Xiaomi-Robotics-0: An Open-Sourced Vision-Language-Action Model with Real-Time Execution

Xiaomi Robotics

Abstract

In this report, we introduce `Xiaomi-Robotics-0`, an advanced vision-language-action (VLA) model optimized for high performance and fast and smooth real-time execution. The key to our method lies in a carefully designed training recipe and deployment strategy. `Xiaomi-Robotics-0` is first pre-trained on large-scale cross-embodiment robot trajectories and vision-language data, endowing it with broad and generalizable action-generation capabilities while avoiding catastrophic forgetting of the visual-semantic knowledge of the underlying pre-trained VLM. During post-training, we propose several techniques for training the VLA model for asynchronous execution to address the inference latency during real-robot rollouts. During deployment, we carefully align the timesteps of consecutive predicted action chunks to ensure continuous and seamless real-time rollouts. We evaluate `Xiaomi-Robotics-0` extensively in simulation benchmarks and on two challenging real-robot tasks that require precise and dexterous bimanual manipulation. Results show that our method achieves state-of-the-art performance across all simulation benchmarks. Moreover, `Xiaomi-Robotics-0` can roll out fast and smoothly on real robots using a consumer-grade GPU, achieving high success rates and throughput on both real-robot tasks. To facilitate future research, code and model checkpoints are open-sourced at https://xiaomi-robotics-0.github.io.

1 Introduction

Vision-language-action (VLA) models have emerged as a new paradigm for effective robot policy learning [3, 24, 75]. Building upon pre-trained vision-language models (VLMs), VLA models provide a unified framework that maps observations and language instructions directly to actions across a wide range of tasks. However, despite their strong performance and generalization capabilities, VLA models suffer from high inference latency due to their large parameter counts, which can scale to billions of parameters. This creates challenges for smoothly chaining actions across consecutive inference steps, leading to out-of-distribution jerky motions if not handled properly [4].

In this report, we introduce `Xiaomi-Robotics-0` (Fig. 1), an advanced vision-language-action (VLA) model that delivers high performance while enabling fast and smooth rollouts on real robots. It is composed of a pre-trained vision-language model (VLM) [1] for processing vision and language inputs and a diffusion transformer [48] for generating actions via flow-matching [34, 39]. The training recipe contains two stages: pre-training and post-training. During pre-training, we train the model with large-scale cross-embodiment robot trajectories and vision-language data. This stage endows the model with broad and generalizable action generation capabilities while maintaining the strong vision-language capabilities in the underlying pre-trained VLM it built upon. During post-training, we introduce novel techniques to enable fast and smooth asynchronous execution during real-robot rollouts. Specifically, we first condition the generation of action

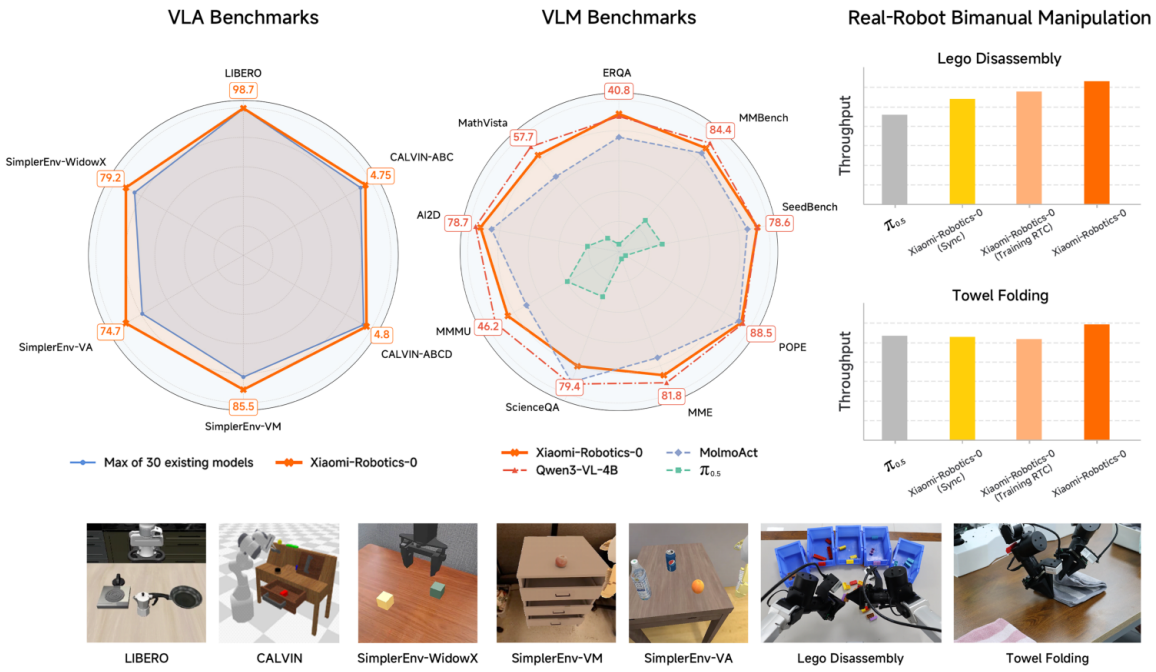


Figure 1 Overview. `Xiaomi-Robotics-0` achieves state-of-the-art performance in three widely-used simulation benchmarks. It also attains high throughput on two challenging real-robot bimanual manipulation tasks. Furthermore, it matches the underlying pre-trained VLM on several VLM benchmarks.

chunks by prefixing it with actions from the previous inference as in [5]. While this conditioning method ensures continuity across consecutively generated chunks, it allows the generation of later-timestep actions to exploit the temporal correlation that successive actions tend to be similar. As a result, policy learning can take a shortcut by simply imitating the action prefix rather than attending to visual and language signals, resulting in less reactive policies and degraded performance. To address this issue, we replace the causal attention mask with a Λ-shape attention mask [16, 20, 70] during post-training, encouraging action generation to pay more attention to visual and language conditions rather than over-relying on the action prefix. During deployment, we carefully align the timesteps of action chunks generated from consecutive inferences to ensure continuous and seamless real-robot rollouts.

We evaluate `Xiaomi-Robotics-0` extensively on both simulation benchmarks and a bimanual real-robot platform. Our model achieves state-of-the-art performance across three widely-used simulation benchmarks. Specifically, it achieves an average success rate of **98.7%** on **Libero** [35]. On **SimplerEnv** [30], it delivers strong performance under the visual matching (**85.5%**) and visual aggregation (**74.7%**) settings in the Google Robot evaluations as well as the WidowX evaluations (**79.2%**). On **CALVIN** [44], `Xiaomi-Robotics-0` improves the average length of completing 5 tasks in a row from 4.54 to **4.75** and from 4.67 to **4.80** on the ABC→D and ABCD→D split, respectively. In real-robot experiments, we evaluate on two challenging tasks that require precise and dexterous bimanual manipulation: Lego disassembly and towel folding. `Xiaomi-Robotics-0` is able to achieve high success rates on both tasks while outperforming state-of-the-art methods [5, 19] on both tasks in terms of throughput, enabling rapid and smooth real-time execution. In addition, our pre-trained model matches the performance of the underlying pre-trained VLM [1] on several general vision-language benchmarks and a benchmark focused on embodied reasoning [63].

We release the pre-trained and post-trained checkpoints, along with the inference code to facilitate future research. We hope these resources serve as a practical foundation for advancing vision-language-action (VLA) models.

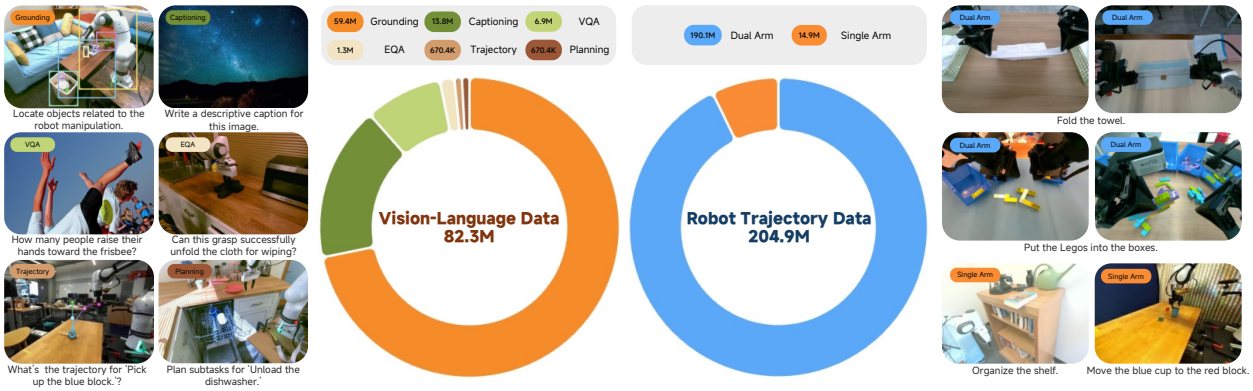


Figure 2 Data. `Xiaomi-Robotics-0` leverages both robot trajectory data and vision-language (VL) data during pre-training.

