

ManualVLA: A Unified VLA Model for Chain-of-Thought Manual Generation and Robotic Manipulation

Chenyang Gu^{*1}, Jiaming Liu^{*†1}, Hao Chen^{*†2}, Runzhong Huang^{*1}, Qingpo Wu¹, Zhuoyang Liu¹, Xiaoqi Li¹, Ying Li¹, Renrui Zhang², Peng Jia³, Pheng-Ann Heng², Shanghang Zhang^{✉1}

¹State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University ²The Chinese University of Hong Kong ³Simplexity Robotics

Project web page: <https://sites.google.com/view/maunallvla>.

Abstract

Vision–Language–Action (VLA) models have recently emerged, demonstrating strong generalization in robotic scene understanding and manipulation. However, when confronted with long-horizon tasks that require defined goal states, such as LEGO assembly or object rearrangement, existing VLA models still face challenges in coordinating high-level planning with precise manipulation. Therefore, we aim to endow a VLA model with the capability to infer the “how” process from the “what” outcomes, transforming goal states into executable procedures. In this paper, we introduce ManualVLA, a unified VLA framework built upon a Mixture-of-Transformers (MoT) architecture, enabling coherent collaboration between multimodal manual generation and action execution. Unlike prior VLA models that directly map sensory inputs to actions, we first equip ManualVLA with a planning expert that generates intermediate manuals consisting of images, position prompts, and textual instructions. Building upon these multimodal manuals, we design a Manual Chain-of-Thought (ManualCoT) reasoning process that feeds them into the action expert, where each manual step provides explicit control conditions, while its latent representation offers implicit guidance for accurate manipulation. To alleviate the burden of data collection, we develop a high-fidelity digital-twin toolkit based on 3D Gaussian Splatting, which automatically generates manual data for planning expert training. ManualVLA demonstrates strong real-world performance, achieving an average success rate 32% higher than the previous hierarchical SOTA baseline on LEGO assembly and object rearrangement tasks.

1. Introduction

Recently, building on internet-scale pretrained vision-language models [1, 34], vision-language-action (VLA) models have emerged [9, 37] and are trained on robot demonstrations to predict control actions. These models exhibit impressive capabilities in robotic scene understand-

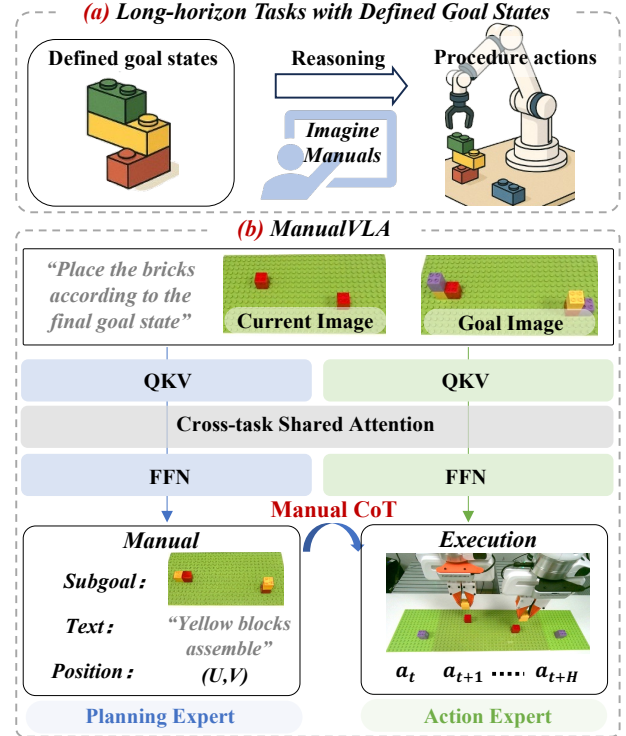


Figure 1. **Overview.** (a) Long-horizon tasks with predefined goal states, such as LEGO assembly or object rearrangement, pose a significant challenge for intelligent robots, as they require not only imagining procedural manuals but also executing precise manipulations based on them. (b) We address such tasks by introducing ManualVLA, a unified VLA model built upon a MoT architecture, which enables coherent collaboration between multimodal manual and action generation via a designed Manual Chain-of-Thought.

ing [46, 48, 89] while demonstrating strong generalization and manipulation performance in out-of-lab scenarios [5, 28, 49]. Such advances have significantly accelerated progress toward developing generalist robotic agents. However, when confronted with long-horizon tasks that require precisely defined goal states (Figure 1 (a)), such as LEGO assembly or object rearrangement, VLA models remain highly challenged. The difficulty arises from two as-

^{*}Equal contribution [†]Project leader [✉]Corresponding author.

pects: (1) the VLA model must execute precise manipulation strictly aligned with the predefined final scene or object configuration, and (2) it must integrate long-horizon planning with fine-grained control, while maintaining generalization to diverse real-world environments.

In contrast, for humans, upon accumulating sufficient manipulation experience, they can perform long-horizon, goal-specific tasks independently, without relying on further demonstrations. For example, humans can assemble LEGO structures by inferring the target configuration, or reorganize household objects according to a predefined spatial arrangement. This ability stems from humans’ intuitive inference of intermediate cues, reasoning over spatial and causal relations, and decomposition of tasks into coherent, subgoal-directed manipulation steps. Recently, some hierarchical methods have attempted to emulate this ability by relying on detailed manuals [51] or human demonstration videos [2, 29, 64, 80]. However, such approaches are often limited in generalization to unseen final goal states and increase the dependence on human involvement. This naturally leads to a question: *“Is it possible to endow a VLA model with the capability to infer the procedural “how” from the desired “what,” thereby transforming a predefined final goal into a sequence of coherent and precise execution steps?”*

To this end, we propose **ManualVLA**, a unified VLA model based on a Mixture-of-Transformers (MoT) [45] architecture, capable of generating multimodal manuals and actions directly from a final goal state. Since only the final state is provided and intermediate steps are unknown, previous VLA models that map sensory inputs directly to actions struggle with such long-horizon tasks. In contrast, as shown in Figure 1 (b), ManualVLA selectively activates planning and action experts within a unified framework for subgoal manual and action generation. Specifically, ManualVLA is equipped with a planning expert to generate intermediate manuals that integrate images, position prompts, and textual instructions. These manuals are then used in our proposed **Manual Chain-of-Thought (ManualCoT)** reasoning strategy, which guides the action expert by treating each subgoal step as an explicit condition for precise execution. Furthermore, a cross-task shared attention mechanism between the two experts enables long-context interactions between manual-generation features and action generation, providing implicit guidance for coherent manipulation.

To train ManualVLA, we first leverage the generative capability of VLM model (e.g., Janus-Pro [84]) and fine-tune it on self-collected and simulation-synthesized manual data to acquire the planning expert capability. Due to the substantial uncertainty in the final goal states, a large amount of data is required to train the model with sufficient world knowledge for effective task planning. To alleviate the burden of data collection, we develop a high-fidelity digital-twin toolkit based on 3D Gaussian Splatting [35], which automat-

ically generates manual data for planning expert training. Meanwhile, we pretrain the action expert on large-scale open-source robotic datasets comprising over 400K trajectories [36, 63]. Benefiting from the rich manual conditions incorporated during action generation, ManualVLA requires only around 100 trajectories to achieve generalizable manipulation while finetuning on downstream tasks. Experimental results demonstrate that, when confronted with long-horizon and complex LEGO assembly and object rearrangement tasks, ManualVLA not only generates accurate manuals but also achieves an average manipulation success rate of 32% higher than existing hierarchical SOTA baseline. Note that ManualVLA also achieves state-of-the-art (SOTA) performance on other general manipulation tasks. In summary, our contributions are as follows:

- We address long-horizon tasks with precisely defined goal states by introducing ManualVLA, a unified VLA model built upon a MoT architecture that supports coherent multimodal manual generation and action execution.
- We design a Manual Chain-of-Thought (ManualCoT) reasoning process that translates generated manuals into precise actions, where each manual step provides explicit control conditions and its latent representation offers implicit guidance for manipulation.
- Equipped with the proposed training strategy, ManualVLA demonstrates strong real-world performance, achieving superior manipulation accuracy and generalization compared with previous SOTA baselines on downstream tasks.

2. Related Work

Vision-language models (VLMs) [1, 34, 69] have achieved strong multimodal reasoning by learning from internet-scale image-text data. Building on this progress, VLA models [9, 37] have emerged as a promising approach for robot learning, enabling end-to-end mapping from multimodal observations to control signals. Subsequent works have further advanced VLA models by incorporating richer sensory understanding [40, 43, 50, 67], exploring more robust action generation [7, 28, 41, 49, 82], and developing optimized inference strategies [48, 65, 83] as well as dual-system paradigms [5, 12, 21, 93]. Notably, recent approaches like MoTVLA [27] and F1-VLA [56] have introduced MoT architectures into general robotic manipulation. However, when faced with long-horizon tasks that require precise goal specifications, these models continue to struggle. To reconcile this issue, several works [26, 44, 90, 98] have incorporated visual world modeling into VLA architectures to enable reasoning about the future images. They typically decouple long-horizon progress into intermediate steps either by generating explicit pixel-level subgoals [62, 85] or by formulating compressed token representations [10, 22, 42, 95]. Nevertheless, these models struggle to capture the relationship between subgoals and fine-grained control. In contrast,

we introduce a comprehensive Chain-of-Thought (CoT) reasoning process that combines both explicit and implicit cues to transform the generated manuals into precise actions.

