{<ACT>}) = \text{BCE}(\text{MLP}(f_\phi^g(o'_t, l')), a^g_{t:t+T}) \\ + \omega_1 \cdot ||\text{MLP}(f_\phi^l(o'_t, l')) - a^l_{t:t+T}|| \\ + \omega_2 \cdot ||\text{MLP}(f_\phi^r(o'_t, l')) - a^r_{t:t+T}||,\end{aligned} \quad (1)$$

where ω_1, ω_2 are hyperparameters to balance the effect of each loss item, and MLP represents linear layer. To reconstruct the sequence of 3D locations and 3D rotations, we apply the L_1 loss. Additionally, we supervise the end-effector opening using binary cross-entropy loss (BCE).

Diffusion Learning. Following the previous diffusion-based approach [8, 17, 45], we train our model using the action denoising objective. During training, we randomly sample a time step t and a diffusion step i, adding noise $\epsilon = (\epsilon^l, \epsilon^r)$ to a ground-truth trajectory τ_t^0. The objective is defined as:

$$\begin{aligned}\mathcal{L}_{policy}(\theta, \texttt{<ACT>}) = \text{BCE}(\pi_\theta^g(o_t, z_t^{\texttt{<ACT>}}, c_t, \tau_t^i, i), a^g_{t:t+T}) \\ + \omega_3 \cdot ||\epsilon_\theta^l(o_t, z_t^{\texttt{<ACT>}}, c_t, \tau_t^i, i) - \epsilon^l_{t:t+T}|| \\ + \omega_4 \cdot ||\epsilon_\theta^r(o_t, z_t^{\texttt{<ACT>}}, c_t, \tau_t^i, i) - \epsilon^r_{t:t+T}||,\end{aligned} \quad (2)$$

where ω_3, ω_4 are also hyperparameters to balance loss items. Please refer to [1] for the details of the loss function.

Two stage training. We adopt a two-stage training approach to train our proposed dual system. In the first stage, to initially align the embedding produced by the VLM with the feature space of the pre-trained policy, we freeze the parameters of the large model and the low-level policy, training only the prompt and projection layers. In the second stage, we keep the large model frozen and unfreeze the low-level

Table 2. **Language-Conditioned Visuomotor Control on CALVIN ABC→D.** We report both the success rates and the average task completion length (out of a total of 5 tasks) per evaluation sequence. The terms *Lang* and *All* indicate whether the models are trained solely on vision-language data pairs or on the full dataset, respectively. * represents we provide our reproduced results.

Method	Train sets	Task completed in a row (%) ↑ 1	2	3	4	5	Avg. Len. ↑
3D Diffusion Policy	Lang	28.7	2.7	0.0	0.0	0.0	0.31
3D Diffuser Actor	Lang	92.2	78.7	63.9	51.2	41.2	3.27
RT-1	Lang	53.3	22.2	9.4	3.8	1.3	0.90
RoboFlamingo	Lang	82.4	61.9	46.6	33.1	23.5	2.48
GR-1	Lang	85.4	71.2	59.6	49.7	40.1	3.06
MCIL	All	30.4	1.3	0.2	0.0	0.0	0.31
HULC	All	41.8	16.5	5.7	1.9	1.1	0.67
SuSIE	All	87.0	69.0	49.0	38.0	26.0	2.69
HiRT*	Lang	93.6	77.6	62.1	50.1	38.1	3.43
RoboSE (Ours)	Lang	**96.0**	**81.3**	**72.0**	**64.0**	**50.7**	**3.64**

policy, fine-tuning it together with the prompt and projection. The objectives in both stages remain unchanged. The only difference between the two stages is whether the low-level policy is frozen. In summary, our loss function includes two components and can be defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{lm} + \mathcal{L}_{policy} \tag{3}$$

5. Experiments

We conduct extensive experiments to assess our method's performance, emphasizing its generalization ability in complex language instructions and dynamic scenarios. Our study focuses on the following research questions: I. Can our method achieve higher success rates in a simulation environment compared to previous methods? II. Does our method exhibit generalization capabilities in both open-form language instruction and dynamic scenarios?

5.1. Comparison with SOTA Methods

In this section, we compare our method with state-of-the-art (SOTA) approaches to demonstrate its effectiveness in long-horizon tasks.

