ANT GROUP

A Pragmatic VLA Foundation Model

Wei Wu*, Fan Lu*, Yunnan Wang*, Shuai Yang*, Shi Liu*, Fangjing Wang*, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, Kecheng Zheng†

*Equal Contribution †Project Lead

Offering great potential in robotic manipulation, a capable Vision-Language-Action (VLA) foundation model is expected to faithfully generalize across tasks and platforms while ensuring cost efficiency (*e.g.*, data and GPU hours required for adaptation). To this end, we develop `LingBot-VLA` with around 20,000 hours of real-world data from 9 popular dual-arm robot configurations. Through a systematic assessment on 3 robotic platforms, each completing 100 tasks with 130 post-training episodes per task, our model achieves clear superiority over competitors, showcasing its ***strong performance*** and ***broad generalizability***. We have also built an ***efficient*** codebase, which delivers a throughput of 261 samples per second per GPU with an 8-GPU training setup, representing a $1.5 \sim 2.8\times$ (depending on the relied VLM base model) speedup over existing VLA-oriented codebases. The above features ensure that our model is well-suited for real-world deployment. To advance the field of robot learning, we provide open access to the code, base model, and benchmark data, with a focus on enabling more challenging tasks and promoting sound evaluation standards.

Website: https://technology.robbyant.com/lingbot-vla
Github: https://github.com/robbyant/lingbot-vla
Checkpoints: https://huggingface.co/robbyant/lingbot-vla

Robbyant

1 Introduction

Vision-Language-Action (VLA) foundation models [5, 6, 27] have emerged as a promising method for enabling robots to perform diverse manipulation tasks guided by natural language instructions. Through large-scale pre-training, these models acquire generalizable skills that can be rapidly adapted to diverse tasks and robotic platforms. Despite the significant progress, there remains a lack of comprehensive empirical studies on how real-robot performance scales with increasingly vast pre-training datasets. Moreover, the community lacks a highly optimized training codebase capable of efficiently conducting these scaling evaluations on massive volumes of data. Consequently, a fundamental question that demands investigation in the real-world setting is: *How do VLA models truly scale with massive real-world robot data?*

Understanding the scaling behavior of VLA models is crucial for robotic learning, especially on vast and diverse real-world datasets. In this work, we provide a systematic empirical investigation into how success rates scale with respect to data volume and diversity during VLA pre-training By scaling pre-training data from 3,000 hours to 20,000 hours, we demonstrate that downstream success rates improve consistently and substantially. Notably, this scaling behavior shows no signs of saturation even at the 20,000-hour mark, suggesting that VLA performance continues to benefit from increased data volume. These results provide the first empirical evidence of favorable scaling properties in real-world robot learning, offering critical insights for future VLA development and large-scale data curation.

While scaling analysis reveals favorable performance trends, translating these insights into reliable, deployable systems necessitates rigorous evaluation on real robotic platforms at a large scale. Thanks to GM-100 [29], which provides 100 carefully designed tasks, we conduct a systematic assessment across 3 robotic platforms, involving 130 episodes per task per embodiment. By emphasizing task diversity and multi-platform consistency, our evaluation framework provides a choice of new standards for sound VLA benchmarking.

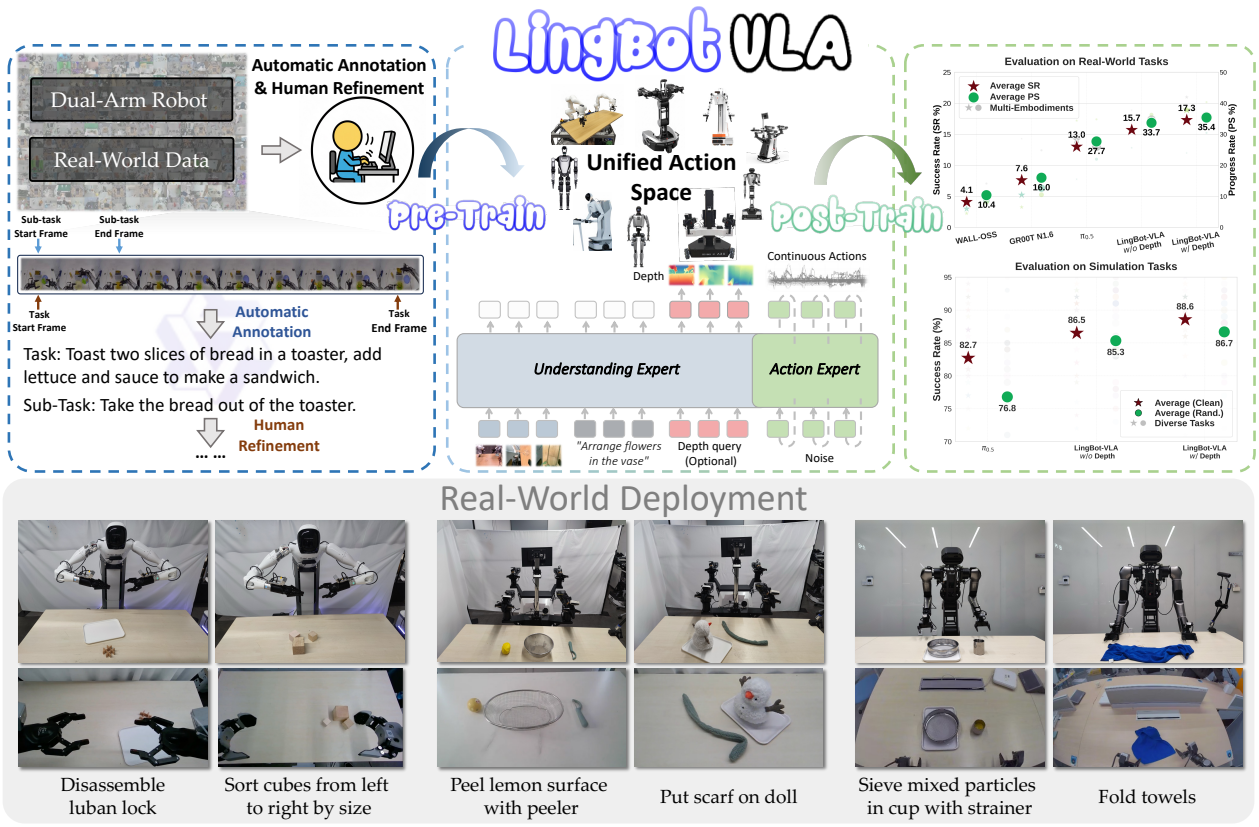


Figure 1. Overview of LingBot-VLA. We scale dual-arm robot data collected in the real world for pre-training. LingBot-VLA can be easily and efficiently transferred to downstream tasks. Moreover, we conduct a systematic assessment across three robotic embodiments, which demonstrates the clear superiority of our model.

In this paper, we present LingBot-VLA, a pragmatic VLA foundation model trained on about 20,000 hours of real-world manipulation data from 9 robotic platforms. Our systematic evaluation on the comprehensive benchmark, demonstrates that LingBot-VLA achieves state-of-the-art performance and exceptional generalization compared to existing methods. Beyond model capabilities, we emphasize that large-scale robot learning necessitates high computational efficiency. To this end, we have developed an optimized codebase that achieves a throughput of 261 samples per second per GPU on an 8-GPU cluster. This efficiency gain substantially shortens training cycles and reduces computational overhead, thereby lowering the overall costs. By combining superior performance, broad generalizability, and computational efficiency, LingBot-VLA is well-positioned for real-world robotic applications. To foster community progress, we provide open access to the code, base model, and benchmark data, with a focus on enabling more challenging tasks and promoting sound evaluation standards.