Final goal conditioned manipulation, where robots must reach specified target states from given initial configurations, such as LEGO assembly or object rearrangement, represents a challenge in embodied AI. A prominent research direction uses human hand videos [2, 29, 75] to present the desired intermediate procedure. Methods such as Vid2Robot [29] and DexCap [80] extract manipulation trajectories from egocentric videos, transferring human dexterity to robotic control. Meanwhile, several works [66, 81, 94] exploit operation manuals or goal-state descriptions as guidance. CheckManual [51] conditions robot policies on predefined instruction manuals, while [87, 91] utilize the final target scene configuration to inform execution goal. However, providing hand videos, human-crafted manuals, or additional reasoning models introduces extra human effort and computational cost, limiting the practicality of these approaches. In contrast, we make the first attempt to address long-horizon, goal-conditioned manipulation through a unified VLA model that enables coherent collaboration between multimodal manual generation and action execution.

3. ManualVLA

In this section, we first introduce the fundamentals of Vision-Language-Action (VLA) models in Section 3.1. Then, Section 3.2 presents the architectural details of ManualVLA, followed by Section 3.3, which elaborates on its working principles. Sections 3.4 and 3.5 describe the training strategies of ManualVLA and the digital-twin toolkit, respectively.

3.1. Preliminary

VLA models integrate visual, linguistic, and proprioceptive inputs to generate robot control signals, exhibiting strong generalization in diverse manipulation tasks [7, 37]. Despite their impressive abilities, existing VLA models often struggle with long-horizon tasks with defined goal states, reflecting their limited world knowledge for planning the intermediate progress required to achieve such goals. To address this limitation, we aim to endow the VLA model with a human-like capability to infer the procedural “how” from the desired “what”. Therefore, we propose ManualVLA π_θ , a unified VLA model equipped with world knowledge that first reasons about multimodal manuals describing the task procedure, and subsequently generates the corresponding actions for execution. As shown in Figure. 2, given the language instruction l together with the images of the current state $\mathcal{I}_t^{\text{current}}$ and the final goal state $\mathcal{I}^{\text{goal}}$, ManualVLA first generates a manual that consists of the textual description of the target objects \hat{l}_t , their target 2D coordinates p_t , and the corresponding subgoal image $\mathcal{I}_t^{\text{subgoal}}$:

$$\pi_\theta(\mathcal{I}_t^{\text{subgoal}}, p_t, \hat{l}_t \mid \mathcal{I}^{\text{goal}}, \mathcal{I}_t^{\text{current}}, l). \quad (1)$$

Based on this manual, we construct a prompted image $\mathcal{I}_t^{\text{prompt}}$ by overlaying the target object’s final position as a mask on the current scene image $\mathcal{I}_t^{\text{current}}$. Finally, the model takes the robot state s_t and the prompted image $\mathcal{I}_t^{\text{prompt}}$, together with $\mathcal{F}_t^{\text{subgoal}}$, \mathcal{F}_t^p , and $\mathcal{F}_t^{\hat{l}}$ (the key and value features stored during the generation of \hat{l}_t , p_t , and $\mathcal{I}_t^{\text{subgoal}}$), as conditional inputs for modeling the action chunk $a_{t:t+h}$, enabling explicit subgoal-guided action generation.

$$\pi_\theta(a_{t:t+h} \mid s_t, \mathcal{I}_t^{\text{prompt}}, \mathcal{F}_t^{\text{subgoal}}, \mathcal{F}_t^p, \mathcal{F}_t^{\hat{l}}). \quad (2)$$

3.2. Model Architecture

ManualVLA adopts Janus-Pro [14] as its foundation model due to its strong capability in general multimodal understanding and generation. Unlike atomic or short-horizon manipulation tasks, our goal is to perform long-horizon, goal-specific tasks, where the model must not only generate detailed manuals but also predict precise actions and enable effective interaction between the two processes. To achieve this, we extend the basic VLM into a Mixture-of-Transformers (MoT) architecture, forming a unified VLA model that integrates collaborative planning and action experts. We next detail the key components of our ManualVLA.

Vision Tokenizer and Encoder. As shown in Figure 2, given the distinct characteristics of discrete and continuous image-injection paradigms, as well as the differing demands of manual and action generation, ManualVLA employs two separate visual modules: a VQ-based vision tokenizer for manual generation and a continuous vision encoder for action generation. For the vision tokenizer, ManualVLA adopts an encoder-quantizer-decoder architecture following VQ-GAN [20]. The encoder and decoder are convolutional networks with a downsampling factor of 16, and the quantizer maintains a codebook $\mathbf{Z} \in \mathbb{R}^{16,384 \times 8}$. For the vision encoder, ManualVLA uses SigLIP-Large [92] with an input resolution of 384 to extract high-dimensional semantic features from input images.

Mixture-of-Transformers LLM. The base language model in ManualVLA is DeepSeek-LLM 1.5B [4]. To integrate the distinct capabilities of manual generation and action generation, we construct a MoT architecture atop this LLM. The proposed MoT extends the standard Transformer by introducing task-specific parameter sets for all non-embedding components, including feed-forward networks (FFN), attention projections, and layer normalizations, yielding the planning and action experts illustrated in Figure 2 (a). When faced with complex and long-horizon tasks, this design enables the VLA model to efficiently handle heterogeneous tasks while preserving its ability to learn cross-task dependencies within a unified framework.

To formally describe this mechanism, we take a single MoT layer as an example. Let $x = (x_1, \dots, x_n)$ denote the input token sequence, where each token x_i is assigned

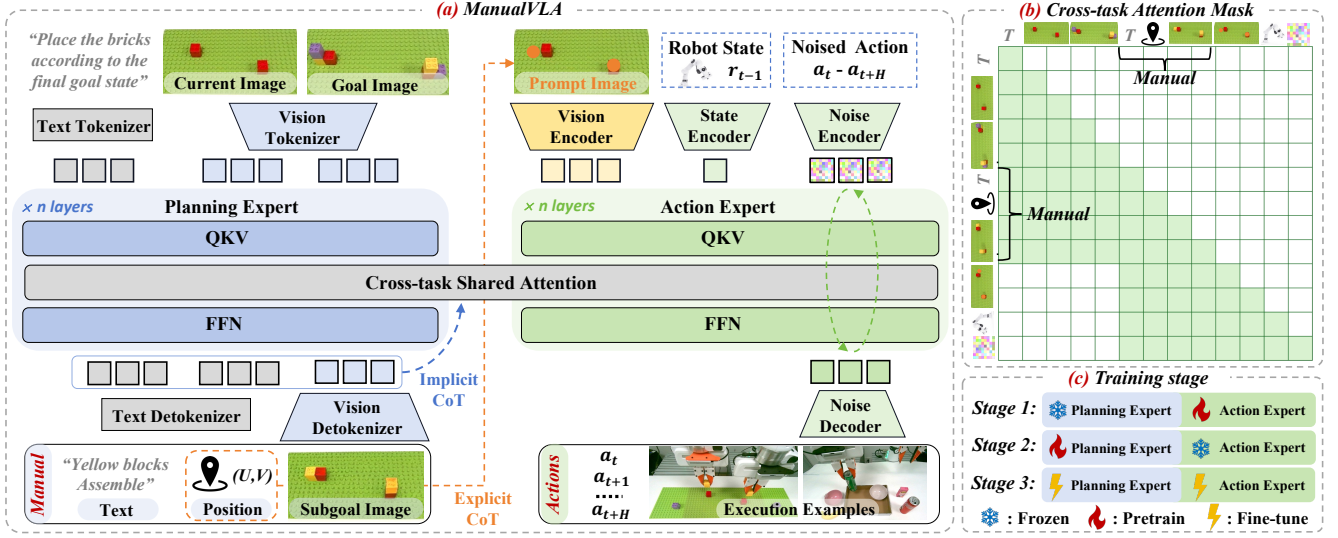


Figure 2. **Framework of ManualVLA.** (a) To accomplish long-horizon tasks with defined goal states, we propose ManualVLA, a unified VLA model built upon a MoT architecture. The framework consists of two experts: a planning expert responsible for generating multimodal manuals, and an action expert responsible for predicting precise actions. The planning expert processes human instructions, the current image, and the final goal image to generate intermediate manuals that combine next-step image, positions, and sub-task instructions. We introduce an explicit CoT reasoning process, where each positional indicator serves as a visual prompt embedded into the observation of the action expert. (b) Along with the cross-task shared attention mechanism and the designed attention mask, the generated manual tokens are also used as conditioning signals for action generation, enabling an implicit CoT reasoning process that effectively guides the action expert. (c) ManualVLA adopts a three-stage training strategy that aligns the planning and action experts for effective collaboration.

to exactly one task category $t_i \in \mathcal{T} = \{\text{manual}, \text{action}\}$. For each task, we define task-dependent operator bundles $\Theta^t = \{\theta_{\text{attn}}^t, \theta_{\text{ffn}}^t\}$ and corresponding mappings $\Phi_{\text{attn}}^t, \Phi_{\text{ffn}}^t : \mathbb{R}^d \rightarrow \mathbb{R}^d$, which are applied token-wise depending on the associated task. Then, a MoT layer acting on a mixed-task sequence can be compactly expressed as:

$$\text{MoT}_{\Theta}(x) = x + \mathcal{N}_{\text{ffn}}^{t(\cdot)} \left(\Phi_{\text{ffn}}^{t(\cdot)} \left(x + \mathcal{N}_{\text{attn}}^{t(\cdot)} \left(\Phi_{\text{attn}}^{t(\cdot)}(x) \right) \right) \right), \quad (3)$$

where the notation $t(\cdot)$ indicates that each token at position i uses its corresponding task parameters and $\mathcal{N}_{\text{attn}}^t, \mathcal{N}_{\text{ffn}}^t$ denote task-specific layer-normalization operators.

Finally, to define the global attention operator [18], let $X \in \mathbb{R}^{n \times d}$ denote the matrix form of the input sequence $x = (x_1, \dots, x_n)$, where each row corresponds to a token embedding. The operator is then given by

$$Q = XW_Q^{t(\cdot)}, \quad K = XW_K^{t(\cdot)}, \quad V = XW_V^{t(\cdot)}, \quad (4)$$

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \quad \Phi_{\text{attn}}(x) = (AV)W_O^{t(\cdot)},$$

where the projections $W_Q^{t(\cdot)}, W_K^{t(\cdot)}, W_V^{t(\cdot)}, W_O^{t(\cdot)}$ are selected per token according to its task t_i , while the attention weight matrix A is computed globally across all tokens. This formulation enables ManualVLA to adaptively allocate computation according to the distinct characteristics of manual and action generation, while maintaining a unified architec-

ture for the collaborative execution of final goal conditioned manipulation tasks.