2 Xiaomi-Robotics-0

`Xiaomi-Robotics-0` is an end-to-end vision-language-action (VLA) model that takes as inputs observation images, a language instruction, and the robot proprioceptive state. It outputs an action chunk [74] to control a bimanual robot in an end-to-end manner.

2.1 Data

We leverage both robot trajectory data and vision-language (VL) data during training. Fig. 2 illustrates the detailed data composition. Our robot trajectory data are sourced from multiple open-sourced robot datasets (*e.g.*, DROID [23] and MolmoAct [26]) as well as in-house data collected by ourselves. Our in-house data consists of teleoperated trajectories for two challenging tasks: lego disassembly and towel folding. In total, we collected 338 and 400 hours of data for these two tasks, respectively. Overall, the entire robot trajectory dataset contains about 200M timesteps for training.

For the vision-language data, we curate a comprehensive corpus of more than 80M samples from two primary sources: general vision-language (VL) datasets [9, 59, 65, 68] and robot datasets [23, 26]. While general VL data preserve broad semantic knowledge, VL data derived from robot trajectories enhance the model's perception on robot-centric images, which are often captured from egocentric perspectives or wrist-mounted cameras. Specifically, we curate data by focusing on four vision-language tasks: (1) visual grounding, (2) visual question answering (VQA), (3) image captioning, and (4) embodied reasoning & planning. For visual grounding, we develop a rigorous cross-validation consensus mechanism using Grounded SAM [55], Grounding DINO 1.5 [54], and LLMDet [15] to ensure pixel-level precision of annotations. VQA and captioning annotation quality is further refined through re-labeling using state-of-the-art pre-trained VLMs [1]. For embodied reasoning & planning, we leverage pre-trained VLMs to generate data from root trajectories, focusing on embodied question answering (EQA), high-level task planning, and point trajectory prediction.

2.2 Model & Training

`Xiaomi-Robotics-0` adopts a mixture-of-transformers (MoT) [32] model architecture. It consists of a pre-trained vision-language model (VLM) (*i.e.*, Qwen3-VL-4B-Instruct [1]) and a diffusion transformer (DiT) [48]. The VLM takes as inputs observation images $\mathbf{o}_t$ of the current timestep, along with a language instruction l provided by the user. The DiT generates a T-step action chunk $\mathbf{a}_{t:t+T}$ [74] via flow-matching, conditioned on the KV cache produced by the VLM and the robot proprioceptive state. In total, the model has 4.7B parameters.

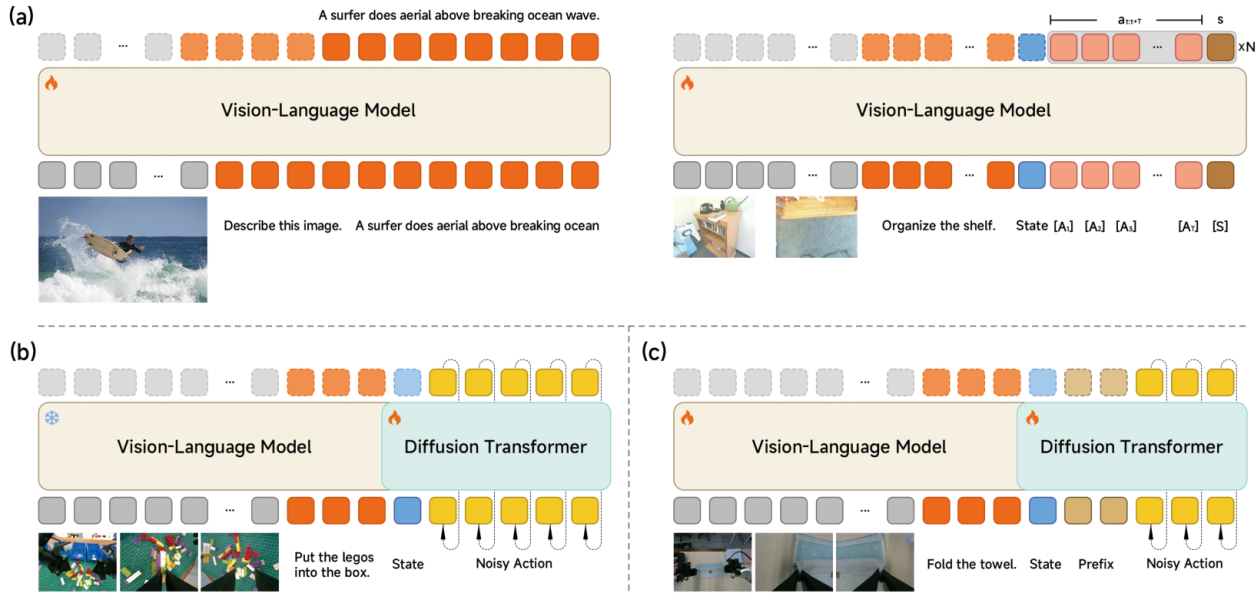


Figure 3 Model & Training. (a) During the first step of pre-training, we train the VLM on both vision-language data (left) and robot trajectory data (right). Vision-language data are trained via a next-token-prediction objective. We adopt the training paradigm in Choice Policies [51] to train the VLM for action prediction on the robot trajectory data. (b) In the second step of pre-training, we freeze the VLM and train the diffusion transformer for generating action via flow-matching. (c) During post-training for asynchrnnous execution, we append clean action prefix in front of the noisy action tokens.

2.2.1 Pre-training

We perform pre-training in two steps. In the first step, our goal is to endow the VLM with action-generation capability by training it to predict action chunks from observation images, language instructions, and robot proprioceptive states. To account for the multi-modality in trajectories, we adopt Choice Policies [51] for action prediction. Specifically, we train the VLM to simultaneously predict N action chunk candidates along with their corresponding N scores (right of Fig. 3(a)). During training, we compute the L_1 distance between each predicted action chunk candidate and the ground truth, and utilizing these values as supervision targets for score prediction. Action prediction is supervised via a winner-takes-all scheme: only the candidate with the lowest L_1 distance is updated via backpropagation.

Architecturally, we encode the robot proprioceptive state $\mathbf{s}_t$ using an MLP. For action prediction, we append T learnable tokens $\texttt{[}A_i\texttt{]}$ to predict N sets of T-step action chunks $\mathbf{a}_{t:t+T}$, and one additional token $\texttt{[}S\texttt{]}$ to predict the score s for each chunk. The input token sequence is: $\mathbf{o}_t, l, \mathbf{s}_t, \texttt{[}A_1\texttt{]}, ..., \texttt{[}A_T\texttt{]}, \texttt{[}S\texttt{]}$. The output of each action token $\texttt{[}A_i\texttt{]}$ is mapped to N predictions of the action at the i-th timestep, while the output of the score token $\texttt{[}S\texttt{]}$ is mapped to N scores.

To avoid catastrophic forgetting of the strong vision-language capabilities of the underlying pre-trained VLM [1], and to improve its visual understanding on robot-centric data, we co-train the model with the entire robot trajectory data *and* vision-language data described in Sec. 2.1. The vision-language data is trained with a next-token-prediction objective (left of Fig. 3(a)). We sample vision-language data and robot trajectory data at a ratio of 1:6.

After the first step of training, the VLM is equipped with the capability to generate actions. In the second step, we freeze the VLM and train the diffusion transformer (DiT) from scratch on the entire robot trajectory data with a flow-matching loss (Fig. 3(b)):

$$L(\theta) = ||\mathbf{v}_\theta(\mathbf{o}_t, l, \mathbf{s}_t, \tilde{\mathbf{a}}^\tau_{t:t+T}, \tau) - \mathbf{u}(\tilde{\mathbf{a}}^\tau_{t:t+T}, \mathbf{a}_{t:t+T}, \tau)||^2_2 \tag{1}$$

$\tau \in [0, 0.999]$ is the flow-matching timestep. $\mathbf{a}^\tau_{t:t+T} = \tau \tilde{\mathbf{a}}_{t:t+T} + (1-\tau)\boldsymbol{\epsilon}$ is the noisy action where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

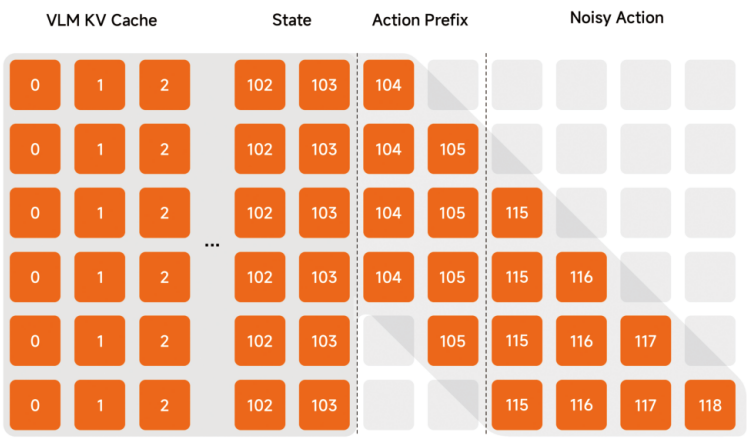


Figure 4 The Λ-Shape Attention Mask for Post-Training. A noisy action token can only attend to the vision and language tokens via the VLM KV cache, the sink token, the state token, and the action tokens of the previous w timesteps. The number in each token indicates the RoPE positional index of the token. Note that we add an offset of 10 for the noisy action tokens to allow the model to distinguish them from the clean action prefix tokens.

