TwinVLA: Data-Efficient Bimanual Manipulation with Twin Single-Arm Vision-Language-Action Models

Hokyun Im[1,2] Euijin Jeong[1] Andrey Kolobov[2] Jianlong Fu[2] Youngwoon Lee[1]
[1]Department of Artificial Intelligence, Yonsei University [2]Microsoft Research
https://jellyho.github.io/TwinVLA/

Abstract

Vision-language-action models (VLAs) trained on large-scale robotic datasets have demonstrated strong performance on manipulation tasks, including bimanual tasks. However, because most public datasets focus on single-arm demonstrations, adapting VLAs for bimanual tasks typically requires substantial additional bimanual data and fine-tuning. To address this challenge, we introduce TwinVLA, a modular framework that composes two copies of a pretrained single-arm VLA into a coordinated bimanual VLA. Unlike monolithic cross-embodiment models trained on mixtures of single-arm and bimanual data, TwinVLA improves both data efficiency and performance by composing pretrained single-arm policies. Across diverse bimanual tasks in real-world and simulation settings, TwinVLA outperforms a comparably-sized monolithic RDT-1B model without requiring *any* bimanual pretraining. Furthermore, it narrows the gap to state-of-the-art model, π_0 which rely on extensive proprietary bimanual data and compute cost. These results establish our modular composition approach as a data-efficient and scalable path toward high-performance bimanual manipulation, leveraging public single-arm data.

1 Introduction

Thanks to publicly available large-scale robotic datasets, vision-language-action models (VLAs) have shown impressive performance in single-arm robotic manipulation, generalizing across diverse tasks, objects, and environments (Zitkovich et al., 2023; Open X-Embodiment Collaboration et al., 2024; Kim et al., 2024; Black et al., 2024). However, extending these successes to *bimanual* manipulation remains challenging, as public bimanual datasets are scarce, and existing approaches often rely on large, proprietary datasets that require thousands of hours of data collection and curation (Black et al., 2024), limiting reproducibility and progress.

Can we build strong bimanual VLAs without collecting or fine-tuning on large bimanual datasets by leveraging existing single-arm data? Recent cross-embodiment learning work typically trains monolithic models on multi-robot datasets (Open X-Embodiment Collaboration et al., 2024) by employing embodiment-specific action decoders (Octo Model Team et al., 2024; Doshi et al., 2024; NVIDIA et al., 2025) or shared, zero-padded action spaces (Liu et al., 2024; Black et al., 2024). Although promising, differences in observation and action spaces introduce heterogeneity, forcing a single model to handle disparate action spaces, and monolithic training underutilizes the modular structure inherent to bimanual tasks.

A modular perspective on bimanual manipulation is supported by neuroscience: human bimanual manipulation is the coordination of arm-specific motor primitives rather than a single monolithic controller. Dedicated neural circuits, such as the Supplementary Motor Area (SMA) and the corpus callosum, orchestrate and synchronize the two arms (Sadato et al., 1997; Swinnen, 2002). Similar principles have proven effective in vision-language modeling, where interaction between modality-specific backbones improves its efficiency and effectiveness (Liang et al., 2024).

Inspired by these insights, we propose TwinVLA, a modular architecture that operationalizes this coordination-centric view. Instead of training from scratch, TwinVLA leverages a pretrained single-arm VLA. Specifically, we first design a lightweight, compact single-arm VLA, which we

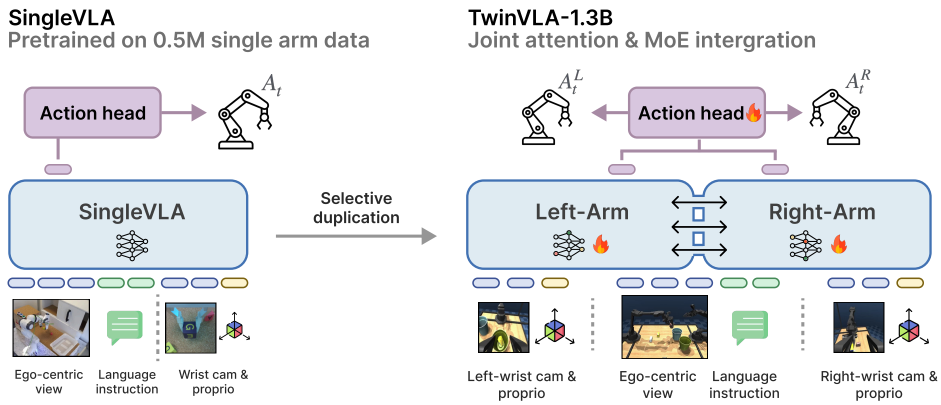


Figure 1: **Overview of TwinVLA.** Inspired by humans' two-arm coordination for bimanual manipulation, TwinVLA duplicates a VLM backbone pretrained on cross-embodiment single-arm data (*Left*) to form two arm-specific branches linked via **Joint Attention** (*Right*). Shared inputs (ego-centric views, language instructions) are routed via a mixture-of-experts (MoE) to improve computational efficiency. Only the VLM backbone is duplicated, keeping the increase in model size minimal.

call SingleVLA (Appendix A). We pretrain SingleVLA for single-arm manipulation on the OXE dataset (Open X-Embodiment Collaboration et al., 2024). We then duplicate this SingleVLA and integrate the two "twin" instances through a lightweight coordination method. This design is highly data-efficient: it eliminates the need for a bimanual pretraining dataset and achieves strong performance with only a small amount of bimanual demonstrations for fine-tuning.

To integrate two SingleVLAs into a bimanual policy, TwinVLA utilizes a joint attention (Liang et al., 2024) across the twin models, as illustrated in Figure 1. This allows the twin SingleVLAs to exchange information and coordinate their actions, while preserving their pretrained capabilities. This approach is made feasible without significant overhead, as we duplicate only the VLM backbone and utilize a Mixture-of-Experts (MoE) to efficiently manage shared inputs. In contrast to monolithic cross-embodiment models (Liu et al., 2024; Octo Model Team et al., 2024; Doshi et al., 2024), our approach yields better performance and data efficiency, significantly reducing the need for large-scale bimanual data collection and compute.

We evaluate TwinVLA across a broad range of environments, including a complex, long-horizon real-world task and a diverse suite of bimanual manipulation tasks in simulations. Despite leveraging only public single-arm data and limited bimanual fine-tuning data, TwinVLA achieves performance comparable to state-of-the-art bimanual policies.

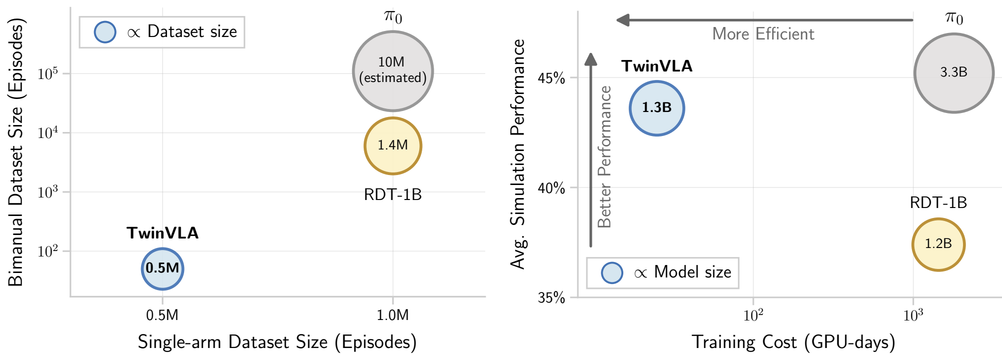
In summary, our main contributions are threefold:

- We propose a novel modular architecture for bimanual manipulation that integrates two copies of a pretrained SingleVLA with a lightweight coordination method based on joint attention with MoE, enabling synchronized two-arm control.
- We present a data-efficient paradigm that adapts our twin architecture into a capable bimanual policy for a target task by fine-tuning on only a small bimanual dataset, crucially without requiring additional pretraining, thereby eliminating the need for large-scale bimanual data.
- Through extensive experiments across real and simulated bimanual tasks, TwinVLA matches or surpasses state-of-the-art models trained on far larger bimanual data and compute.

Together, these findings identify our modular SingleVLA composition approach as a scalable, efficient path to high-performance bimanual manipulation.

2 Related Work

Vision-Language-Action models (VLAs) represent a promising approach to robotic control. They adapt large pretrained Vision-Language Models (VLMs) (Liu et al., 2023b; Karamcheti et al., 2024;



(a) Data efficiency comparison

(b) Compute efficiency comparison

Figure 2: **(a) Data efficiency.** While RDT-1B and π_0 use 1M + single-arm data with sizable bimanual data, TwinVLA uses $\sim$0.5M single-arm data and only 50 target bimanual data. **(b) Compute efficiency.** RDT-1B and π_0 require high compute (exceeding 1,000 H100 GPU-days), whereas TwinVLA achieves higher or comparable performance at substantially lower compute (25 H100 GPU-days).