Action and Robot State Components. ManualVLA employs a diffusion-based approach for action modeling, therefore, a noise encoder is introduced to inject the noised actions into the action expert, and a noise decoder predicts the noise from the latent representations. Both modules are implemented as two-layer MLPs. In addition, the robot state is incorporated into the action expert through another two-layer MLP (state encoder), enabling the model to condition action generation on the current proprioceptive state.

3.3. Manual and Action Generation via CoT

Given the final goal state of a task, ManualVLA generates subgoal manuals at key steps. To better produce corresponding action sequences based on the current observation and the generated manuals, we introduce a Manual Chain-of-Thought (ManualCoT) reasoning process including both explicit and implicit CoT. Specifically, we introduce the details of manual generation in Section 3.3.1, followed by the method for generating executable actions through ManualCoT in Sections 3.3.2. Finally, in Sections 3.3.3, we describe how both processes are unified within a single token sequence for the end-to-end training.

3.3.1. Subgoal Manual Generation

To accomplish long-horizon manipulation tasks with pre-defined goal states, we design a multimodal manual that

consists of (1) textual descriptions for subtask reasoning, (2) next-step images providing semantically rich conditioning, and (3) low-level position prompts for precise manipulation guidance. As shown in Figure 2 (a), upon receiving the language instruction along with the current and final state images, ManualVLA first generates a textual component that describes object attributes and actions. For target positions, we represent them using the pixel-level (U, V) coordinates of each object’s centroid. Finally, subgoal image generation helps the model better model the physical world dynamics.

We assume that accomplishing a long-horizon task does not require extreme dense temporal subgoals. Instead, providing guidance only at key frames where the task state changes is sufficient, such as when placing a pair of bricks onto the board. ManualVLA generates a new manual only after the completion of the previous subgoal, ensuring efficient planning without redundant intermediate guidance. To achieve this, we first generate the text description in the manual. If the generated description of the manipulated objects differs from the previous planning output, ManualVLA proceeds to produce an entirely new manual. For example, the text output changes from “yellow blocks” to “purple blocks”. Otherwise, it reuses the previously generated manual for subsequent action generation.

3.3.2. Manual-Conditioned Action Generation

Based on the generated subgoal manual, ManualVLA executes actions in a closed-loop manner, progressively generating the action sequence until the subgoal is achieved. As shown in Figure 2 (a), leveraging the effectiveness of visual prompts for manipulation, we use the predicted (U, V) coordinates to overlay a mask on the current image, highlighting the affordance region that serves as input to the action expert. This construction of a prompt image to guide the action learning is defined as **explicit CoT reasoning**. Meanwhile, we introduce an implicit CoT reasoning process within the shared attention module. As shown in Figure 2 (b), in the latent space, the subgoal manual serves as a conditioning signal for action modeling through our constructed cross-task attention mask. This conditioning information first informs the model about “what” object to manipulate, then specifies “where” the object should be placed, and finally provides the anticipated visual outcome after the manipulation. This CoT reasoning process performed in the latent space is referred to as **implicit CoT reasoning**. By incorporating both explicit and implicit CoT reasoning processes, ManualVLA significantly improves its success rate in long-horizon tasks, as demonstrated in Section 4.3.

3.3.3. Token Sequence Design

After introducing the processes of manual and action generation, this section describes how ManualVLA jointly learns both tasks within a unified token sequence, while employing

planning and action experts to specialize in different tasks. As shown in Figure 2, the language instruction, along with the current and goal scene images, is first inserted into the token sequence as the condition for subgoal manual generation. The subsequent tokens represent the generated manual, including object descriptions, target coordinates, and subgoal image, all of which are processed by the planning expert. Following this, the sequence incorporates the prompt image used in explicit CoT reasoning, as well as the robot state and noised action embeddings, which are handled by the action expert. As shown in Figure 2 (b), a cross-task shared attention mechanism is designed to allow the action expert to attend to the subgoal manual representations while masking out earlier inputs, thereby enabling effective information exchange between the two experts and fostering coherent reasoning across planning and action generation.

3.4. Training Strategy

As shown in Figure 2 (c), we train ManualVLA in three stages. Before training, we initialize ManualVLA with the pretrained parameters of Janus-Pro [14], and duplicate its LLM to separately initialize the planning and action experts.

Stage 1: Action expert pretraining. During this pretraining stage, we curated an assembly dataset by carefully filtering large-scale cross-embodiment datasets [36, 63, 86] to update all parameters of the action expert. As detailed in the Appendix A, the resulting dataset comprises over 400K trajectory samples. ManualVLA was trained on this dataset for five epochs, where the only conditioning inputs are the language instruction, a current scene image, and robot state. Following diffusion policy [15], the training objective is the mean squared error (MSE) between the predicted noises $\hat{\epsilon}^i$ at the i -th denoising steps and the ground-truth noises ϵ , defined as: $\mathcal{L}_{\text{action}} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), i} \|\hat{\epsilon}^i - \epsilon\|_2^2$. **Stage 2: Manual expert pretraining.** At this stage, we train only the manual expert using data synthesized with our digital-twin toolkit, resulting in a dataset of over 10K frames for each task. Following Janus-Pro [14], the model is supervised with a cross-entropy loss $\mathcal{L}_{\text{manual}}$ applied to the subgoal manual, which includes the object description, target position, and subgoal image tokens.

Stage 3: Joint manual-action fine-tuning. Benefiting from the stable generation capabilities acquired during pretraining, we collect 100 demonstrations for each downstream task using master-puppet teleoperation. Objects are placed at diverse locations on the table to ensure sufficient variation. Each demonstration includes both action execution data and automatically extracted manual data. All components of ManualVLA are then jointly trained using the token sequences defined in Section 3.3.1. The final objective is defined as: $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{manual}} + \mathcal{L}_{\text{action}}$.

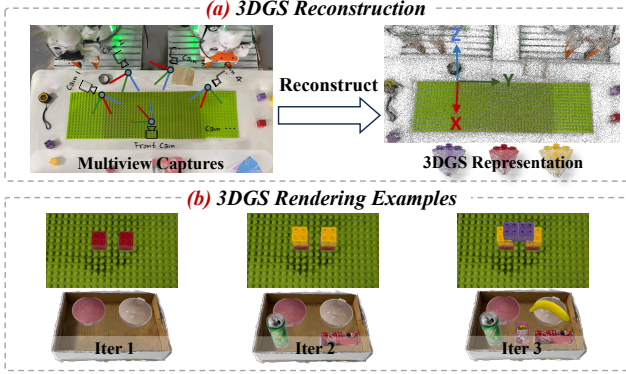


Figure 3. **Digital-twin example.** (a) We reconstruct 3D Gaussian Splatting representations, which are then decomposed into the LEGO board and individual bricks. (b) We iteratively place the bricks on the board or objects on the box.

3.5. Real-to-Render Data Generation

To construct downstream task data, we develop a high-fidelity digital-twin toolkit based on 3D Gaussian Splatting [35] to automatically generate training data with intermediate target states. For example, in Figure 3 (a), we first reconstruct 3D assets of the LEGO board and individual bricks through multi-view capture, and then align them to a unified Cartesian coordinate system for consistent spatial referencing. Subsequently, we follow an iterative placement procedure as shown in Figure 3 (b), where given an initial state and a set of available bricks, we sequentially place each brick by randomly sampling a valid position on the board. At each intermediate state, we render the current configuration from a front-view camera perspective. This process produces photorealistic images for each assembly or rearrangement step, along with position and textual information. More details are provided in Appendix A.

4. Experiments

We organize our experiments as follows. Section 4.1 defines the tasks and introduces the baseline methods. Section 4.2 compares ManualVLA with these baselines on manual generation and manipulation performance. In Section 4.3, we assess the effectiveness of our MoT architecture and CoT reasoning mechanism. Finally, Section 4.4 evaluates the generalization of ManualVLA to unseen object shapes, backgrounds, and lighting conditions. All the above experiments are performed on a dual-arm Franka robotic platform. Additionally, we validate the advantages of our method on general manipulation tasks on RL Bench [30] Benchmark in Section 4.5.

4.1. Experimental Setup

4.1.1. Task Definition

We design three long-horizon tasks with defined goal states, challenging the model’s procedural reasoning and manipula-

tion capabilities. **(1) 2D LEGO Assembly:** The task begins with several LEGO bricks of different colors placed on a planar board. Given the final 2D assembled structure as the goal, the model must infer a sequence of intermediate manipulation actions and execute them through coordinated bimanual control. **(2) 3D LEGO Assembly:** The task extends the 2D LEGO Assembly task to a more challenging 3D setting, where the final configuration transitions from a planar layout to a 3D structure. This upgraded task imposes greater demands on the model’s spatial reasoning abilities. **(3) Object Rearrangement:** The task begins with several objects of diverse shapes, sizes, and semantics scattered around a box. Given a goal state in which all objects are placed at their designated positions inside the box, the model must progressively generate manipulation actions, alternating control of the left and right arms to prevent collisions.

4.1.2. Baselines

We compare ManualVLA against three categories of strong VLA baselines that represent the state of the art in robotic manipulation. **(1) First category:** π_0 [7], $\pi_{0.5}$ [28], and FAST [65], which adopt robust action-generation paradigms. We load the official pretrained weights provided by each method and strictly follow their fine-tuning protocols, except that we additionally embed the final goal image into the model as the target condition. **(2) Second category:** CoT-VLA [96], which not only incorporates the final goal image as an additional condition, but also predicts key subgoal future images. For fair comparison, we align the supervision of subgoal image generation with that used in ManualVLA. **(3) Third category:** We introduce a hierarchical baseline that combines a VLM [14] with $\pi_{0.5}$. The VLM is trained to generate visual and language prompts similar to our method, while $\pi_{0.5}$ is trained to interpret these prompts and generate actions. This baseline can be regarded as a hierarchical variant of our manual-generation approach.