2 Related Work

2.1 Vision-Language-Action Models

Foundation VLA. Vision-language-action foundation models typically adopt a powerful pre-trained vision-language model [2, 3] as the semantic backbone, coupled with diffusion-based action head. Recent VLA foundation models [4–7, 13, 26, 27, 32, 33] have demonstrated enhanced multi-task execution capabilities and superior multi-embodiment adaptability, following pre-training on larger-scale and increasingly diverse datasets. Distinguishing itself from the datasets utilized in preceding VLA foundation models, our model is pre-trained on an extensive corpus approximate 20,000 hours of multi-embodiment data. This massive-scale dataset, characterized by its high behavioral diversity, significantly bolsters the model's generalization capabilities across various robotic manipulation tasks.

Spatial VLA. While traditional VLA models excel at semantic understanding, they often struggle with precise geometric reasoning and depth perception required for complex spatial manipulation. To address this, several works [9, 11, 14, 21, 23, 25, 32] have integrated spatial representations into the VLA framework. Several research [9, 26, 32] initiatives have focused on bolstering the spatial awareness of VLMs within embodied scenarios to enhance the spatial manipulation capabilities of VLAs in downstream tasks. Others explicitly or implicitly incorporate depth information during the VLA training phase. Spatial Forcing [14] employs a streamlined alignment strategy that compels the integration of VLA visual embeddings with spatial representations, thereby significantly improving the model's spatial comprehension.

2.2 Evaluation on Robot Policy

Current evaluation methodologies for robot policies are primarily bifurcated into two categories: simulation-based [8, 15, 17, 19, 20] and real-world embodiment-based [1, 31]. Simulation-based benchmark provide a rapid and convenient means to evaluate the capabilities of policies, enabling large-scale parallel testing across vast and diverse interaction scenarios at very low cost. Although simulation environments typically employ idealized physical models, their results often do not fully represent the complexity of the real physical world. The another real-world evaluations' efficiency is often bottlenecked by the requirement for extensive hardware parallelism. Consequently, the majority of prior VLA studies have been confined to comparing a limited number of methods across only a few tasks. To more comprehensively evaluate the real-world performance of policies, this work conducts an assessment across three distinct robotic platforms, with 100 tasks executed on each platform. We further provide a thorough analysis of how mainstream VLA models adapt to the diversity encountered in real-world scenarios.

2.3 Efficient VLA Training

The rapid iteration of VLA models has catalyzed the development of specialized training infrastructure. Several well-designed open-source codebases have recently emerged in the community, each catering to different research priorities. For instance, the OpenPI [6] repository provides a versatile framework supporting both JAX and PyTorch for training the π series models. StarVLA [22] introduces a modular and user-friendly codebase specifically optimized for the co-training of VLAs and VLMs, facilitating the transfer of semantic knowledge to robotic control. Additionally, Dexbotic [30] is designed as a unified and efficient solution to streamline the development lifecycle of VLAs, focusing on standardizing the pipeline from data ingestion to model deployment. Despite these advancements, training large-scale VLA models on multi-node clusters remains a significant challenge due to data I/O bottlenecks and communication overheads. To bridge this gap, we present `LingBot-VLA`, a high-performance open-source codebase engineered for large-scale VLA training. Unlike existing frameworks, our codebase implements systemic optimizations in data loading, distributed training strategies, and operator-level acceleration. These enhancements lead to a comprehensive improvement in training throughput and scalability, providing a more efficient foundation for the community to explore the scaling limits of robotic foundation models.

3 Pre-training Dataset

3.1 Data Collection

The pre-training dataset is built upon large-scale teleoperated data collected from 9 popular dual-arm robot embodiments, as shown in Fig. 2. We discuss these embodiments below:

- **AgiBot G1.** This setup has two 7-DoF arms with three RGB-D cameras. Robot data are collected via VR-based teleoperation on this setup.
- **AgileX.** This setup is equipped with three cameras and two 6-DoF arms. Robot control is achieved using isomorphic arms during the data collection process.
- **Galaxea R1Lite.** This setup has two 6-DoF arms, with one stereo camera and two wrist cameras.
- **Galaxea R1Pro.** Two 7-DoF arms, one stereo camera, and two wrist cameras are used in this setup.
- **Realman Rs-02.** This setup uses three cameras and features a 16-dimensional configuration and action space: two 7-DoF arms and two parallel grippers.

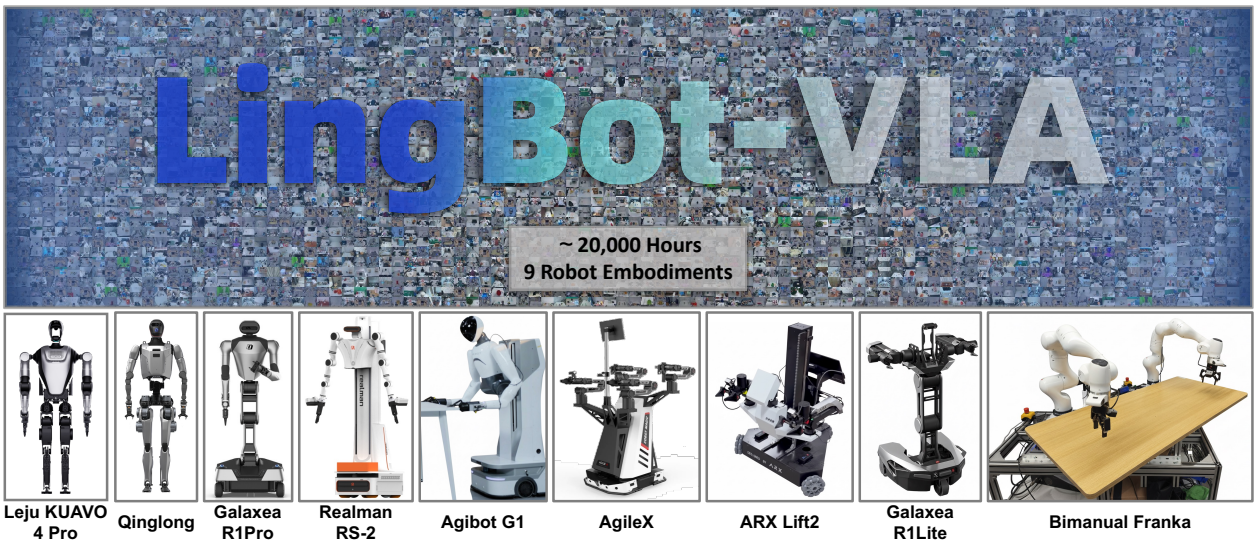


Figure 2. Visualization of pre-training dataset used by LingBot-VLA.

- **Leju KUAVO 4 Pro.** This setup features two 7-DoF arms, two parallel grippers, one camera on the head, and two cameras on the wrists.
- **Qinglong.** A humanoid robot with two 7-DoF arms and three cameras: one on the head and one on each wrist.
- **ARX Lift2.** This setup uses three cameras and two 6-DoF arms.
- **Bimanual Franka.** This setup uses two 7-DoF arms and two parallel grippers, forming a 16-dimensional action space, with three cameras.

3.2 Data Labeling

To obtain precise language instructions, we perform the following annotations: (1) *Video Segment*. Videos from multiple viewpoints, captured by robots, are jointly decomposed into clips by human annotators according to predefined atomic actions. Besides, to reduce the redundant information within videos, static frames at the start and end of the videos are eliminated at this stage. (2) *Instruction Annotation*. After obtaining videos containing the robots' full motion trajectories and video clips for each atomic action, we employ Qwen3-VL-235B-A22B [2] for precise annotation of task and sub-task instructions, as shown in Fig. 1.