Chen et al., 2024) to robotic control (Open X-Embodiment Collaboration et al., 2024; Khazatsky et al., 2024) by fine-tuning on action-labeled data. This process enables them to translate language instructions and visual context into grounded actions. Early VLA models explore various strategies for connecting VLMs with action generation (Ahn et al., 2022). RT-2 (Zitkovich et al., 2023) tokenizes actions as part of the model's output vocabulary, whereas RoboFlamingo (Li et al., 2024b) uses a separate continuous action head. Recent work, such as π_0 (Black et al., 2024) and CogAct (Li et al., 2024a), further integrates a denoising policy with a VLM to improve action execution accuracy.

Bimanual manipulation policies are essential for enabling robots to perform complex tasks that require coordinated two-handed control, such as folding laundry (Bersch et al., 2011; Avigal et al., 2022), assembling parts (Stavridis & Doulgeri, 2018), or wiping the plate Black et al. (2025); Chi et al. (2024b). Learning effective bimanual policies is challenging due to high-dimensional, tightly coupled action spaces and the scarcity of high-quality bimanual demonstrations (Lee et al., 2020; Xie et al., 2020). Consequently, specialist methods, such as Diffusion policy (Chi et al., 2023) and ACT (Zhao et al., 2023), trained only on target-task demonstrations, struggle on precise, long-horizon bimanual tasks (Black et al., 2024).

To mitigate data scarcity, prior work has largely pursued a straightforward approach: training a single, unified model through large-scale bimanual data collection and computationally intensive pretraining. RDT-1B (Liu et al., 2024) is fine-tuned on over 6K bimanual episodes after extensive pretraining on the large-scale OXE dataset (Open X-Embodiment Collaboration et al., 2024), reportedly requiring a month on 48 H100 GPUs. Similarly, π_0 (Black et al., 2024) relies on a $10,000$-hour proprietary dataset with substantial computational cost. Moreover, because their data are not publicly accessible, reproducibility and broader adoption are limited.

In contrast to such monolithic cross-embodiment policies and compute-heavy bimanual pretraining, our approach adopts a modular, coordination-centric design. We first train a SingleVLA on large-scale public single-arm data, duplicate the pretrained SingleVLA and couple them via joint attention, and then fine-tune it on bimanual tasks—allowing each stage to benefit from the most suitable data (see Figure 2). This composition-based approach avoids bimanual pretraining, requires only a small amount of bimanual fine-tuning, better preserving the capabilities of single-arm policies, and improves data and compute efficiency.

3 Preliminaries

This paper aims to develop a data-efficient framework for learning bimanual manipulation policies by building upon pretrained single-arm Vision-Language-Action (SingleVLA) models. This section formalizes the single-arm and bimanual settings, and describes the conditional flow-matching objective for training action heads of VLAs.

3.1 Formulating the Bimanual VLA Policy

Our goal is to extend a pretrained SingleVLA π_{single} into a bimanual policy π_{twin} applicable to target bimanual tasks. A VLA $\pi(A_t \mid o_t)$ predicts an *action chunk* $A_t = (a_t, a_{t+1}, \ldots, a_{t+T-1})$ of length T from an observation o_t. For single-arm manipulation, the observation $o_t^{\text{single}} = ((l, I_{\text{ego}})_t, (I_{\text{wrist}}, d)_t)$ includes a language prompt l, a ego-centric image I_{ego} (shared input), and an arm-specific wrist image I_{wrist} with proprioception d (arm-specific input). We train $\pi_{\text{single}}(A_t \mid o_t^{\text{single}})$ to predict the action chunk for one arm. For bimanual manipulation, the observation aggregates both right (R) and left (L) arm-specific input, $o_t^{\text{twin}} = \big((l, I_{\text{ego}})_t, (I^R_{\text{wrist}}, d^R)_t, (I^L_{\text{wrist}}, d^L)_t\big)$, and the policy $\pi_{\text{twin}}(A^R_t, A^L_t \mid o_t^{\text{twin}})$ outputs a joint action chunk for right and left arms.

3.2 Training VLAs with Conditional Flow Matching

We aim for the VLA model to predict continuous robot actions from observations. Each observation o_t is tokenized: language via a language tokenizer, images via a vision encoder, and proprioception via an MLP encoder. We append a learnable *readout* token r_t to the observation token sequence, which signals the model to produce actions. These tokens are fed into the VLM backbone, and the embedding h_t corresponding to the readout token r_t is obtained from the final hidden state.

To enable continuous action prediction from VLM outputs, we attach an action head using conditional flow matching for both π_{single} and π_{twin}. The action head $v_\theta(A_t^\tau, h_t, d_t)$ is trained with the following flow-matching loss function:

$$\mathcal{L}^T(\theta) = \mathbb{E}_{p(A_t \mid o_t), q(A_t^\tau \mid A_t)} \| v_\theta(A_t^\tau, h_t, d_t) - \mathbf{u}(A_t^\tau \mid A_t) \|^2, \tag{1}$$

where h_t is the VLM output hidden state, d_t is proprioception. The objective trains the action head to predict the reference flow from a noised action chunk A_t^τ to the target action chunk A_t, where $\tau \in [0, 1]$ denotes the flow matching timesteps. We adopt a simple linear Gaussian path $q(A_t^\tau \mid A_t) = N(\tau A_t, (1-\tau)I)$, which has demonstrated robust performance across domains.

Specifically, we first sample a noisy action $A_t^\tau = \tau A_t + (1-\tau)\epsilon$, where the noise $\epsilon \sim N(0, I)$ and the timestep $p(\tau) = Beta(\frac{0.999-\tau}{0.999}; 1.5, 1)$, following π_0 (Black et al., 2024). The action head $v_\theta(A_t^\tau, h_t, d_t)$ and the VLM are jointly trained end-to-end to predict the reference flow $u(A_t^\tau \mid A_t) = \epsilon - A_t$.

During inference, we sample actions using the forward Euler integration method. Starting from $A_0 \sim N(0, I)$, we iteratively update the action using the learned flow $v_\theta(A_t^\tau, h_t, d_t)$:

$$A_t^{\tau+\delta} = A_t^\tau + \delta v_\theta(A_t^\tau, h_t, d_t), \tag{2}$$

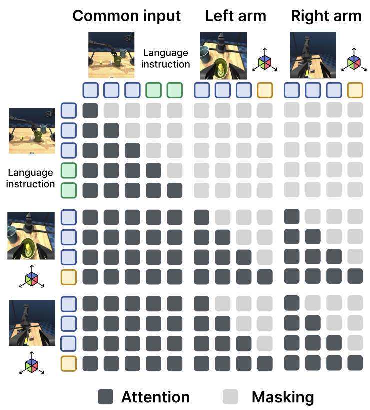
where we set the sampling step $n = 10$ and use $\delta = \frac{1}{n}$. The entire model, including the vision encoder and VLM backbone, is trained in an end-to-end manner.

4 TwinVLA

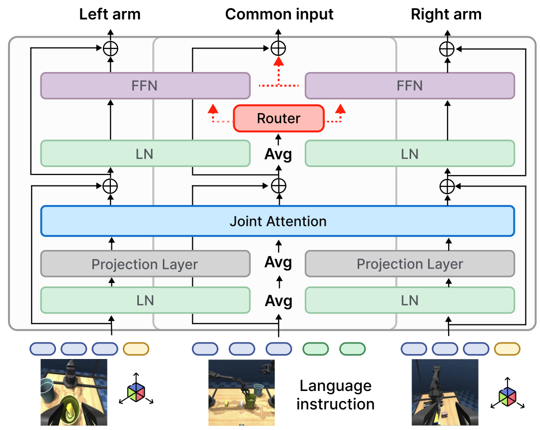
TwinVLA is a modular architecture that transforms a pretrained SingleVLA into a coordinated bimanual policy through three core principles. First, we selectively duplicate modules from a pretrained SingleVLA to form a bimanual policy (Section 4.1). We then introduce a joint attention mechanism to enable effective cross-arm coordination between the duplicated VLMs (Section 4.2). Finally, we integrate a Mixture-of-Experts (MoE) layer for shared observations to enhance computational efficiency without sacrificing performance (Section 4.3).

4.1 Single-Arm Policy Duplication

To construct TwinVLA from SingleVLA, we initialize the twin policies for the left and right arms by copying the pretrained SingleVLA. However, instead of duplicating the full model, we share the vision encoder and DiT (Peebles & Xie, 2022) action head while fully replicating the VLM. Each arm



(a) Causal joint attention mask



(b) Transformer block of TwinVLA

Figure 3: **(a) Causal attention mask for joint attention.** It preserves causality while processing shared, left, and right inputs in parallel. **(b) TwinVLA joint attention mechanism.** The two VLMs share information, and the shared modality $(l, I_{\text{ego}})_t$ is further processed by MoE to more efficiently leverage both VLMs.

has its own lightweight proprioception encoder. This design yields a compact 1.3B-parameter model, comparable to the 1.2B-parameter RDT-1B, without significantly increasing computational cost.

Visual inputs are processed by the shared encoder, and each VLM produces readout tokens that are jointly decoded by the shared DiT. This design is motivated by the principle that general visual understanding (image encoding) and low-level motor control (action decoding) are largely embodiment-agnostic skills that can be effectively shared for both arms. In contrast, the VLM, which decides output action given encoded observation, is fully replicated to allow for specialized control.