Experimental Setup. CALVIN [28] is an open-source benchmark designed for learning long-horizon tasks conditioned on language. The environment simulates a 7-DOF Franka robotic arm equipped with a gripper employed on a desk with various objects for interaction. The robot must complete a series of complex 6D manipulation tasks guided by a sequence of language instructions. Each subtask is paired with a specific instruction, and the robot moves on to the next subtask with a new instruction after successful completion. CALVIN consists of four different environments A, B, C, and D, with a shared set of language instructions and subtasks.

Baselines. We compare our method with three types of approaches: small models, large models, and dual systems. The 3D Diffusion Policy [45] and 3D Diffuser Actor [17] are small policies based on 3D depth information. RT-1 [5], RoboFlamingo [23], and GR-1 [40] are 2D policies based on large-scale transformers. MCIL [26], HULC [27], SuSIE [3], and HiRT [46] are hierarchical 2D policies, while LCB [33] is a hierarchical 3D policy. These methods generate latent features or images of subgoals based on language instructions, which are then passed to a low-level policy. We present results for 3D Diffuser Actor, HULC, RoboFlamingo, SuSIE, and GR-1 as reported in their respective papers. The results for 3D Diffusion Policy, RT-1, and MCIL are referenced from [17]. We report the reproduced results of HiRT.

Results. The results are shown in Table 2. We can observe that our method outperforms previous approaches, as our average length has improved from 3.27 to 3.64, and the success rate of completing five consecutive tasks has increased by 9.5%.

5.2. Generalization Ability

Additionally, we further investigate whether our method demonstrates improved generalization in understanding enriched instructions and handling dynamic scenarios.

Experimental Setup. We evaluate all methods in two generalization settings. (1) **CALVIN-E:** The model is evaluated in unfamiliar environments based on free-form instructions enriched by GPT-4, which were not encountered during training. (2) **CALVIN-D:** We selected the relatively simple rotate task from CALVIN's 34 tasks for augmentation. Specifically, we added four additional types of motion to the stationary blocks in the CALVIN: leftward movement, upward movement, diagonal movement, and circular motion.

Baselines. We carefully choose some representative methods for comparison, including small model: 3D Diffuser Actor [17], Large model: RoboFlamingo [23], and dual system: LCB [33] and HiRT [46].

Table 3. **Evaluations on generalization capability to GPT-4 enriched language instructions (CALVIN-E).** * represents we provide our reproduced results.

Method	Task completed in a row (%) ↑					Avg. Len. ↑
	1	2	3	4	5	
RoboFlamingo	63.0	33.0	16.4	8.6	3.6	0.40
3D Diffuser Actor	65.2	39.1	20.3	11.7	6.1	1.42
LCB	73.6	50.2	28.5	16.0	9.9	1.78
HiRT*	72.0	37.0	21.0	11.0	5.0	1.46
RoboSE (Ours)	**83.8**	**56.8**	**48.6**	**29.7**	**24.3**	**2.43**

Table 4. **Evaluations on generalization capabilities to dynamic scenario (CALVIN-D).** * represents we provide our reproduced results.

Method	Task completed (%) ↑					Avg. Suc. ↑
	Static	Left	Forward	Diagonal	Circle	
3D Diffuser Actor	82	84	46	67	80	71.8
HiRT*	90	78	64	72	79	76.6
RoboSE (Ours)	**88**	**90**	**68**	**74**	**82**	**80.4**

Instruction Generalization. As shown in Table 3, our method outperforms previous approaches in the enriched language instruction setting, with an average length increase of 0.65 and a 14.4% improvement in the success rate of completing five consecutive tasks. This is because the prompt-tuning strategy fully preserves the pre-trained knowledge of MLLM, making RoboSE have a good generalization capability to free-form language instruction.

Dynamic Environment Generalization. As shown in Table 4, our method outperforms previous approaches in the dynamic scenario setting, with a 3.8% improvement in the average success rate of completing five dynamic tasks. This is because multimodal reasoning learning fully leverages the generalization capability of MLLM, making RoboSE more robust to variations in vision.