Following [3, 19], we sample τ from a Beta distribution, placing more weight on noisier timesteps during training. We leverage adaptive normalization layers (adaLN) [48, 49] to inject the flow-matching timestep condition into the DiT for action generation. The robot proprioceptive state and noisy actions are encoded with MLPs. We add a learnable attention sink token at the front of the state and noisy action tokens to stabilize the attention distribution during training. The input tokens for the DiT are sequenced as: $\texttt{[SINK]}, \mathbf{s}_t, \tilde{\mathbf{a}}_t, \ldots, \tilde{\mathbf{a}}_{t+T-1}$, where $\tilde{\mathbf{a}}_{t+k}$ corresponds to a noisy action. To account for the temporal relationship between actions at different timesteps, we use causal attention in the DiT. To reduce inference latency, we use a 16-layer DiT and condition it on the KV cache from the last 16 layers of the VLM. During this step, we leverage the VLM as a frozen multimodal conditioner for providing visual-language features, while the DiT learns to generate action chunks conditioned on these features. The inputs to the VLM contains only the observation images $\mathbf{o}_t$ and language l without the newly introduced tokens for action prediction as in the first step.

2.2.2 Post-training

In post-training, we adapt `Xiaomi-Robotics-0` to a specific robot by training solely on its native trajectory data. We describe two post-training methods for synchronous and asynchronous execution, respectively. For synchronous execution, we simply unfreeze the entire model, *i.e.*, the VLM and DiT, and continue training on predicting actions via flow-matching as in the second step of pre-training.

However, when deploying on real robots, inference latency becomes non-negligible due to the large number of parameters, causing pauses in synchronous execution where the robot remains idle until the next inference is completed. Asynchronous execution enables the robot to continue rolling out trajectories during model inference. In this setting, it is crucial to maintain consistency across consecutively inferred action chunks and ensure smooth transitions between them, since inconsistency can induce jerky motions and drive the robot into out-of-distribution regimes [4]. To address this problem, prior work proposes real-time chunking (RTC) [4] and training-time RTC [5], which condition action generation on previously committed actions. In this work, we follow training RTC [5] and condition action generation on Δt_c previously committed actions by prefixing them to the noisy action tokens in DiT (Fig. 3(c)). The input token sequence of the DiT thus becomes: $\texttt{[SINK]}, \mathbf{s}_t, \mathbf{a}_t, \ldots, \mathbf{a}_{t+\Delta t_c-1}, \tilde{\mathbf{a}}^{\tau}_{t+\Delta t_c}, \ldots, \tilde{\mathbf{a}}^{\tau}_{t+T-1}$. While this approach reduces inconsistency, it also enables predictions of later-timestep actions to exploit the temporal correlation between successive actions. As a result, policy learning may take a shortcut by simply copying the action prefix instead of attending to visual and language inputs, leading to less reactive policies and degraded performance.

We propose simple techniques to alleviate this issue. We first simply add an offset to the RoPE positional

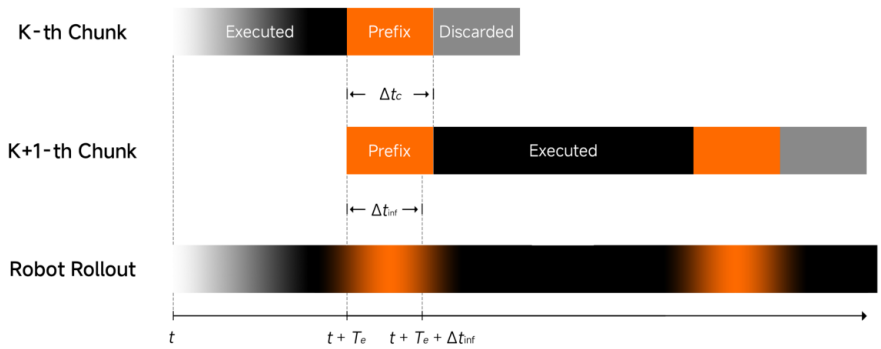


Figure 5 Asynchronous Execution. We show two consecutive chunks and how they are stitched together during robot rollout. See Sec. 2.3 for more details.

indices of the noisy action tokens to enable the model to distinguish tokens of noisy actions from those of the clean action prefix. In addition, we change the original causal attention mask of the DiT to a Λ-shape attention mask [16, 20, 70] (Fig. 4). Since the noisy action tokens immediately following the tokens of the action prefix can attend to them, the generated actions can smoothly transition from the action chunk produced by the previous inference. In contrast, noisy action tokens of later timesteps cannot attend to the tokens corresponding to the conditioned action prefix, forcing them to attend to other signals (*e.g.*, visual observations and languages), thus ensuring reactivity in the predicted actions.

During training, we sample Δt_c from the set $\{0, 1, \cdots, 6\}$. When $\Delta t_c > 0$, we dynamically re-weight the flow-matching loss based on the L_1 error between *online-predicted actions* and the ground-truth actions. This strategy prioritizes samples with larger deviations, directing the model to focus on correcting significant execution errors.

2.3 Deployment

We describe deployment methods for synchronous and asynchronous execution, respectively.

Synchronous Execution. For synchronous execution, we control the robot to execute the first T_e steps of actions within the T-step predicted action chunk. Once these actions have been rolled out, we immediately start inferring the next action chunk using the latest observation images and proprioceptive state. The robot remains idle until the inference completes.

Asynchronous Execution. We visualize the asynchronous execution in Fig. 5. For each inferred action chunk, we first similarly roll out T_e steps before triggering the subsequent inference cycle. However, instead of staying idle, the robot continues executing the remaining actions of the current chunk while the next chunk is being inferred. We condition the next inference by prefixing the noisy actions with the actions from step T_e to step $T_e + \Delta t_c - 1$ of the current chunk. Upon completion, the newly generated chunk is executed starting from step Δt_{inf}, where Δt_{inf} is the inference latency. We set $\Delta t_c \geq \Delta t_{\text{inf}}$ so that the action prefix covers the entire inference window. As a result, there are always actions available for execution throughout the entire inference, enabling seamless transition between consecutive control cycles.

During inference, we initialize the action chunk by sampling from a standard Gaussian distribution, $\mathbf{a}_{t:t+T}^{\tau=0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then perform 5 flow-matching steps and integrate τ from 0 to 1 to obtain the predicted action chunk. Deployed on an NVIDIA GeForce RTX 4090 GPU, the model achieves an inference latency of $t_{\text{inf}} = 80$ms. To ensure consistency with the training distribution, we synchronize all input modalities by resampling them to a unified 30 Hz timeline using timestamps. At each clock tick, the temporally nearest measurements from all sensors are aggregated to form a synchronized model input.

3 Experiments

Method	Libero-Spatial	Libero-Object	Libero-Goal	Libero-Long	Average
OpenVLA [24]	84.7%	88.4%	79.2%	53.7%	76.5%
OpenVLA-OFT [25]	97.6%	98.4%	97.9%	94.5%	97.1%
π_0 [3]	96.8%	98.8%	95.8%	85.2%	94.2%
π_0-FAST [50]	96.4%	96.8%	88.6%	60.2%	85.5%
$\pi_{0.5}$ [19]	98.8%	98.2%	98.0%	92.4%	96.9%
GR00T-N1 [45]	94.4%	97.6%	93.0%	90.6%	93.9%
UniVLA [8]	95.4%	98.8%	93.6%	94.0%	95.4%
Discrete Diffusion VLA [33]	97.2%	98.6%	97.4%	92.0%	96.3%
MemoryVLA [60]	98.4%	98.4%	96.4%	93.4%	96.7%
FLOWER [58]	97.2%	99.3%	96.9%	92.4%	93.4%
EO-1 [52]	**99.7**%	99.8%	**99.2**%	94.8%	98.2%
`Xiaomi-Robotics-0` (Ours)	98.8%	**100.0**%	98.8%	**97.2**%	**98.7**%

Table 1 Results on the LIBERO benchmark.