4.2 Joint Attention for Cross-arm Fusion

We integrate arm-specific inputs using a joint attention mechanism inspired by Mixture of Transformers (MoT) (Liang et al., 2024). As illustrated in Figure 3(b), this is achieved by sharing only the self-attention layers across the VLM backbones, while other components like the projection networks operate independently for each arm. Unlike π_0 (Black et al., 2024), which links VLM with an action head, we connect two VLMs directly. The detailed pseudocode of joint attention is presented in Algorithm 1.

Causal joint attention mask. Effective joint attention requires appropriate attention masking. Standard LLMs use a lower-triangular attention mask for causal prediction. To support joint attention among the shared and arm-specific inputs, we designed the attention mask for TwinVLA as shown in Figure 3a. Specifically, we embed lower-triangular masks within each arm's region while treating the shared modality as fully accessible. Each arm also attends to half of the other's tokens, enabling symmetric cross-arm interaction without violating autoregressive constraints.

4.3 Mixture-of-Experts Integration

In TwinVLA, each VLM receives both shared and arm-specific inputs to produce single-arm actions. Feeding the shared modality $(l, I_{\text{ego}})_t$ redundantly to both VLMs is inefficient, as it increases token length and VRAM usage in a way that hinders the practical training of large models. To address this, we apply a Mixture-of-Experts (MoE) mechanism that routes the shared modality tokens between both VLMs, reducing token length while preserving representational power.

While the MoE effectively shares the feedforward networks (FFNs) between the twin VLM backbones, other key components like projection layers remain duplicated. To address this, we implement an effective form of task arithmetic (Tang et al., 2024). We process an input through the corresponding component in each VLM and then average their outputs. This output-averaging technique functionally simulates a single shared layer without averaging the parameter itself. This is illustrated in the center of Figure 3b. Thanks to this change, we enable training the model with a batch size of 8 on a single 40GB GPU. Detailed implementation is provided in Appendix C.3.

Attention re-weighting. A potential side effect of introducing new arm-specific tokens is that the model's learned attention patterns can be disrupted, shifting focus away from the pretrained shared modalities. To mitigate this and preserve the valuable pretrained knowledge, we re-scale the attention scores for the shared modality. This maintains pretrained modality importance, allowing the model to bypass an initial adaptation phase and focus directly on the target task—a benefit evidenced by a lower initial loss and converged loss during fine-tuning.

5 EXPERIMENTS

In this paper, we propose TwinVLA to achieve strong bimanual manipulation performance with minimal bimanual data by fully leveraging a single-arm VLA pretrained on abundant single-arm data. Our empirical studies aim to answer the following questions:

- How does TwinVLA compare to state-of-the-art methods across diverse bimanual tasks, without any bimanual pretraining (Sections 5.2 and 5.3)?
- How quickly can TwinVLA adapt to new bimanual tasks (Section 5.4)?
- Does TwinVLA retain core VLA properties—language-following and robustness to unseen scenes and instructions (Sections 5.5 and 5.6)?
- How much does each key design choice contribute to overall performance (Section 5.7)?

5.1 COMPARED METHODS

We evaluate TwinVLA against three bimanual manipulation policies, each representing a different point in the design space.

- **RDT-1B** (Liu et al., 2024): This serves as our direct baseline. With a comparable size(1.2B vs. TwinVLA's 1.3B parameters), it represents the standard monolithic approach that requires substantially larger resources (1.4M trajectories, $\sim$1,440 H100 days vs. 0.5M single-arm data, $\sim$25 H100 days).
- **π_0** (Black et al., 2024): We include this as a skyline, as this is 3.3B-parameter VLA trained on over 10K hours of proprietary robot data. Our goal is to assess how closely TwinVLA can approach this performance ceiling with far greater efficiency.
- **Diffusion Policy (DP)** (Chi et al., 2023): This is a strong baseline method in low-data regime with 271M parameters, used to demonstrate the crucial benefits of pretraining.

(a) Real-world tasks (b) Simulation tasks

Figure 4: **Experimental setups.** (a) We evaluate TwinVLA on three real-world bimanual tasks using an Anubis robot. (b) We further analyze TwinVLA on a large suite of simulation tasks: 5 tasks in Tabletop-Sim and 50 tasks in RoboTwin 2.0.

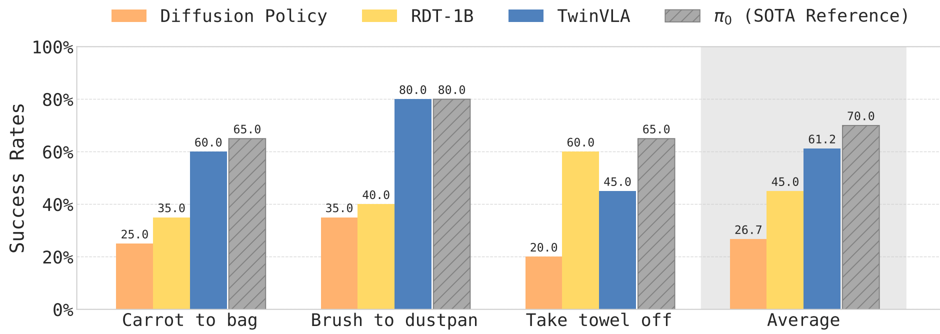


Figure 5: **Success rates on real-world tasks.** TwinVLA outperforms RDT-1B and DP on average. Moreover, TwinVLA shows comparable performance with π_0 while trained only on target data.

5.2 Real-World Experiments

Environment. For real-world experiments, we use a dual-arm robot, Anubis (Kang et al., 2025), as shown in Figure 4a. Anubis has two 6 DoF arms with parallel-jaw grippers. The robot is equipped with two wrist-mounted cameras and a single ego-centric view camera.

Tasks. We design three long-horizon tabletop manipulation tasks which requires careful coordination and accurate motions: `carrot to bag`, `brush to dustpan`, and `take towel off`. We collect 50 episodes for each task using absolute EEF control. We fine-tune all methods for each task, and evaluate them with 20 rollouts per task.

Results. As presented in Figure 5, our model, TwinVLA, significantly outperforms its direct competitor, RDT-1B. This result is particularly noteworthy given that TwinVLA was pretrained on a substantially smaller single-arm dataset (0.5M vs. RDT-1B's 1.4M) with far less computational cost, demonstrating the data efficiency of our approach. The comparatively low performance of DP further underscores the critical importance of pretraining. Among all methods, π_0 ultimately demonstrated the best overall performance.

5.3 Simulation Experiments

RoboTwin 2.0. We use the RoboTwin 2.0 benchmark (Chen et al., 2025a), consisting of 50 bimanual tasks. Adhering to the official evaluation protocol, we fine-tune a model per task with 50 generated demonstrations and perform 100 test rollouts under both "Easy" and "Hard" settings. For Easy tasks, test scenes match the training data, but the instructions are novel. The Hard tasks introduce variations in texture, object position, and height. For compared methods, we use the results reported from RoboTwin 2.0 (Chen et al., 2025a).

Tabletop-Sim. To assess dexterous scenarios beyond tasks in RoboTwin, we develop Tabletop-Sim[1], a tabletop simulation environment based on `dm_control` (Tunyasuvunakool et al., 2020) and assets from ALOHA2 (Team et al., 2024) and GSO object dataset (Downs et al., 2022). We design 5 representative tasks that require precise bimanual coordination. Specifically, we define four single-tasks and one multi-task: `dish-drainer`, `handover-box`, `shoes-table`, `lift-box`, and `put X box into Y pot`. In the "Hard" tasks, we vary background textures and objects. We collect 50 episodes on each task using absolute EEF control, and fine-tune a model per task, and perform 500 evaluation rollouts for both "Easy" and "Hard" settings.

Results. The results in Figure 6 show the average success rates of TwinVLA and compared methods. DP, trained from scratch, shows the worst performance, highlighting the importance of pretraining. Once again, we observe that TwinVLA outperforms RDT-1B in most scenarios, except for the RoboTwin Hard tasks, and achieves comparable performance with π_0 by effectively leveraging single-arm data and modularity of bimanual manipulation. Notably, in Tabletop-Sim Easy tasks, TwinVLA even outperforms π_0, which is trained on an extensive corpus of high-quality bimanual

[1]Our simulation setup is similar to the concurrent work Aloha-Sim, released by Google DeepMind (Google DeepMind, 2025).

