
Fast-in-Slow: A Dual-System VLA Model Unifying Fast Manipulation within Slow Reasoning

Hao Chen^{* 1, 2} Jiaming Liu^{*,† 2} Chenyang Gu^{* 2, 4} Zhuoyang Liu^{* 2} Renrui Zhang^{† 1} Xiaoqi Li²
Xiao He³ Yandong Guo³ Chi-Wing Fu¹ Shanghang Zhang^{2, 4}  Pheng-Ann Heng¹

¹The Chinese University of Hong Kong ²State Key Laboratory of Multimedia Information Processing,
School of Computer Science, Peking University

³AI²Robotics ⁴Beijing Academy of Artificial Intelligence (BAAI)

Abstract

Generalized policy and execution efficiency constitute the two critical challenges in robotic manipulation. While recent foundation policies benefit from the common-sense reasoning capabilities of internet-scale pretrained vision-language models (VLMs), they often suffer from low execution frequency. To mitigate this dilemma, dual-system approaches have been proposed to leverage a VLM-based System 2 module for handling high-level decision-making, and a separate System 1 action module for ensuring real-time control. However, existing designs maintain both systems as separate models, limiting System 1 from fully leveraging the rich pretrained knowledge from the VLM-based System 2. In this work, we propose Fast-in-Slow (FiS), a unified dual-system vision-language-action (VLA) model that embeds the System 1 execution module within the VLM-based System 2 by partially sharing parameters. This innovative paradigm not only enables high-frequency execution in System 1, but also facilitates coordination between multimodal reasoning and execution components within a single foundation model of System 2. Given their fundamentally distinct roles within FiS-VLA, we design the two systems to incorporate heterogeneous modality inputs alongside asynchronous operating frequencies, enabling both fast and precise manipulation. To enable coordination between the two systems, a dual-aware co-training strategy is proposed that equips System 1 with action generation capabilities while preserving System 2’s contextual understanding to provide stable latent conditions for System 1. For evaluation, FiS-VLA outperforms previous state-of-the-art methods by 8% in simulation and 11% in real-world tasks in terms of average success rate, while achieving a 117.7 Hz control frequency with action chunk set to eight. **Project web page:** fast-in-slow.github.io.

1 Introduction

The fundamental objective of robotic manipulation learning [3, 4, 5, 6] is to convert real-world sensory data and human instructions into precise control signals. Simultaneously, enabling robots to execute a broad spectrum of tasks while adapting to variations in objects and environments remains the core challenge. Recently, some works [7, 8, 9, 10, 11, 12] have sought to leverage the pretrained knowledge of foundational vision-language-models (VLMs) [13, 14, 15, 16, 17, 18] to enable generalized manipulation by fine-tuning these models on robotic datasets [19, 20], giving rise to the vision-language-action (VLA) models. However, these methods, with their billion-scale parameters and autoregressive action generation, lead to low operating frequencies, which constrain responsive closed-loop control and hinder real-world application.

^{*}Equal contribution, [†]Project lead,  Corresponding author.

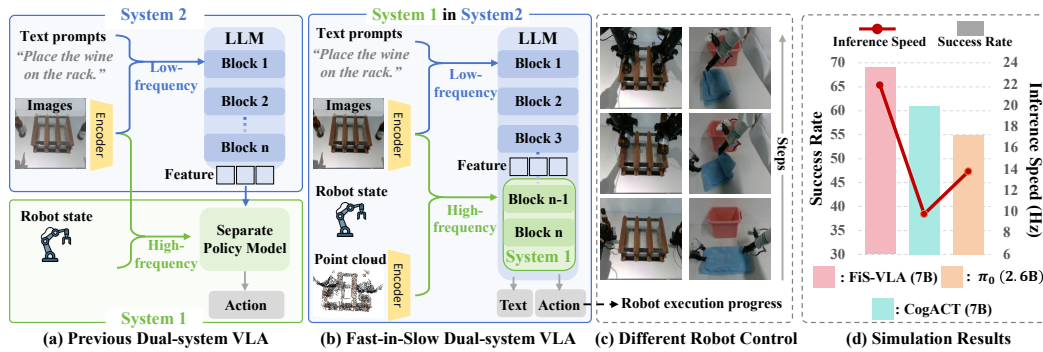


Figure 1: **Overview of FiS-VLA.** (a) Unlike previous dual-system VLA methods [1, 2] that attach a separate policy head as System 1, FiS-VLA (b) repurposes the final transformer blocks of an intact VLM as System 1, while retaining the full model for System 2 reasoning. Under this paradigm, FiS-VLA achieves superior performance and high-frequency control, as shown in (c) and (d).

Drawing inspiration from Kahneman’s dual-system theory [21] that “*System 1 is fast, intuitive, and unconscious, while System 2 is slow, logical, and involves deliberate reasoning*”, recent works have explored incorporating dual-system design into VLA models. Most recent end-to-end approaches [22, 23, 24] leverage VLM as System 2 for high-level feature extraction, while appending an additional policy head as System 1 to transform VLM outputs into executable action poses. Building on a similar architecture, methods such as [2, 1, 25] design dual-system frameworks with asynchronous operating frequencies, further clarifying the distinct roles of System 1 and System 2. While these methods improve execution efficiency, their System 1, as a lightweight separate model, lacks internet-scale pretrained knowledge and depends solely on feature representations extracted by System 2, thus failing to fully leverage the reasoning capabilities within System 2’s VLM. Considering these limitations, and motivated by the functional abstraction of Kahneman’s dual-system theory, we raise a question: “*If a VLM model serves as the central decision-making module of the robot, can it integrate System 1 and System 2 processes to enable coordinated reasoning and execution?*”

To this end, we propose Fast-in-Slow (FiS), a VLA foundation model that integrates the fast execution capabilities of System 1 into a pretrained VLM, while preserving its inherent System 2 contextual understanding and generation capabilities. As shown in Figure 1, unlike prior dual-system VLA approaches [1, 25] that attach System 2 with an independent policy model as System 1, FiS-VLA repurposes the final transformer blocks of System 2 into a high-efficiency execution module, serving as System 1. Under this dual-system paradigm, FiS-VLA enables seamless coordination between the two systems, as both are derived from the same foundation model without altering its connectivity structure. Since System 2 handles understanding and reasoning while System 1 focuses on rapid action execution, we design the two systems with heterogeneous modality inputs and asynchronous operating frequencies. For System 2, it operates at a lower frequency, processing 2D observations and language instructions into multimodal latent representations that guide System 1’s actuation. For System 1, we systematically investigate the impact of various high-frequency inputs for robot control, including the robot state, 2D images, and 3D point clouds. Notably, since 3D geometric information is critical for precise manipulation [26, 27], we utilize a fast 3D embedding strategy that tokenizes point clouds [28] and processes them through a shared vision encoder to extract spatial features, which directly condition the System 1 for geometry-aware interaction.

To jointly optimize the reasoning and execution components in FiS-VLA, we introduce a dual-aware co-training strategy. For the execution component (System 1), we adopt the probabilistic and continuous nature of diffusion modeling [3, 29, 30] by injecting noised actions as latent vectors into the embedding space of System 1 to learn action generation. For the multimodal comprehension component (System 2), we exploit an autoregressive next-token prediction objective to maintain its discrete action generation or high-level language planning capabilities and preserve the overall coherence and integrity of System 2. Under this co-training approach, FiS-VLA first undergoes large-scale pretraining on open-source robotic datasets [31, 20, 32] comprising more than 860K trajectories. It is then fine-tuned on high-quality, self-collected real-world and simulation data [33]. In both real-world and simulated experiments, FiS-VLA achieves state-of-the-art (SOTA) manipulation performance. Meanwhile, FiS-VLA demonstrates strong generalization to unseen objects, complex

backgrounds, and diverse lighting conditions, regardless of the robot type. With a 1:4 operating frequency ratio between System 2 and System 1, FiS-VLA achieves a 117.7 Hz control frequency on an NVIDIA 4090 GPU with action chunk set to eight. In summary, our contributions are as follows:

- We propose Fast-in-Slow (FiS), a unified dual-system VLA model that embeds System 1 execution within a pretrained VLM while preserving its inherent System 2 reasoning capabilities, thereby enabling seamless coordination between both systems.
- Given that System 2 and System 1 serve fundamentally distinct roles within FiS-VLA, we systematically design them with heterogeneous modality inputs and asynchronous operating frequencies, enabling both fast and precise manipulation.
- We propose a dual-aware co-training strategy to jointly optimize System 2 and System 1 in FiS-VLA. Our model demonstrates SOTA performance in both single-arm simulation and dual-arm real-world experiments, while maintaining a high execution frequency.

2 Related Work

Vision-language-action models. Early approaches for robot manipulation primarily relied on reinforcement learning with reward functions derived from proprioceptive signals [34, 35, 36, 37], as well as imitation learning based on visual observations [38, 39, 40, 3]. More recently, increasing attention has been directed toward integrating vision-language models (VLMs) into robotic systems [41, 42, 43, 44, 12, 11, 45, 46, 30], leading to the emergence of Vision-Language-Action (VLA) models. These models leverage the reasoning capabilities of VLMs to directly predict low-level SE(3) poses for manipulation tasks. A common approach among prior works [47, 48, 7, 49] involves autoregressive next-token prediction to generate action sequences. However, such methods often suffer from action discontinuities and low execution frequency. To mitigate this, some VLA models [50, 51, 52, 53] incorporate a policy head to enable continuous action prediction. Recent studies [27, 54, 55, 56, 57] emphasize the value of 3D spatial information for robotic manipulation, making 3D observations a common strategy to boost spatial understanding and accuracy. Furthermore, it has been demonstrated [7, 22, 30, 10] that pretraining VLA models on large-scale robotic datasets [19, 20, 32, 58] can significantly improve their generalization capability. However, these VLA methods commonly suffer from low action generation frequency and still lack the adaptability to adjust their behavior with low latency in response to dynamic task requirements.

Dual-system design in VLA. To improve the execution frequency of VLA models, some recent methods split the framework into two systems. System 2 is responsible for high-level task reasoning, while System 1 focuses on low-level action generation. Methods such as [23, 22, 59, 24] adopt a VLM as System 2 to produce a latent feature. This latent feature is then used as a condition for a diffusion-based action head (System 1). However, in these methods, System 1 and System 2 typically operate at the same frequency, limiting System 1’s potential for high-frequency action generation. We refer to this design as a synchronous dual-system architecture. Additionally, [60, 1, 61, 2, 25] adopt a similar architecture but operate System 2 and System 1 at different frequencies. This asynchronous design further improves the overall action generation frequency of the VLA model. Moreover, some methods [62, 63] attempt to incorporate subtask decomposition within a synchronous dual-system architecture to enhance task planning. Nevertheless, all of these methods introduce a new untrained System 1 policy head and rely solely on features extracted by the System 2 VLM, thus failing to fully leverage the VLM’s pretrained knowledge and reasoning capabilities [64]. In this work, we propose FiS-VLA, an asynchronous architecture that integrates a System 1 execution module into a System 2 VLM, enabling seamless coordination between the two systems within a single pretrained model.