4.2. Main Results

4.2.1. Manual Generation

Table 1. Quantitative results of ManualVLA in generating subgoal images and (U, V) coordinates across the three downstream tasks.

Task	Subgoal Image		(U, V)
	PSNR \uparrow	FID \downarrow	MAE \downarrow
2D LEGO Assembly	29.01	36.39	3.23
3D LEGO Assembly	28.68	34.63	3.58
Object Rearrangement	28.11	24.46	6.21

We first evaluate the capability of the planning expert in ManualVLA to generate high-fidelity manuals on 300 unseen test samples. As shown in Table 1, our model produces satisfactory intermediate images across all three tasks, achieving high PSNR scores, indicating strong structural and

Table 2. **Comparison of ManualVLA and baselines.** We report the manipulation success rate (S.R.) for the complete long-horizon tasks using 20 unseen test goal states, and additionally report the success rate of key intermediate steps.

Method	2D LEGO Assembly				3D LEGO Assembly				Object Rearrangement			
	2 bricks →	2 bricks →	2 bricks →	S.R.	2 bricks →	2 bricks →	2 bricks →	S.R.	2 objects →	2 objects →	2 objects →	S.R.
π_0 [7]	0.25	0.20	0.15	0.15	0.25	0.15	0.10	0.10	0.35	0.20	0.10	0.10
$\pi_{0.5}$ [28]	0.30	0.25	0.25	0.20	0.30	0.20	0.15	0.15	0.45	0.25	0.15	0.15
FAST [65]	0.20	0.15	0.10	0.10	0.15	0.15	0.05	0.05	0.20	0.15	0.05	0.05
CoT-VLA [96]	0.40	0.35	0.30	0.30	0.35	0.30	0.25	0.25	0.60	0.40	0.30	0.30
VLM + $\pi_{0.5}$	0.75	0.70	0.65	0.60	0.65	0.45	0.35	0.35	0.90	0.55	0.65	0.50
ManualVLA	0.95	0.90	0.85	0.85	0.90	0.75	0.65	0.65	0.90	0.70	0.80	0.65

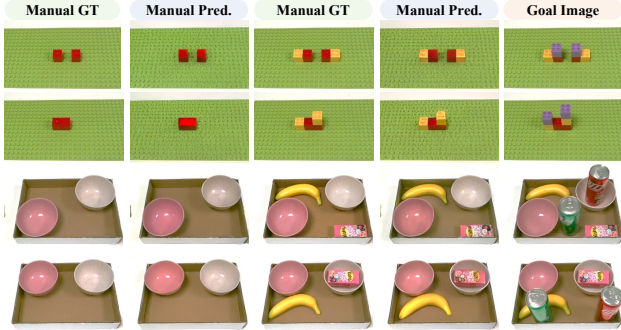


Figure 4. For each task, we visualize three components: (1) manual ground truth (GT), (2) manual predictions (Pred.) generated by ManualVLA, and (3) the final goal image.

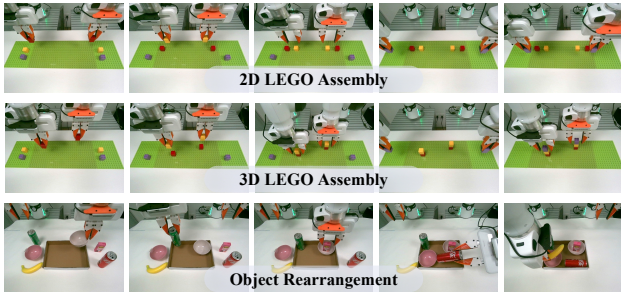


Figure 5. Visualization of real-world experiments on Franka Research 3 dual-arm robots, executed from left to right.

pixel-level consistency with the ground truth. Furthermore, the low FID scores, particularly in the Object Rearrangement task, demonstrate that the generated image distribution closely matches that of real images, confirming their realism and fidelity. The remarkably low MAE scores further highlight ManualVLA’s precision in predicting the position of target objects. For language descriptions, we evaluate accuracy by checking the predicted object nouns, all of which are correctly generated on unseen test samples.

Figure 4 presents qualitative results of ManualVLA on 2D and 3D LEGO assembly as well as object rearrangement tasks. The model accurately predicts intermediate steps that align with the ground truth, effectively capturing both spatial arrangements and object identities. In LEGO assembly, ManualVLA sequentially reconstructs the correct brick

placements and colors, demonstrating precise step-wise reasoning. For object rearrangement, it gradually progresses toward the final goal configuration and accurately generates the spatial relationships between objects. Overall, these results highlight ManualVLA’s strong intermediate reasoning capabilities in long-horizon tasks, establishing a reliable foundation for the action expert to generate accurate actions.

4.2.2. Action Generation

Across all three real-world long-horizon tasks, ManualVLA achieves the highest success rates, markedly outperforming all baselines. As shown in Table 2, we report both step-wise subgoal accuracy and end-to-end task success. Compared with the strongest hierarchical baseline, ManualVLA improves the final task completion rate by 15%-30%. While baseline models often succeed in the early stages of a long-horizon pipeline, they typically fail to sustain this performance through the whole sequence. In contrast, ManualVLA mitigates this degradation by decomposing complex tasks into structured subgoal manuals and grounding them into precise actions through a combination of explicit and implicit reasoning, enabling consistent performance throughout the entire task. Note that, as reflected in the manual generation results, the generated manuals may contain minor inaccuracies. Nevertheless, with the ManualCoT strategy and the capacity of our MoT architecture, ManualVLA remains robust and can still produce reliable actions even under moderate manual errors. In Appendix B, we also validate the advantages of our method on general manipulation tasks.

The qualitative rollouts in Figure 5 further corroborate these results. ManualVLA generates structured and interpretable intermediate states that reliably guide the dual-arm system through precise grasping and relocation motions. In both LEGO assembly tasks, the robot maintains accurate brick alignment across all stages, while in the object-rearrangement task, it robustly manipulates objects with varying shapes, textures, and occlusions. More visualizations, failure case analyses, and execution videos are provided in Appendix C, Appendix D, and the supplementary material.

4.3. Ablation Study

We conduct detailed ablation studies on the 2D LEGO Assembly task, reporting the long-horizon task success rate. (a)

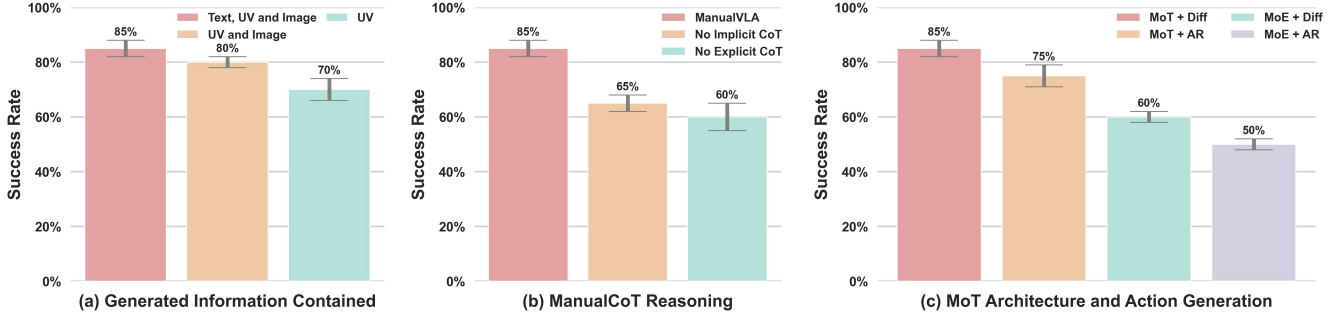


Figure 6. **Ablation study.** We investigate the impact of (a) the information contained in the generated manuals, (b) explicit and implicit CoT reasoning, (c) the MoT architecture design, and (d) the action generation paradigm on long-horizon manipulation success rates.

What information should a generated manual contain?

We explore three variants: generating only the target positions (U, V), generating both target positions and subgoal images, and generating the full set of multimodal manual. As shown in Figure 6 (a), increasing the amount of multimodal information in the manuals leads to improved manipulation performance. Note that in this experiment, we consistently use explicit CoT visual-prompt images as inputs to the action expert. The results demonstrate that high-level textual descriptions for subtask understanding, next-step images for semantically reasoning, and position prompts for accurate localization all serve as critical implicit conditions that enable precise manipulation. **(b) Importance of Explicit CoT and Implicit CoT.** We examine two variants of our ManualCoT reasoning: (1) No Explicit CoT, where the action expert receives only the latent features from the planning expert together with the current image; and (2) No Implicit CoT, where the action expert receives only the visual-prompt image. As shown in Figure 6 (b), both variants lead to a noticeable performance degradation compared to ManualVLA, demonstrating that explicit and implicit CoT reasoning are jointly indispensable for solving long-horizon, goal-defined manipulation tasks. **(c) The MOT Architecture and Action Generation Design.** In Figure 6 (c), we compare our MoT architecture with a standard Mixture-of-Experts (MoE) architecture (duplicate only FFNs in LLM) [18]. The results show that using an MoE strategy fails to produce high-quality manuals and actions simultaneously, both of which are crucial for long-horizon tasks. Meanwhile, we find that for precise manipulation tasks, diffusion-based action generation yields superior performance. More ablations on how manual quality influences manipulation performance are provided in Appendix B.