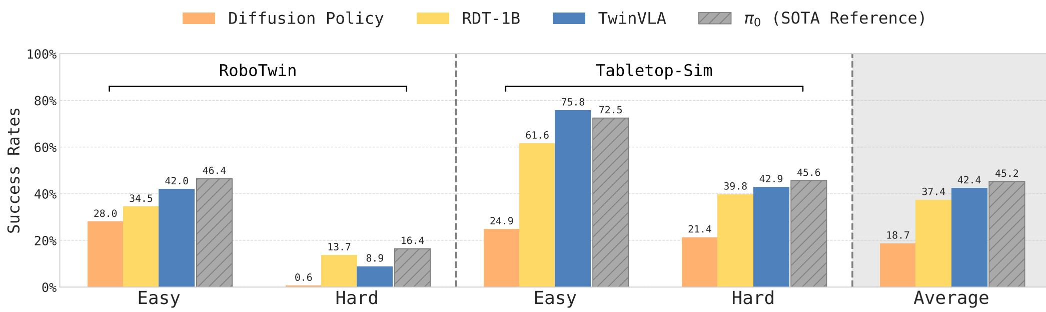


Figure 6: **Average success rates for diverse bimanual tasks.** Despite being pretrained solely on single-arm datasets, **TwinVLA** outperforms other methods except π_0.

pretraining data. This demonstrates TwinVLA's advantages in scenarios demanding higher dexterity and significant bimanual coordination.

5.4 Data Efficiency

TwinVLA exhibits data efficiency in two key aspects: pretraining and fine-tuning. For pretraining, it is efficient because it does not require supplemental bimanual data. For fine-tuning, it learns new tasks rapidly because its structural inductive bias facilitates the efficient transfer and application of its pretrained single-arm knowledge. We validate this efficiency in Tabletop-Sim Easy environment, comparing model's average success rates with varying amounts of demonstration data. As illustrated in Figure 7, TwinVLA exhibits a steep learning curve. Despite a modest start with 20 demonstrations, it quickly surpasses the performance of RDT with just 50 demonstrations, highlighting its exceptional data efficiency.

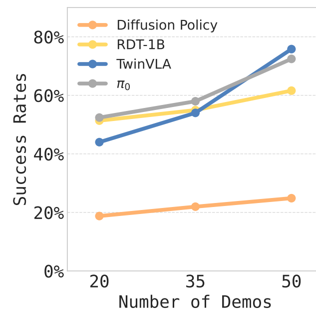


Figure 7: **Average success rates on the Tabletop-Sim Easy tasks.** Models are evaluated after fine-tuning with 20, 35, and 50 demonstrations.

5.5 Language Following Evaluations

A known challenge is that fine-tuning VLMs on robotic data can degrade their ability to faithfully follow nuanced instructions. We therefore evaluate how effectively our model preserves this core capability in a multi-task setting. The "`Put X box into Y pot`" task involves 3 box colors (`X`) and 2 pot colors (`Y`), resulting in a total of 6 possible instruction combinations. As observed in Figure 8a, TwinVLA outperforms both RDT-1B and π_0. We believe this performance stems from effectively preserving the knowledge acquired during single-arm pretraining through careful fine-tuning.

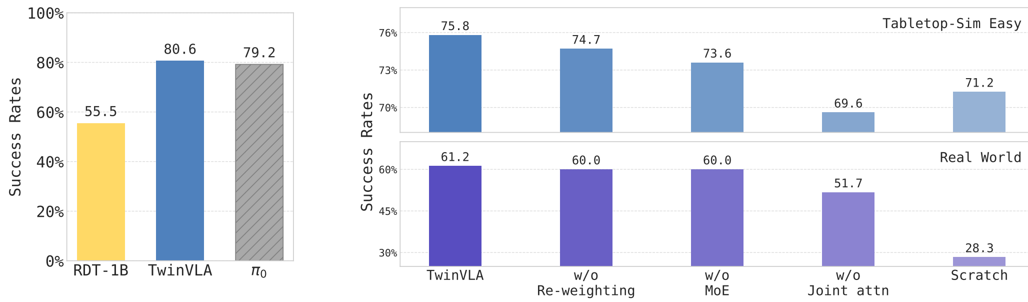
5.6 Policy Robustness

One of the advantages of VLAs is their robustness to unseen situations and novel language instructions, thanks to pretraining. As shown in Figure 6, TwinVLA outperforms RDT-1B by 3.3% even in the Hard setup of Tabletop-Sim, which involves different textures and objects.

The RoboTwin benchmark, both in the Easy and Hard setups, uses evaluation language instructions that are unseen during training. Here, TwinVLA again shows 7.48% better performance than RDT-1B in the Easy setup. Although TwinVLA's performance on the RoboTwin Hard tasks is 3.72% lower than that of RDT-1B, it still outperforms a non-pretrained Diffusion policy by 9.38%. This result demonstrates that TwinVLA possesses sufficient robustness as a bimanual VLA, even without being pretrained on large-scale bimanual manipulation data.

5.7 Ablations

In this section, we conduct a sequential ablation study to analyze the cumulative impact of our key design choices on performance. Starting from the full TwinVLA model, we progressively remove



(a) Language task results

(b) Ablation results

Figure 8: **Language following task and ablation results.** (a) We evaluate average success rates on the language following tasks in Tabletop-Sim. (b) Ablation studies in the real world and Tabletop-Sim Easy tasks.

each component in a specific order: first Attention Re-weighting, followed by MoE integration, and finally Joint Attention. This method reveals how performance degrades as each component of our architecture is stripped away. The results on our real-world and Tabletop-Sim Easy tasks are reported in Figure 8b.

Attention re-weighting. Removing the attention re-weighting mechanism (*w/o Re-weighting*) increased the initial fine-tuning loss by **40%** and decreased final performance by **1.1%** and **1.2%** in simulation and real world, respectively. This demonstrates that our re-weighting strategy successfully mitigates the input distribution shift between pretraining and fine-tuning.

MoE integration. Building on the previous ablation, we next remove the MoE integration (*w/o MoE*). This additional change increased the token sequence length by **28%** and increased VRAM usage by **21%**, making VLA training more burdensome. Surprisingly, it also further decreases the success rate by **1.1%** in simulation, suggesting that MoE integration eliminates redundant processing of shared inputs while maintaining the performance.

Joint attention. Lastly, removing the joint attention mechanism (*w/o Joint attn*) cause the most significant additional performance drops by **4.0%** and **8.3%** in simulation and real world, respectively. This impact is greater than other components, confirms that the joint attention is the critical mechanism for the bimanual manipulation requires coordination between both arms.

Effect of single-arm pretraining. As a separate, foundational experiment, we assess the role of pretraining by training a model from scratch without OXE dataset (*Scratch*). This resulted in a **4.6%** performance drop in simulation and a stark **32.9%** in real world. This result confirms that effective cross-arm coordination is essential for bimanual manipulation and validates joint attention as the critical mechanism for achieving it in our model.

Twin structure. While we have confirmed that joint attention effectively connects the two modules, a crucial question remains: how does this approach compare to a monolithic model that is inherently unified from the start? To answer this, we revisit our comparison against RDT-1B, a monolithic model of a comparable 1.2B parameter size. The results are telling: TwinVLA outperforms RDT-1B by **16.2%** in the real world, **5.0%** in simulation, and **25.1%** in language-following tasks. This provides strong evidence that the inductive bias from the Twin Structure itself is highly beneficial for bimanual manipulation, validating our design choice over a monolithic approach.

6 LIMITATIONS

Since our work adapts a pre-trained single-arm VLA's *representations* for bimanual tasks, a current limitation is that the model forgets its single-arm manipulation skills after fine-tuning. Future research into mechanisms that prevent this forgetting could address data scarcity by integrating diverse data, while also improving model explainability and better generalization capability to unseen tasks.

Moreover, the choice of action space is critical for VLA models, particularly for knowledge transfer. We adopt absolute end-effector (EEF) pose control because it provides an embodiment-agnostic representation essential for our single-arm transfer strategy. In contrast, joint positions are specific to a robot's degrees of freedom (DOF), making them unsuitable for direct policy transfer. We believe that exploring relative absolute actions (Chi et al., 2024a) or developing a shared representation across diverse embodiments could enable more efficient transfer.

7 CONCLUSION

In this paper, we introduce TwinVLA, a data-efficient VLA model for bimanual manipulation. TwinVLA provides a new perspective on solving bimanual manipulation under scarce bimanual data by leveraging abundant single-arm datasets. From a small amount of bimanual demonstration data, TwinVLA learns to coordinate two copies of a SingleVLA pretrained on large-scale single-arm data via our proposed joint attention. Through exhaustive experiments both in the real world and simulation, TwinVLA demonstrates its data-efficient learning of bimanual tasks compared to prior monolithic approaches. We believe this principle of bridging data availability gaps via leveraging modularity opens up exciting possibilities for other complex robotic domains, such as mobile manipulation, thereby broadening the impact of large-scale robotic learning.

ACKNOWLEDGMENTS

This project was supported in part by Microsoft Research Asia and the Microsoft Accelerate Foundation Models Research (AFMR) grant program. This research was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean Government (MSIT) (RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); RS-2024-00436680, Global Research Support Program in the Digital Field Program), the Alchemist Project (RS-2024-00432143) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (24ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling, and evolving ways). LG Electronics provided the Anubis robot, which was used for the experiments.

REFERENCES

N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8. IEEE, 2022.

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, 2023.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. In *Proceedings of the 7th Conference on Robot Learning*, 2023.

Christian Bersch, Benjamin Pitzer, and Sören Kammel. Bimanual robotic cloth manipulation for laundry folding. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1413–1419. IEEE, 2011.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *Robotics: Science and Systems*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=vlhoswksBO.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2022.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *CoRR*, 2024.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *CoRR*, 2025b.