3 Fast-in-Slow Dual-System VLA

In this section, we introduce our proposed FiS-VLA framework, as shown in Figure 2. We begin in section 3.1 with a formal problem formulation. In section 3.2, we describe the overall architecture of FiS-VLA. The core idea of our approach is to retain an intact VLM for System 2 reasoning, while repurposing its final transformer blocks into a System 1 execution module. This design constructs System 1 not as an independently injected module, but as a component that inherits the VLM’s pretrained knowledge and maintains coherent understanding of the intermediate reasoning outputs of System 2, while meeting the low-latency demands of real-time control. In Section 3.3, we present the motivation and detail the mechanisms for designing the two systems to operate at

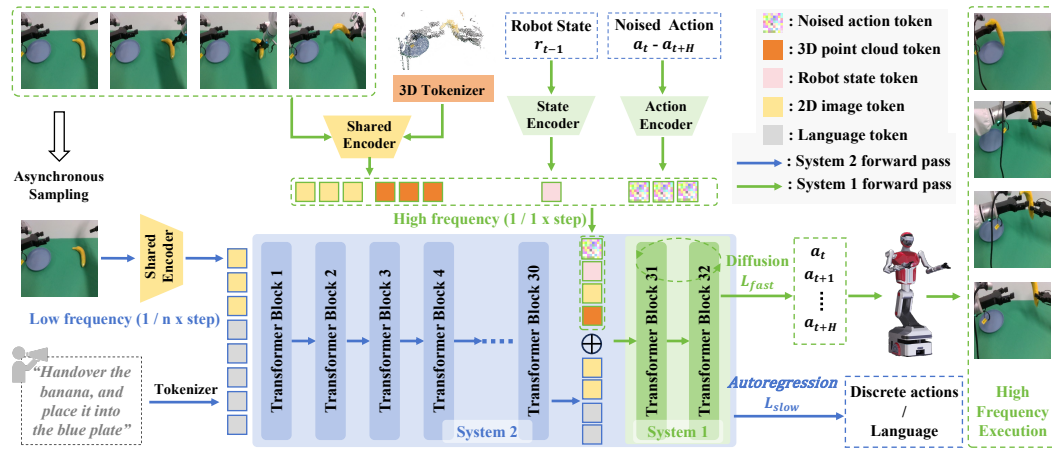


Figure 2: **Framework of FiS-VLA.** FiS-VLA leverages an intact VLM for System 2 reasoning while repurposing the final transformer blocks of the LLM for System 1 execution module. System 2 handles low-frequency inputs such as 2D images and language instructions and produces intermediate latent features that serve as conditioning information for System 1. Instead of being conditioned solely on these periodically updated high-level representations, System 1 processes high-frequency inputs including 3D point clouds, 2D images, and robot states to produce stable and responsive actions. For joint optimization, we introduce a dual-aware co-training strategy that combines a diffusion denoising objective with an autoregressive objective which enables FiS-VLA to support fast action generation while retaining System 2’s multimodal reasoning capabilities.

asynchronous frequencies with different input modalities. Finally, in Section 3.4, we show our dual-aware co-training strategy that jointly optimizes both systems. Within this training framework, FiS-VLA leverages System 1 for continuous action generation while employing System 2 for discrete action or language generation.

3.1 Problem Formulation

Following [23, 22], VLA models typically learn robotic control policies through imitation learning on heterogeneous demonstration datasets $\mathcal{D}$. The training objective is to maximize the likelihood of generating temporally extended action sequences $a_{t:t+H}$, conditioned on multimodal observations o_{t-1} and language instructions l . In this work, we construct comprehensive observations, including the robot state, multi-view images, and 3D point clouds. Formally, given the policy model π_θ , this corresponds to the optimization problem:

$$\max_{\theta} \mathbb{E}_{(a_{t:t+H}, o_{t-1}, l) \sim \mathcal{D}} [\log \pi_\theta(a_{t:t+H} \mid o_{t-1}, l)].$$

The action a can represent different control spaces and control modes. In this work, we employ 7-DoF end-effector pose control for the single-arm Franka Panda robot in simulation, consisting of 3-DoF for relative positional offsets ($[\Delta x, \Delta y, \Delta z] \in \mathbb{R}^3$), 3-DoF for rotation (represented as Euler angles, $\in \mathbb{R}^3$), and 1-DoF for gripper state (open/closed, $\in \mathbb{R}^1$). For real-world experiments, to validate our model’s robustness across different robot embodiments and control modes, we employ 14-DoF control on the AgileX and 16-DoF control on the AlphaBot dual-arm robots, under the end-effector pose control and joint position control, respectively.

3.2 FiS-VLA Architecture

We begin by presenting an overview of the FiS-VLA architecture, as shown in Figure 2. Similar to prior VLA methods [7, 22], FiS-VLA inherits the base architecture and initializes pretrained parameters from Prismatic VLMs [16]. The model primarily consists of a vision encoder and a LLM, with an additional lightweight 3D tokenizer introduced to efficiently process point cloud inputs.

Vision encoder. We employ both SigLIP [65] and DINOv2 [66] to jointly extract visual representations that capture high-level semantic features and local spatial details. Specifically, for each

input image, we first resize it to 224×224 pixels. The image is then processed by both encoders, yielding two distinct feature representations $f^{\text{SigLIP}} \in \mathbb{R}^{N_v \times 1024}$ and $f^{\text{DINO}} \in \mathbb{R}^{N_v \times 1152}$, where N_v represents the token dimension. These two features are concatenated along the channel dimension, resulting in a unified visual embedding for further processing.

Point cloud encoder. To investigate the impact of 3D geometric information on robotic manipulation, we incorporate point cloud data $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^{N_p}$, which is derived from a single-view depth map using camera intrinsics and extrinsics. N_p denotes the number of points. Unlike some approaches [54, 27] that directly process point clouds with newly injected 3D encoders, our method first transforms the point cloud into high-dimensional tokens using a lightweight 3D tokenizer [67]. Specifically, the 3D tokenizer consists of three blocks, each containing farthest point sampling [68] for downsampling, the k-nearest neighbors algorithm for local aggregation, and a learnable linear layer for feature encoding. The tokenized representation is then processed by our shared vision encoder to extract local spatial features, following [28]. This design offers two key advantages: first, it effectively projects 3D information into the LLM’s embedding space by leveraging the vision encoder of the pretrained VLM with vision-language alignment capabilities; and second, it avoids obvious parameter increase and maintains computational efficiency.

LLM backbone. The 7B LLaMA2 [69] model is adopted as the LLM backbone for FiS-VLA. LLaMA2 is a decoder-only transformer architecture consisting of 32 blocks, where the input and output of each block can be viewed as high-dimensional representations of a token sequence. Previous works [24, 70] find that, in VLA models, leveraging intermediate LLM representations instead of the final layer for action generation improves downstream policy success rates without degrading multimodal representation quality. Therefore, we repurpose the final few blocks of the LLM for System 1 to condition on intermediate latent features from System 2, enabling efficient, low-latency responses. To ensure that System 2 maintains its full reasoning capability, we incorporate the complete LLM as its core component, forming a “*fast system within slow system*” architecture that balances rapid action generation with deep contextual reasoning.

MLP components. To further clarify the FiS-VLA architecture, we describe the remaining auxiliary components, all of which are implemented as MLPs. First, a pretrained vision-language projector is employed to map 2D and 3D features into the LLM’s textual embedding space, which is initialized from the pretrained VLM. In parallel, the robot proprioceptive state is encoded using a state encoder. Given that we adopt diffusion-based action generation for System 1, two additional MLPs are incorporated to project the timestep and noised actions as continuous vectors.

3.3 Dual-System Coordination

Asynchronous frequency design. FiS-VLA is structured into two components: a slow System 2 and a fast System 1, inspired by Kahneman’s dual-system theory [21]. Since System 2 VLM with billion-scale parameters, it operates at a low frequency to perform high-level semantic understanding and contextual reasoning. In our framework, it interprets task-relevant visual observations and language instructions, and produces a comprehension output in the form of latent features from an intermediate block of the LLM. Building on previous action chunking methods [39, 3], the instruction and scene observation at time step t can provide guidance for a future horizon of action steps ($a_{t:t+H}$). Consequently, System 2’s intermediate output serves as a latent condition that temporally guides System 1’s action generation across the following H time steps. In contrast, System 1 focuses on generating executable actions in real time. At each time step, it leverages the most recent observation to generate actions, while being conditioned on the periodically updated high-level reasoning output from System 2. This behavior resembles intuitive and reactive responses, positioning System 1 as a high-frequency action generation module.

To achieve this, we investigate the coordination frequency between the two systems. A central question is that “*How many future action steps can be effectively guided by the intermediate comprehension output from System 2?*” We empirically explore the effect of varying horizon lengths (e.g., 1, 2, 4, ..., n) in the ablation study. This corresponds to setting the operating frequency ratio between System 2 and System 1 to 1: n , as our robot’s hardware does not support deploying the two systems on separate GPUs for parallel inference, unlike the implementation in Helix [25]. While parallel inference can further improve model speed, we focus on the fundamental research question of identifying the optimal coordination ratio between two systems. In Figure 2, to ensure that System 1 can effectively interpret the latent conditions produced by System 2 from earlier horizon steps, we

employ asynchronous sampling during training to reduce the operating frequency of System 2. This encourages the System 1 execution module to maintain temporal consistency in task understanding.

Heterogeneous modality input. The two systems in FiS-VLA are designed to serve fundamentally distinct purposes. System 2 is responsible for high-level task understanding and scene reasoning, whereas System 1 is optimized for fast, reactive control. In line with these different objectives, we propose that each system should be provided with input modalities specifically tailored to its function. Since the System 2 VLM has undergone internet-scale pretraining on image-text paired data, we provide it with both language instructions and 2D visual observations to fully exploit high-level semantic reasoning capabilities. In contrast, System 1 is tasked with generating executable actions in real time, conditioned on a comprehensive representation of the robot’s current environment. We carefully explore the information required for accurate and responsive control. First, System 1 must receive low-latency 2D images of the current scene. To enhance temporal consistency in closed-loop control, the robot’s current state is also essential. Furthermore, since the robot must reason about spatial relationships and interact with complex spatial configurations, we additionally provide 3D point cloud data to support precise manipulation. Ultimately, the System 1 execution module integrates the three input modalities with the periodically updated latent feature from System 2, jointly serving as the conditioning context for diffusion-based action generation. Our experimental results confirm that each modality contributes meaningfully to the success of the manipulation tasks.