4 Model Training

4.1 Architecture

To leverage well-trained vision-language representations, LingBot-VLA integrates the pre-trained VLM (*i.e.*, Qwen2.5-VL [2]) with an initialized action generation module called 'action expert'. These components are organized via a Mixture-of-Transformers (MoT) architecture like BAGEL [10], where vision-language and action modalities are processed through distinct transformer pathways, coupled by a shared self-attention mechanism for layer-wise unified sequence modeling. This MoT framework ensures that high-dimensional semantic priors from the VLM provide continuous guidance across all layers, while simultaneously mitigating cross-modal interference by maintaining modality-specific processing. The architecture of LingBot-VLA is illustrated in Fig. 1. Multi-view operational images and the related task instruction are uniformly encoded through a VLM to establish multimodal conditioning for subsequent action generation. Concurrently, the robot's proprioceptive sequences, specifically initial states and action chunks, are fed into the action expert for the prediction of action generation. We employ Flow Matching [16] for continuous action modeling, which facilitates fluid and smooth robotic control, ensuring high-precision execution across complex tasks and diverse robots.

In LingBot-VLA, the VLM and the action expert interact through a shared self-attention mechanism, facilitating a unified layer-wise representation. Consequently, the joint modeling sequence at timestamp t is formulated as the concatenation of the observation conditions $\mathbf{O}_t$ and the action chunk $\mathbf{A}_t$. Specifically, the observation context is defined as:

$$\mathbf{O}_t = [\mathbf{I}_t^1, \mathbf{I}_t^2, \mathbf{I}_t^3, \mathbf{T}_t, \mathbf{s}_t], \tag{1}$$

which incorporates tokens from three-view operational images $\mathbf{I}_t^{1,2,3}$ of dual-arm robots, the task instruction $\mathbf{T}_t$, and the robot state $\mathbf{s}_t$. The corresponding action sequence is denoted as:

$$\mathbf{A}_t = [\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+T-1}], \tag{2}$$

where T represents the action chunk length, *i.e.*, the temporal horizon of the predicted trajectory, which is set to 50 during our pre-training stage. Therefore, the training objective is to characterize the conditional distribution $p(\mathbf{A}_t|\mathbf{O}_t)$ through conditional flow matching. For a flow timestep $s \in [0, 1]$, we define a probability path through linear interpolation between the Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the ground-truth action $\mathbf{A}_t$, obtaining the intermediate action $\mathbf{A}_{t,s} = s\mathbf{A}_t + (1 - s)\epsilon$. The conditional distribution of $\mathbf{A}_{t,s}$ is formulated as:

$$p(\mathbf{A}_{t,s}|\mathbf{A}_t) = \mathcal{N}(s\mathbf{A}_t, (1 - s)\mathbf{I}). \tag{3}$$

The action expert v_θ is trained to predict the conditional vector field by minimizing the Flow Matching objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{s\sim\mathcal{U}[0,1],\mathbf{A}_t,\epsilon} \left\| v_\theta(\mathbf{A}_{t,s}, \mathbf{O}_t, s) - (\mathbf{A}_t - \epsilon) \right\|^2, \tag{4}$$

where the target velocity is given by the ideal vector field $\mathbf{A}_t - \epsilon$ derived from the linear probability path.

Following π_0 [6], we implement blockwise causal attention for modeling the joint sequence $[\mathbf{O}_t, \mathbf{A}_t]$. The sequence can be partitioned into three distinct functional blocks: $[\mathbf{I}_t^1, \mathbf{I}_t^2, \mathbf{I}_t^3, \mathbf{T}_t]$, $[\mathbf{s}_t]$ and $[\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+T-1}]$. A causal mask is applied among these blocks, such that tokens in each block can only attend to themselves and those in preceding blocks. Conversely, all tokens within the same block employ bidirectional attention and can attend to each other. This configuration ensures that the action expert can leverage all available observation knowledge, while preventing information leakage from future action tokens into the current observation representations.

To explicitly capture spatial awareness within manipulation environments and further enhance the robot's execution robustness, we adopt a vision distillation approach inspired by recent works [12, 28]. Specifically, we apply the learnable queries $[\mathbf{Q}_t^1, \mathbf{Q}_t^2, \mathbf{Q}_t^3]$ corresponding to three-view operational images. To integrate depth information, these queries are processed by VLM and then aligned with the depth tokens $[\mathbf{D}_t^1, \mathbf{D}_t^2, \mathbf{D}_t^3]$ from LingBot-Depth [24]. We align the VLM learnable queries and LingBot-Depth tokens by minimizing the distillation loss $\mathcal{L}_{distill}$:

$$\mathcal{L}_{distill} = \mathbb{E}_{\mathbf{Q}_t} \left| \text{Proj}(\mathbf{Q}_t) - \mathbf{D}_t \right|, \tag{5}$$

where $\text{Proj}(\cdot)$ is a projection layer that applies cross-attention for dimensional alignment. This integration infuses geometric information into the LingBot-VLA model, enabling precise perception for complex manipulation tasks.

4.2 Training Efficiency Optimization

Given that action data is inherently high-frequency, establishing a highly efficient pipeline encompassing distributed training and operator optimization is imperative. Our optimization methodology is structured as follows:

Distributed Strategy: While VLA models typically possess a moderate parameter count, achieving an optimal trade-off between GPU memory occupancy and training throughput remains essential. We employ Fully Sharded Data Parallel (FSDP)—a highly efficient PyTorch implementation of the Zero Redundancy Optimizer (ZeRO)—to shard optimizer states, model parameters, and gradients, thereby minimizing memory footprint. Drawing inspiration from the Hybrid Sharded Data Parallel (HSDP) approach proposed in VeOmni [18], we construct specific "shard groups" exclusively for the action expert modules. This strategy effectively mitigates the communication overhead associated with excessive parameter sharding. Additionally, we implement a mixed-precision policy: performing reductions in `torch.float32` to ensure numerical stability, while utilizing `torch.bfloat16` for storage and communication.

Operator-Level Optimization: The multimodal fusion of vision, language, and action within our architecture is fundamentally a sparse attention process. To address this, we leverage FlexAttention to optimize computation. Furthermore, we apply operator fusion (via `torch.compile`) to reduce kernel launch overhead and maximize memory bandwidth utilization.

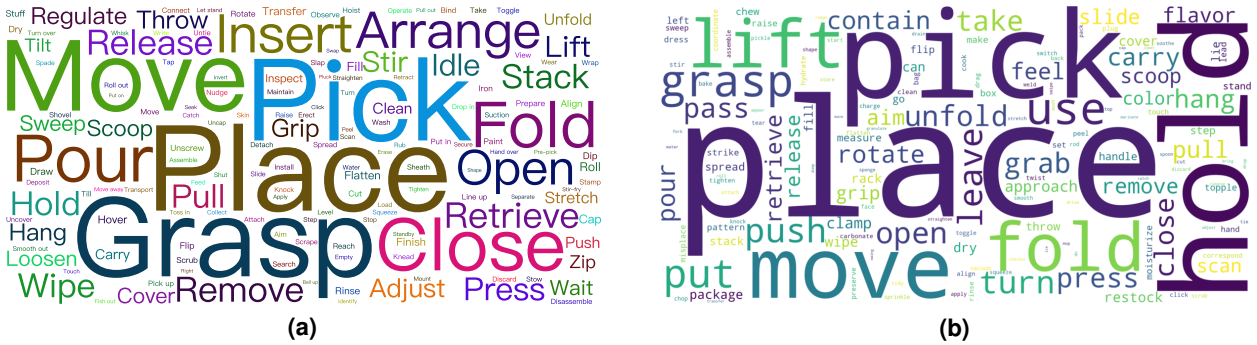


Figure 3. Word cloud of atomic actions in (a) Pre-training datasets and (b) Benchmark.

5 Experiments

5.1 Large-scale Real-world Benchmark

We conduct a large-scale empirical evaluation of LingBot-VLA designed to rigorously assess multi-embodiment generalization and real-world robustness. Our experimental framework comprises three core components: (1) 25 physical robots spanning 3 distinct commercial platforms, (2) GM-100 [29] benchmark featuring 100 diverse manipulation tasks with 39,000 expert demonstrations and (3) a controlled evaluation protocol generating 22,500 trials comparing LingBot-VLA against three state-of-the-art baselines under identical training and testing conditions.