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*, pp. 02783649241273668, 2023.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024a. doi: 10.15607/RSS.2024.XX.045.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *CoRR*, 2024b.

Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. Clvr jaco play dataset, 2023. URL `https://github.com/clvrai/clvr_jaco_play_dataset`.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *8th Annual Conference on Robot Learning*, 2024.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

Google DeepMind. Aloha-sim. `https://github.com/google-deepmind/aloha_sim`, 2025. Accessed: 2025-10-24.

Minho 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.

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 *5th Annual Conference on Robot Learning*, 2021.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Jim Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16923–16930, 2025. doi: 10.1109/ICRA55743.2025.11127809.

Taewoong Kang, Joonyoung Kim, Shady Nasrat, Dongwoon Song, Gijae Ahn, Minseong Jo, Seonil Lee, and Seung-Joon Yi. Anubis: A compact, low-cost, compliant humanoid mobile manipulation robot. In *24th International Conference on Humanoid Robots (Humanoids)*, 2025. URL `https://ras.papercept.net/conferences/scripts/rtf/ICHR25_ContentListWeb_2.html`.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

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, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon 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. In *Robotics: Science and Systems*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems*, 2025.

Youngwoon Lee, Jingyun Yang, and Joseph J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations*, 2020.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *CoRR*, 2024a.

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. In *The Twelfth International Conference on Learning Representations*, 2024b.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An Huang, Timo Roman, Tong Lu, Jose M. Alvarez, Bryan Catanzaro, Jan Kautz, Andrew Tao, Guilin Liu, and Zhiding Yu. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *CoRR*, 2025.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. In *Conference on Parsimony and Learning*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

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*, 2023c.

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. In *International Conference on Learning Representations*, 2024.

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. *The International Journal of Robotics Research*, 44(4):592–606, 2025. doi: 10.1177/02783649241276017. URL `https://doi.org/10.1177/02783649241276017`.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023.

Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version), 2025. URL `https://arxiv.org/abs/2409.02920`.

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.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL `https://arxiv.org/abs/2503.14734`.

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

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin

Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation*, 2024.

William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2022. URL `https://api.semanticscholar.org/CorpusID:254854389`.

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. In *Robotics: Science and systems*, 2025.

Gabriel Quere, Annette Hagengruber, 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)*, pp. 7, Paris, France, 2020.

Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. In *Conference on Robot Learning*, 2022.

Norihiro Sadato, Yoshiharu Yonekura, Atsuo Waki, Hiroki Yamada, and Yasushi Ishii. Role of the supplementary motor area and the right premotor cortex in the coordination of bimanual finger movements. *Journal of Neuroscience*, 17(24):9667–9674, 1997. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.17-24-09667.1997. URL `https://www.jneurosci.org/content/17/24/9667`.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=PwqiqaaEzJ`.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. URL `https://openreview.net/forum?id=B1ckMDqlg`.

Sotiris Stavridis and Zoe Doulgeri. Bimanual assembly of two parts with relative motion generation and task related optimization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7131–7136. IEEE, 2018.

Stephan P. Swinnen. Intermanual coordination: From behavioural principles to neural-network interactions. *Nature Reviews Neuroscience*, 3(5):348–359, 2002. doi: 10.1038/nrn807.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

ALOHA 2 Team, Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, Wayne Gramlich, Torr Hage, Alexander Herzog, Jonathan Hoech, Thinh Nguyen, Ian Storz, Baruch Tabanpour, Leila Takayama, Jonathan Tompson, Ayzaan Wahid, Ted Wahrburg, Sichun Xu, Sergey Yaroshenko, Kevin Zakka, and Tony Z. Zhao. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation, 2024. URL `https://arxiv.org/abs/2405.02292`.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL `https://www.sciencedirect.com/science/article/pii/S2665963820300099`.

Jörn Vogel, Annette Hagengruber, Maged Iskandar, Gabriel Quere, Ulrike Leipscher, Samuel Bustamante, Alexander Dietrich, Hannes Hoeppner, Daniel Leidner, and Alin Albu-Schäffer. Edan - an emg-controlled daily assistant to help people with physical disabilities. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

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. In *Conference on Robot Learning*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *CoRR*, 2024.

Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 12156–12163. IEEE, 2024.

Fan Xie, Alexander Chowdhury, M De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation. *Advances in neural information processing systems*, 33:2327–2337, 2020.

Sudhir Pratap Yadav, Rajendra Nagar, and Suril V. Shah. Learning vision-based robotic manipulation tasks sequentially in offline reinforcement learning settings. *Robotica*, 42(6):1715–1730, 2024. doi: 10.1017/S0263574724000389.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738–5746, 2019. doi: 10.1109/CVPR.2019.00589.

Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingxiao Huo, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. `https://sites.google.com/berkeley.edu/fanuc-manipulation`, 2023.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *Conference on Robot Learning*, 2022a.

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, 2022b. doi: 10.1109/LRA.2022.3146589.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

APPENDIX

A SINGLEVLA: EFFICIENT SINGLE-ARM POLICY DESIGN AND PRETRAINING

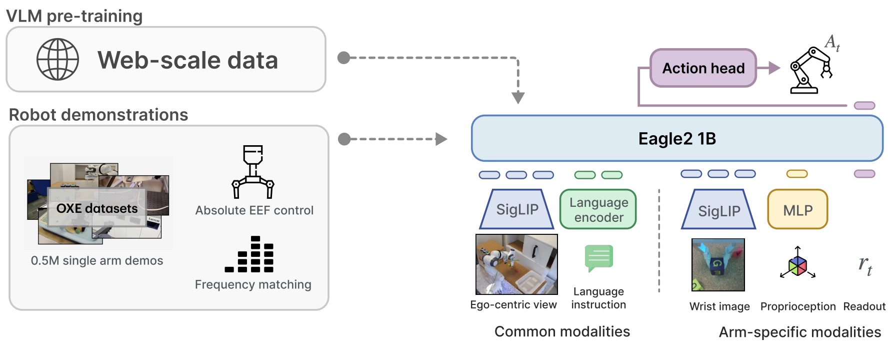


Figure 9: **Overview of SingleVLA architecture design and pretraining method.**

This section presents the design of the SingleVLA π_{single}. While SingleVLA follows established VLA conventions, our key novelty is a duplication strategy that enables the construction of TwinVLA. Prior 7B-scale models (Kim et al., 2024; 2025; Li et al., 2024a) are prohibitively large for such duplication, motivating a more efficient, lightweight SingleVLA (Fig. 9). To acquire generalizable knowledge, we pretrain SingleVLA on a 0.5M-trajectory subset of the OXE mix, enabling transfer across diverse environments and embodiments. Pretraining ran for **120k** steps and took about **5 days** on a cluster with **5**$\times$ **H100** GPUs.

To ensure effective transfer to bimanual manipulation, it is crucial to choose an appropriate *action space*. Heterogeneous joint configurations across robots induce incompatible action spaces and complicate joint training. Prior work mitigates this with robot-specific decoders or high-dimensional zero-padded spaces (NVIDIA et al., 2025; Doshi et al., 2024; Octo Model Team et al., 2024; Black et al., 2024; Liu et al., 2024). Instead, we convert all actions into absolute end-effector (EEF) poses, providing a consistent, semantically meaningful representation across robots that naturally extends to bimanual control. For rotation, we adopt a 6D representation (Zhou et al., 2019), which is well suited for neural network learning.

A.1 PRETRAINING

SingleVLA is pretrained on an OXE subset (0.5M trajectories); dataset composition and sampling rates appear in Table 1. We adopt the dataset loader from the OpenVLA (Kim et al., 2024) codebase and apply sampling according to the designated weights. Because some datasets (e.g., Kuka and BC-Z) include failed trajectories, we pre-process to retain only successful ones. All actions are converted to absolute EEF control with 6D rotations, resulting in a 10-Dimensional action space. We further apply *frequency matching* as described below.

Frequency matching. Robotic datasets differ in control frequency, making fixed-length action-chunk prediction misaligned in real time. For example, a 20-step chunk spans $\sim$ 7 seconds in RT-1 (Brohan et al., 2022) (3 Hz) but only $\sim$ 1.3 seconds in DROID (Khazatsky et al., 2024) (15 Hz). Mixing low-frequency data like OXE (Open X-Embodiment Collaboration et al., 2024) with high-frequency datasets can degrade pretraining quality. Inspired by π_0-FAST (Pertsch et al., 2025), which uses DCT (Ahmed et al., 1974) to map 1-second actions into a consistent space, we perform frequency matching via interpolation: all datasets are resampled to 20 Hz, improving temporal alignment and transfer to high-frequency bimanual tasks.

A.2 HYPERPARAMETERS AND COMPUTE

Table 2 summarizes training hyperparameters for SingleVLA and TwinVLA. SingleVLA pretraining used $5\times$ H100 GPUs for about 5 days. TwinVLA fine-tuning used $1\times$ L40S GPU for about 2 days.