3.1 Simulation Benchmarks

We evaluate our method on three widely-used simulation benchmarks.

- **LIBERO** [35]: LIBERO features a robot arm performing various manipulation in the simulation. We use the filtered expert demonstrations from [24], which remove unsuccessful trajectories, for training. Following [50], we train the model on data from all four splits, *i.e.*, Libero-Spatial, Libero-Object, Libero-Goal, and Libero-Long. We follow the standard evaluation protocol in OpenVLA [24] and report success rates on each split and the average success rate across all splits. We set the action chunk length $T = 10$.

- **CALVIN** [44]: CALVIN contains four different environments (*i.e.*, A, B, C, and D) in total. It is designed for multi-task learning and long-horizon manipulation. We follow the standard evaluation protocol [44] and evaluate on the ABCD→D and ABC→D splits. For ABCD→D, the model is trained on data collected from environments A, B, C, and D; for ABC→D, it is trained on data collected from environments A, B, and C only. In both settings, evaluation is conducted in environment D. Therefore, ABCD→D measures in-distribution performance, whereas ABC→D quantifies out-of-distribution generalization capabilities. During evaluation, the model is prompted with 1000 unique instruction chains, each containing five instructions. For each chain, the model outputs actions to control the robot to solve tasks specified by the instructions sequentially. We report success rates of completing 1, 2, 3, 4, and 5 tasks in a row and the average length of tasks completed per chain. We set the action chunk length $T = 10$.

- **SimplerEnv** [30]: SimplerEnv is a real-to-sim benchmark featuring two robot platforms: Google Robot and WidowX. It enables a pipeline where policies are trained on real-world robot trajectories and subsequently evaluated in simulated environments. The Google Robot environment provides two evaluation settings: Visual Matching and Variant Aggregation. Visual Matching aligns the visual appearance between real-robot scenes and their simulated counterparts, whereas Variant Aggregation introduces visual randomization to test robustness. Following the standard evaluation protocol of SimplerEnv [30], we train our policy on the RT-1 Fractal dataset [6] and evaluate it on four tasks in the Google Robot environment under both settings. For the WidowX environment, we train our policy on the Bridge dataset [66]. For each policy, we report success rates for all tasks and variants, as well as the average success rate across tasks. We set the action chunk length $T = 4$.

Across all three simulation benchmarks, `Xiaomi-Robotics-0` achieves state-of-the-art (SoTA) performance. On **LIBERO** (Tab. 1), using the standard evaluation protocol, we obtain an average success rate of **98.7%**, outperforming all the comparing baseline methods. On **CALVIN** (Tab. 2), our method showcases clear advantages in both multi-task long-horizon manipulation (ABCD→D) and zero-shot environment generalization (ABC→D). Measured by the average number of tasks completed in a row of 5, it achieves **4.80** and **4.75** in the two settings, respectively, substantially outperforming prior baseline methods. On **SimplerEnv** (Tab. 5 & 4), `Xiaomi-Robotics-0` achieves average success rates of **85.5%** and **74.7%** on Google Robot evaluations under

Method	Setting	Tasks Completed in a Row					Avg. Len. ↑
		1	2	3	4	5	
RoboFlamingo [28]	ABCD→D	96.4%	89.6%	82.4%	74.0%	66.0%	4.08
GR-1 [69]	ABCD→D	94.9%	89.6%	84.4%	78.9%	73.1%	4.21
MoDE [56]	ABCD→D	97.1%	92.5%	87.9%	83.5%	77.9%	4.39
RoboVLMs [29]	ABCD→D	96.7%	93.0%	89.9%	86.5%	82.6%	4.49
MDT [57]	ABCD→D	98.6%	95.8%	91.6%	86.2%	80.1%	4.52
UniVLA [8]	ABCD→D	98.5%	96.1%	93.1%	89.9%	85.1%	4.63
FLOWER [58]	ABCD→D	99.2%	96.9%	96.9%	92.3%	88.3%	4.67
`Xiaomi-Robotics-0` (Ours)	ABCD→D	**99.7%**	**98.0%**	**96.7%**	**94.2%**	**91.8%**	**4.80**
RoboFlamingo [28]	ABC→D	82.4%	61.9%	46.6%	33.1%	23.5%	2.47
SuSIE [2]	ABC→D	87.0%	69.0%	49.0%	38.0%	26.0%	2.69
GR-1 [69]	ABC→D	85.4%	71.2%	59.6%	49.7%	40.1%	3.06
3DDA [21]	ABC→D	93.8%	80.3%	66.2%	53.3%	41.2%	3.35
MoDE [56]	ABC→D	96.2%	88.9%	81.1%	71.8%	63.5%	4.01
GR-MG [27]	ABC→D	96.8%	89.3%	81.5%	72.7%	64.4%	4.04
RoboVLMs [29]	ABC→D	98.0%	93.6%	85.4%	77.8%	70.4%	4.25
Seer-Large [64]	ABC→D	96.3%	91.6%	86.1%	80.3%	74.0%	4.28
VPP [17]	ABC→D	95.7%	91.2%	86.3%	81.0%	75.0%	4.29
UniVLA [8]	ABC→D	98.9%	94.8%	89.0%	82.8%	75.1%	4.41
FLOWER [58]	ABC→D	99.3%	95.9%	90.5%	84.8%	77.5%	4.54
`Xiaomi-Robotics-0` (Ours)	ABC→D	**100.0%**	**98.3%**	**96.0%**	**92.6%**	**88.1%**	**4.75**

Table 2 Results on the CALVIN benchmark.

Visual Matching and Variant Aggregation, respectively, as well as **79.2%** on WidowX evaluations, surpassing all compared baselines. This consistently strong performance demonstrates robust visual generalization, especially given the substantial visual gap between the real-world training data and the simulated evaluation environments.

3.2 Real-Robot Experiments

3.2.1 Evaluation Details

To validate real-world performance, we perform experiments with a bimanual robot equipped with two 6-DoF robotic arms (Fig. 6). In total, we use three cameras for observation: two wrist-mounted cameras for close-up views and one external camera for a global view. We evaluate our method on two representative tasks:

- **Lego Disassembly** (Fig. 6(a)): The robot is tasked with 1) disassembling Lego structures into individual bricks, and 2) sorting each brick into the corresponding storage bins according to its color. This task requires precise and coordinated bimanual grasping under contact, followed by accurate placement of individual bricks.
- **Towel Folding** (Fig. 6(b)): In towel folding, the robot needs to 1) pick out a towel from a tray, 2) flatten the towel, 3) fold the towel in half twice, and 4) place the towel to a staging area. This task is challenging because towels are deformable and exhibit complex, partially observable dynamics (*e.g.*, wrinkles and occlusions), requiring accurate and coordinated bimanual grasping, and continuous shape control throughout the long-horizon folding sequence.

For Lego Disassembly, we evaluate under two settings: large-assembly (LA) and multi-assembly (MA), as illustrated in Fig. 6(a). The LA setting evaluates the model's ability to handle increasing complexity within a single structure. It includes three sizes, *i.e.*, LA-5, LA-10, and LA-20, which comprises 5, 10, and 20 bricks, respectively. For each size, we evaluate three different assembly configurations and run three trials per configuration. The MA setting consists of 34 bricks in total, including both single bricks and groups of two