4.4. Generalization Analysis

First, in Section 4.2, all final goal states used for evaluation differ from those in the teleoperated manipulation training set. Therefore, our main results simultaneously validate the generalization capability of ManualVLA with respect to both final goal states and object positions. In addition,

Table 3. **Generalization.** We report the mean success rate and performance degradation ratio for each task over 20 rollouts under variations in background, object shape, and lighting.

	Background	Shape	Lighting	
2D LEGO	Origin	Background	Shape	Lighting
VLM + $\pi_{0.5}$	0.60	0.45(-25%)	0.35(-46%)	0.50(-17%)
ManualVLA	0.85	0.65(-23%)	0.60(-29%)	0.70(-17%)

we evaluate the robustness of ManualVLA under variations in background, object shape, and lighting. As shown in Table 3, these unseen perturbations are introduced in the 2D LEGO Assembly task and differ from all configurations seen during training. ManualVLA exhibits only a modest performance drop, which can be attributed to the rich guidance provided by our proposed manual generation expert and the CoT reasoning strategy during action prediction. Moreover, our proposed digital-twin toolkit provides large-scale manual generation data, allowing the model to produce accurate manuals even in unseen scenarios.

4.5. Simulation Experiment

We evaluate ManualVLA on the RL Bench [30] benchmark, comparing it against state-of-the-art (SOTA) VLA baselines. The results demonstrate ManualVLA’s robust capabilities in predicting future images and generating precise actions.

4.5.1. Simulation benchmark.

To assess the fundamental manipulation capabilities of our method across common manipulation tasks, we conduct experiments on 10 tasks in the RL Bench [30] benchmark based on the CoppeliaSim simulator. The task suite includes *Close box*, *Close Laptop*, *Toilet seat down*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out*, *Take frame off hanger*, *Place wine at rack*, and *Water plants*. All tasks are executed on a Franka Panda robot equipped with a front-

Table 4. **Comparison of ManualVLA and baselines on RL Bench.** We train all methods in the multi-task setting [77] and report the success rates (S.R.) and variances (Var.).

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
FAST [65]	0.85	0.70	0.90	0.45	0.60	0.15	0.15	0.25	0.45	0.15	0.47 \pm 0.03
π_0 [7]	0.85	0.95	0.90	0.85	0.80	0.25	0.20	0.65	0.65	0.25	0.63 \pm 0.01
$\pi_{0.5}$ [28]	0.90	0.80	0.90	0.50	0.60	0.15	0.25	0.35	0.75	0.35	0.56 \pm 0.03
CoT-VLA [96]	0.90	0.85	0.90	0.60	0.70	0.20	0.30	0.55	0.55	0.30	0.59 \pm 0.03
ManualVLA (ours)	0.90	0.90	0.90	1.00	0.85	0.30	0.50	0.70	0.65	0.35	0.70 \pm 0.02

view RGB camera to get the visual input. We collect the data by following pre-defined waypoints and utilizing the Open Motion Planning Library [78]. Building upon the frame-sampling technique employed in previous studies [23, 32, 77], we construct a training dataset where each task contains 100 trajectories. To generate ground-truth (U, V) labels, we follow the key-frame extraction procedure from prior work. Specifically, we extract the end-effector poses of the key frames and then use the camera parameters to project their world-coordinate positions into (U, V) coordinates.

4.5.2. Training and evaluation details.

We compare ManualVLA against four state-of-the-art (SOTA) VLA models, including FAST [65], π_0 [7], $\pi_{0.5}$ [28], and CoT-VLA [96]. While the former three adopt robust action-generation paradigms, CoT-VLA conditions on the final goal image and additionally predicts future subgoal images. Specifically, FAST [65] utilizes autoregressive action outputs, π_0 [7] and $\pi_{0.5}$ [28] employ flow matching, and CoT-VLA [96] combines autoregressive image generation with diffusion-based action prediction. For all baselines, we initialize with the official pretrained parameters and strictly adhere to their original fine-tuning configurations. To ensure a fair comparison, we align the subgoal supervision in CoT-VLA to match the formulation used in ManualVLA. For ManualVLA’s input, the single-view RGB image is resized to 384×384 , with text instructions derived directly from the simulation environment and the robot state is aligned with the predicted actions. ManualVLA model is trained for 500 epochs using the AdamW optimizer [52] and CosineAnnealingLR [53] on 8 NVIDIA H20 GPUs, with mixed-precision training employed. Following [23, 41], 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.

4.5.3. Quantitative results.

As presented in Table 4, ManualVLA achieves an average success rate of 70% across 10 diverse tasks, surpassing the previous SOTA methods π_0 [6] and CoT-VLA [96] by margins of 7% and 11%, respectively. Specifically, ManualVLA attains superior performance on 8 out of 10 tasks, highlighting the advantage of ManualCoT strategy in guiding pre-

cise action generation. By generating sub-goal images and constructing visual prompt images, ManualVLA effectively leverages the fine-grained affordance guidance provided by explicit CoT reasoning. Furthermore, the MoT architecture, equipped with a shared attention module, enables robust task understanding and action generation conditioned on the sub-goal manual within the latent space. Through the integration of both explicit and implicit CoT reasoning, ManualVLA demonstrates substantial improvements in tasks requiring precise actions, such as *sweep to dustpan* and *take out umbrella*, compared to π_0 and $\pi_{0.5}$.

5. Conclusion

In this work, we address the challenge of enabling robots to autonomously perform long-horizon tasks with defined goal states, such as LEGO assembly and object rearrangement. To this end, we introduce ManualVLA, a unified VLA model built on a Mixture-of-Transformers architecture that couples multimodal manual generation with action execution. Central to the model is a Manual Chain-of-Thought (ManualCoT) process, which converts subgoal manuals into precise actions by using them as explicit control conditions and implicit manipulation cues. To support scalable training, we further develop a 3D Gaussian Splatting-based digital-twin pipeline that automatically produces large amounts of manual data. Experimental results on long-horizon tasks show that ManualVLA achieves a 32% higher average success rate than existing VLA methods.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Barr, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.
- [2] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.
- [3] Suneel Belkale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.
- [4] Xiao Bi, Deli Chen, Guanting Chen, Shanhua Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiusi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

- [5] 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.
- [6] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.
- [7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [8] 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.
- [9] 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. In *arXiv preprint arXiv:2307.15818*, 2023.
- [10] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.
- [11] 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.
- [12] Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025.
- [13] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. <https://sites.google.com/view/berkeley-ur5/home>.
- [14] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.
- [15] 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.
- [16] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiqullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.
- [17] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023.
- [18] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.
- [19] 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.
- [20] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.
- [21] figureai. Helix: A vision-language-action model for generalist humanoid control. <https://www.figure.ai/news/helix>. Accessed 2025.5.7.
- [22] Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.
- [27] Wenhui Huang, Changhe Chen, Han Qi, Chen Lv, Yilun Du, and Heng Yang. Motvla: A vision-language-action model with unified fast-slow reasoning, 2025.
- [28] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.
- [29] Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024.
- [30] 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.
- [31] 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.
- [32] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilve Wang, Longzan Luo, Xiaoqi Li, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d policy: Lifting 2d foundation models for robust 3d robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17347–17358, 2025.
- [33] 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.
- [34] 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.
- [35] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.
- [36] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, et al. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models, 2025.
- [41] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozhen 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.
- [42] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.
- [43] 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.
- [44] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.
- [45] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.
- [46] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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.
- [51] Yuxing Long, Jiyao Zhang, Mingjie Pan, Tianshu Wu, Tae-whan Kim, and Hao Dong. Checkmanual: A new challenge and benchmark for manual-based appliance manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22595–22604, 2025.
- [52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [53] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.
- [54] 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.
- [55] 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.
- [56] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiang-miao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

- [57] Corey Lynch, Ayzaan Wahid, Jonathan Thompson, 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.
- [58] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emons, 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.
- [59] 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.
- [60] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023.
- [61] 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.
- [62] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. 2024.
- [63] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023.
- [64] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+ x: Retrieval and execution from everyday human videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8284–8290. IEEE, 2025.
- [65] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, et al. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.
- [66] Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. Generating physically stable and buildable brick structures from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14798–14809, 2025.
- [67] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.
- [68] Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freck 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.
- [69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [70] 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.
- [71] 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.
- [72] 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.
- [73] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.
- [74] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MU-TEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023.
- [75] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. *ArXiv*, abs/2212.04498, 2022.
- [76] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools, 2023.
- [77] 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.
- [78] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.
- [79] 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.
- [80] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.
- [81] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Chin-Yi Cheng, and Jiajun Wu. Translating a visual lego manual to a machine-executable plan. In *European Conference on Computer Vision*, pages 677–694. Springer, 2022.
- [82] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024.
- [83] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

- [84] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.
- [85] 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. In *International Conference on Learning Representations*, 2024.
- [86] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS) 2025*. Robotics: Science and Systems Foundation, 2025.
- [87] Mingdong Wu, Fangwei Zhong, Yulong Xia, and Hao Dong. Targf: Learning target gradient field to rearrange objects without explicit goal specification. *Advances in Neural Information Processing Systems*, 35:31986–31999, 2022.
- [88] Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens Dataset. 2023.
- [89] Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*, 2025.
- [90] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejun Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.
- [91] Yiming Zeng, Mingdong Wu, Long Yang, Jiyao Zhang, Hao Ding, Hui Cheng, and Hao Dong. Lvdifffusor: Distilling functional rearrangement priors from large models into difffusor. *IEEE Robotics and Automation Letters*, 2024.
- [92] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.
- [93] 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.
- [94] Jiahao Zhang, Anoop Cherian, Cristian Rodriguez, Weijian Deng, and Stephen Gould. Manual-pa: Learning 3d part assembly from instruction diagrams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6304–6314, 2025.
- [95] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025.
- [96] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.
- [97] 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.
- [98] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.
- [99] 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.
- [100] 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.
- [101] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023.