Table 1: **SingleVLA pretraining datasets and sampling percentages.**

Dataset	Sample Percentage
RT-1 (Brohan et al., 2022)	24.49%
Kuka (filtered) (Yadav et al., 2024)	12.40%
BridgeV2 (Walke et al., 2023)	13.74%
Taco Play (Rosete-Beas et al., 2022)	3.10%
Jaco Play (Dass et al., 2023)	0.50%
Viola (Zhu et al., 2022a)	1.00%
Berkeley Autolab UR5 (Chen et al., 2023)	1.28%
Stanford Hydra (Belkhale et al., 2023)	4.73%
Austin Buds (Zhu et al., 2022b)	0.22%
NYU Franka Play (Cui et al., 2022)	0.88%
FurnitureBench (Heo et al., 2023)	2.40%
Austin Sailor (Nasiriany et al., 2022)	2.33%
Austin Sirius (Liu et al., 2023c)	1.84%
DLR EDAN (shared control) (Vogel et al., 2020; Quere et al., 2020)	0.05%
UT Austin Mutex (Shah et al., 2023)	2.38%
Berkeley FANUC manipulation (Zhu et al., 2023)	0.82%
CMU Stretch (Bahl et al., 2023; Mendonca et al., 2023)	0.16%
BC-Z (filtered) (Jang et al., 2021)	7.90%
FMB (Luo et al., 2025)	7.40%
Dobb-E (Shafiullah et al., 2023)	1.50%
DROID (Khazatsky et al., 2024)	10.70%

Table 2: **Key hyperparameters for TWINVLA training.**

Hyperparameter	SingleVLA	TwinVLA
Global batch size	256	8
Precision	FP32/BF16 (mixed)	FP32/BF16 (mixed)
Gradient clipping (L_2)	1.0	1.0
Learning rate	1×10^{-4}	1×10^{-4}
LR scheduler	cosine	cosine
Warm-up ratio	0.01	0.05
Total steps	120k	100k
Optimizer	AdamW	AdamW
Weight decay	1×10^{-5}	1×10^{-5}
Adam ϵ	1×10^{-8}	1×10^{-8}
Vision backbone frozen	true	true
Image augmentation	true	false

A.3 SINGLEVLA VLM ABLATION

We validate SingleVLA's VLM choice in the LIBERO (Liu et al., 2023a) environment using several VLMs. The LIBERO actions are converted to absolute EEF 6D control. Due to computational limits, we directly fine-tune the pretrained VLM checkpoints on LIBERO (i.e., without additional pretraining on LIBERO). Each model is evaluated with 500 rollouts per task suite under identical random seeds. Results are shown in Table 3.

Table 3: **Performance of different VLMs on LIBERO.**

VLM	Spatial	Object	Goal	Long	Average
Qwen2VL-2B (Wang et al., 2024)	80.4%	88.6%	83.8%	43.0%	73.9%
InternVL2.5-1B (Chen et al., 2025b)	64.6%	84.8%	78.4%	46.2%	68.5%
Eagle2-1B (Li et al., 2025)	73.4%	85.4%	90.8%	46.6%	**74.0%**

Although Qwen2VL is widely regarded as robust, Eagle2-1B achieves comparable or slightly better results while using roughly half the parameters and providing significantly faster inference. We therefore select **Eagle2-1B** as the VLM backbone for SingleVLA.

Table 4: **Performance of pretrained SingleVLA on LIBERO.**

Method	Spatial	Object	Goal	Long	Average
SingleVLA (Eagle2-1B, no pretraining)	73.4%	85.4%	90.8%	46.6%	74.0%
SingleVLA (pretrained)	**92.4%**	**94.5%**	**93.5%**	**63.7%**	**86.0%**
OpenVLA (Kim et al., 2024)	84.7%	88.4%	79.2%	53.7%	76.5%
Octo (Octo Model Team et al., 2024)	78.9%	85.7%	84.6%	51.1%	75.1%

After pretraining SingleVLA with Eagle2-1B, we fine-tune it on LIBERO to assess single-arm capability. As shown in Table 4, the pretrained SingleVLA substantially improves performance and even surpasses the 7B model OpenVLA, indicating that the learned single-arm policy is both effective and sufficiently strong to benefit the bimanual policy.

B Implementation Details

Table 5: **Training hyperparameters for baseline models.**

Method	# of params	Learning rate	Lr scheduler	Batch size	Training steps
TwinVLA	1.3B	1e-4	cosine	8	100k
RDT-1B	1.2B	1e-4	constant	8	100k
DP	271M	2e-5	cosine	8	100k
π_0	3.3B	2.5e-5	cosine	8	100k

We use the official implementation of RDT-1B. Diffusion Policy and π_0 are evaluated via the public LeRobot release (Cadene et al., 2024), with two modifications for a fair comparison. First, the LeRobot evaluation script normalized images differently from training; we corrected this to match the training pipeline.

All models are fine-tuned with the same number of steps and batch size so that the total number of training samples is consistent across methods. For learning rates, we began with each model's default and tuned within a similar compute budget. In practice, defaults worked well for DP and RDT-1B. For π_0, we observed better final returns by slowing the cosine decay; we therefore extended the LR schedule from 30k to 100k steps.

C TwinVLA Details

C.1 Joint Attention

The joint attention in TwinVLA is fundamentally almost identical to the implementation in the Mixture-of-Transformers (MoT) (Liang et al., 2024) study. While MoT has transformers for text, image, and speech inputs, in TwinVLA, the inputs for the left and right arms correspond to these.

Furthermore, MoT requires an operation to group mixed inputs by modality and then restore their original order. However, this process is unnecessary in TwinVLA because the inputs are fed in a fixed sequence: left arm, then right arm. The detailed computation process is shown in Algorithm 1.

C.2 Attention Re-weighting

Attention re-weighting is a technique we employ to improve the efficiency of adapting a pretrained SingleVLA into a bimanual TwinVLA. Constructing TwinVLA involves adding a second set of arm-specific modality tokens. During operation, input tokens are processed by their corresponding arm's VLM backbone, pass through a joint attention layer, and then flow back to the individual VLMs.

Algorithm 1 Joint Attention Computation

Let $x = (x_1, \ldots, x_n)$ be the input sequence with $x_i \in \mathbb{R}^d$; let $m_i \in \{\text{left arm}, \text{right arm}\}$ denote the modality of x_i.
Let $\mathcal{M} = \{\text{left arm}, \text{right arm}\}$.
for each modality $m \in \mathcal{M}$ **do**
  $I_m \leftarrow \{i : m_i = m\}$ ▷ Indices for modality m
  $X_m \leftarrow \{x_i : i \in I_m\}$ ▷ Group tokens by modality
  $Q_m \leftarrow W_Q^m X_m, \; K_m \leftarrow W_K^m X_m, \; V_m \leftarrow W_V^m X_m$ ▷ Modality-specific projections
end for
$Q \leftarrow \bigcup_{m \in \mathcal{M}} Q_m, \; K \leftarrow \bigcup_{m \in \mathcal{M}} K_m, \; V \leftarrow \bigcup_{m \in \mathcal{M}} V_m$ ▷ Concatenate
$A \leftarrow \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$ ▷ Global self-attention
for each modality $m \in \mathcal{M}$ **do**
  $O_m \leftarrow W_O^m A|_{I_m}$ ▷ Modality-specific output projection
  $H_m \leftarrow X_m + \mathrm{LayerNorm}^m_{\text{attn}}(O_m)$ ▷ Residual & norm
  $F_m \leftarrow \mathrm{FFN}_m(H_m)$
  $Y_m \leftarrow H_m + \mathrm{LayerNorm}^m_{\text{ffn}}(F_m)$
end for
return $\{Y_m : m \in \mathcal{M}\}$

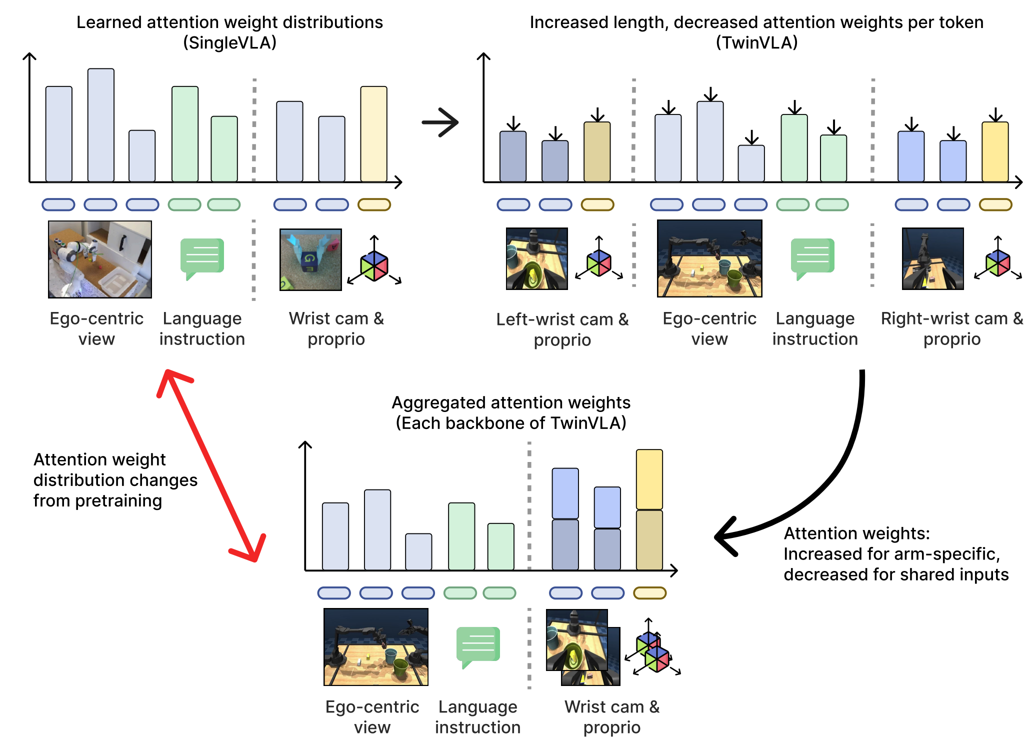


Figure 10: Due to the increased token length and softmax normalization, each VLM of TwinVLA refers to arm-specific inputs more than during pretraining, requiring the model to adapt.