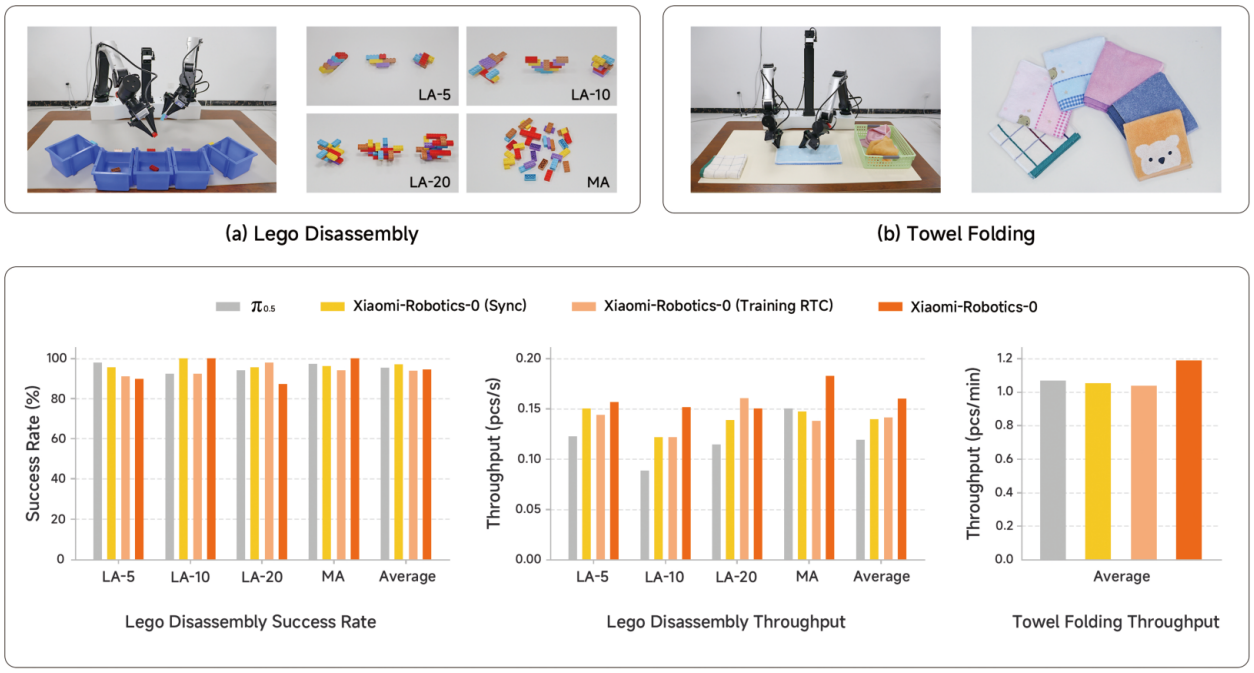


Figure 6 Real-Robot Experiments. (a) We show the setting for Lego Disassembly evaluation. (b) We show the setting for Towel Folding evaluation and all the towels used during evaluation. (c) Quantitative results of different methods on the two tasks.

or three bricks assembled together. We evaluate three trials for this setting. We report the average success rate, defined as the ratio of correctly sorted bricks to the total number of bricks, as well as the throughput, computed as the number of correctly sorted bricks divided by the total rollout time. For Towel Folding, we evaluate the policy using six different towels, as shown in Fig. 6(b). We perform two continuous 30-minute rollouts for each method. During evaluation, if a single folding attempt exceeds a 2-minute threshold, it is considered a failure. We evaluate the performance by reporting the throughput, calculated as the total number of successfully folded towels divided by the rollout duration.

3.2.2 Implementation Details

We compare our method with a state-of-the-art baseline and several ablation variants:

- $\pi_{0.5}$ [19]: A state-of-the-art VLA baseline. We follow the official OpenPi[1] fine-tuning protocol and adapt the released base model to our specific tasks. We use identical training settings as in our experiments.
- `Xiaomi-Robotics-0`: Our main method, which incorporates asynchronous execution during post-training to achieve smooth and responsive real-time execution.
- `Xiaomi-Robotics-0` (Sync): A synchronous variant of our model for quantifying the performance gains from asynchronous execution.
- `Xiaomi-Robotics-0` (Training RTC): The baseline asynchronous variant that leverages training RTC [5] during post-training.

We follow the pre-training and post-training procedures described in Sec. 2. We pre-train the model for 40k steps with a batch size of 32,768. We post-train the model with a batch size of 2,048 for 40k steps on Lego Disassembly and 80k steps on Towel Folding. We use AdamW [41] as the optimizer and DeepSpeed ZeRO-2 for training. We set the action chunk length T, corresponding to 1 second of actions.

[1] https://github.com/Physical-Intelligence/openpi

3.2.3 Results

Experiment results are summarized in Fig. 6(c). In Lego Disassembly, all methods are comparable in terms of average success rates, with the two synchronous methods—$\pi_{0.5}$ and `Xiaomi-Robotics-0` (Sync)—performing slightly better than the other two asynchronous counterparts. This is because asynchronous methods are less reactive in motion, leading to less precise grasps and high tension between bricks and the gripper fingers, which can cause bricks to eject away from the workspace. In terms of throughput, `Xiaomi-Robotics-0` (Sync) surpasses $\pi_{0.5}$, indicating that our method is more efficient in this task. `Xiaomi-Robotics-0` achieves the highest throughput among all methods, surpassing the training RTC variant which is also deployed asynchronously. This shows that our proposed post-training techniques are effective in improving execution efficiency in this task that requires high precision.

For Towel Folding, $\pi_{0.5}$, `Xiaomi-Robotics-0` (Sync), and `Xiaomi-Robotics-0` (Training RTC) achieve comparable throughputs of 1 pcs/min. `Xiaomi-Robotics-0` outperforms these three methods, achieving a throughput of 1.2 pcs/min. These results demonstrate that our method enables fast and robust performance in the challenging deformable object manipulation. The Training RTC variant often gets stuck when it inadvertantly grasps multiple layers of the towel during the flinging motion, preventing the motion from flattening the towel. Rather than re-grasping to correct this, the policy falls into a repetitive loop, repeatedly executing the flinging motion. This observation suggests that the action-prefixing mechanism introduces a shortcut in policy learning, allowing later-timestep action predictions to simply copy the prefixed actions rather than attend to signals from other modalities. In contrast, `Xiaomi-Robotics-0` are able to effectively avoid such repetitive failures.

3.3 Preservation of Vision-Language Capabilities

During pre-training, we jointly train `Xiaomi-Robotics-0` on both vision-language data and robot trajectory data (Sec. 2.2). This enables the model to avoid catastrophic forgetting of the vision-language capabilities of the underlying pre-trained VLM and enhance visual perception on robot-centric data. To validate this, we evaluate the VLM of `Xiaomi-Robotics-0` after pre-training on a comprehensive suite of vision-language (VL) benchmarks [14, 22, 31, 40, 42, 47, 61, 63, 72, 73], covering tasks from general QA to hallucination detection. We further report results on ERQA [63], a benchmark designed for evaluating embodied reasoning capabilities. We compare with two state-of-the-art vision-language-action (VLA) models, $\pi_{0.5}$ [19] and MolmoAct [26], which also incorporate vision-language data during training. We also include comparison with VLA models that *do not* incorporate VL data during training to understand how the VL capabilities degrade in the absence of the corresponding data. Specifically, we compare with π_0 [3] and a variant of our method which removes VL data during pre-training (denoted as `Xiaomi-Robotics-0` (w/o VL data)).

Model	ERQA	SEED	POPE	AI2D	MMBench	MME	MMMU	TextVQA	SciQA	ChartQA
π_0 [3]	0.0	0.0	0.0	0.0	0.0	0.1	0.1	1.4	0.0	0.0
$\pi_{0.5}$ [19]	0.0	21.5	0.0	14.4	22.1	0.0	19.9	0.0	28.0	0.5
MolmoAct [26]	33.5	72.7	86.6	72.0	80.1	69.5	38.0	67.3	**91.1**	57.1
`Xiaomi-Robotics-0` (w/o VL data)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
`Xiaomi-Robotics-0` (Ours)	**40.8**	**78.6**	**88.5**	**78.7**	**84.4**	**81.8**	**46.2**	**72.0**	79.4	**59.2**
Qwen3-VL-4B-Instruct [1]	40.0	78.8	89.7	81.6	88.7	87.1	51.7	78.0	92.7	76.8

Table 3 Quantitative results on general vision-language and embodied reasoning benchmarks. See Appendix C for detailed definitions and evaluation metrics for each benchmark.

Results are shown in Tab. 3. `Xiaomi-Robotics-0` outperforms all the comparing VLA baselines in all but one benchmarks. It is able to effectively preserve the vision-language capabilities of the underlying pre-trained VLM, trailing slightly behind it on most general VL benchmarks. It showcases strong performance in the challenging object hallucination evaluations (the POPE series) and OCR-related tasks (*e.g.*, AI2D). Surprisingly, `Xiaomi-Robotics-0` slightly surpasses Qwen3-VL-4B-Instruct on the ERQA benchmark (40.8 v.s. 40.0). We hypothesize that this gain stems from incorporating vision-language data derived from robot trajectory data into the training mixture, which strengthens visual perception on robot-centric inputs. π_0 achieves near-zero performance on most VL tasks, while `Xiaomi-Robotics-0` (w/o VL data) attains zero

performance across all tasks. These results indicate that without explicit vision-language supervision, training on robot trajectories alone fails to retain the general-purpose VL knowledge, leading to severe catastrophic forgetting. We provide additional qualitative results of `Xiaomi-Robotics-0` on different evaluation benchmarks in App. A.