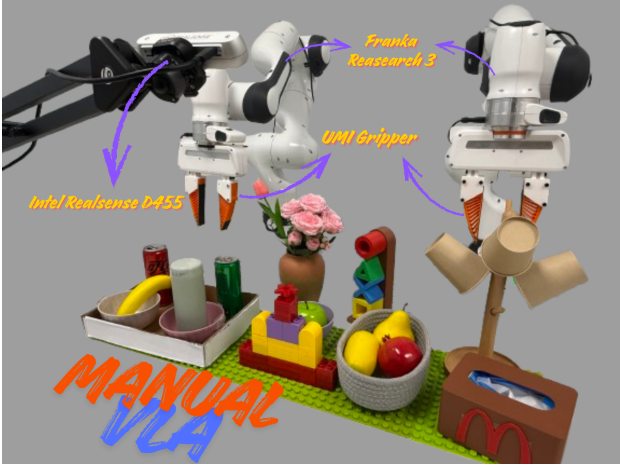


Figure 7. **Real-World Assets and Experimental Settings.** We provide visualizations of the assets used and the hardware settings for Dual-Arm Franka Platform tasks.

A. Additional Dataset Details

This section provides further details on our pretraining and real-world datasets used in the multi-stage training process. Furthermore, we elaborated in detail on the digital-twin data generation pipeline.

A.1. Pretraining Data Curation

As outlined in Stage 1 of our training strategy, the action expert is pretrained on a large-scale, cross-embodiment assembly dataset. To build this dataset, we carefully aggregate and curate demonstrations from public robotic manipulation datasets, including Open X-Embodiment [63], Droid [36], and Robomind [86]. These datasets contain millions of trajectories from a diverse set of robotic platforms, sensor configurations, and task settings. However, only a subset of these demonstrations is directly relevant to the domain of robotic assembly and rearrangement. Therefore, a careful multi-stage filtering and unification process is required before pretraining. We first apply a task-level filtering pipeline designed to extract trajectories involving low-level manipulation primitives that commonly appear in assembly and rearrangement settings, such as precise object-manipulation, and general pick-place. After filtering and integration, the assembly pretraining dataset contains more than 400,000 high-quality trajectory samples, each representing a structured demonstration of object manipulation. Although this is a fraction of the raw data available in the source corpora, the curated subset is specifically optimized for assembly-centric skills while remaining a robust foundation for the action expert to learn a wide range of manipulation primitives before fine-tuning on downstream tasks.

A.2. Real-World Data Collection

For real-world experiments, we evaluate three downstream tasks (2D LEGO Assembly, 3D LEGO Assembly, Object

Table 5. **The dataset name and sampling weight used in our mixed large-scale pretraining dataset.**

Training Dataset Mixture	
Fractal [8]	6.8%
Kuka [33]	10.5%
Bridge[19, 79]	4.9%
Taco Play [59, 71]	2.5%
Jaco Play [17]	0.4%
Berkeley Cable Routing [54]	0.2%
Roboturk [58]	2.0%
Viola [101]	0.8%
Berkeley Autolab UR5 [13]	1.0%
Toto [97]	1.7%
Language Table [57]	3.7%
Stanford Hydra Dataset [3]	3.8%
Austin Buds Dataset [100]	1.8%
NYU Franka Play Dataset [16]	0.7%
Furniture Bench Dataset [25]	2.1%
UCSD Kitchen Dataset [88]	<0.1%
Austin Sailor Dataset [61]	1.9%
Austin Sirius Dataset [47]	1.5%
DLR EDAN Shared Control [68]	<0.1%
IAMLab CMU Pickup Insert [72]	0.7%
UTAustin Mutex [74]	1.9%
Berkeley Fanuc Manipulation [99]	0.6%
CMU Stretch [60]	0.1%
BC-Z [31]	6.3%
FMB Dataset [55]	6.0%
DobbE [73]	1.2%
DROID [36]	14.2%
Stanford Kuka Dataset [39]	0.3%
Stanford Robocook Dataset [76]	0.2%
Columbia Cairlab Pusht Real [15]	<0.1%
UCSD Pick and Place	0.8%
Maniskill [24]	7.5%
Berkeley RPT [70]	<0.1%
QUT Dexterous Manipulation [11]	<0.1%
RoboSet [38]	5.2%
BridgeData V2 [79]	9.3%
RoboMind [86]	1.2%

Rearrangement) on the dual-arm Franka platform. Below, we detail the hardware configurations, task settings, and data protocols.

Hardware Configurations. We equip the dual-arm Franka experimental environment with two Franka Research 3 arms each with a 3D-printed UMI gripper, the configurations of which is summarized in Table 6. As shown in Figure 7, we utilize an Intel RealSense 455 camera to capture a static third-person view at the speed of 30Hz.

Task Settings. We provide a detailed explanation of the

Table 6. The hardware setups of the Franka Research 3, including joint position limits and velocity limit.

Joint Number	Position Limits	Velocity Limits
J1	$-166^\circ \sim +166^\circ$	$150^\circ/s$
J2	$-105^\circ \sim +105^\circ$	$150^\circ/s$
J3	$-166^\circ \sim +166^\circ$	$150^\circ/s$
J4	$-176^\circ \sim -7^\circ$	$150^\circ/s$
J5	$-165^\circ \sim +165^\circ$	$301^\circ/s$
J6	$+25^\circ \sim +265^\circ$	$301^\circ/s$
J7	$-175^\circ \sim +175^\circ$	$301^\circ/s$

assembly and rearrangement tasks and their success conditions. **2D Assembly:** The final state of the task is that LEGO blocks of different colors are randomly inserted into the same layer position on the planar board, without any stacking of blocks. Given the final 2D assembled structure as the goal, the robot arms need to pick up the LEGO blocks from both sides of the board in turn and insert them into the correct corresponding position in the center of the board. We only consider it a success if the current Lego blocks on the board match the position of the final structure at each key intermediate step. For a complete evaluation, only if all key intermediate steps are successful will it be counted as successful. **3D Assembly:** Based on the 2D Assembly Task, the 3D Assembly Task allows for more complex placement situations such as stacking between LEGO blocks. For evaluation, we do not require the placement order of LEGO blocks, but we still require that the LEGO blocks placed at each key intermediate step conform to the placement position of the corresponding color LEGO block in the final state, and the final 3D LEGO shape needs to match the given shape. **Object Rearrangement:** Unlike Assembly tasks, we use bowls, bananas, beverage cans and other common objects in life. The goal of the task is to pack the objects on the table into the box in turn and conform to the given placement pattern. In the rearrangement task, the order of placement is very important since there should be no situation where the objects in the bowl are placed before the bowl during execution. We follow the same evaluation settings as Assembly tasks at key intermediate steps and final states.

Data Protocols. For each task, we collect 100 demonstrations using 3DConnexion Spacemouse to teleoperate each Franka arm with target positions randomized on the table and box to promote data diversity. Language instructions are manually created and diversified via augmentation. Each trajectory is recorded at a frequency of 15hz, and each step contains a third-angle image shaped 640x480, the dual-arm end effector poses and the grippers discrete opening and closing. For each trajectory, we automatically filter key intermediate steps by grippers status to obtain the subgoal image, and use pixel matching to obtain the low-level (U, V) coordinates corresponding to each image.

A.3. Digital-Twin Data Generation

To train the planning expert (Stage 2) without the prohibitive cost of large-scale human annotation, we developed a high-fidelity digital-twin toolkit based on 3D Gaussian Splatting (3DGS). The pipeline consists of two main steps:

1. **Asset Reconstruction:** We first reconstruct high-fidelity 3D assets of all relevant objects, including the LEGO board, individual bricks of various colors, and the objects used in the rearrangement task. This is achieved by capturing multi-view images of each object and using them to train a 3DGS [35] model. The resulting representations are then decomposed and aligned to a unified Cartesian coordinate system for consistent spatial referencing.
2. **Iterative Scene Generation:** Given an initial state and a set of available objects, the toolkit iteratively and automatically generates intermediate task states. For LEGO assembly, it sequentially places each brick by randomly sampling a valid position on the board. At each intermediate step, we render a photorealistic image of the current scene from a fixed front-view camera perspective. This process provides the necessary data for training the planning expert: the rendered image serves as the subgoal image, the brick’s board position provides the (U, V) coordinates, and a corresponding textual description (e.g., “Yellow blocks assemble”) is generated via templates.

This automated pipeline enabled us to generate a dataset of over 10,000 frames for each task, providing the rich data needed to effectively pretrain the planning expert. More visualizations of the generated sim-to-real data are shown in Figure 8 and Figure 9.

B. Additional Experimental Details

In this section, we report the additional ablation studies on the impact of manual generation quality and token sequence arrangement on model’s action accuracy.

B.1. Additional Ablation Studies

This section will present additional ablation studies to further validate our design choices.

Table 7. Comparison of manual generation quality impact on action generation.