However, the softmax normalization within this joint attention layer presents a challenge. Although the total sequence length doubles, the number of tokens for shared inputs remains unchanged. Consequently, the proportion of attention allocated to these shared inputs is significantly diluted compared to the pretraining phase, creating a distribution shift for each VLM backbone's inputs, as illustrated in Figure 10.

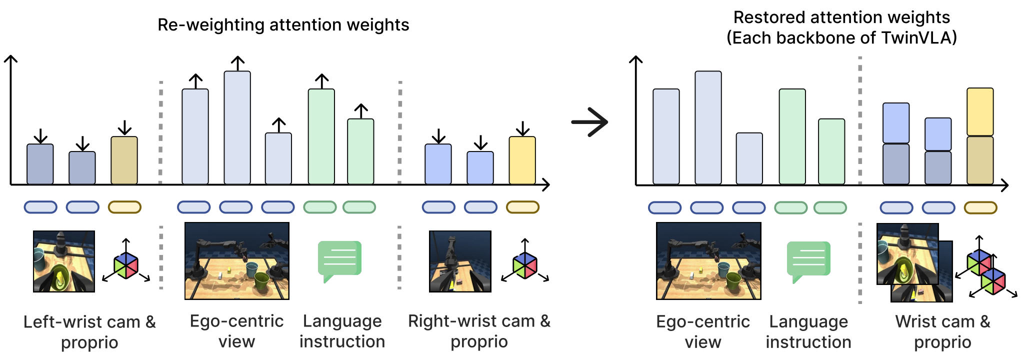


Figure 11: By re-weighting the attention weights, we can make each VLM refer to each modality identically to its pretraining stage, resulting in no adaptation and a lower initial loss.

This discrepancy requires greater adaptation effort for TwinVLA during fine-tuning on bimanual tasks. To address this, we introduce a simple re-weighting trick immediately after the attention scores are calculated. Specifically, we double the attention weights corresponding to the shared modality tokens and then re-normalize all weights to sum to one. This adjustment effectively restores the proportional attention each VLM backbone assigns to the shared inputs, aligning it with the pretraining conditions (see Figure 11). Applying this method reduced the initial fine-tuning loss by approximately 40%. While TwinVLA could learn bimanual manipulation without this technique, the required adaptation period would be substantially longer. This simple trick makes the process significantly more efficient and faster. We illustrate our implementation with simple pseudocode in Algorithm 2.

Algorithm 2 Attention Re-weighting

Input: Attention weights $\mathbf{A}$, Scale factor $\alpha = 2$
Output: Re-weighted attention weights $\mathbf{A}'$
function APPLYREWEIGHTING($\mathbf{A}$, $\mathbf{c}$, α)
  Create mask $\mathbf{M}$ ▷ Create a mask for arm-specific inputs
  $\mathbf{A}_{\mathbf{reweighted}} \leftarrow \mathbf{A} \odot (\mathbf{M} + \alpha \cdot \neg \mathbf{M})$ ▷ Apply scaling to attention weights using the mask
  $\mathbf{A}_{\mathbf{reweighted}} \leftarrow \text{Normalize}(\mathbf{A}_{\mathbf{reweighted}})$ ▷ Normalize the new weights
  return $\mathbf{A} + (\mathbf{A}_{\mathbf{reweighted}} - \mathbf{A})$ ▷ Return weights as a residual update for gradient flow
end function

C.3 MoE INTEGRATION

To enable sharing of the shared inputs between the two-arm models, we duplicated the entire VLM transformer. This necessitates different strategies for sharing the FFNs and the other components. This section details the strategy used for each component of the transformer.

Feed-Forward Networks. To share FFNs, we adopt the common approach of using a gating-based MoE. In standard MoE, multiple FFNs are included within a transformer, and a gating mechanism activates a subset for each input. In TwinVLA, the two VLMs act as distinct FFN experts. Because shared inputs (e.g., egocentric views or language prompts) may have asymmetric relevance for each arm, the gating mechanism learns how much each FFN should contribute to processing the shared input. This approach is widely used and has been shown to improve training stability and preserve information more effectively than simple averaging (Shazeer et al., 2017).

Other Components. Beyond FFNs, elements such as layer normalization and projection layers also require integration. For these, we apply task arithmetic (Tang et al., 2024), merging the two VLMs via simple parameter averaging with weight $\lambda = 0.5$. This extends MoE-style computation to the full transformer architecture.

D Real-World Robot Experiment Details

D.1 Initial Distribution

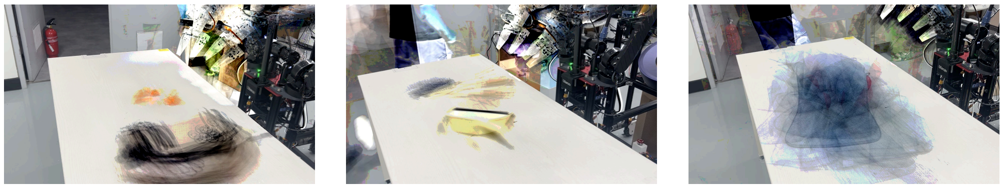


Figure 12: Initial distribution of each tasks in real-world.

To illustrate the diversity of initial configurations in our dataset, Figure 12 shows an overlay of the first frames from all 50 demonstrations. For each demonstration, the position and orientation of the objects were randomized, resulting in a unique starting setup.

D.2 Quantitative Results

Table 6: **Success rates for each model across all subtasks.** The best overall performance is highlighted in bold. As π_0 is included as a skyline, as this is excluded from this direct comparison.

Task	Subtask	DP	TwinVLA	RDT-1B	π_0
`Carrot to bag`	`Pick up carrot`	0.50	1.00	0.75	0.85
	`Put carrot`	0.20	0.70	0.40	0.65
	`Close bag`	0.15	**0.65**	0.35	**0.65**
`Brush to dustpan`	`Move the brush`	0.70	1.00	1.00	1.00
	`Pick up the brush`	0.65	1.00	1.00	1.00
	`Put onto dustpan`	0.35	**0.80**	0.40	**0.80**
`Take towel off`	`Dragging`	0.40	0.90	0.80	0.95
	`Half off`	0.35	0.70	0.70	0.85
	`Entirely off`	0.20	0.55	0.60	**0.65**

We provide the quantitative results on real-world experiments in subtask-level detail in Table 6. The results reveal the main bottleneck in each long-horizon task. The `Carrot to bag` task is challenging when inserting the carrot, which requires precisely opening the bag. The `Brush to dustpan` task's bottleneck is the high-precision insertion of the brush into the dustpan. Lastly, in `Take towel off rack`, the final unfolding is difficult—unlike the simple initial steps—as it requires a successful switch between the arms. In the next subsection, we show qualitative results from these specific bottleneck phases.

D.3 Qualitative Results

Figure 13 presents qualitative results highlighting challenging situations for each task. A check mark was used when the model succeeded with a probability above 0.5, an X mark for probabilities below 0.3, and an exclamation mark icon for intermediate cases.

- `Carrot to bag.` π_0 showed the highest success rate, followed by TwinVLA, RDT, and DP. DP failed to interact meaningfully with the bag, especially struggling to grasp the cover properly. RDT failed to complete the task successfully, primarily due to its inability to accurately localize and grasp the bag's opening.
- `Brush to dustpan.` DP struggled either to grasp the brush itself or to successfully insert it. Interestingly, the RDT managed to grasp the brush well but lacked precision during the insertion. In this task, TwinVLA and π_0 demonstrated the same success rate.