4 Related Work

Recently, vision-language-action (VLA) models have emerged as a new paradigm for robot policy learning [3, 6, 8, 10, 19, 24, 25, 29, 45, 50, 75]. By leveraging large scale of robot data collected across diverse embodiments, tasks, and environments [7, 12, 23, 66], VLA models can effectively solve a broad range of tasks and showcase strong generalization capabilities on handling various kinds of out-of-distribution settings, including novel environments, instructions and objects [10, 19, 75]. Typically, VLAs are built upon pre-trained vision-language models (VLMs) that have been trained to capture broad visual semantic knowledge. A straightforward approach is to convert actions to discretized tokens and train the VLM to generate actions via a next-token-prediction objective [24, 50, 75]. However, action tokenization can introduce quantization error and reduce control precision. Another effective method of modeling the complex trajectory distribution is to leverage the expressive power of flow matching [3, 10, 19, 45, 52] or diffusion [11, 36–38, 67]. To effectively transfer knowledge from the pre-trained VLMs to VLA models, it is crucial to preserve the VLMs' capabilities throughout VLA training. One simple method is to jointly train on both vision-language (VL) data and robot trajectories [19, 26, 75]. Recent work [13] further proposes detaching the flow-matching objective from the VLM backbone, thereby preventing gradient backpropagation into it. We build `Xiaomi-Robotics-0` by integrating a pre-trained VLM [1] with a diffusion transformer [48] that generates actions via flow matching. During pre-training, we jointly train on both robot trajectories and VL data in the first stage and freeze the VLM while training the DiT in the second stage to avoid catastrophic forgetting of the vision-language knowledge.

Given the large number of parameters—often up to billions—the inference latency of large VLA models is non-negligible during real-robot rollouts [4]. A simple execution strategy is to roll out policies synchronously [3], where the robot remains idle until the next inference completes, causing pauses and discontinuous actions. Another method is to accelerate inference to achieve real-time performance [43]. Recent work explores asynchronous execution, where the robot continues executing while the model performs inference [3, 19, 62]. A line of work proposes to prefix previously generated and committed actions in the prediction of the next action chunk. Real-Time Chunking (RTC) [4] leverages a training-free inpainting algorithm that "freezes" the prefixed action and "inpaints" the rest in a way that is consistent with the frozen prefix. Training RTC [5] incorporates the prefixed actions during training. However, conditioning on prefixed actions during training allows later-timestep predictions to leverage the shortcut of exploiting temporal correlations between consecutive actions, resulting in less reactive behavior. In this work, we propose several practical techniques to address this issue, achieving high throughput on challenging tasks that require precise and dexterous manipulation.

5 Conclusions

We introduce `Xiaomi-Robotics-0`, a powerful vision-language-action (VLA) model designed for both high performance and fast and smooth real-time execution. `Xiaomi-Robotics-0` is pre-trained on large-scale robot trajectories and vision-language data, enabling broad and generalizable action generation while preventing catastrophic forgetting of the visual-semantic knowledge in the underlying pre-trained VLM. In post-training, we develop several practical techniques to train the VLA model for asynchronous execution, allowing continuous and reactive real-time execution on real robots. We validate `Xiaomi-Robotics-0` on extensive simulation benchmarks and two real-robot tasks requiring precise, dexterous bimanual manipulation. Results showcase that the proposed method delivers state-of-the-art performance on all simulation benchmarks. In addition, `Xiaomi-Robotics-0` runs fast and smoothly on real robots with a consumer-grade GPU, achieving high success rates and strong throughput in both real-world tasks. In the future, we plan to explore training the model on larger and more diverse robot datasets and continue to improve its robustness and generalization capabilities in real-world tasks.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. arXiv preprint arXiv:2511.21631, 2025.

[2] 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.

[3] 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. arXiv preprint arXiv:2410.24164, 2024.

[4] Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. arXiv preprint arXiv:2506.07339, 2025.

[5] Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. arXiv preprint arXiv:2512.05964, 2025.

[6] 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.

[7] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. AgiBot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv preprint arXiv:2503.06669, 2025.

[8] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. arXiv preprint arXiv:2505.06111, 2025.

[9] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In CVPR, 2021.

[10] Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, et al. GR-3 technical report. arXiv preprint arXiv:2507.15493, 2025.

[11] 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, 44(10-11):1684–1704, 2025.

[12] 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, Muhammad Zubair Irshad, 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, Vitor Guizilini, 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. https://arxiv.org/abs/2310.08864, 2023.

[13] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. arXiv preprint arXiv:2505.23705, 2025.

[14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. In The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2025.

[15] Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. LLMDet: Learning strong open-vocabulary object detectors under the supervision of large language models. arXiv preprint arXiv:2501.18954, 2025.

[16] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Zero-shot extreme length generalization for large language models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3991–4008, 2024.

[17] 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.

[18] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. ThinkAct: Vision-language-action reasoning via reinforced visual latent planning. arXiv preprint arXiv:2507.16815, 2025.

[19] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. arXiv preprint arXiv:2504.16054, 2025.

[20] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. MInference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. Advances in Neural Information Processing Systems, 37:52481–52515, 2024.

[21] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3d scene representations. arXiv preprint arXiv:2402.10885, 2024.

[22] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In European conference on computer vision, pages 235–251. Springer, 2016.

[23] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. arXiv preprint arXiv:2403.12945, 2024.

[24] 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.

[25] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645, 2025.

[26] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. MolmoAct: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917, 2025.

[27] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. GR-MG: Leveraging partially annotated data via multi-modal goal-conditioned policy. IEEE Robotics and Automation Letters, 2025.

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

[29] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. arXiv preprint arXiv:2412.14058, 2024.

[30] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941, 2024.

[31] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL https://openreview.net/forum?id=xozJw0kZXF.

[32] 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.

[33] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. arXiv preprint arXiv:2508.20072, 2025.

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.

[35] 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, 2023.

[36] 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.

[37] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. arXiv preprint arXiv:2410.07864, 2024.

[38] Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, Xiao Ouyang, Hang Su, and Jun Zhu. RDT2: Exploring the scaling limit of umi data towards zero-shot cross-embodiment generalization. arXiv preprint arXiv:2602.03310, 2026.

[39] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003, 2022.

[40] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In European conference on computer vision, pages 216–233. Springer, 2024.

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[42] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In The 36th Conference on Neural Information Processing Systems (NeurIPS), 2022.

[43] Yunchao Ma, Yizhuang Zhou, Yunhuan Yang, Tiancai Wang, and Haoqiang Fan. Running vlas at real-time speed. arXiv preprint arXiv:2510.26742, 2025.

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

[45] NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, 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. In ArXiv Preprint, March 2025.

[46] 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.

[47] Abdelghny Orogat, Isabelle Liu, and Ahmed El-Roby. CBench: Towards Better Evaluation of Question Answering Over Knowledge Graphs. Proceedings of the VLDB Endowment (PVLDB), 14(8), 2021.

[48] William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4195–4205, 2023.

[49] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[50] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747, 2025.

[51] Haozhi Qi, Yen-Jen Wang, Toru Lin, Brent Yi, Yi Ma, Koushil Sreenath, and Jitendra Malik. Coordinated humanoid manipulation with choice policies. arXiv preprint arXiv:2512.25072, 2025.

[52] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. EO-1: Interleaved vision-text-action pretraining for general robot control. arXiv preprint arXiv:2508.21112, 2025.

[53] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830, 2025.

[54] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding DINO 1.5: Advance the "edge" of open-set object detection, 2024.

[55] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling open-world models for diverse visual tasks, 2024.

[56] Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. arXiv preprint arXiv:2412.12953, 2024.

[57] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. arXiv preprint arXiv:2407.05996, 2024.

[58] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-action flow policies. arXiv preprint arXiv:2509.04996, 2025.

[59] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Proceedings of ACL, 2018.

[60] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. arXiv preprint arXiv:2508.19236, 2025.

[61] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8317–8326, 2019.

[62] Jiaming Tang, Yufei Sun, Yilong Zhao, Shang Yang, Yujun Lin, Zhuoyang Zhang, James Hou, Yao Lu, Zhijian Liu, and Song Han. VLASH: Real-time vlas via future-state-aware asynchronous inference. arXiv preprint arXiv:2512.01031, 2025.

[63] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini Robotics: Bringing ai into the physical world. arXiv preprint arXiv:2503.20020, 2025.

[64] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. arXiv preprint arXiv:2412.15109, 2024.

[65] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms, 2024. URL https://arxiv.org/abs/2406.16860.

[66] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData v2: A dataset for robot learning at scale. In Conference on Robot Learning, pages 1723–1736. PMLR, 2023.

[67] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-language model with plug-in diffusion expert for general robot control. arXiv preprint arXiv:2502.05855, 2025.

[68] Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. FineVision: Open data is all you need, 2025. URL https://arxiv.org/abs/2510.17269.

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

[70] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453, 2023.

[71] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 14203–14214, 2025.

[72] Jie Ying, Zihong Chen, Zhefan Wang, Wanli Jiang, Chenyang Wang, Zhonghang Yuan, Haoyang Su, Huanjun Kong, Fan Yang, and Nanqing Dong. SeedBench: A multi-task benchmark for evaluating large language models in seed science. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31395–31449, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.1516/.