3.4 Training Objective and Recipe

Dual-aware co-training strategy. The core objective of FiS-VLA is to generate accurate and executable actions. To this end, we leverage the continuous nature of diffusion modeling, which typically yields more reliable actions than discrete prediction approaches [22, 23]. Given an initial action sequence $\tilde{a}$, we inject Gaussian noise $\eta \sim \mathcal{N}(0, I)$ at a randomly chosen timestep $\tau \sim \mathcal{U}(1, T)$, where $\tau \in \mathbb{Z}$ and $T = 100$. The forward process adds noise in closed form: $\tilde{a}_\tau = \sqrt{\beta_\tau}\tilde{a} + \sqrt{1 - \beta_\tau}\eta$, where β_τ denotes the noise scaling factor according to a predefined schedule [71]. To train System 1 π_{θ_f} , we formulate the learning process as an optimization problem over the following objective:

$$\mathcal{L}_{\text{fast}} = \mathbb{E}_{\tau, c, \tilde{a}, \eta} \left[\|\eta - \pi_{\theta_f}(\sqrt{\beta_\tau}\tilde{a} + \sqrt{1 - \beta_\tau}\eta, c, \tau)\|^2 \right], \quad (1)$$

where c denotes the conditioning sources. In FiS-VLA, c consists of two components: the low-frequency latent feature extracted from System 2, and the high-frequency input to System 1. Since the System 1 execution module is embedded within the System 2 VLM, exclusively training the model for diffusion-based action generation may lead to catastrophic forgetting of its autoregressive reasoning capability. To mitigate this issue, we propose a joint training objective to the entire VLA model that preserves System 2’s reasoning ability by incorporating next token prediction with a cross-entropy loss. The autoregressive supervision signal can be either discrete actions [7, 49] or language-based plans [50, 8], depending on the construction of the robotic training data. As an example with discrete actions, we define the objective as follows:

$$\mathcal{L}_{\text{slow}} = - \sum_{i=1}^{D_t} \log P(\hat{a}_i \mid (\text{context}), \theta), \quad (2)$$

where D_t represents the total length of discrete action tokens, $\hat{a}_i$ denotes the i -th ground-truth action token, and $P(\hat{a}_i \mid \text{context}, \theta)$ is the probability predicted by the LLM given the input context and model parameters θ ($\theta_f \subseteq \theta$). Finally, we derive the overall training objective to update the FiS-VLA model.

$$\mathcal{L}_{\text{FiS-VLA}} = \mathcal{L}_{\text{fast}} + \mathcal{L}_{\text{slow}}. \quad (3)$$

Pretraining recipe. Prior to pretraining FiS-VLA, we initialize the model with parameters from a pretrained VLM [16], following the method established in [7, 22]. We curated a specialized pretraining dataset by carefully processing and filtering large-scale cross-embodiment datasets including Open X-Embodiment [19], DROID [20], ROBOMIND [32], and so on. As detailed in Appendix A, this dataset comprises over 860K trajectory samples. FiS-VLA was trained on this dataset for five epochs, with both system inputs consisting solely of a single image as observation. Since the pretraining data contains no subgoal-level language instructions, we initially supervise System 2 using discrete action sequences. During fine-tuning, we enhance the System 2 objective with additional language supervision by incorporating manually annotated sub-task plans and applying automated augmentation.

Table 1: **Comparison of FiS-VLA and baselines on RL Bench.** All methods are trained in the multi-task setting [73], and we report success rates (S.R.) based on the evaluation criteria defined in RL Bench. Inference speed is evaluated on an NVIDIA 4090 GPU with action chunk set to one.

Models	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var	Infer. speed
ManipLLM [48]	0.50	0.80	0.40	0.20	0.80	0.35	0.10	0.25	0.15	0.20	0.38 ± 0.04	2.2 Hz
OpenVLA [7]	0.65	0.40	0.75	0.50	0.80	0.20	0.35	0.15	0.10	0.10	0.40 ± 0.04	6.3 Hz
π_0 [23]	0.90	0.80	0.95	0.30	0.85	0.30	0.30	0.70	0.10	0.30	0.55 ± 0.03	13.8 Hz
CogACT [22]	0.90	0.80	0.95	0.50	0.85	0.50	0.55	0.45	0.30	0.25	0.61 ± 0.04	9.8 Hz
FiS-VLA	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 ± 0.03	21.9 Hz

4 Experiments

In Section 4.1, we compare the manipulation performance and inference speed of FiS-VLA with prior methods in simulated environments. The effectiveness of each component is evaluated in Section 4.2 and Appendix B. Section 4.3 presents both quantitative and qualitative results for FiS-VLA on real-world manipulation tasks, including dual-arm control under different robot configurations. Finally, in Section 4.4, we demonstrate the generalization capabilities of FiS-VLA by assessing its performance on previously unseen objects, backgrounds, and lighting conditions.

4.1 Simulation Experiment

Simulation benchmark. In order to fully evaluate our method, we tested on 10 various manipulation tasks in the RL Bench [33] benchmark based on the CoppeliaSim simulator, including *Close box*, *Close Laptop*, *Toilet seat down*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out*, *Frame off hanger*, *Wine at rack*, and *Water plants*. All the tasks were performed on a Franka Panda robot, using the front-view camera to get the input RGB image and point cloud. We collect the data by following pre-defined waypoints and utilizing the Open Motion Planning Library [72]. Building upon the frame-sampling technique employed in previous studies [73, 5, 28], we construct a training dataset where each task contains 100 trajectories.

Training and evaluation details. We compare FiS-VLA against four state-of-the-art (SOTA) VLA models, including ManipLLM [48], OpenVLA [7], π_0 [23], and CogACT[22], where the latter two are dual-system methods but operate with synchronous frequencies. For baselines, we load the official pretrained parameters provided by each method and adhere to their respective fine-tuning settings. For FiS-VLA’s input, the single-view RGB image is resized to 224×224 , the point cloud is downsampled to 1024 points, the text instruction is derived from simulation, and the robot state is aligned with the predicted actions. FiS-VLA model is trained for 300 epochs using the AdamW optimizer [74] on 8 NVIDIA A800 GPUs, with mixed-precision training employed. All methods utilized the officially provided pre-trained parameters and underwent full fine-tuning. Following [22, 5], we evaluate all methods using 20 rollouts from the latest epoch checkpoint, repeating the evaluation three times for each task and reporting the average success rate along with the variance.

Quantitative results. As shown in Table 1, FiS-VLA achieves an average success rate of 69% across 10 diverse tasks, surpassing the previous SOTA methods CogACT and π_0 by margins of 8% and 14%, respectively. In particular, FiS-VLA achieves superior performance on 8 out of 10 tasks, highlighting the robustness of its action generation capabilities. By embedding the System 1 execution module within the intact VLM (System 2), FiS-VLA leverages the VLM’s pretrained knowledge for action generation and enables more effective interpretation of System 2’s latent feature guidance. In terms of control frequency, FiS-VLA operates at 21.9 Hz, over 2 $\times$ faster than CogACT (9.8 Hz) and more than 1.6 $\times$ faster than π_0 (13.8 Hz), with action chunk set to one. Note that π_0 employs a 2.6B-parameter LLM, while both CogACT and our FiS-VLA are based on a 7B-parameter LLM. The results demonstrate that our asynchronous input frequency design significantly improves the inference speed of VLA models. Moreover, the Fast-in-Slow framework facilitates effective coordination between the two systems, leading to enhanced manipulation accuracy.

4.2 Ablation Study

To analyze the impact of each component on overall performance within the FiS-VLA, we conduct ablation experiments on 10 RL Bench tasks using the same settings as the simulation experiments.

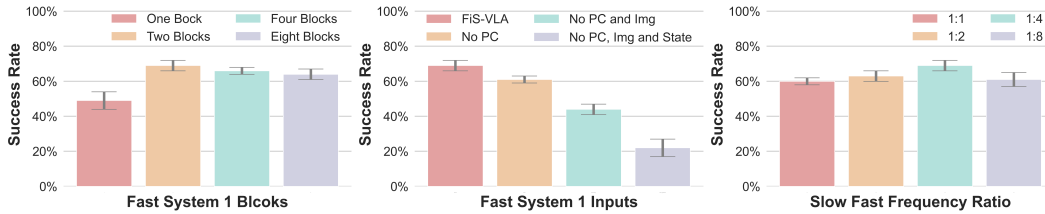


Figure 3: **Ablation study.** We investigate the impact of (1) the parameters of System 1’s shared blocks within System 2, (2) different modality inputs to System 1, and (3) the operating frequency ratio between the two systems on final manipulation success rates.

(1) The parameters of System 1’s shared blocks within System 2. In this exploration, we set the operating frequency ratio between the two systems to 1:4 and use all modality inputs. By gradually increasing the number of shared transformer blocks reconstructed from the VLM-based System 2 into the fast System 1 (from 1 to 8), we observe an improvement in manipulation performance, which tends to saturate when two blocks are used. These results show that embedding System 1 within the VLM-based System 2 enables it to inherit rich pretrained knowledge, achieving stable manipulation with relatively few parameters while maintaining high inference speed. **(2) Different modality inputs to System 1.** We compare the combinations of input information into fast System 1, evaluating the cases of using only latent features from slow System 2, adding robot state, and further incorporating 2D images and 3D point clouds. System 1 is composed of 2 transformer blocks, and the asynchronous frequency ratio between the two systems is set to 1:4. The results show that each modality substantially contributes to improving manipulation performance. The robot state provides access to the robot’s internal status, while 3D point clouds enhance the understanding of geometric structure and spatial relationships. **(3) The operating frequency ratio between the two systems.** We empirically set different asynchronous frequency ratios between System 2 and System 1 (from 1:1 to 1:8). Note that in this experiment, we use 2 transformer blocks for System 1 while retaining all modality inputs. The results show that when the ratio is 1:4, FiS-VLA excels the best performance, striking a perfect balance between slow reasoning and fast action generation. This validates that the asynchronous coordination frequency design not only improves the action generation rate but also increases the informational richness of observations provided to the execution module. **(4) Training strategy.** If $\mathcal{L}_{\text{slow}}$ is removed during training, manipulation performance drops from 69% to 62%. This result underscores the importance of our dual-aware co-training strategy, which preserves the integrity and inherent reasoning capabilities of the System 2 model, thereby providing more effective latent guidance for System 1 execution. To further assess whether preserving System 2’s reasoning improves FiS-VLA’s action generation, we replaced discrete action supervision with task plans automatically generated by Gemini [75] and manually verified. FiS-VLA achieves a 73% average success rate with plan-based co-training, outperforming the 69% obtained using discrete actions. This suggests that explicit reasoning supervision leads to more accurate conditioning of System 1 and improved performance. More ablation experiments can be found in Appendix B.