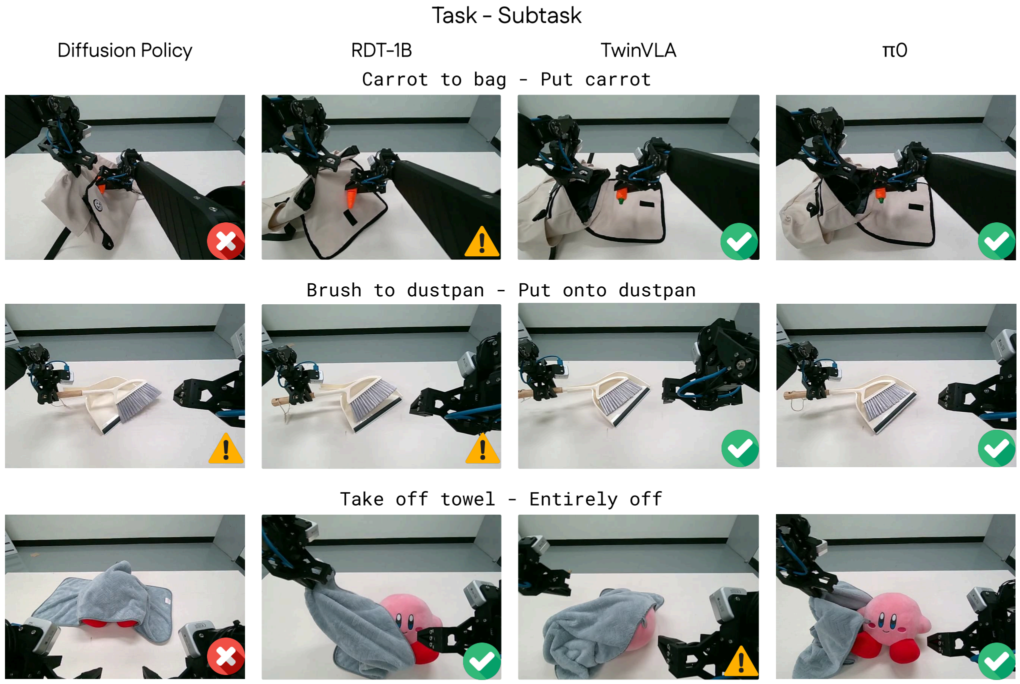


Figure 13: **Qualitative visualization of real world experiments.**

- `Take towel off.` DP mostly failed to pull the doll from a distant position toward the center, while the other models succeeded in pulling it to the center but showed differences in towel removal. Both RDT and π_0 tended to successfully remove one side of the towel and then easily remove the other side as well. In contrast, TwinVLA struggled with removing the remaining part and repeated the same action. This is likely because the longer action chunk length of RDT and π_0 helped them overcome the multimodality challenge.

D.4 Robot Hardware Spec

We conduct our real-world experiments using a custom-built robot named Anubis. The platform features a teleoperation system inspired by the Mobile ALOHA setup (Fu et al., 2024). Each arm has 6 DoF and is equipped with a parallel gripper and a wrist-mounted camera. At the center of the robot, an Intel RealSense camera is mounted on a height-adjustable mechanism, serving as the ego-centric view camera. Details are described in Table 7. Anubis is equipped with a 3-wheel omni-directional base that supports planar locomotion; however, in this work, the mobility feature is not utilized.

Table 7: **Anubis Robot Hardware Specifications.**

Component	**Specification**
Base Type	3-wheel omni-directional chassis
Mobility DOF	3 (X, Y, Yaw)
Arm DOF	2 × (6 DOF + gripper) = 14
Total Action Space	17 DOF
Wrist Cameras	Intel RealSense D405
Gripper	Parallel transparent gripper (hole design, ALOHA-style)
Power System	3 × Greenworks 40V 5.0Ah batteries (PC, wheels & leader/follower)
Frame	3D-printed custom components

Figure 14: **The Anubis robot.**

E Simulation Experiment Details

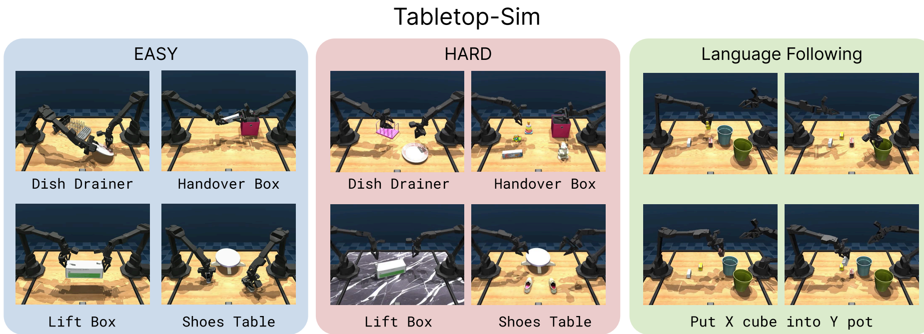


Figure 15: **Task list of Tabletop-Sim.**

E.1 Tabletop-Sim

To test bimanual policies in simulation, we developed Tabletop-Sim, a new benchmark specifically engineered to evaluate dexterous manipulation skills, in contrast to other benchmarks (Mu et al., 2025) that primarily focus on task diversity. The benchmark comprises four single-task environments and one multi-task setup. Our task selection was guided by the taxonomy in DexMimicGen (Jiang et al., 2025), which categorizes bimanual tasks into: (1) parallel (two arms are doing separate tasks simultaneously), (2) coordinated (two arms are closely working together), and (3) sequential (one arm completes the task, and the other arm takes over) interactions. Using a custom controller similar to GELLO (Wu et al., 2024), we collected 50 demonstrations for each single-task and 60 for the multi-task environment.

The multi-task setup is a language-following task requiring the policy to place a specific box (out of three) into a designated pot (out of two) based on a language instruction. This task is designed to rigorously assess a model's instruction-following capabilities, as Vision-Language-Action (VLA) models often disregard instructions after fine-tuning.

Furthermore, to evaluate policy robustness, we established two difficulty settings for the four single-tasks. The original tasks are designated as the Easy setting, while a Hard variant for each task incorporates challenging variations such as different textures, object models, and the presence of distractor objects. Figure 15 presents snapshots of each task.

To ensure reproducibility and support future research, we will fully open-source this simulation and dataset.

E.2 Quantitative Results

This section describes the detailed results for the simulation tasks. The results for **Tabletop-Sim** are listed in Table 8, while the results for the **RoboTwin 2.0** benchmark are in Table 9. For RoboTwin, the results for other baselines were referenced from the official benchmark results.

Although π_0 achieves the highest overall performance, this result is unsurprising considering its larger model size and pretraining dataset. Meanwhile, **TwinVLA** demonstrates consistently superior performance compared to **RDT-1B**, a model of a similar scale.

Table 8: **Performance comparison on the Tabletop-Sim benchmark.**

	Tabletop-Sim								
	Dish drainer		Handover box		Lift box		Shoes table		Put X cube in to Y pot
Model	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	
DP	0.686	0.590	0.180	0.086	0.100	0.006	0.028	0.260	-
RDT-1B	0.810	0.780	0.694	0.508	0.300	0.076	0.660	0.192	0.555
TwinVLA	**0.954**	**0.836**	0.780	**0.530**	0.452	0.044	**0.848**	0.306	**0.806**
PI-0	0.774	0.520	**0.788**	0.444	**0.512**	**0.136**	0.824	**0.660**	0.792

Table 9: **Success rates of TwinVLA for** 50 **bimanual tasks in RoboTwin 2.0.**

Task Name	Easy	Hard	Task Name	Easy	Hard
adjust bottle	0.97	0.35	place can basket	0.40	0.00
beat block hammer	0.77	0.10	place cans plasticbox	0.47	0.08
blocks ranking rgb	0.58	0.00	place container plate	0.77	0.04
blocks ranking size	0.03	0.00	place dual shoes	0.18	0.03
click alarmclock	0.33	0.01	place empty cup	0.50	0.01
click bell	0.58	0.13	place fan	0.34	0.00
dump bin bigbin	0.80	0.34	place mouse pad	0.50	0.00
grab roller	0.96	0.22	place object basket	0.48	0.03
handover block	0.17	0.00	place object scale	0.06	0.00
handover mic	0.84	0.02	place object stand	0.20	0.02
hanging mug	0.10	0.05	place phone stand	0.34	0.02
lift pot	0.87	0.07	place shoe	0.48	0.04
move can pot	0.45	0.05	press stapler	0.62	0.26
move pillbottle pad	0.32	0.02	put bottles dustbin	0.08	0.04
move playingcard away	0.61	0.35	put object cabinet	0.39	0.16
move stapler pad	0.11	0.00	rotate qrcode	0.54	0.03
open laptop	0.80	0.17	scan object	0.11	0.04
open microwave	0.03	0.01	shake bottle horizontally	0.96	0.55
pick diverse bottles	0.16	0.08	shake bottle	0.93	0.58
pick dual bottles	0.18	0.12	stack blocks three	0.00	0.00
place a2b left	0.27	0.05	stack blocks two	0.26	0.00
place a2b right	0.15	0.01	stack bowls three	0.77	0.15
place bread basket	0.11	0.03	stack bowls two	0.84	0.11
place bread skillet	0.20	0.01	stamp seal	0.16	0.01
place burger fries	0.67	0.13	turn switch	0.25	0.15
Average					
Diffusion Policy	0.280	0.006			
RDT-1B	0.345	0.137			
TwinVLA	0.420	0.089			
π_0	0.464	0.163			