[73] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In Proceedings of CVPR, 2024.

[74] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705, 2023.

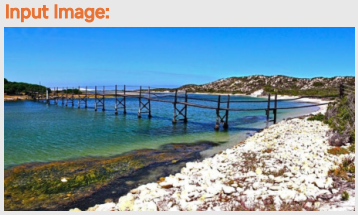
[75] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning, pages 2165–2183. PMLR, 2023.

Appendix

A Case Study for Preservation of Vision-Language Capabilities

Dense Caption

Input Image:



User:
Provide a detailed description of everything in the image.

Xiaomi-Robotics-0:
A weathered wooden jetty extends across the turquoise waters of a sheltered cove, its aged planks and rusted metal supports anchored in the clear, shallow water that reveals patches of submerged vegetation and rocky seabed. The foreground features a rugged shoreline of white limestone rocks interspersed with sparse, drought-resistant shrubs, while the jetty leads the eye toward a distant sandy beach bordered by low-lying, scrub-covered hills under a cloudless azure sky. The calm water transitions from emerald near the shore to deeper sapphire hues farther out, reflecting the bright sunlight that illuminates the arid, undulating landscape beyond the cove.

Attribute Recognition

Input Image:

User:
What color is the bird in the image?

Xiaomi-Robotics-0:
The bird in the image has a combination of colors: its upper parts (back, wings, and head) are brownish-gray, while its underparts (chest and belly) are white. Additionally, it has a dark (black) beak and greenish legs.

Fine-grained Classification

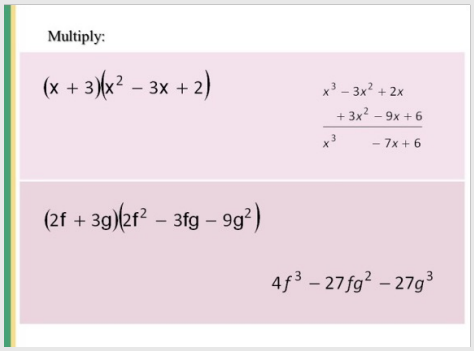
Input Image:

User:
Provide a detailed description of everything in the image.

Xiaomi-Robotics-0:
The man is holding a food item that appears to be a **corn dog** or a similar fried snack, which is consistent with the food offerings advertised by the "Charlie's Italian Sausage" stand (e.g., "CORN DOGS" is listed on the stand's signage).

OCR

Input Image:



User:
Could you please depict the text and formula in the picture?

Xiaomi-Robotics-0:
Two polynomial multiplication problems are shown, each with step-by-step solutions. The first multiplies $(x+3)$ by (x^2-3x+2), distributing to obtain x^3-3x^2+2x, then adding $3x^2-9x+6$ to yield the simplified result x^3-7x+6. The second multiplies $2f+3g$ by $2f^2-3fg-9g^2$, distributing to form $4f^3-27fg^2-27g^3$, with no intermediate steps shown. Both problems are presented in a structured format with the title "Multiply:" above the equations.

Dense Captioning

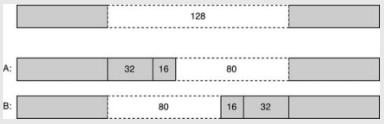
Input Image:

User:
What could be a potential reason for the sunglasses placed on the table?

Xiaomi-Robotics-0:
A weathered wooden jetty extends across the turquoise waters of a sheltered cove, its aged planks and rusted metal supports anchored in the clear, shallow water that reveals patches of submerged vegetation and rocky seabed. The foreground features a rugged shoreline of white limestone rocks interspersed with sparse, drought-resistant shrubs, while the jetty leads the eye toward a distant sandy beach bordered by low-lying, scrub-covered hills under a cloudless azure sky. The calm water transitions from emerald near the shore to deeper sapphire hues farther out, reflecting the bright sunlight that illuminates the arid, undulating landscape beyond the cove.

Chart Understanding

Input Image:



User:
In Diagram B, what is the ratio of the largest segment to the smallest segment? A) 5:1 B) 2:1 C) 4:1 D) 1:1

Xiaomi-Robotics-0:
The largest segment in Diagram B is 80, and the smallest segment is 16. The ratio of the largest to the smallest is: $\frac{80}{16}=5$. Thus, the ratio is 5:1.

Answer: A

Figure 7 Qualitative results of `Xiaomi-Robotics-0` on vision-language tasks (I).

Grounding

Input Image:

User:
What are the bounding boxes for vessels?

Xiaomi-Robotics-0:
The vessels are found in the following regions:
- (65, 302, 905, 582),
- (425, 658, 555, 708),
- (712, 692, 820, 742).

Embodied Grounding with Reasoning

Input Image:

User:
In this frame, on which thing can I find the knives? Point out its position in form of bounding box.

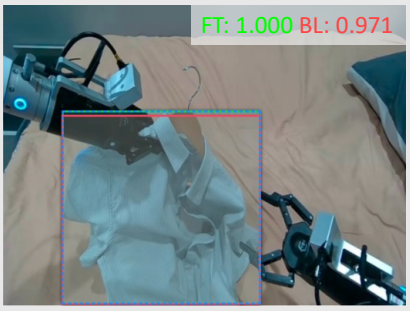
Xiaomi-Robotics-0:
On the cutting board, which is ... at (485, 308, 667, 397).

Baseline (Qwen3-VL-4B):
There are no knives in this frame. The image shows a kitchen scene with a cutting board on the counter, but no visible knives are present.

Ground Truth:
(487.0, 306.0, 665.0, 395.0)

Embodied Grounding

Input Image:



User:
In the current observation, what is the position of shirt?

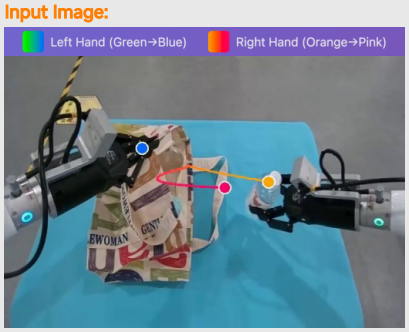
Xiaomi-Robotics-0:
The shirt is ... at (146, 356, 640, 997).

Baseline (Qwen3-VL-4B):
(144, 370, 640, 999)

Ground Truth:
(146.0, 356.0, 640.0, 997.0)

Trajectory Prediction

Input Image:



User:
For the task 'Place the picked-up small bottle mineral water into the colorful canvas bag on the desktop with the right arm.', output synchronized left and right arm paths in normalized image coordinates (0–1000). Format: left: ...; right: ...

Xiaomi-Robotics-0:
left: (350, 419), (350, 419), (350, 419), (350, 419), (350, 419); right: (634, 560), (400, 510), (400, 510), (400, 510), (541, 480).

Embodied Grounding

Input Image:

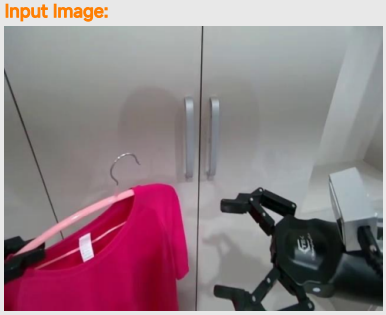
User:
Where is the red bell pepper in the image? Please provide the bounding box in format (x1, y1, x2, y2).

Xiaomi-Robotics-0:
The red bell pepper is ... at (171, 450, 321, 638).

Ground Truth:
(180, 457, 344, 598)

Embodied Planning

Input Image:



User:
Provide a detailed plan to accomplish:
Open the wardrobe and hang the clothes.

Xiaomi-Robotics-0:
['Grip the wardrobe door handle with the right arm.', 'Open the wardrobe with the right arm.', 'Advance towards the open wardrobe with the clothes hanger held in the left arm.', 'Hang the held clothes in the wardrobe with the left arm.', 'Step back a bit away from the wardrobe.', 'Turn to the open wardrobe door.', 'Close the wardrobe door with the right arm.']

Figure 8 Qualitative results of `Xiaomi-Robotics-0` on vision-language tasks (II). "..." indicates omitted content for space constraints.