4.3 Real-World Experiment

Self-collected data. For dual-arm tasks, we evaluate four tasks on the Agilex Robot and AlphaBot respectively, each equipped with three camera views: a static exterior view, a right-wrist view, and a left-wrist view. On the Agilex Robot, we conduct the following four tasks: 1) *Pick objects and place in basket*, 2) *Lift ball and place in basket*, 3) *Place bottles at rack*, 4) *Wipe blackboard*. On the AlphaBot, we perform another set of four tasks: 1) *Pick bowl and place object*, 2) *Handover object and place*, 3) *Pour water and move cup*, 4) *Fold towel and place in bucket*. For each task, we collect 100 demonstrations via master-puppet teleoperation, with objects placed in varying positions on the table to ensure diversity. Additional implementation details can be found in the Appendix A.

Training and evaluation details. We evaluate FiS-VLA against π_0 [23], using the same training setup as in simulation, with the exception of three-view RGB inputs for real-world dual-arm tasks. Evaluation is conducted using the final checkpoint over 20 rollouts across varied tabletop positions. Note that we control the Agilex Robot using end-effector poses and the AlphaBot using joint positions, demonstrating our model’s effectiveness across different robot control paradigms.

Table 2: **Comparison of FiS-VLA and π_0 in real-world scenarios.** We train all methods in a single-task setting [26] and report the success rates. Success is determined by human evaluation based on whether the task is completed.

Models	Agilex Dual-Arm Robot Task					AlphaBot Dual-Arm Robot Task				
	Pick and place	Lift ball and place	Place bottles at rack	Wipe blackboard	Mean S.R. $\uparrow$	Pick bowl and place object	Handover and place	Pour water and move	Fold towel and place	Mean S.R. $\uparrow$
π_0 [23]	0.70	0.75	0.55	0.35	0.59	0.65	0.75	0.65	0.40	0.61
FiS-VLA	0.80	0.75	0.70	0.45	0.68	0.80	0.80	0.75	0.60	0.74

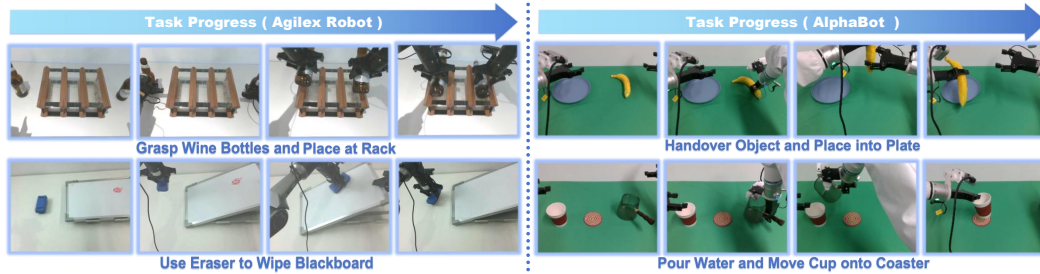


Figure 4: Visualization of real-world experiments with Agilex and AlphaBot dual-arm robots.

Quantitative and qualitative results. As shown in Table 2, FiS-VLA consistently outperforms the baseline π_0 across eight real-world tasks. On the Agilex Robot, FiS-VLA achieves a mean success rate of 68%, compared to 59% for π_0 . Notably, FiS-VLA achieves significantly higher success rates in complex manipulation tasks requiring precise spatial reasoning. For example, in the *Place Bottles at Rack* task, our method attains a 70% success rate compared to π_0 55%. Similarly, on the AlphaBot platform, FiS-VLA achieves a higher mean success rate of 74%, surpassing π_0 61%. The greatest improvement is seen in the *Fold towel and put* task, which involves manipulating deformable objects. Qualitative results in Figure 4 showcase FiS-VLA’s ability to execute diverse tasks across robots, including sequential bottle manipulation and blackboard erasing on Agilex, as well as fine-grained actions like pouring water on AlphaBot. These outcomes highlight the model’s effective coordination of high-level reasoning and low-latency control, enabling adaptive behavior in real-world settings. Additional visualizations and failure cases are provided in Appendix C and D, respectively.

4.4 Generalization Experiment

To evaluate the generalization of FiS-VLA in real-world settings, we conduct three test scenarios involving unseen manipulated objects, complex backgrounds, and varying lighting conditions, as shown in Table 3 and Figure 5. We compare FiS-VLA with the baseline model π_0 on two tasks: *place bottles at rack* using the Agilex platform and *Pick bowl and place object* using the AlphaBot platform. **(1) Unseen manipulated objects.** This experiment evaluates the generalization of FiS-VLA to novel object instances. For example, the banana is replaced with a visually distinct hot dog bun. FiS-VLA demonstrates a smaller performance drop compared to π_0 across both platforms. Notably, on AlphaBot, FiS-VLA experiences only a 19% reduction in accuracy, whereas π_0 suffers a 38% drop. These results demonstrate that under the proposed FiS-VLA dual-system paradigm, embedding the System 1 execution module within the VLM-based System 2 allows it to better inherit the rich pretrained knowledge of the VLM and more effectively interpret the high-level reasoning latent features provided by System 2. **(2) Complex backgrounds.** To simulate distracting environments, we introduce visually cluttered scenes containing irrelevant objects such as mugs, hamburgers, and bottles. These test whether the model can comprehend human instructions and task-relevant information while ignoring distractions. FiS-VLA demonstrates more stable performance than π_0 , with only a 25% drop in accuracy on AlphaBot and a 29% drop on Agilex. This validates that System 2 of FiS-VLA excels at focusing on semantically relevant objects through contextual reasoning, while System 1 ensures execution remains aligned with real-time visual cues. **(3) Varying lighting conditions.** Lighting variation is a common real-world challenge that often negatively impacts the model’s perception. In this setting, FiS-VLA still demonstrates strong generalization capabilities, achieving over 50% manipulation success on both robotic platforms. These results highlight the

Table 3: **Generalization experiments.** “Object”, “Background”, and “Lighting” refer to unseen manipulated objects, complex backgrounds, and illumination disruption, respectively.

Task Robot	Place Bottles at Rack Agilex Robot		Pick Bowl and Place Object AlphaBot	
Models	FiS-VLA	π_0 [23]	FiS-VLA	π_0 [23]
Original	0.70	0.55	0.80	0.65
Object	0.55 (-21%)	0.40 (-27%)	0.65 (-19%)	0.40 (-38%)
Background	0.50 (-29%)	0.35 (-36%)	0.60 (-25%)	0.40 (-38%)
Lighting	0.50 (-29%)	0.40 (-27%)	0.55 (-31%)	0.35 (-46%)

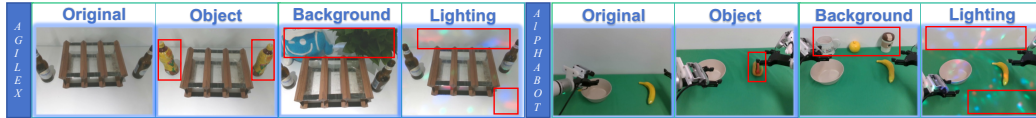


Figure 5: Visualization of generalization setting with key differences highlighted using red box.

importance of the heterogeneous modality input design in FiS-VLA’s dual systems, which enhances robustness to perceptual perturbations.

5 Conclusion and Limitation

In this paper, we introduce Fast-in-Slow, a novel dual-system VLA foundation model that embeds a fast execution module (System 1) seamlessly within a VLM-Based slow reasoning model (System 2), thereby achieving high-frequency action generation while maintaining the reasoning capability of pre-trained VLMs. We conducted a comprehensive investigation into the dual-system architecture, analyzing their divergent task objectives, asynchronous operating frequencies, and heterogeneous input modalities. Furthermore, we propose a novel dual-aware co-training strategy that enables joint optimization of both systems. However, FiS-VLA statically configures the shared parameters of System 1 within System 2 and the collaboration frequency between the two systems. We hypothesize that enabling dynamic adaptation of these factors based on task demands and environmental complexity could lead to a more robust and generalizable model, which will be a key focus of our future work. Finally, the social impact of our work is detailed in Appendix E.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62476011), and in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project AoE/E-407/24-N, as well as the InnoHK Clusters via Hong Kong Center for Logistics Robotics.

References

- [1] 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.
- [2] 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, 2025.
- [3] 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.
- [4] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.
- [5] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.
- [6] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.
- [7] 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.
- [8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspier Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.
- [9] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.
- [10] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.
- [11] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.
- [12] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

- [13] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.
- [14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- [15] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [16] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.
- [17] Senqiao Yang, Jiaming Liu, Renrui Zhang, Mingjie Pan, Ziyu Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Hongsheng Li, Yandong Guo, et al. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9247–9255, 2025.
- [18] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.
- [19] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023.
- [20] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma,

Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minh Heo, Kyle Hsu, Jiaheng Hu, Donovan Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

- [21] Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.
- [22] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.
- [23] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [24] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- [25] figureai. Helix: A vision-language-action model for generalist humanoid control. <https://www.figure.ai/news/helix>. Accessed 2025.5.7.
- [26] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.
- [27] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.
- [28] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024.
- [29] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.
- [30] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.
- [31] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.
- [32] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

- [33] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.
- [34] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.
- [35] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2023.
- [36] Shirin Joshi, Sulabh Kumra, and Ferat Sahin. Robotic grasping using deep reinforcement learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1461–1466. IEEE, 2020.
- [37] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.
- [38] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.
- [39] 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.
- [40] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [41] Xiaoqi Li, Lingyun Xu, Jiaming Liu, Mingxu Zhang, Jiahui Xu, Siyuan Huang, Iaroslav Ponomarenko, Yan Shen, Shanghang Zhang, and Hao Dong. Crayonrobo: Toward generic robot manipulation via crayon visual prompting. 2024.
- [42] Chuyan Xiong, Chengyu Shen, Xiaoqi Li, Kaichen Zhou, Jiaming Liu, Ruiping Wang, and Hao Dong. Autonomous interactive correction mllm for robust robotic manipulation. In *8th Annual Conference on Robot Learning*, 2024.
- [43] Ran Xu, Yan Shen, Xiaoqi Li, Ruihai Wu, and Hao Dong. Naturalvlm: Leveraging fine-grained natural language for affordance-guided visual manipulation. *arXiv preprint arXiv:2403.08355*, 2024.
- [44] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation. *arXiv preprint arXiv:2405.17418*, 2024.
- [45] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.
- [46] Xiaoqi Li, Jingyun Xu, Mingxu Zhang, Jiaming Liu, Yan Shen, Iaroslav Ponomarenko, Jiahui Xu, Liang Heng, Siyuan Huang, Shanghang Zhang, et al. Object-centric prompt-driven vision-language-action model for robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27638–27648, 2025.
- [47] 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.