Training Frames	PSNR \uparrow	2D LEGO Assembly				
		2 bricks	→ 2 bricks	→ 2 bricks	→ 2 bricks	→ S.R.
0.5K	25.71	0.35	0.25	0.20	0.20	
1K	26.61	0.45	0.35	0.30	0.25	
3K	27.16	0.65	0.65	0.60	0.60	
6K	28.29	0.85	0.80	0.80	0.80	
10K	29.01	0.95	0.90	0.85	0.85	

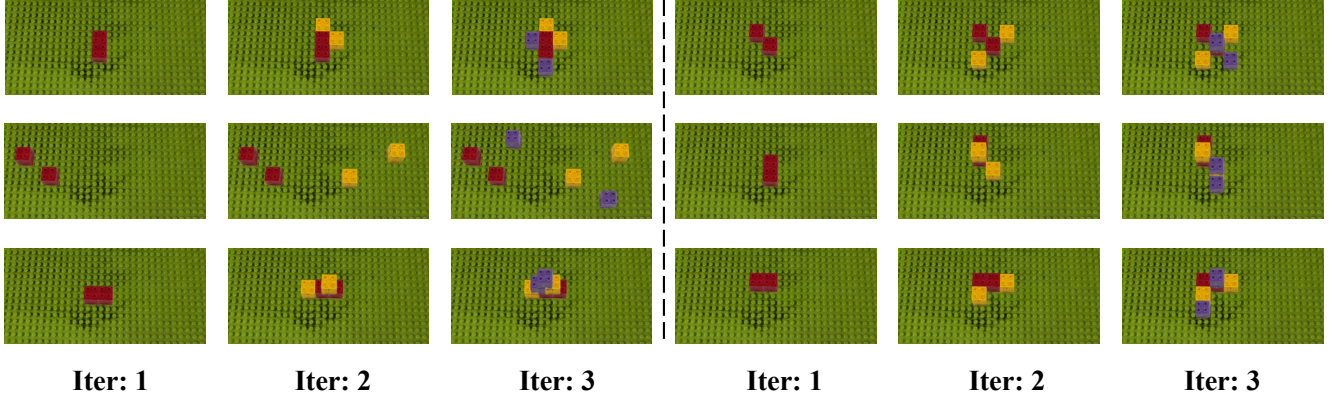


Figure 8. **Iterative manual generation examples for LEGO Assembly.** Each row shows a sequence where two bricks are progressively stacked per iteration. Scenes are rendered at each step using our digital-twin toolkit.

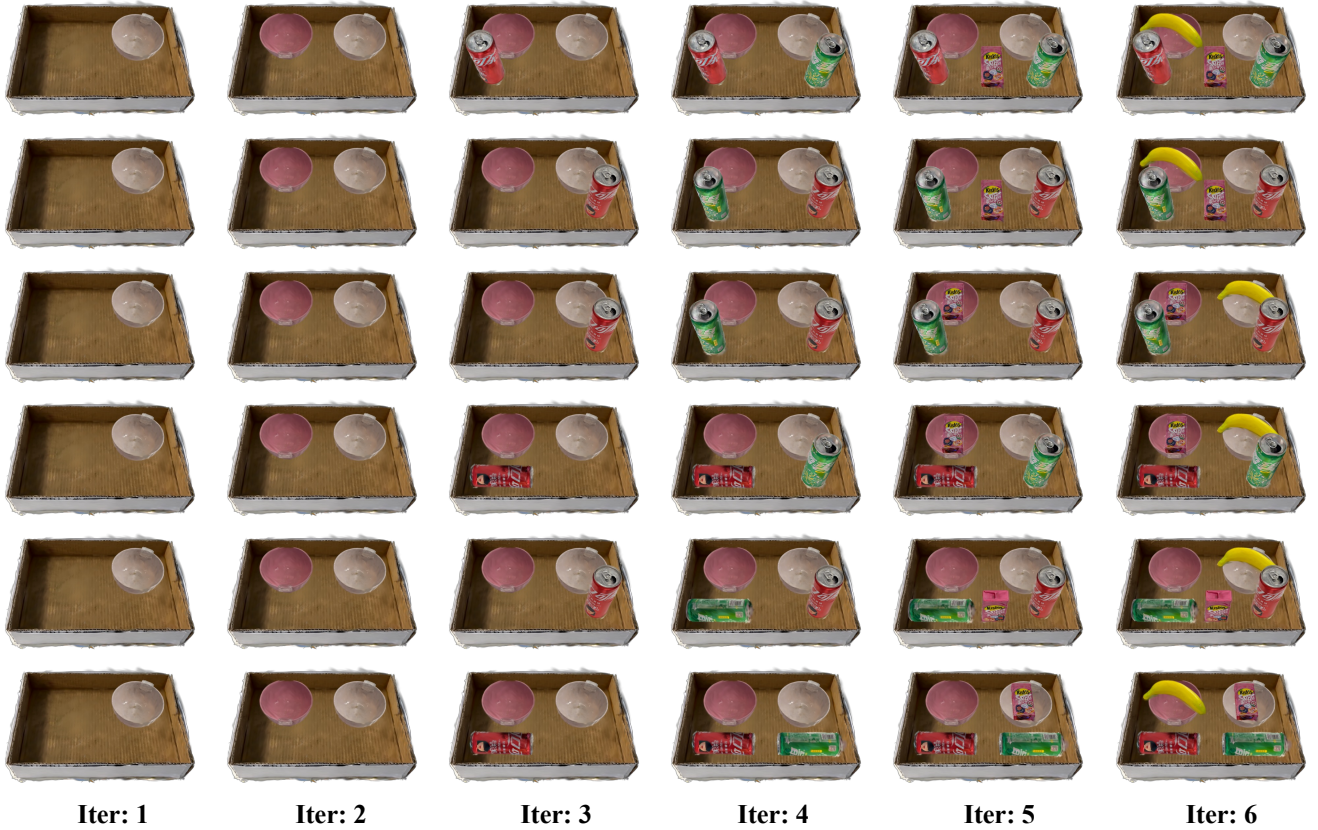


Figure 9. **Iterative manual generation examples for Object Rearrangement.** Each row shows a sequential process where objects are placed into the box one at a time. Scenes are rendered at each step using our digital-twin toolkit.

B.1.1. Impact of Manual Generation Quality on Action

To evaluate the robustness of the action expert under varying manual-generation quality, we compare five versions of our planning expert trained on datasets of 0.5K, 1K, 3K, 6K and 10K frames, respectively, for the 2D LEGO Assembly task.

All these training frames are generated using our high-fidelity digital-twin toolkit. These planning experts produce manuals with differing levels of fidelity, quantified by the PSNR of the generated subgoal images. We condition ManualVLA on these manuals and measure the resulting task success rate

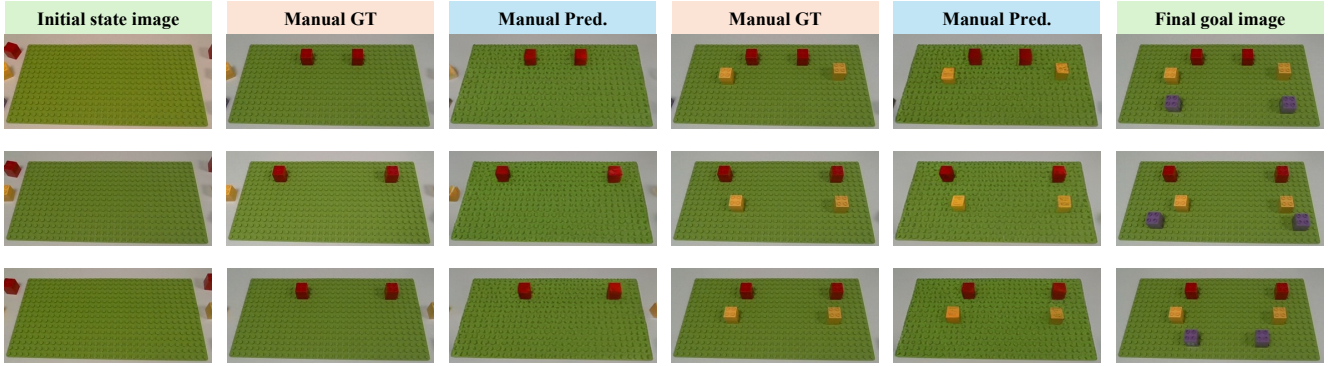


Figure 10. **Iterative manual generation examples for 2D LEGO Assembly.** Pred refers to the predictions generated by our model, while GT denotes the ground truth in the test set.

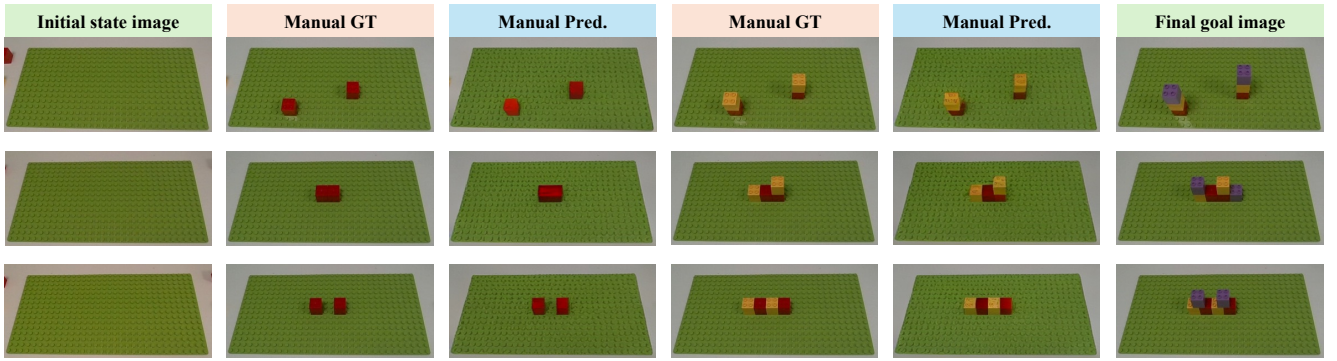


Figure 11. **Iterative manual generation examples for 3D LEGO Assembly.** Pred refers to the predictions generated by our model, while GT denotes the ground truth in the test set.