5.1.1 Hardware Platforms

We conduct experiments across 3 distinct robotic platforms: AgileX, Agibot G1 and Galaxea R1Pro. All three embodiments feature a dual-arm configuration equipped with parallel-jaw grippers. To ensure robust perception, each robot is outfitted with multiple cameras: two wrist-mounted cameras and a head-mounted camera to capture an egocentric, human-eye perspective. All tasks are tabletop-based, with the embodiment's chassis and waist securely fixed in place.

5.1.2 Data Collection and Processing

For each GM-100 task [29], we collect expert demonstrations via teleoperation following a standardized protocol designed to ensure high data quality and environmental diversity. *Trajectory Volume:* 150 raw trajectories are collected per task across three platforms. The top 130, ranked by execution quality (task completion, motion smoothness, and protocol adherence), are retained for training. All trajectories strictly follow GM-100 task specifications. *Standardized Objects:* task objects are standardized and sourced according to GM-100 material specifications to ensure reproducibility across sites. *Environmental Diversity:* object poses (positions and orientations) are randomized within the workspace for each trajectory to prevent overfitting to specific spatial configurations and encourage learning of task-relevant invariances. *Teleoperation Guidelines:* (1) maintaining clearance between the end-effector and workspace surfaces to avoid collisions, (2) reducing velocity during object contact phases for smooth manipulation, and (3) ensuring distinct image observations at episode start and termination for reliable policy training. *Automated Filtering:* an algorithmic screening procedure automatically excludes episodes exhibiting technical anomalies. *Manual Review:* human reviewers validate the filtered dataset using synchronized multi-view video streams. Episodes are removed if they include extraneous objects or deviate from task protocols.

To analyze the semantic distribution and diversity of action categories, we visualized the most prevalent atomic actions in the training and testing sets using word clouds, as shown in Figs. 3a and 3b. Quantitative analysis reveals that approximately 50% of the atomic actions in the test set are absent from the top 100 most frequent training actions. This significant discrepancy underscores the diversity of our test set and ensures a rigorous assessment of the model's generalization capabilities.

Table 1. Experiment results of real-world evaluation on GM-100 [29] benchmark. 'SR' refers to success rate, and 'PS' refers to progress score.

Platform	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
Agibot G1	2.99%	8.75%	5.23%	12.63%	7.77%	21.98%	**12.82%**	30.04%	11.98%	**30.47%**
AgileX	2.26%	8.16%	3.26%	10.52%	17.20%	34.82%	15.50%	36.31%	**18.93%**	**40.36%**
Galaxea R1Pro	6.89%	14.13%	14.29%	24.83%	14.10%	26.14%	18.89%	34.71%	**20.98%**	**35.40%**
Average	4.05%	10.35%	7.59%	15.99%	13.02%	27.65%	15.74%	33.69%	**17.30%**	**35.41%**

Table 2. Experiment results of simulation evaluation on RoboTwin 2.0 [8] benchmark.

(a). Clean Scenes

	$\pi_{0.5}$	Ours *w/o* depth	Ours *w/* depth
Average SR	82.74%	86.50%	**88.56%**

(b). Randomized Scenes

	$\pi_{0.5}$	Ours *w/o* depth	Ours *w/* depth
Average SR	76.76%	85.34%	**86.68%**

5.1.3 Benchmarking and Evaluation Protocol

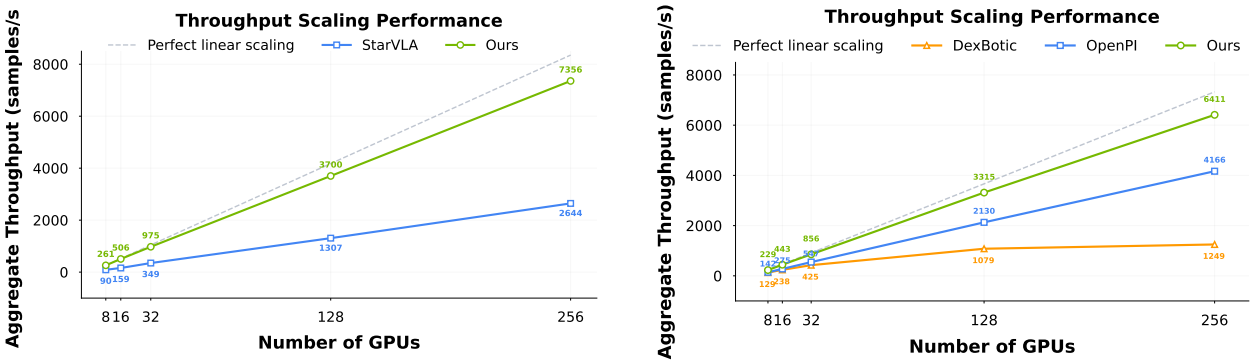
We systematically compare LingBot-VLA with three state-of-the-art VLA models: $\pi_{0.5}$, GR00T N1.6, and WALL-OSS, under strict experimental controls to isolate architectural performance. *Standardized Training:* All models are fine-tuned from publicly available pre-trained checkpoints using the same post-training pipeline. The verified dataset (130 filtered trajectories per task) and consistent hyperparameters (*i.e.*, batch sizes=256, epochs=20) are applied to ensure fair comparisons. *Strict Machine-Task Pairing:* To eliminate hardware-induced variance, evaluations are conducted on the exact robot units used during data collection. All models are tested sequentially on the same hardware-task pair in randomized order. For example, in the "Stack Bowls" task, all models are evaluated on the same unit across AgileX, Agibot G1, and Galaxea R1pro platforms. *Controlled Evaluation Setup:* Testing conditions follow standardized protocols, mirroring data collection procedures with randomized object positions and orientations while maintaining consistent task specifications. This ensures evaluation of generalization rather than memorization. *Inference and Recording:* Each model undergoes 15 trials per task-robot pair for statistical robustness. Evaluation environments are kept constant, and comprehensive data (*e.g.*, third-person views, robot states, and model predictions) are recorded in rosbag format for transparency. These recordings will be open-sourced to establish verifiable benchmarks.

5.1.4 Evaluation Metrics

We evaluate model performance using two metrics capturing both task completion and partial progress. *Success Rate (SR):* the proportion of trials where the model completes all task steps within a 3-minute time limit. This primary metric reflects the model's real-world deployment viability. *Progress Score (PS):* measures partial task completion by tracking progress through sequential subtask checkpoints: For example, in a 6-step "Stack Bowls" task, completing steps 1–4 but failing at step 5 results in a score of $\frac{4}{6} \approx 0.67$. This diagnostic metric highlights failure modes and rewards partial success. *Termination Criteria:* A trial ends if: (1) three consecutive subtask failures occur, or (2) safety-critical events (*e.g.*, collisions) arise. Progress is scored based on subtasks completed before termination. We report overall SR and PS across 100 tasks, and per-platform metrics stratified by robot type to assess cross-embodiment generalization.

5.2 Comparison on Real-world Benchmark

As shown in Tab. 1, we compare our two LingBot-VLA variants with three strong baselines across three platforms. On all platforms, LingBot-VLA *w/o* depth significantly outperforms WALL-OSS and GR00T N1.6 in both SR and PS metrics. By incorporating depth-based spatial information, LingBot-VLA *w/* depth achieves an average SR improvement of 4.28% and a PS increase of 7.76% over $\pi_{0.5}$ across the three embodiments. Notably, GR00T N1.6 performs average on the Agibot G1 and AgileX embodiment but achieves SR and PS comparable to $\pi_{0.5}$ on the Galaxea R1Pro platform. This is due to the extensive inclusion of Galaxea R1Pro data during its pre-training, indicating that pre-training can significantly enhance performance on downstream tasks with high structural similarity. The complete and detailed test results can be found in the Appendix Tab. S1- S6.