- [48] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18061–18070, 2024.
- [49] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.
- [50] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110, 2024.
- [51] Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. *arXiv preprint arXiv:2403.11289*, 2024.
- [52] 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.
- [53] 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.
- [54] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models, 2025.
- [55] Xiaoqi Li, Liang Heng, Jiaming Liu, Yan Shen, Chenyang Gu, Zhuoyang Liu, Hao Chen, Nuowei Han, Renrui Zhang, Hao Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025.
- [56] Zhuoyang Liu, Jiaming Liu, Jiadong Xu, Nuowei Han, Chenyang Gu, Hao Chen, Kaichen Zhou, Renrui Zhang, Kai Chin Hsieh, Kun Wu, et al. Mla: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. *arXiv preprint arXiv:2509.26642*, 2025.
- [57] Sixiang Chen, Jiaming Liu, Siyuan Qian, Han Jiang, Lily Li, Renrui Zhang, Zhuoyang Liu, Chenyang Gu, Chengkai Hou, Pengwei Wang, et al. Ac-dit: Adaptive coordination diffusion transformer for mobile manipulation. *arXiv preprint arXiv:2507.01961*, 2025.
- [58] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@CoRL2023*, 3:5, 2023.
- [59] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025.
- [60] Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: Latent codes as bridges in hierarchical robot control, 2024.

- [61] ByungOk Han, Jaehong Kim, and Jinhyeok Jang. A dual process vla: Efficient robotic manipulation leveraging vlm, 2024.
- [62] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.
- [63] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.
- [64] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.
- [65] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *International Conference on Computer Vision (ICCV)*, 2023.
- [66] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [67] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Jiaming Liu, Han Xiao, Chaoyou Fu, Hao Dong, and Peng Gao. No time to train: Empowering non-parametric networks for few-shot 3d scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2024.
- [68] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [69] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [70] Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *arXiv preprint arXiv:2411.02359*, 2024.
- [71] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [72] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.
- [73] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.
- [74] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [75] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

- [76] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.
- [77] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.
- [78] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.
- [79] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.
- [80] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.
- [81] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023.
- [82] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv preprint arXiv:2307.08927*, 2023.
- [83] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. *CoRR*, abs/1811.02790, 2018.
- [84] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023.
- [85] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. <https://sites.google.com/view/berkeley-ur5/home>.
- [86] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark, 2023.
- [87] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.
- [88] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.
- [89] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.
- [90] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncured robot data. *arXiv preprint arXiv:2210.10047*, 2022.
- [91] Minh Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.
- [92] Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens Dataset. August 2023.

- [93] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.
- [94] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.
- [95] Gabriel Quere, Annette Hagenhuber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Joern Vogel. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, page 7, Paris, France, 2020.
- [96] Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *7th Annual Conference on Robot Learning*, 2023.
- [97] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023.
- [98] Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingyu Ding, Wei Zhan, and Masayoshi Tomizuka. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. 2023.
- [99] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023.
- [100] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.
- [101] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.
- [102] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.
- [103] Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, 2019.
- [104] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools, 2023.
- [105] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023.
- [106] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.
- [107] Federico Ceola, Lorenzo Natale, Niko Sünderhauf, and Krishan Rana. Lhmanip: A dataset for long-horizon language-grounded manipulation tasks in cluttered tabletop environments. *arXiv preprint arXiv:2312.12036*, 2023.
- [108] Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Abhishek Gupta, and Aravind Rajeswaran. Robohive: A unified framework for robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [109] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.
- [110] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not have theoretical proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We did.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We did not pose such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We did.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We did not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We did not involve with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We didn't.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix A Additional Dataset Details. In this section, the construction of real-world datasets for large-scale pretraining is described. Subsequently, the self-collected simulator and real-world datasets used for downstream task fine-tuning are introduced.

Appendix B Additional Quantitative Results. In this section, we present additional ablation studies, which include an investigation into the impact of action chunking on manipulation performance and control frequency, as well as a deeper exploration of heterogeneous modality inputs for System 1 and System 2. Moreover, we conducted experiments to validate the effectiveness of both the shared parameters in FiS-VLA and the proposed method using a small-scale LLM backbone. Finally, we report the detailed success rates for each task across all ablation experiments, including those presented in both the main paper and the appendix.

Appendix C Additional Qualitative Results. In this section, we provide additional visualizations of both simulation and real-world tasks. Compared to the main paper, this part offers a more detailed illustration of the execution process for each task.

Appendix D Failure Case Analysis. In this section, we analyze failure cases observed when deploying FiS-VLA to control dual-arm robots in real-world scenarios.

Appendix E Broader Impact. A brief discussion on the potential broader impact of our work.

A Additional Dataset Details

A.1 Large-scale pretraining dataset

Similar to RDT [29] and CogACT [22], we assemble a large-scale pre-training dataset by integrating existing open-source robotic datasets. Our pre-training corpus consists of 37 datasets, totaling 860k trajectories and 36 million frames. By including both single-arm and recent dual-arm datasets such as RDT and RoboMIND [32], our pre-training corpus enhances the model’s ability to generalize across diverse robotic control configurations. Table 4 provides a comprehensive list of all datasets used in pre-training along with their corresponding sampling weights. Both the number of trajectories and sampling weights can be automatically adjusted during dataset assembly. Following the preprocessing pipeline introduced in [7], we reformulate the dataset to preserve both end-effector trajectory control and joint position control for robot actions. Regarding observations, due to structural discrepancies across datasets, we use only single-view 2D RGB images as visual inputs during pre-training. During fine-tuning, FiS-VLA supports both single-view and multi-view inputs, depending on the task requirements and robot hardware configuration. For instance, in AgileX and AlphaBot dual-arm robot tasks, we use three camera views: one exterior camera and two wrist cameras, in order to mitigate occlusions caused by the robot arms. Furthermore, leveraging our heterogeneous modality design, the Fast System 1 of FiS-VLA is equipped to process point cloud data derived from exterior-view depth maps, computed using the camera’s intrinsic and extrinsic parameters. It is worth noting that, although the number and modality of input images differ between pre-training and fine-tuning, the training objectives and overall training recipe remain consistent. Consequently, this variation does not degrade downstream manipulation performance; instead, the integration of multi-view and multimodal inputs contributes to a more robust manipulation policy.

A.2 Simulation dataset

We follow the simulation setup used in PerAct and RVT, employing CoppeliaSim to collect 10 RL Bench [33] tabletop tasks, which are executed using a Franka Panda robot equipped with a two-finger parallel gripper. These tasks cover pick-and-place, tool use, articulated object manipulation, and several precise control tasks, including: *Close box*, *Close laptop*, *Toilet seat down*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out*, *Frame off hanger*, *Wine at rack*, and *Water plants*, similar to prior work [30, 28]. Although the simulator environment includes multiple RGB-D cameras, we only leverage the front-view camera to obtain RGB images and point cloud inputs. Following previous work [5, 73], we collect 100 trajectories per task using pre-defined waypoints and the Open Motion Planning Library, and apply the same frame-sampling method to extract keyframes for building the training dataset. The visualizations of the execution process in simulation are shown in Figure 8 and Figure 9.

Table 4: The dataset name and sampling weight used in our mixed large-scale pretraining dataset.

Training Dataset Mixture	
Fractal [40]	6.8%
Kuka [76]	10.5%
Bridge[77, 78]	4.9%
Taco Play [79, 80]	2.5%
Jaco Play [81]	0.4%
Berkeley Cable Routing [82]	0.2%
Roboturk [83]	2.0%
Viola [84]	0.8%
Berkeley Autolab UR5 [85]	1.0%
Toto [86]	1.7%
Language Table [87]	3.7%
Stanford Hydra Dataset [88]	3.8%
Austin Buds Dataset [89]	1.8%
NYU Franka Play Dataset [90]	0.7%
Furniture Bench Dataset [91]	2.1%
UCSD Kitchen Dataset [92]	<0.1%
Austin Sailor Dataset [93]	1.9%
Austin Sirius Dataset [94]	1.5%
DLR EDAN Shared Control [95]	<0.1%
IAMLab CMU Pickup Insert [96]	0.7%
UTAustin Mutex [97]	1.9%
Berkeley Fanuc Manipulation [98]	0.6%
CMU Stretch [99]	0.1%
BC-Z [100]	6.3%
FMB Dataset [101]	6.0%
DobbE [102]	1.2%
DROID [20]	14.2%
Stanford Kuka Dataset [103]	0.3%
Stanford Robocook Dataset [104]	0.2%
Columbia Cairlab Pusht Real [3]	<0.1%
UCSD Pick and Place	0.8%
Maniskill [105]	7.5%
Berkeley RPT [106]	<0.1%
QUT Dexterous Manipulation [107]	<0.1%
RoboSet [108]	5.2%
BridgeData V2 [78]	9.3%
RoboMind [32]	1.2%

A.3 Self-collected real-world dataset

For real-world experiments, we evaluate four tasks each on the Agilex Robot and the AlphaBot robot. Below, we detail the hardware configurations, data collection protocols, and task setting for both platforms.

Agilex robot setup. As summarized in Table 5, the Agilex Robot is equipped with two 6-DoF arms mounted on a mobile base. As shown in Figure 6, two Orbbec DABAI cameras capture the left and right wrist views, while a RealSense 435 camera mounted overhead provides exterior-view RGB images and point cloud data. All cameras record at 30 Hz. For trajectory recording and control, we use end-effector poses. The four tasks conducted on the Agilex Robot are as follows:

1) *Pick objects and place in basket.* The robot uses both arms to pick up two objects according to a language command and place them into a container. This task assesses the model’s understanding of spatial positioning.