6. Related Work

Generalist Models. Generalist models in robotic manipulation leverage massive parameters and extensive cross-task training datasets to enhance task adaptability and generalization. These models can handle complex, multimodal inputs and support multi-task functionality. For instance, the RT-X series (e.g., RT-1 [5] and RT-2 [4]) utilizes Vision-Language-Action (VLA) models to facilitate knowledge transfer across tasks, enabling strong generalization in diverse environments. OpenVLA [18] further integrates pre-trained vision-language models with large datasets to boost cross-domain adaptability. However, large models' high computational and inference costs make deployment challenging in resource-limited systems. While large models are well-suited for scenarios requiring high-level abstract understanding and cross-domain transfer, they face real-time responsiveness and computational efficiency limitations.

Specialist Models. Specialist models [1–3, 7, 9, 10, 12, 13, 15, 16, 19–22, 25, 29–31, 35–39, 41–44, 47–49] are designed for specific tasks, offering simpler architectures and higher computational efficiency, achieving high precision and low latency in defined scenarios. Models like Diffusion Policy [8, 17, 32, 45], based on diffusion mechanisms, can precisely manage multimodal action distributions, showing strong control accuracy and stability, making them ideal for low-dimensional, task-specific operations. Additionally, some specialist policies [11, 14, 34] leverage 3D representations to enhance performance in low-dimensional control tasks, achieving higher success rates in complex multi-task manipulation settings. However, small models have a limited scope, lack broad generalization, and are often inadequate for adapting to dynamic, varying environments.

Dual systems. Dual systems combine the generalization capacity of large foundation models with the execution efficiency of small policy, achieving a balance through layered or cooperative systems. Representative works such as LCB [33] and HiRT [46] bridge task understanding and execution through latent space encoding and hierarchical control. At the same time, RoboDual [6] employs a synergistic dual-system structure, leveraging high-level task comprehension from large models and real-time control from small models to adapt flexibly in complex task environments. Dual systems effectively handle the intricate demands of multi-task, multimodal scenarios, enhancing performance in diverse environments while optimizing response speed and resource efficiency, providing robust solutions for efficient robotic manipulation. In this work, we provide a holistic ablation of different design choices for the dual system.

7. Conclusion

In summary, this paper propose a simple-yet-effective dual-system for robot learning, dubbed RoboSE, which leverages the strengths of dual-system VLAs. There are three important innovation in RoboSE: (1) we employ **pre-alignment** of the semantic space to bridge the high-level MLLM with the low-level policy model. (2) **prompt tuning** is applied to the entire framework without tuning the MLLM parameters, ensuring training efficiency while enhancing generalization and performance. (3) we introduce an auxiliary task to compel the MLLM to engage in **multimodal reasoning**. Besides, we also construct a new benchmark CALVIN-D with dynamic tasks to evaluate multimodal reasoning and decision-making in real-time environments for RoboSE. Extensive experimental results demonstrate RoboSE significantly outperforms existing SOTA methods, and solid analysis highlight the critical importance of each component within the model.

References

[1] Christopher Agia, Toki Migimatsu, Jiajun Wu, and Jeannette Bohg. Stap: Sequencing task-agnostic policies. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, page 7951–7958. IEEE, 2023. 8

[2] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.

[3] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023. 7, 8

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 1, 8

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. 1, 7, 8

[6] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024. 1, 8

[7] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023. 8

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 1, 6, 8

[9] D., E., and Whitney. The mathematics of coordinated control of prosthetic arms and manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 94(4):303–309, 1972. 8

[10] Haritheja Etukuru, Norihito Naka, Zijin Hu, Seungjae Lee, Julian Mehu, Aaron Edsinger, Chris Paxton, Soumith Chintala, Lerrel Pinto, and Nur Muhammad Mahi Shafiullah. Robot utility models: General policies for zero-shot deployment in new environments. *arXiv preprint arXiv:2409.05865*, 2024. 8

[11] Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024. 8

[12] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. 8

[13] Shihao Ge, Beiping Hou, Wen Zhu, Yuzhen Zhu, Senjian Lu, and Yangbin Zheng. Pixel-level collision-free grasp prediction network for medical test tube sorting on cluttered trays. *IEEE Robotics and Automation Letters*, 8(12):7897–7904, 2023. 8