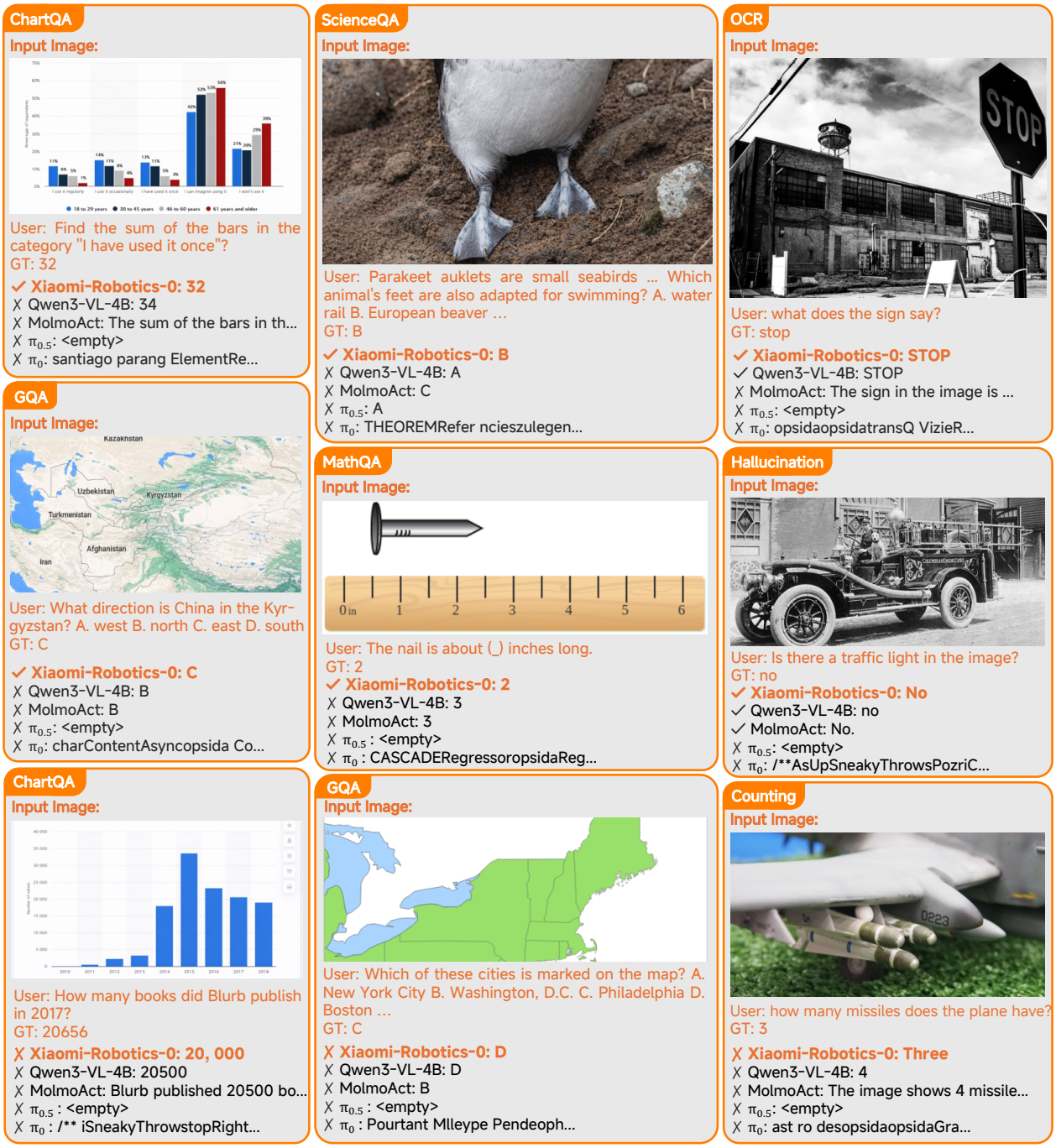


Figure 9 Qualitative comparison of `Xiaomi-Robotics-0` against baseline methods. In particular, the bottom row illustrates challenging failure cases, highlighting limitations in complex numerical reasoning on dense charts and minor format-following errors (*e.g.*, outputting words instead of digits in counting tasks).

B Detailed results on SimplerEnv

WidowX	Put Spoon on Towel	Put Carrot on Plate	Stack Blocks	Put Eggplant in Basket	Overall
RT-1-X [12]	0%	4.2%	0%	0%	1.1%
OpenVLA [24]	0%	0%	0%	4.1%	1.0%
Octo-Base [46]	12.5%	8.3%	0%	43.1%	16.0%
Octo-Small [46]	47.2%	9.7%	4.2%	56.9%	29.5%
Magma [71]	37.5%	29.2%	20.8%	91.7%	44.8%
RoboVLMs [29]	45.8%	20.8%	4.2%	79.2%	37.5%
π_0 [3]	83.8%	52.5%	52.5%	87.9%	69.2%
π_0-FAST [50]	29.1%	21.9%	10.8%	66.6%	32.1%
SpatialVLA [53]	16.7%	25.0%	29.2%	**100**%	42.7%
ThinkAct [18]	58.3%	37.5%	8.7%	70.8%	43.8%
EO-1 [52]	63.6%	54.5%	**81.8**%	90.9%	72.7%
`Xiaomi-Robotics-0` (Ours)	**95.8**%	**62.5**%	75.0%	83.3%	**79.2**%

Table 4 Results on the WidowX evaluations of SimplerEnv.

Visual Matching	Pick Coke Can	Move Near	Open/Close Drawer	Drawer Apple	Overall
Octo-Base [46]	17.0%	4.2%	22.7%	0%	11.0%
OpenVLA [24]	16.3%	46.2%	35.6%	0%	24.5%
RT-1 [6]	85.7%	44.2%	73.0%	6.5%	52.4%
RT-1-X [12]	56.7%	31.7%	59.7%	40.7%	47.2%
RT-2-X [12]	78.7%	77.9%	25.0%	7.4%	47.3%
Magma [71]	75.0%	53.0%	58.9%	8.3%	48.8%
RoboVLMs [29]	77.3%	61.7%	43.5%	24.1%	51.7%
SpatialVLA [53]	86.0%	77.9%	57.4%	0%	55.3%
π_0 [3]	97.9%	78.7%	62.3%	46.6%	71.4%
π_0-FAST [50]	75.3%	67.5%	42.9%	0%	46.4%
ThinkAct [18]	92.0%	72.4%	50.0%	-	-
MolmoAct [26]	77.7%	77.1%	60.0%	-	-
EO-1 [52]	98.0%	83.8%	71.3%	52.8%	76.5%
`Xiaomi-Robotics-0` (Ours)	**98.7**%	**88.8**%	**79.6**%	**75.0**%	**85.5**%
Visual Aggregation	**Pick Coke Can**	**Move Near**	**Open/Close Drawer**	**Drawer Apple**	**Overall**
Octo-Base [46]	0.6%	3.1%	1.1%	0%	1.2%
OpenVLA [24]	54.5%	47.7%	17.7%	0.0%	30.0%
RT-1 [6]	89.8%	50.0%	32.3%	2.6%	43.7%
RT-1-X [12]	49.0%	32.3%	29.4%	10.1%	30.2%
RT-2-X [12]	82.3%	79.2%	35.3%	20.6%	54.4%
Magma [71]	68.6%	78.5%	59.0%	24.0%	57.5%
RoboVLMs [29]	75.6%	60.0%	10.6%	0%	36.6%
π_0 [3]	90.1%	80.7%	27.6%	20.5%	54.7%
π_0-FAST [50]	77.6%	68.2%	31.3%	0%	44.3%
SpatialVLA [53]	88.0%	72.7%	41.8%	6.3%	52.2%
ThinkAct [18]	84.0%	63.8%	47.6%	-	-
MolmoAct [26]	76.1%	61.3%	**78.8**%	-	-
EO-1 [52]	**91.6**%	**81.7**%	55.0%	23.8%	63.0%
`Xiaomi-Robotics-0` (Ours)	88.2%	76.8%	67.2%	**66.7**%	**74.7**%

Table 5 Results on the Google Robot evaluations of SimplerEnv.

C VLM Benchmark Details

Benchmark	Samples	Evaluation Focus
Comprehensive Multi-modal Capabilities		
MMBench [40]	4,329	Comprehensive VLM ability with circular evaluation
SEED-Bench [72]	14,233	Fine-grained image-text understanding & spatial relations
MME [14]	2,374	Holistic perception and cognition evaluation
Object Hallucination Evaluation		
POPE [31]	9,000	Object existence polling (Random/Popular/Adversarial)
Reasoning and Expert Knowledge		
ERQA [63]	400	Embodied AI reasoning in physical scenarios
ScienceQA [42]	2,017	Scientific question answering with chain-of-thought
MMMU [73]	900	Multi-discipline expert-level reasoning
Fine-grained Visual Perception		
AI2D [22]	3,088	Scientific diagram structure understanding
ChartQA [47]	2,500	Data visualization interpretation
TextVQA [61]	1,731	OCR-based reasoning in natural scenes
Total	**≈ 40,500**	**Diverse Generalization Assessment**

Table 6 Summary of Vision-Language Benchmarks. We select a diverse set of 10 benchmarks covering comprehensive capabilities, hallucination, reasoning, and fine-grained perception to evaluate the vision-language capabilities of our model.