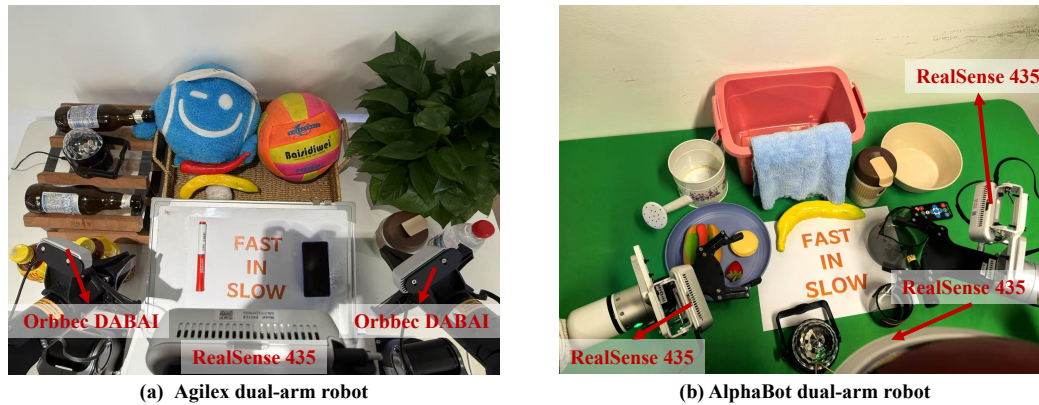


Figure 6: **Real-world assets and camera configurations.** We present visualizations of the real-world assets and camera setups used in the Agilex and AlphaBot dual-arm robot tasks, respectively.

Table 5: **The hardware setups of the two dual-arm robots, including the number of joints and their corresponding angle ranges of motion.**

Agilex dual-arm robot		AlphaBot dual-arm robot	
Joint number	Angle range	Joint number	Angle Range
J1	$-154^{\circ} \sim +154^{\circ}$	J1	$-178^{\circ} \sim +178^{\circ}$
J2	$0^{\circ} \sim +195^{\circ}$	J2	$-130^{\circ} \sim +130^{\circ}$
J3	$-175^{\circ} \sim 0^{\circ}$	J3	$-178^{\circ} \sim +178^{\circ}$
J4	$-106^{\circ} \sim +106^{\circ}$	J4	$-135^{\circ} \sim +135^{\circ}$
J5	$-75^{\circ} \sim +75^{\circ}$	J5	$-178^{\circ} \sim +178^{\circ}$
J6	$-100^{\circ} \sim +100^{\circ}$	J6	$-128^{\circ} \sim +128^{\circ}$
-	-	J7	$-180^{\circ} \sim +180^{\circ}$

2) *Lift ball and place in basket.* The robot must synchronize both arms to grasp a ball held between the grippers and transport it without slippage. This task evaluates dual-arm coordination.

3) *Place bottles at rack.* Each arm grasps a bottle from its side, rotates it, and aligns it parallel to the rack. This task tests inter-object relationship reasoning and precise rotational manipulation.

4) *Wipe blackboard.* One arm holds the board while the other erases red marker using an eraser. This setup tests precise, coordinated actions in dual-arm scenarios.

AlphaBot robot setup. As shown in Table 5, the AlphaBot leverages two 7-DoF arms mounted on a mobile base. As shown in Figure 6, three RealSense 435 cameras are used to capture the left wrist, right wrist, and exterior views, while only the exterior view is used for point cloud generation. All modalities are recorded at 30 Hz. To evaluate the model’s robustness to different control schemes, we adopt joint position control for both trajectory collection and inference execution. For each task, we collect 100 demonstrations using master-puppet teleoperation, with object positions randomized on the table to promote data diversity. Language instructions are manually created and diversified via augmentation. The four tasks evaluated on the AlphaBot include:

1) *Pick bowl and place object.* The robot uses its left arm to pick up a bowl and its right arm to pick up an object, placing the object into the bowl. This task involves coordinated dual-arm manipulation, where each arm performs distinct, asymmetric roles.

2) *Handover object and place.* The right arm picks up an object and hands it to the left arm. The arms must avoid collisions and ensure proper grasp alignment. The left arm then places the object into a plate. This task serves as a comprehensive benchmark for evaluating the model’s capabilities in 3D perception, grasp reasoning, and dual-arm motion planning. It poses significant challenges while remaining highly practical for real-world bimanual manipulation scenarios.

3) *Pour water and move cup.* The robot grasps a cup handle with its right arm, rotates it to pour water into another cup, then moves the receiving cup to a coaster. This task combines high-precision

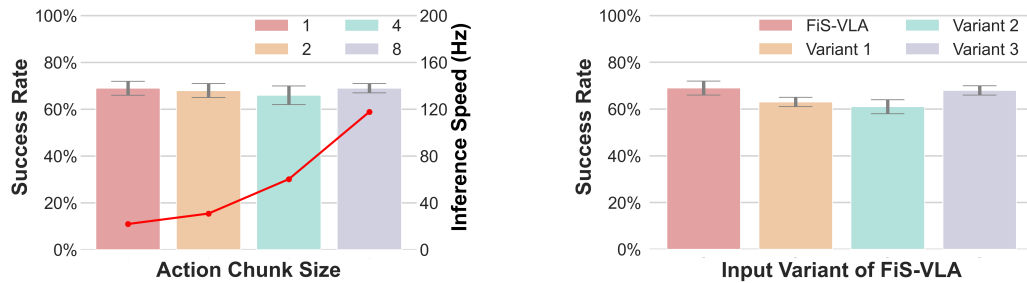


Figure 7: **Ablation studies on action chunk size and input variants of FiS-VLA.** (Left) Impact of different action chunk sizes on success rate and inference speed. While increasing action chunk size leads to improved inference speed, success rate remains relatively stable. (Right) Comparison of success rates among FiS-VLA and its input variants, showing FiS-VLA achieves the best performance.

pose control, physical reasoning, and multi-stage planning, making it a representative benchmark for evaluating precision-oriented manipulation capabilities

4) *Fold towel and place in bucket.* The robot folds a deformable towel using both arms, then places it into a bucket. This task evaluates coordinated manipulation of deformable objects.

B Additional Quantitative Results

B.1 Action chunking for robust and high-Frequency robot control

In closed-loop control of robots, a key challenge is the compounding of errors, where an early mistake can cascade into subsequent decisions, ultimately driving the system’s observations far from the training distribution and leading to unrecoverable failures [109]. To mitigate this issue, inspired by the concept of *action chunking*, researchers have explored predicting multiple actions at once [29, 22, 23, 110]. This approach reduces the number of decision points along a trajectory, thereby decreasing the opportunity for error accumulation. Moreover, it enables higher effective control frequencies, resulting in smoother and more continuous robot motions. By considering sequences of actions jointly, the model can enforce temporal consistency and avoid abrupt changes that could physically damage the robot. In this work, we investigate the effect of action chunking by predicting future action sequences of length H ranging from one to eight, as illustrated in Figure 7 and Table 11. We observe that the performance of FiS-VLA remains stable across different values of H , while the control frequency increases proportionally. Notably, when predicting eight future actions in a single step, the theoretical control frequency reaches up to **117.7 Hz**, demonstrating the potential of our method for high-speed, high-fidelity robotic control.

B.2 Multi-modal input configuration analysis

In our study, we found that incorporating multi-modal inputs consisting of 2D images, 3D point clouds, and robot state information into System 1 of FiS-VLA significantly improves execution accuracy. Based on this observation, we conducted a more comprehensive investigation to evaluate the impact of different combinations of these modalities when provided to System 1 and System 2. We refer to these configurations as the **input variants of FiS-VLA**. The results are presented in Figure 7 and Table 12. In **Variant 1**, System 2 receives language instructions, 2D images, and 3D point clouds, while System 1 takes 2D images and robot state as input. This configuration leads to slightly lower control accuracy compared to the original FiS-VLA. In **Variant 2**, the robot state input is moved from System 1 to System 2, resulting in a marginal performance drop relative to Variant 1. Finally, in **Variant 3**, we explored a configuration where both System 1 and System 2 receive 2D images and 3D point clouds. Additionally, System 1 receives the robot state and System 2 receives the language instruction. This setup achieves performance that is nearly equivalent to the original FiS-VLA. These results demonstrate the robustness and flexibility of the FiS-VLA architecture in integrating multi-modal information for high-precision robotic control.

B.3 Effectiveness of Parameter Sharing

To evaluate the effectiveness of our parameter-sharing design, we conducted the following ablation studies. **Experiment 1:** The VLM serves as System 2, and the last two LLM layers are duplicated to form an independent System 1. This configuration supports both autoregressive generation from System 2’s VLM and diffusion-based action prediction through the duplicated LLM layers. The entire system is trained using identical settings with FiS-VLA. **Experiment 2:** This experiment mirrors the setup of Experiment 1, with the difference being that four LLM layers are duplicated instead of two. As shown in the Table 6, both variants underperform compared to FiS-VLA. We attribute this to feature misalignment. The LLM was pretrained across all 32 layers, enabling hierarchical (layer-by-layer) processing. Duplicating the last few layers breaks this chain (e.g., in Experiment 1, the new 31st layer in System 1 must process outputs from System 2’s 32nd layer), despite never being pretrained to do so. This disrupts feature compatibility and limits performance. In contrast, FiS-VLA’s parameter-sharing keeps the LLM’s original structure and information flow intact. System 1 remains embedded within System 2, preserving pretrained knowledge while enabling action execution. This design proves more efficient and effective than separated architectures.

Table 6: Results of additional Transformer blocks as System 1 on RL Bench.

	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
Experiment 1	0.80	0.95	0.95	0.55	0.80	0.55	0.40	0.70	0.30	0.10	0.61 $\pm$ 0.03
Experiment 2	0.90	0.85	0.90	0.35	0.90	0.45	0.45	0.50	0.45	0.10	0.59 $\pm$ 0.05
FiS-VLA	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 $\pm$ 0.03

B.4 Performance of FiS-VLA with a Smaller-Scale LLM

We further conducted an additional comparison using the 2.7B Phi-2 model as the LLM backbone. To ensure fairness, FiS-VLA (2.7B) was also pretrained on the same assembled robotic datasets under identical settings. As shown in the Table below, the compact-sized FiS-VLA also achieves satisfactory performance, which demonstrates the effectiveness of our approach and its generalizability to different VLM backbones.

Table 7: Results of FiS-VLA with 2.7B and 7B LLMs on RL Bench.

	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
FiS-VLA (2.7B)	0.90	0.85	0.90	0.55	0.80	0.55	0.40	0.65	0.45	0.15	0.62 $\pm$ 0.03
FiS-VLA (7B)	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 $\pm$ 0.03

B.5 The detailed results for each experimental setting

We have presented all the results of the ablation studies in both the main paper and the appendix in a fine-grained manner, as shown in Table 8 to Table 12.

Table 8: Results of different fast System 1 blocks on RL Bench. The results in this table correspond to the first subplot of Figure 3 in the main paper.