[14] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024. 8

[15] Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. Mail: Improving imitation learning with mamba. *arXiv preprint arXiv:2406.08234*, 2024. 8

[16] Daniel Kappler, Franziska Meier, Jan Issac, Jim Mainprice, Cristina Garcia Cifuentes, Manuel Wüthrich, Vincent Berenz, Stefan Schaal, Nathan Ratliff, and Jeannette Bohg. Real-time perception meets reactive motion generation, 2017. 8

[17] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 1, 4, 6, 7, 8

[18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1, 3, 8

[19] George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, page 1015–1023, Red Hook, NY, USA, 2009. Curran Associates Inc. 8

[20] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *International Journal of Robotics Research*, 31(3):360–375, 2012.

[21] Youngwoon Lee, Shao-Hua Sun, Sriram Somasundaram, Edward S. Hu, and Joseph J. Lim. Composing complex skills by learning transition policies. In *Proceedings of International Conference on Learning Representations*, 2019.

[22] Youngwoon Lee, Joseph J. Lim, Anima Anandkumar, and Yuke Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization, 2021. 8

[23] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 3, 7

[24] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang,

Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *ICLR*. OpenReview.net, 2024. 2

[25] Chien Erh Lin, Minghan Zhu, and Maani Ghaffari. Se3et: Se(3)-equivariant transformer for low-overlap point cloud registration. *IEEE Robotics and Automation Letters*, 9(11): 9526–9533, 2024. 8

[26] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020. 7

[27] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7 (4):11205–11212, 2022. 7

[28] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. 7

[29] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation, 2019. 8

[30] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[31] Richard P Paul. *Robot Manipulators : Mathematics, Programming and Control*. Robot Manipulators : Mathematics, Programming and Control, 1981. 8

[32] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024. 1, 8

[33] Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: Latent codes as bridges in hierarchical robot control. *arXiv preprint arXiv:2405.04798*, 2024. 1, 2, 7, 8

[34] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 8

[35] Theodoros Stouraitis and Michael Gienger. Predictive and robust robot assistance for sequential manipulation. *IEEE Robotics and Automation Letters*, 8(12):8026–8033, 2023. 8

[36] Jan Ole von Hartz, Eugenio Chisari, Tim Welschehold, Wolfram Burgard, Joschka Boedecker, and Abhinav Valada. The treachery of images: Bayesian scene keypoints for deep policy learning in robotic manipulation. *IEEE Robotics and Automation Letters*, 8(11):6931–6938, 2023.

[37] M. Vukobratovic and V. Potkonjak. *Dynamics of Manipulation Robots: Theory and Application*. Springer Berlin Heidelberg, 2012.

[38] Yixiao Wang, Yifei Zhang, Mingxiao Huo, Ran Tian, Xiang Zhang, Yichen Xie, Chenfeng Xu, Pengliang Ji, Wei Zhan, Mingyu Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. *arXiv preprint arXiv:2407.01531*, 2024.

[39] Junjie Wen, Yichen Zhu, Minjie Zhu, Jinming Li, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, et al. Object-centric instruction augmentation for robotic manipulation. *arXiv preprint arXiv:2401.02814*, 2024. 8

[40] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 7

[41] Sizhe Yang, Yanjie Ze, and Huazhe Xu. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 36, 2024. 8

[42] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[43] Yanjie Ze, Nicklas Hansen, Yinbo Chen, Mohit Jain, and Xiaolong Wang. Visual reinforcement learning with self-supervised 3d representations. *IEEE Robotics and Automation Letters*, 8(5):2890–2897, 2023.

[44] Yanjie Ze, Yuyao Liu, Ruizhe Shi, Jiaxin Qin, Zhecheng Yuan, Jiashun Wang, and Huazhe Xu. H-index: Visual reinforcement learning with hand-informed representations for dexterous manipulation. *Advances in Neural Information Processing Systems*, 36, 2024. 8

[45] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024. 1, 6, 7, 8

[46] Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024. 1, 2, 7, 8

[47] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*. 8

[48] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, et al. Language-conditioned robotic manipulation with fast and slow thinking. *arXiv preprint arXiv:2401.04181*, 2024.

[49] Yichen Zhu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Retrieval-augmented embodied agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17985–17995, 2024. 8