(a). Qwen2.5-VL-3B-π model **(b).** PaliGemma-3B-pt-224-π model

Figure 4. Training throughput analysis of the (a) Qwen2.5-VL-3B-π and (b) PaliGemma-3B-pt-224-π models.

5.3 Comparison on Simulation Benchmark

In Tab. 2, we evaluate simulation performance across 50 representative manipulation tasks within the RoboTwin 2.0 suite. Starting from pretrained checkpoints, each model was further finetuned on the RoboTwin dataset. To assess multi-task generalization, we train all models on $2,500$ demonstrations from clean scenes (50 per task) and $25,000$ from highly randomized scenes (500 per task). Randomization factors encompass varied backgrounds, table-top clutter, table-height perturbations, and diverse lighting conditions. Compared to the $\pi_{0.5}$ baseline, `LingBot-VLA` demonstrates marked advancements in RoboTwin 2.0 multi-task settings. Specifically, `LingBot-VLA` *w/o* depth yields absolute success rate increases of over 3.76% in clean environments and 8.58% in randomized scenarios. By employing learnable query-based alignment, the integration of depth information enables `LingBot-VLA` to effectively extract rich spatial priors from LingBot-Depth model. The approach surpasses the baseline model by absolute margins of 5.82% and 9.92% in clean and randomized configurations, respectively. Please refer to Tab. S7 of Appendix for detailed results.

5.4 Training Throughput Analysis

To comprehensively evaluate the training efficiency of VLA models across different frameworks, we selected three open-source codebases (*i.e.,* StarVLA, Dexbotic, and OpenPI) as the baselines for comparison. To ensure a fair comparison, all experiments were conducted on the Libero dataset using a standardized π-like model architecture.

Given the variations in the VLM implementations across different codebases, we reproduced both Qwen2.5-VL-3B-π and PaliGemma-3B-pt-224-π models within our own codebase to facilitate a direct alignment and comparison with the baselines. Regarding the training configuration, the local batch size was standardized to 32 for all experiments.

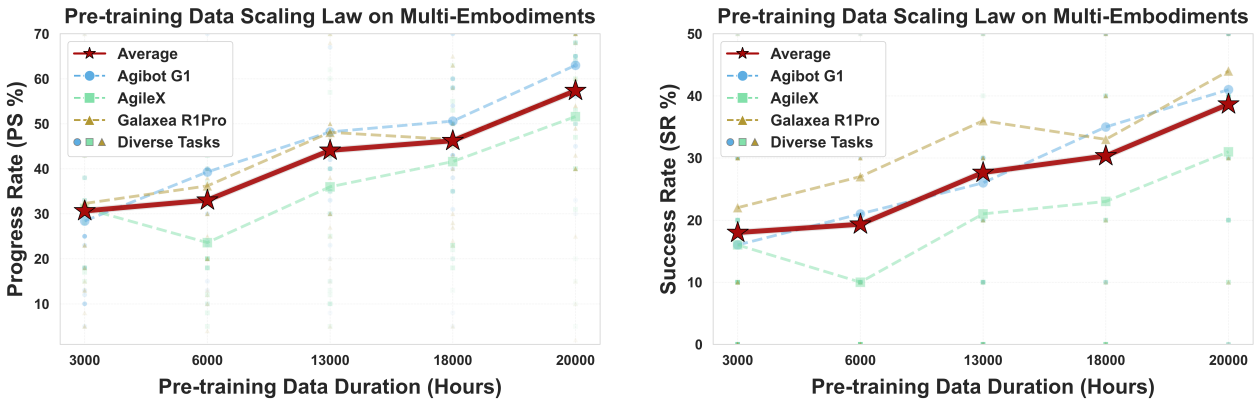
It is worth noting that while StarVLA and Dexbotic default to ZeRO for distributed training, our codebase employs the comparable FSDP2 strategy. In contrast, OpenPI utilizes DDP, which inherently incurs lower communication overhead. We adopted sample throughput (samples/s) as the primary evaluation metric.

Figures 4a and 4b illustrate the training efficiency comparison between our codebase and the baselines for the Qwen2.5-VL-3B-π and PaliGemma-3B-pt-224-π models, respectively. The results demonstrate that our codebase achieved the fastest training speeds in both model settings. Furthermore, the figures detail the training throughput across configurations of 8, 16, 32, 128, and 256 H200 GPUs, alongside the theoretical linear scaling limit. The data indicates that our solution not only delivers superior throughput but also exhibits excellent scaling efficiency that closely follows the theoretical limit as the number of GPUs increases.

5.5 Ablation Studies

5.5.1 Scaling Experiments

To assess the scaling laws of pre-training data, we conduct experiments on a subset of 25 representative tasks drawn from the benchmark. As shown in Figs. 5a and 5b, both the progress rate and success rate demonstrate a consistent upward trend as the pre-training data duration increases from 3,000 to 20,000 hours. This indicates that scaling up



(a). Progress Rate (PS) **(b).** Success Rate (SR)

Figure 5. Scaling behavior across dataset size. With increased data scale, our model exhibits scaling laws in terms of success rate and progress rate.

real-world pre-training data contributes to improved generalization and performance across diverse downstream tasks and embodiments. Furthermore, the individual trends of the three embodiments (*i.e.,* Agibot G1, AgileX, and Galaxea R1Pro) generally align with the aggregated performance, suggesting the observed scaling law is robust and not specific to a single platform. These results validate the effectiveness of our scaling approach in enhancing the capabilities of the generalist policy.

5.5.2 Data-efficient Analysis

Following the large-scale real-world benchmarking protocols, we selected eight representative tasks from GM-100 dataset to conduct data-efficient post-training experiments on the Agibot G1 platform. As illustrated in Fig. 6, with a limited budget of only 80 demonstrations per task, LingBot-VLA outperforms $\pi_{0.5}$ using the full 130-demonstration set in both Progress Rate and Success Rate. Notably, the performance margin between LingBot-VLA and $\pi_{0.5}$ widens significantly as the volume of post-training data increases, demonstrating superior data efficiency and scalability.

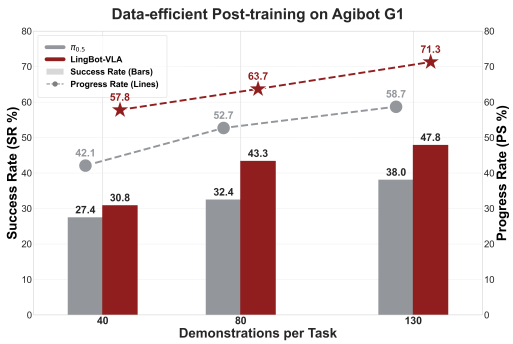


Figure 6. Data efficiency of LingBot-VLA post-training.

6 Conclusion

We have introduced LingBot-VLA, a foundation model that achieves superior generalizability and training efficiency through large-scale real-world data and an optimized codebase. Our comprehensive evaluation across 100 tasks demonstrates that our model achieves clear superiority over competitors, showcasing its strong performance and broad generalizability. To foster open science, we release our code, model, and benchmark data. Future research will focus on scaling the model versatility by integrating single-arm and mobile robotic data, paving the way for more diverse and mobile manipulation capabilities in unconstrained environments.

Acknowledgment. We thank Zhengyu He, Han Zhang, Haidan Zhou, Chongjun Zhong, Yida Zou, Siyuan Li, Zhikun Luo, Yuanqi Chen, Yingying Zhang, Yijun Zheng, Wanting Xu, Hongfei Niu, Yan Zha, Jialiang Zheng, Liping Zhang, Zhen Liu, Rundong Zhou, Yuan Guan, Haitao Wang, Weilun Yao, Zhiwei Liang, Jiahao Fan, Jingran Xu, Linyu Su, Haiwei Liang, Yixiang Gao, Yingmin Li, Yongqiang Wen, Yuanzhe Guo, Yijun Zheng, Fengrui Zhang, Lin Wang, Min Yao, Fei Lu, Jingyun Tian, Ting Huang, Xinyang Wang, Jianxue Qian and Wenhui Shi for help with data, evaluation experiments, training infrastructure, robot hardware and robot software. We also gratefully acknowledge Galaxea Team, AgileX Robotics, and Leju(Shenzhen) Robotics Technology Co., Ltd. for help with the data collection and benchmark.