Fast System 1 blocks	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
One block	0.70	0.55	0.95	0.55	0.80	0.05	0.20	0.70	0.30	0.10	0.49 $\pm$ 0.05
Two blocks (FiS-VLA)	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 $\pm$ 0.03
Four blocks	1.00	0.90	1.00	0.45	0.85	0.35	0.60	0.75	0.40	0.25	0.66 $\pm$ 0.02
Eight blocks	0.90	0.80	0.95	0.55	0.95	0.45	0.45	0.65	0.40	0.30	0.64 $\pm$ 0.03

Table 9: **Results of different fast System 1 input on RL Bench.** The results in this table correspond to the second subplot of Figure 3 in the main paper.

Fast System 1 input	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
FiS-VLA	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 ± 0.03
No PC	0.90	0.90	0.85	0.35	0.85	0.20	0.55	0.80	0.50	0.15	0.61 ± 0.02
No PC and Img	0.45	0.45	0.95	0.30	0.75	0.10	0.50	0.50	0.30	0.10	0.44 ± 0.03
No PC, Img and State	0.50	0.30	0.15	0.00	0.65	0.05	0.55	0.45	0.00	0.00	0.22 ± 0.05

Table 10: **Results of different slow fast frequency ratio on RL Bench.** The results in this table correspond to the third subplot of Figure 3 in the main paper.

Frequency ratio	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
1:1	0.95	0.80	0.85	0.30	1.00	0.40	0.40	0.65	0.45	0.20	0.60 ± 0.02
1:2	0.90	0.85	1.00	0.30	0.90	0.30	0.55	0.70	0.45	0.30	0.63 ± 0.03
1:4 (FiS-VLA)	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 ± 0.03
1:8	0.85	0.90	0.95	0.55	0.95	0.30	0.45	0.85	0.15	0.10	0.61 ± 0.04

Table 11: **Results of different action chunk size on RL Bench.** The results in this table correspond to the first subplot of Figure 7 in the appendix.

Action chunk size	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
1	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 ± 0.03
2	1.00	0.85	1.00	0.50	0.85	0.40	0.75	0.65	0.35	0.40	0.68 ± 0.03
4	1.00	0.90	1.00	0.25	0.90	0.70	0.65	0.55	0.25	0.40	0.66 ± 0.04
8	0.70	0.90	0.95	0.30	0.90	0.70	0.65	0.60	0.50	0.65	0.69 ± 0.02

Table 12: **Results of different input variants of FiS-VLA on RL Bench.** The results in this table correspond to the second subplot of Figure 7 in the appendix.

Input variant	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. & Var
FiS-VLA	1.00	1.00	0.95	0.55	0.90	0.50	0.50	0.70	0.55	0.20	0.69 ± 0.03
Variant 1	0.95	0.90	1.00	0.45	0.85	0.40	0.55	0.70	0.45	0.05	0.63 ± 0.02
Variant 2	0.90	0.90	0.95	0.35	0.80	0.50	0.45	0.55	0.45	0.20	0.61 ± 0.03
Variant 3	1.00	0.90	0.95	0.50	0.85	0.70	0.50	0.65	0.55	0.20	0.68 ± 0.02

C Additional Visualizations

This section presents keyframe visualizations of FiS-VLA performing tasks in the RL Bench simulator and on two real-world robotic platforms: the Agilex Robot and AlphaBot. These visualizations complement the experimental results discussed in the main paper. Figures 8 and 9 depict the execution of tasks by the Franka Panda Arm within the RL Bench simulation environment. In this simulated setting, ten representative tasks are demonstrated, each broken down into key execution steps. These keyframes intuitively illustrate FiS-VLA’s action selection and execution logic at various stages, showcasing its robust capabilities in sequential action prediction and gripper state control.

In real-world scenarios, FiS-VLA is evaluated across eight diverse tasks on two distinct robotic platforms. Figures 10 and 11 provide keyframe snapshots of task execution by the Agilex Robot and AlphaBot, respectively. On the Agilex Robot, tasks such as *Place bottles at rack* and *Wipe blackboard* highlight FiS-VLA’s ability to perform reliable dual-arm coordination and spatial generalization in cluttered, unstructured environments. Furthermore, we evaluate FiS-VLA on long-horizon, multistage tasks, such as *Handover object and place* and *Pour water and move cup*, which require managing sequential dependencies and diverse manipulation skills. In these scenarios, FiS-VLA demonstrates consistent performance across stages and effectively utilizes dual-arm collaboration when necessary, enabling the successful execution of tasks that require both synchronized actions and long-term planning. These results collectively validate FiS-VLA’s strong generalization across domains and platforms, reinforcing its promise as a versatile and scalable visuomotor policy for real-world robotic manipulation.

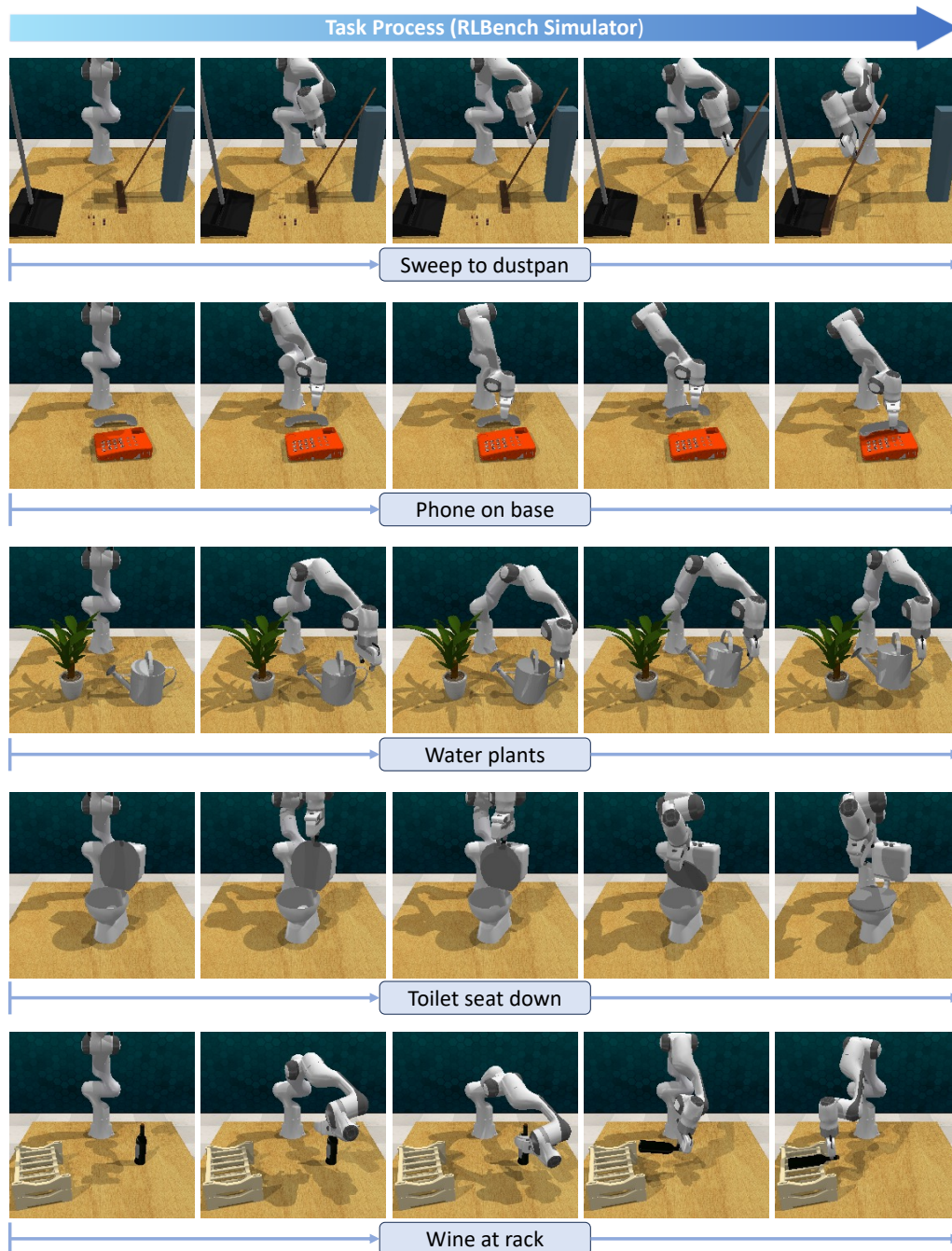


Figure 8: **RLBench visualization.** We visualize key frames of the agent's execution process from the front perspective.

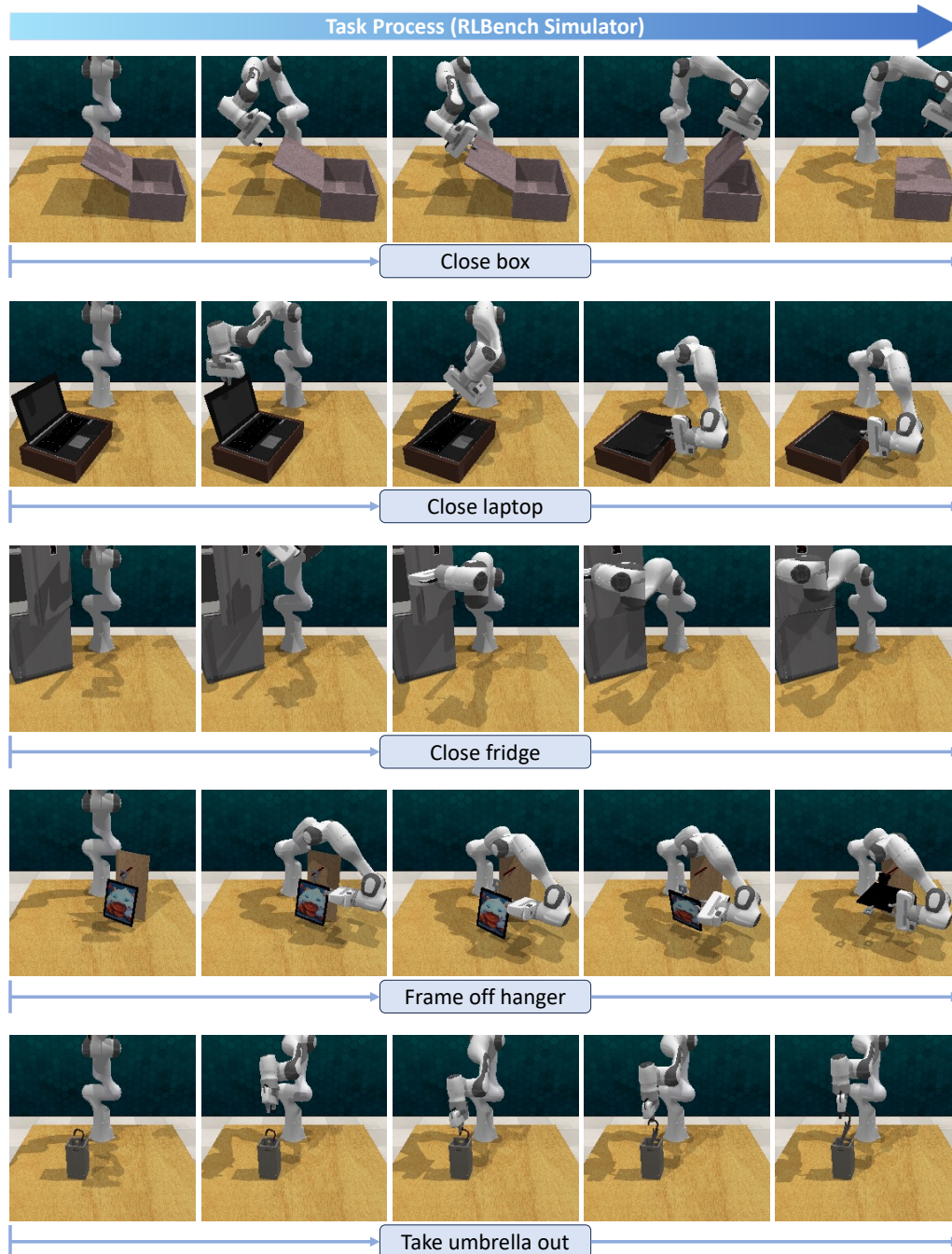


Figure 9: **RLBench visualization.** We visualize key frames of the agent's execution process from the front perspective.