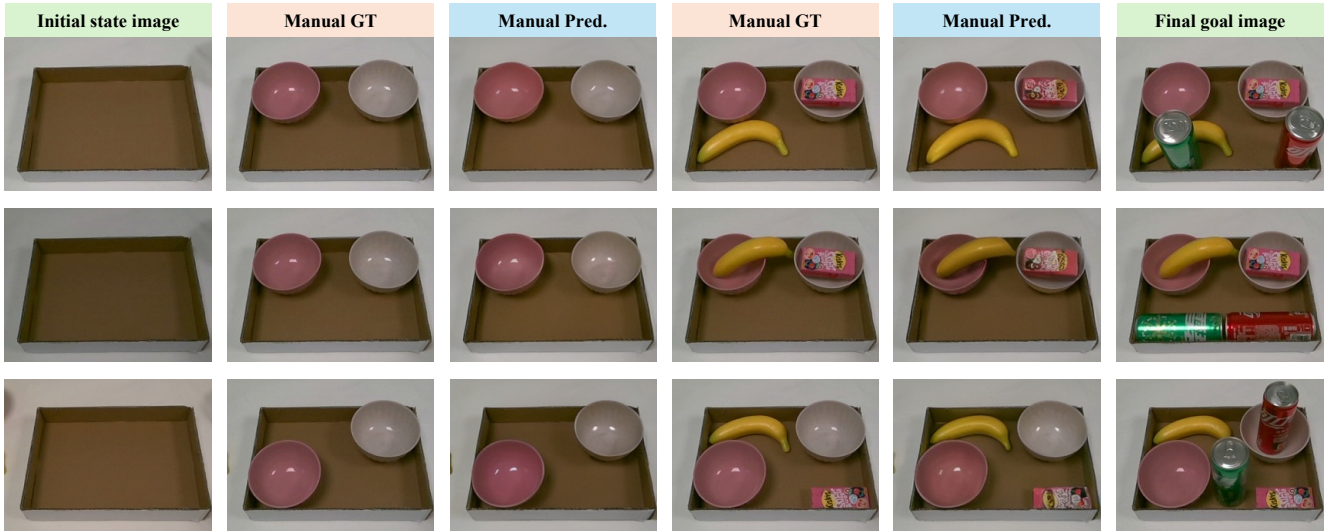


Figure 12. **Iterative manual generation examples for objects rearrangement.** Pred refers to the predictions generated by our model, while GT denotes the ground truth in the test set.

over 20 rollouts, as reported in Table 7. The results indicate that low-quality subgoal images lead to substantial error accumulation during action generation, significantly degrading overall performance. Once the training set reaches 1,000 samples, yielding manuals with PSNR above 27, the action expert exhibits stable and reliable behavior. This trend highlights ManualVLA’s robustness: when the planning expert is sufficiently trained, its explicit and implicit chain-of-thought reasoning reliably supports consistent action generation.

B.1.2. Impact of Token Sequence Arrangement

Table 8. **Comparison of different token sequence order impact on action generation.** Here, T and P denote the textual description and coordinate pairs (U, V) in the manual, while I_{subgoal} and I_{prompt} refer to the subgoal image and the visual prompt image.

Token Sequence	2D LEGO Assembly			
	2 bricks \rightarrow	2 bricks \rightarrow	2 bricks \rightarrow	S.R.
$P \rightarrow I_{\text{subgoal}} \rightarrow T \rightarrow I_{\text{prompt}}$	0.90	0.85	0.80	0.80
$I_{\text{subgoal}} \rightarrow P \rightarrow T \rightarrow I_{\text{prompt}}$	0.75	0.75	0.70	0.70
$T \rightarrow I_{\text{subgoal}} \rightarrow P \rightarrow I_{\text{prompt}}$	0.85	0.80	0.80	0.80
$T \rightarrow P \rightarrow I_{\text{subgoal}} \rightarrow I_{\text{prompt}}$	0.95	0.90	0.85	0.85

To further analyze how the ordering of multimodal tokens in the generated manual influences action generation, we evaluate four different sequence arrangements, as shown in Table 8. The modalities contained in the manual serve complementary purposes: the text instruction provides high-level semantic goals, the coordinate tokens (U, V) specify the future spatial locations of the objects to be manipulated, and the generated manual image offers step-wise visual cues synthesized by the planning expert. Finally, the visual prompt image is conditioned on the generated coordinates (U, V) , and therefore we place it at the end of the token sequence. This ordering naturally forms a pipeline in which the model first performs implicit CoT reasoning, followed by explicit CoT reasoning, before producing the final action sequence. Because the effectiveness of the action expert depends on how well it integrates semantic and visual information, the ordering of these tokens can substantially affect downstream policy performance. Our study on different generation orders of the three types of manual information reveals that the sequence of generating text first, then coordinates, and finally subgoal images yields the best task success rate. Meanwhile, the other sequence configurations introduce only minor performance degradation, demonstrating both the robustness of ManualVLA and the critical role of the combined implicit–explicit CoT reasoning process in enabling strong action-generation performance.

C. Additional Qualitative Results

This section provides more qualitative results for manual generation and real-world robot rollouts.

C.1. Manual Generation Visualization

Figure 10, Figure 11, and Figure 12 provide additional visualizations of the manuals generated by our planning expert across all three downstream tasks. These examples showcase the model’s ability to generate structured and interpretable intermediate states that accurately guide the subsequent action generation. For the LEGO assembly tasks, ManualVLA sequentially reconstructs the correct brick placements and colors, demonstrating precise step-wise reasoning. Similarly, for object rearrangement, it progressively generates subgoals that accurately capture the spatial relationships between objects, moving step-by-step toward the final goal configuration. Overall, these results highlight ManualVLA’s strong intermediate reasoning capabilities, establishing a reliable foundation for the action expert to generate accurate actions.

C.2. Real-World Rollout Visualization

The qualitative rollouts in Figure 13 further corroborate our quantitative findings, illustrating keyframes of the dual-arm real-world execution processes. The visualizations demonstrate that ManualVLA can follow the internally generated manuals to reliably guide the action expert in producing precise grasping, insertion, and placement motions. In both the 2D and 3D LEGO assembly tasks, compared with the final goal image, the robot consistently maintains accurate brick placement throughout all stages. For the object-rearrangement task, also compared with the final goal image, it stably manipulates objects with varying shapes and occlusions. These results collectively validate ManualVLA’s strong action generation capabilities, demonstrating its potential as a robust policy for real-world, long-horizon robotic manipulation.

D. Failure Case Analysis

Although ManualVLA demonstrates strong overall performance, it is not without limitations. Through our experiments, we identified two primary failure modes, as illustrated in Figure 14:

1. **Occasional Erroneous LEGO Placement.** While ManualVLA is generally successful in accomplishing the LEGO assembly task, it can still produce incorrect placements. As shown in Cases 1 and 2 of Figure 14, the system places the yellow bricks in incorrect positions due to model errors. Notably, however, the system is often able to recover from such mistakes and correctly place subsequent bricks.
2. **Placement Errors Under Large Rotation Angles.** In the Objects Rearrangement task, certain scenarios require the robot arm to perform large rotations to achieve the correct placement orientation. ManualVLA may fail in these situations, as illustrated by Case 3 in Figure 14,



Figure 13. **Real-World Task Execution Progress Visualization.** We provide visualizations of three real world tasks including assembly and rearrangement evaluated on dual-arm Franka robot platform.

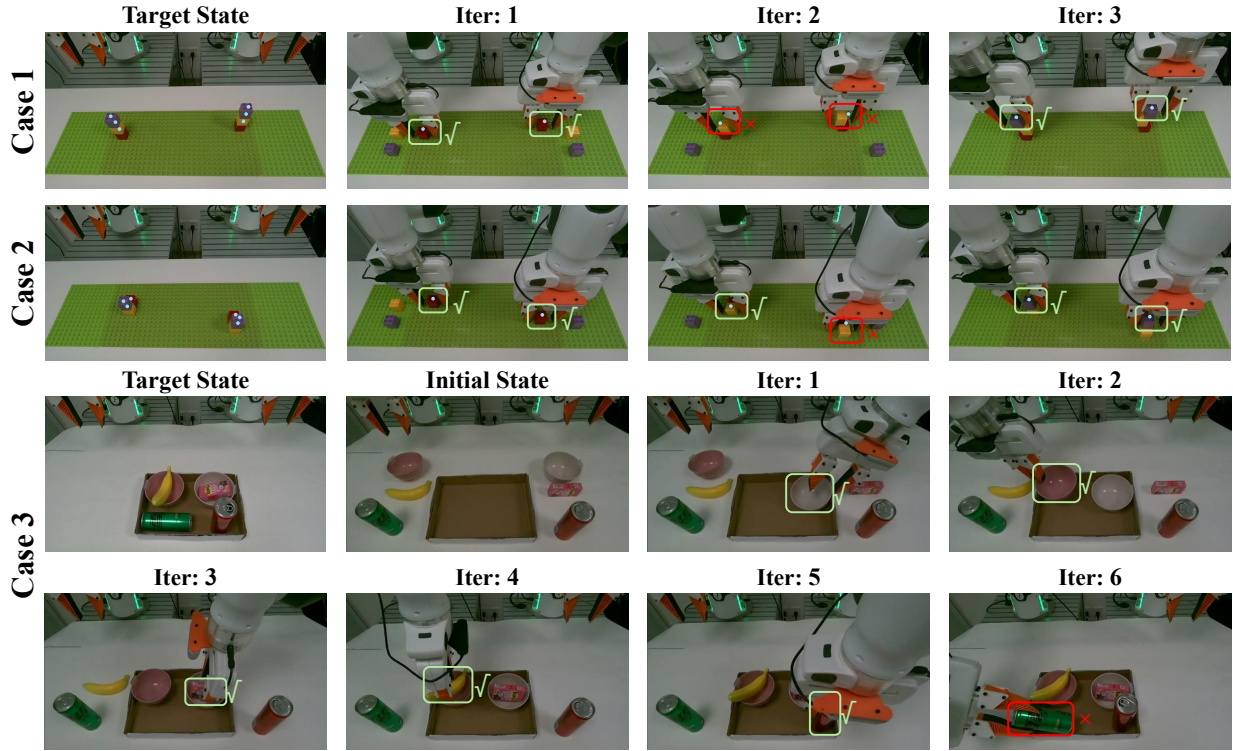


Figure 14. **Failure cases in our two tasks: LEGO assembly and objects rearrangement.** The top two rows illustrates two LEGO failure cases and the bottom two rows shows a failure case of objects rearrangement task.

where the robot arm fails to place the spirit can into the box. We hypothesize that these failures stem from the limited number of such extreme rotation cases in the training data.