References

[1] Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. *arXiv preprint arXiv:2506.18123*, 2025.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[4] 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.

[5] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning*, 2025.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025.

[7] 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.

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

[9] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

[10] 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.

[11] Heyu Guo, Shanmu Wang, Ruichun Ma, Shiqi Jiang, Yasaman Ghasempour, Omid Abari, Baining Guo, and Lili Qiu. OmniVLA: Physically-grounded multimodal vla with unified multi-sensor perception for robotic manipulation. *arXiv preprint arXiv:2511.01210*, 2025.

[12] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. MLLMs need 3D-aware representation supervision for scene understanding. *arXiv preprint arXiv:2506.01946*, 2025.

[13] Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and G0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

[14] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[15] 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.

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

[17] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Adv. Neural Inform. Process. Syst.*, 36:44776–44791, 2023.

[18] Qianli Ma, Yaowei Zheng, Zhelun Shi, Zhongkai Zhao, Bin Jia, Ziyue Huang, Zhiqi Lin, Youjie Li, Jiacheng Yang, Yanghua Peng, et al. Veomni: Scaling any modality model training with model-centric distributed recipe zoo. *arXiv preprint arXiv:2508.02317*, 2025.

[19] 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.

[20] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[21] 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.

[22] starVLA Contributors. StarVLA: A lego-like codebase for vision-language-action model developing, 2025.

[23] Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. GeoVLA: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

[24] Bin Tan, Changjian Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen Shen, and Nan Xue. Masked depth modeling for spatial perception. https://technology.robbyant.com/lingbot-depth, 2026.

[25] Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini Robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025.

[26] 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.

[27] NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_6/, 2025.

[28] Yunnan Wang, Fan Lu, Kecheng Zheng, Ziyuan Huang, Ziqiang Li, Wenjun Zeng, and Xin Jin. Vision-centric activation and coordination for multimodal large language models. *arXiv preprint arXiv:2510.14349*, 2025.

[29] Ziyu Wang, Chenyuan Liu, Yushun Xiang, Runhao Zhang, Qingbo Hao, Hongliang Lu, Houyu Chen, Zhizhong Feng, Kaiyue Zheng, Dehao Ye, Xianchao Zeng, Xinyu Zhou, Boran Wen, Jiaxin Li, Mingyu Zhang, Kecheng Zheng, Qian Zhu, Ran Cheng, and Yong-Lu Li. The Great March 100: 100 detail-oriented tasks for evaluating embodied ai agents, 2026.

[30] Bin Xie, Erjin Zhou, Fan Jia, Hao Shi, Haoqiang Fan, Haowei Zhang, Hebei Li, Jianjian Sun, Jie Bin, Junwen Huang, et al. Dexbotic: Open-source vision-language-action toolbox. *arXiv preprint arXiv:2510.23511*, 2025.

[31] Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, Erjin Zhou, Fan Jia, Haitao Yang, Haoqiang Fan, Haowei Zhang, Hongyang Peng, et al. RoboChallenge: Large-scale real-robot evaluation of embodied policies. *arXiv preprint arXiv:2510.17950*, 2025.

[32] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14203–14214, 2025.

[33] Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting VLMs toward the embodied space. *arXiv preprint arXiv:2509.11766*, 2025.

Appendix

A Experiment

This section provides a comprehensive breakdown of the experimental results. Specifically, Table S1, Table S2, Table S3, Table S4, Table S5 and Table S6 below present the detailed performance on GM-100 real-world benchmark. Table S7 presents the detailed performance on Robotwin 2.0 benchmark. These tables serve as the basis for the aggregated mean results reported in the main text (see Sec. 5.2 and Sec. 5.3).

Table S1. Real-world evaluation on GM-100 [29] benchmark (AgileX, Part I).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#001	7%	17%	20%	23%	53%	57%	40%	67%	47%	73%
#002	0%	3%	0%	0%	0%	23%	0%	15%	0%	2%
#003	0%	1%	0%	0%	20%	53%	7%	42%	0%	19%
#006	0%	3%	0%	0%	0%	20%	0%	0%	0%	17%
#007	0%	16%	0%	7%	0%	41%	20%	61%	27%	70%
#008	0%	8%	0%	0%	0%	1%	0%	39%	0%	66%
#009	0%	20%	27%	47%	0%	0%	13%	33%	47%	73%
#010	0%	2%	0%	0%	0%	0%	13%	13%	7%	13%
#011	0%	15%	7%	7%	53%	77%	7%	67%	20%	67%
#012	0%	0%	0%	0%	0%	0%	7%	45%	33%	59%
#013	0%	0%	0%	0%	0%	15%	0%	0%	7%	20%
#014	0%	3%	0%	5%	0%	2%	0%	40%	0%	54%
#015	53%	53%	67%	67%	40%	40%	80%	80%	100%	100%
#016	0%	0%	7%	15%	7%	18%	0%	7%	0%	27%
#017	7%	11%	20%	63%	53%	75%	27%	71%	0%	60%
#018	0%	0%	0%	3%	0%	2%	0%	0%	0%	20%
#019	0%	12%	0%	0%	13%	57%	0%	55%	7%	35%
#020	0%	3%	0%	0%	0%	27%	7%	37%	0%	32%
#021	0%	12%	0%	5%	7%	25%	0%	27%	0%	23%
#022	0%	0%	0%	1%	0%	26%	0%	59%	7%	63%
#023	0%	3%	0%	0%	0%	3%	0%	22%	0%	12%
#024	0%	0%	0%	4%	0%	15%	0%	9%	0%	10%
#025	0%	0%	0%	0%	0%	5%	0%	0%	0%	0%
#026	0%	0%	0%	0%	0%	4%	0%	0%	0%	0%
#027	0%	0%	0%	3%	53%	68%	20%	59%	20%	59%
#028	0%	0%	0%	3%	27%	60%	40%	43%	0%	8%
#029	0%	15%	7%	15%	47%	67%	0%	43%	13%	44%
#030	0%	0%	53%	53%	60%	62%	67%	87%	53%	78%
#031	0%	0%	0%	2%	27%	35%	0%	3%	0%	5%
#032	0%	13%	0%	24%	0%	0%	20%	33%	0%	27%
#033	0%	0%	0%	24%	0%	53%	0%	31%	0%	27%
#034	0%	0%	0%	7%	87%	96%	67%	84%	53%	85%
#035	0%	0%	0%	0%	0%	0%	0%	7%	60%	60%
#036	0%	11%	0%	9%	13%	75%	0%	51%	0%	51%
#037	0%	1%	0%	3%	7%	16%	0%	6%	0%	8%
#038	0%	0%	0%	0%	0%	10%	0%	1%	0%	6%
#040	0%	14%	0%	0%	80%	96%	80%	91%	93%	98%
#041	0%	0%	0%	2%	0%	8%	0%	2%	0%	0%
#042	0%	2%	0%	0%	40%	65%	7%	22%	0%	22%
#043	0%	18%	0%	22%	7%	58%	27%	78%	33%	78%
#044	0%	0%	0%	0%	0%	14%	0%	5%	0%	3%
#045	0%	0%	0%	3%	0%	22%	0%	11%	0%	23%
#046	0%	0%	0%	5%	13%	29%	7%	60%	73%	89%
#047	0%	6%	0%	19%	7%	40%	7%	59%	7%	37%
#048	0%	7%	0%	9%	7%	39%	7%	63%	0%	52%
#049	0%	0%	0%	5%	0%	18%	0%	22%	0%	23%
#050	0%	3%	0%	4%	0%	25%	0%	15%	0%	33%
#051	0%	0%	0%	2%	0%	2%	0%	40%	0%	58%
#053	0%	16%	0%	13%	7%	60%	0%	48%	13%	20%
#054	0%	3%	0%	3%	0%	33%	7%	20%	20%	43%
#055	0%	0%	0%	3%	0%	13%	0%	5%	0%	0%