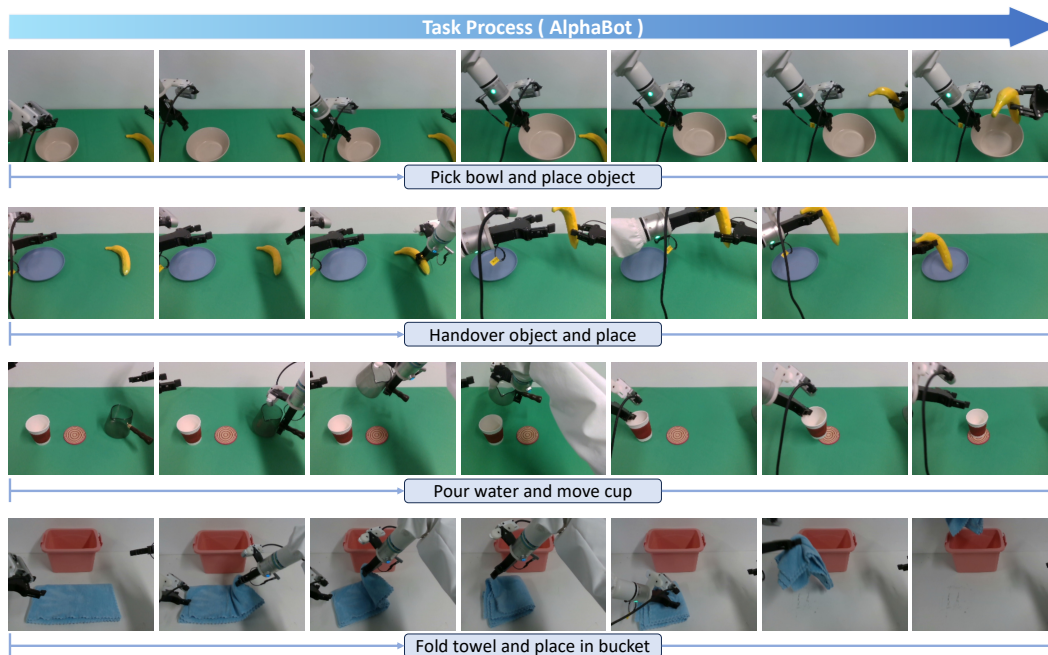


Figure 10: **Agilex robot task execution visualization.** We visualize key frames of the agent's execution process from a static exterior view.

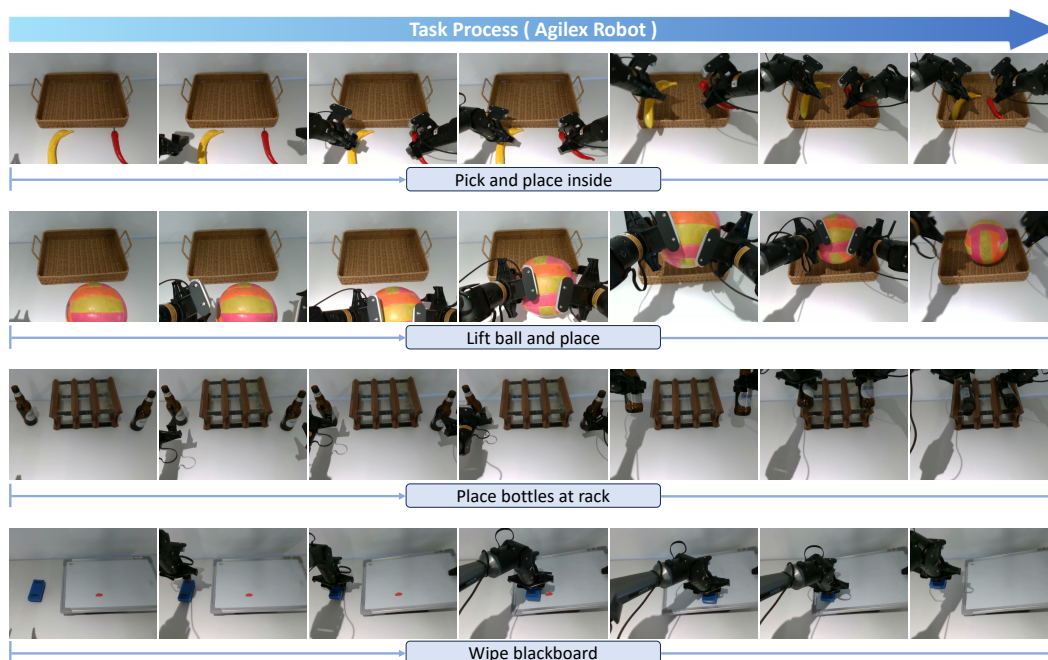


Figure 11: **AlphaBot task execution visualization.** We visualize key frames of the agent's execution process from a static exterior view.

D Failure Case Analysis.

Through real-world experiments on the AlphaBot platform, we observe four specific failure cases encountered by our proposed FiS-VLA, as visualized in Figure 12. Red bounding boxes highlight the critical error frames during each execution sequence.

- 1) The first case involves a **bimanual collision** during the *Handover object and place into plate* task. The left and right arms interfere with each other while attempting to transfer the object, indicating insufficient inter-arm motion coordination and suboptimal wrist camera placement.
- 2) The second case, observed in the *Fold towel and place in bucket* task, is related to an error in **manipulation height**. The predicted joint positions fail to control gripper contact with the towel, revealing the difficulty of height prediction when dealing with thin, deformable objects.
- 3) The third case, from the *Pick bowl and place object* task, reflects a failure in **manipulation position**. The robot mispredicts the location of the banana, resulting in a failed grasp attempt.
- 4) The fourth case presents a **handover rotation error** in the *Handover object and place into plate* task. The right arm rotates the object into an unsuitable orientation, preventing the left arm from executing a stable handover grasp.

These issues can be mitigated by collecting more high-quality demonstrations and incorporating efficient constraints during training to improve robustness in real-world control. Furthermore, enabling our System 2 to recognize and correct failure actions will be a key direction for future work.

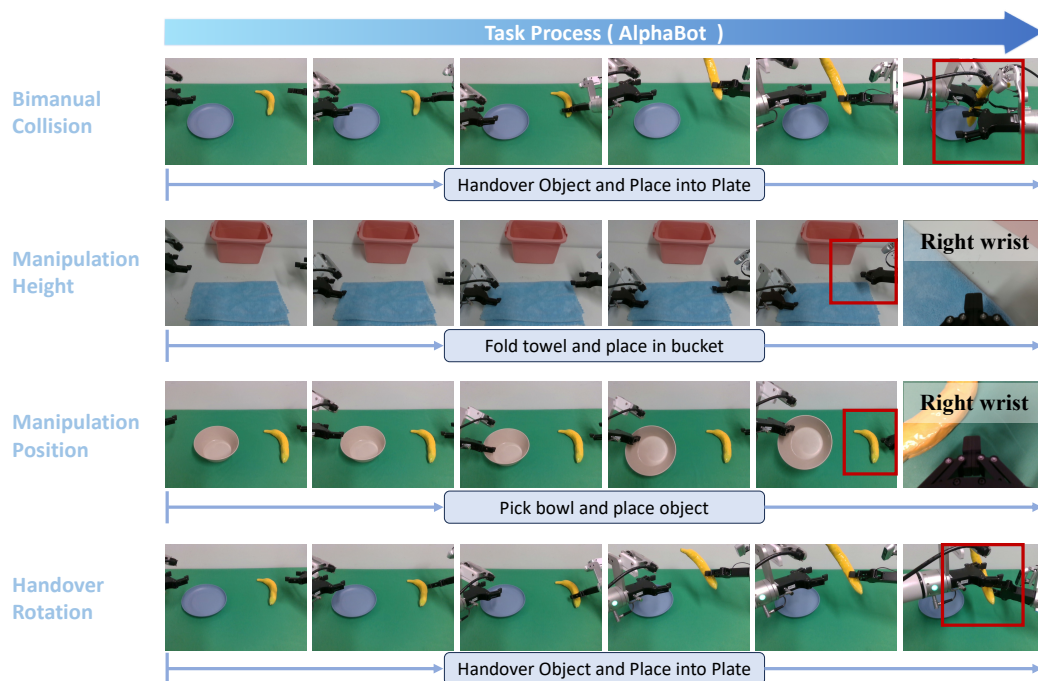


Figure 12: **Failure case visualization.** We visualize the failure cases observed in four real-world experiments, with key error frames during execution highlighted using red bounding boxes.

E Broader Impact

Our work proposes a foundation model for robotic manipulation that integrates high-level reasoning and low-latency action execution within a unified end-to-end Vision-Language-Action (VLA) framework. While the FiS-VLA model improves control efficiency and leverages pretrained reasoning capabilities, it may introduce potential risks when deployed in real-world environments. These risks include safety concerns in high-speed closed-loop control and unsafe behaviors resulting from the misinterpretation of human instructions. To mitigate such risks, future deployments should incorporate strict safety constraints and task-specific operational boundaries. Furthermore, our framework provides robust control and generalizable reasoning capabilities for robotic assistance in domains such as elder care and home automation, where responsiveness and reliability are critical.