Table S2. Real-world evaluation on GM-100 [29] benchmark (AgileX, Part II).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#056	0%	28%	0%	7%	67%	88%	73%	93%	80%	92%
#057	0%	2%	0%	1%	0%	13%	0%	19%	0%	27%
#058	0%	37%	7%	44%	7%	58%	7%	63%	20%	72%
#060	0%	8%	20%	42%	93%	93%	27%	57%	13%	50%
#061	53%	77%	7%	33%	47%	68%	93%	97%	93%	97%
#062	0%	2%	0%	0%	0%	8%	0%	4%	0%	1%
#063	0%	0%	7%	7%	13%	50%	0%	0%	0%	0%
#064	0%	0%	0%	2%	0%	5%	0%	0%	0%	5%
#065	0%	22%	0%	12%	0%	23%	33%	55%	20%	62%
#066	0%	7%	0%	3%	7%	45%	13%	55%	0%	43%
#067	0%	0%	0%	2%	0%	28%	7%	20%	0%	27%
#068	7%	27%	0%	0%	47%	53%	20%	30%	60%	63%
#069	0%	2%	0%	10%	7%	8%	0%	17%	0%	12%
#070	0%	0%	0%	0%	0%	0%	0%	3%	7%	7%
#071	0%	3%	0%	5%	0%	5%	0%	0%	0%	5%
#072	0%	0%	0%	0%	7%	27%	0%	9%	13%	38%
#073	0%	0%	0%	55%	0%	0%	27%	57%	13%	23%
#074	0%	0%	0%	3%	0%	19%	0%	22%	0%	21%
#075	7%	9%	0%	13%	0%	20%	0%	4%	13%	56%
#076	0%	27%	7%	43%	7%	33%	7%	53%	33%	67%
#077	7%	7%	7%	7%	0%	2%	47%	47%	40%	42%
#078	0%	0%	0%	0%	80%	83%	87%	87%	73%	77%
#079	0%	15%	0%	20%	27%	49%	47%	73%	33%	80%
#080	0%	0%	0%	0%	0%	0%	0%	7%	0%	10%
#081	0%	8%	0%	3%	0%	15%	0%	26%	13%	50%
#082	0%	23%	0%	15%	20%	57%	20%	67%	27%	43%
#083	0%	20%	0%	0%	7%	50%	7%	28%	53%	77%
#084	0%	0%	0%	16%	0%	2%	40%	58%	33%	62%
#085	0%	9%	13%	40%	67%	85%	13%	36%	27%	42%
#086	0%	0%	0%	10%	60%	70%	100%	100%	60%	73%
#087	0%	3%	0%	0%	13%	15%	20%	72%	60%	78%
#088	0%	0%	0%	5%	13%	42%	20%	55%	7%	40%
#089	0%	2%	0%	22%	20%	47%	0%	28%	13%	41%
#090	0%	1%	0%	1%	0%	43%	0%	19%	0%	24%
#091	13%	37%	13%	25%	33%	77%	33%	42%	7%	17%
#092	47%	55%	0%	17%	67%	82%	60%	68%	93%	97%
#093	13%	14%	27%	27%	87%	87%	27%	27%	87%	87%
#094	0%	0%	0%	0%	0%	20%	7%	23%	0%	12%
#095	0%	2%	0%	18%	20%	44%	20%	38%	13%	39%
#096	7%	7%	0%	3%	7%	13%	0%	0%	13%	17%
#097	0%	4%	0%	0%	0%	4%	27%	44%	7%	24%
#098	0%	0%	0%	0%	0%	0%	0%	15%	7%	15%
#099	0%	0%	0%	3%	0%	12%	0%	0%	0%	0%
#100	0%	2%	13%	18%	27%	42%	0%	31%	13%	27%
#102	0%	2%	0%	13%	13%	29%	0%	33%	27%	60%
#103	7%	43%	0%	27%	80%	90%	73%	77%	47%	60%
#104	0%	15%	0%	7%	13%	31%	7%	40%	20%	53%
#105	0%	3%	0%	0%	0%	23%	0%	15%	0%	20%
#106	0%	1%	0%	4%	0%	36%	0%	13%	7%	31%
#107	0%	9%	0%	0%	53%	73%	27%	62%	27%	58%

Table S3. Real-world evaluation on GM-100 [29] benchmark (AgibotG1, Part I).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#001	7%	20%	0%	0%	0%	23%	0%	3%	0%	17%
#002	0%	0%	0%	0%	13%	35%	0%	0%	0%	5%
#003	0%	3%	0%	3%	0%	35%	7%	19%	0%	0%
#006	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
#007	0%	0%	0%	18%	7%	44%	0%	25%	13%	52%
#008	0%	0%	0%	10%	0%	7%	0%	2%	0%	0%
#009	0%	41%	0%	30%	0%	33%	13%	57%	7%	47%
#010	0%	0%	0%	10%	0%	11%	7%	44%	7%	18%
#011	0%	0%	7%	23%	20%	43%	53%	88%	0%	43%
#012	0%	0%	0%	0%	7%	28%	13%	22%	13%	37%
#013	0%	0%	20%	69%	20%	71%	33%	71%	0%	45%
#014	0%	0%	0%	0%	0%	0%	0%	5%	0%	0%
#015	80%	80%	100%	100%	93%	93%	100%	100%	93%	93%
#016	0%	0%	0%	0%	0%	20%	0%	3%	0%	2%
#017	0%	0%	0%	3%	27%	45%	0%	51%	0%	55%
#018	0%	3%	0%	0%	0%	8%	7%	67%	33%	72%
#019	0%	8%	0%	12%	0%	5%	0%	20%	0%	0%
#020	0%	0%	0%	0%	0%	10%	13%	37%	33%	47%
#021	33%	54%	0%	3%	0%	17%	13%	63%	60%	87%
#022	0%	7%	0%	1%	0%	13%	0%	12%	0%	9%
#023	0%	2%	0%	3%	0%	10%	0%	3%	0%	35%
#024	0%	0%	0%	23%	7%	18%	0%	9%	0%	29%
#025	7%	12%	0%	0%	0%	2%	0%	5%	13%	18%
#026	0%	0%	0%	0%	0%	2%	0%	0%	0%	7%
#027	0%	0%	0%	5%	0%	24%	27%	65%	47%	68%
#028	0%	5%	0%	0%	0%	43%	67%	77%	60%	78%
#029	0%	0%	0%	1%	0%	17%	7%	19%	7%	19%
#030	0%	0%	0%	0%	47%	47%	73%	78%	27%	76%
#031	0%	3%	0%	14%	0%	12%	7%	7%	0%	7%
#032	0%	0%	0%	0%	0%	7%	27%	44%	7%	27%
#033	0%	2%	0%	20%	0%	16%	0%	31%	7%	42%
#034	7%	7%	13%	50%	13%	45%	0%	42%	20%	67%
#035	0%	0%	0%	1%	0%	20%	0%	24%	7%	44%
#036	0%	1%	0%	5%	0%	11%	0%	2%	0%	5%
#037	0%	0%	0%	0%	0%	17%	0%	8%	0%	0%
#038	0%	4%	0%	5%	13%	34%	0%	10%	0%	4%
#040	0%	6%	0%	0%	0%	24%	40%	66%	47%	64%
#041	0%	0%	0%	5%	0%	0%	0%	0%	0%	3%
#042	0%	0%	0%	0%	47%	49%	0%	0%	0%	40%
#043	0%	2%	0%	23%	0%	13%	0%	35%	13%	52%
#044	0%	0%	0%	0%	7%	7%	0%	2%	0%	0%
#045	0%	0%	0%	0%	0%	0%	0%	3%	0%	15%
#046	0%	0%	0%	9%	7%	35%	47%	73%	27%	42%
#047	7%	30%	0%	0%	0%	5%	0%	7%	7%	23%
#048	0%	1%	0%	0%	0%	3%	0%	0%	0%	11%
#049	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%
#050	0%	2%	0%	0%	0%	5%	0%	10%	0%	8%
#051	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
#053	0%	4%	0%	5%	0%	19%	0%	36%	13%	62%
#054	0%	0%	7%	10%	0%	3%	0%	3%	7%	17%
#055	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table S4. Real-world evaluation on GM-100 [29] benchmark (AgibotG1, Part II).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#056	0%	0%	0%	17%	0%	15%	7%	29%	7%	22%
#057	0%	0%	0%	1%	0%	3%	0%	13%	0%	8%
#058	0%	32%	0%	27%	20%	77%	7%	63%	27%	75%
#060	7%	38%	0%	22%	27%	67%	0%	37%	0%	17%
#061	47%	50%	0%	3%	40%	67%	93%	97%	60%	80%
#062	0%	0%	0%	2%	0%	2%	0%	2%	0%	2%
#063	0%	18%	0%	0%	0%	8%	0%	3%	0%	17%
#064	0%	0%	0%	0%	0%	2%	0%	13%	0%	2%
#065	0%	20%	20%	42%	7%	52%	40%	68%	80%	90%
#066	0%	3%	0%	7%	0%	17%	0%	15%	0%	15%
#067	0%	0%	0%	0%	0%	3%	0%	2%	0%	0%
#068	0%	0%	73%	83%	33%	37%	40%	60%	33%	40%
#069	0%	2%	0%	2%	0%	5%	0%	0%	0%	3%
#070	0%	0%	0%	0%	7%	7%	47%	60%	33%	53%
#071	7%	37%	0%	17%	0%	12%	27%	58%	13%	52%
#072	0%	0%	0%	0%	7%	13%	0%	0%	0%	0%
#073	0%	0%	0%	0%	0%	53%	0%	36%	0%	68%
#074	0%	7%	7%	41%	0%	1%	0%	21%	0%	25%
#075	0%	43%	0%	29%	0%	24%	0%	37%	0%	11%
#076	0%	37%	0%	7%	0%	22%	0%	50%	0%	43%
#077	0%	0%	53%	56%	67%	71%	73%	82%	53%	53%
#078	33%	37%	0%	3%	7%	10%	27%	37%	40%	50%
#079	0%	16%	0%	7%	0%	4%	0%	44%	7%	39%
#080	0%	0%	0%	0%	0%	0%	0%	7%	7%	17%
#081	0%	2%	0%	3%	0%	2%	0%	10%	0%	10%
#082	0%	0%	0%	2%	0%	18%	0%	17%	0%	20%
#083	7%	55%	60%	87%	60%	85%	20%	63%	0%	38%
#084	0%	11%	0%	12%	27%	66%	0%	38%	7%	46%
#085	0%	0%	67%	69%	0%	2%	80%	80%	27%	45%
#086	0%	7%	0%	10%	0%	0%	0%	20%	7%	23%
#087	33%	52%	0%	0%	47%	65%	60%	63%	67%	83%
#088	0%	3%	0%	4%	0%	1%	0%	45%	13%	44%
#089	0%	0%	0%	0%	0%	9%	7%	18%	7%	33%
#090	0%	4%	0%	13%	20%	42%	33%	62%	0%	32%
#091	0%	3%	0%	8%	20%	77%	40%	68%	47%	75%
#092	0%	29%	0%	15%	40%	57%	7%	18%	13%	35%
#093	0%	0%	87%	87%	7%	7%	0%	0%	13%	13%
#094	0%	0%	0%	3%	7%	27%	0%	19%	0%	12%
#095	0%	13%	7%	13%	0%	8%	13%	20%	7%	18%
#096	0%	0%	0%	3%	0%	0%	0%	0%	0%	0%
#097	20%	40%	0%	51%	0%	27%	7%	38%	13%	42%
#098	0%	0%	0%	10%	0%	10%	0%	3%	0%	2%
#099	0%	0%	0%	0%	0%	7%	0%	0%	0%	37%
#100	0%	4%	7%	11%	0%	7%	13%	27%	40%	53%
#102	0%	0%	0%	2%	0%	10%	33%	71%	7%	23%
#103	0%	17%	0%	33%	7%	27%	53%	70%	7%	40%
#104	7%	7%	0%	0%	0%	31%	7%	24%	0%	18%
#105	0%	0%	0%	0%	0%	12%	0%	20%	0%	13%
#106	0%	0%	0%	1%	7%	15%	7%	30%	7%	18%
#107	0%	0%	0%	12%	0%	7%	0%	28%	0%	27%

Table S5. Real-world evaluation on GM-100 [29] benchmark (Galaxea R1Pro, Part I).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#001	13%	17%	13%	20%	20%	43%	20%	27%	27%	47%
#002	33%	45%	0%	2%	0%	2%	0%	10%	0%	8%
#003	7%	49%	0%	0%	0%	0%	73%	89%	33%	44%
#006	0%	1%	0%	12%	0%	0%	7%	28%	0%	6%
#007	0%	2%	0%	8%	13%	18%	0%	0%	33%	39%
#008	0%	4%	0%	14%	0%	20%	0%	43%	0%	45%
#009	0%	20%	80%	83%	13%	57%	33%	63%	27%	53%
#010	7%	8%	0%	2%	7%	15%	0%	15%	13%	43%
#011	33%	63%	80%	82%	73%	88%	100%	100%	80%	95%
#012	27%	33%	47%	62%	53%	72%	33%	35%	0%	2%
#013	0%	2%	0%	0%	40%	40%	87%	96%	80%	80%
#014	0%	0%	0%	3%	7%	7%	0%	25%	7%	35%
#015	47%	47%	93%	93%	40%	40%	53%	53%	47%	47%
#016	0%	0%	0%	0%	13%	30%	0%	8%	0%	38%
#017	0%	4%	0%	7%	0%	27%	0%	3%	0%	15%
#018	0%	17%	7%	20%	7%	28%	0%	10%	73%	87%
#019	0%	5%	0%	0%	0%	18%	0%	0%	0%	0%
#020	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
#021	0%	0%	7%	28%	20%	30%	20%	68%	0%	10%
#022	0%	1%	0%	16%	7%	33%	0%	18%	0%	28%
#023	0%	0%	0%	8%	7%	22%	0%	30%	0%	23%
#024	0%	0%	0%	5%	0%	0%	13%	32%	0%	16%
#025	0%	5%	0%	0%	0%	0%	0%	7%	0%	0%
#026	0%	4%	0%	7%	0%	0%	0%	2%	0%	0%
#027	0%	19%	20%	43%	47%	56%	7%	35%	13%	45%
#028	13%	32%	47%	57%	33%	58%	60%	73%	27%	55%
#029	0%	12%	20%	45%	0%	0%	0%	7%	0%	7%
#030	0%	2%	27%	55%	93%	98%	53%	56%	73%	78%
#031	7%	28%	7%	22%	0%	3%	0%	37%	20%	58%
#032	0%	4%	0%	2%	0%	33%	0%	15%	0%	0%
#033	0%	15%	13%	29%	7%	33%	27%	33%	33%	33%
#034	40%	75%	27%	53%	47%	64%	73%	82%	67%	71%
#035	0%	1%	0%	3%	0%	5%	0%	0%	0%	0%
#036	0%	1%	0%	6%	0%	22%	0%	11%	0%	17%
#037	0%	0%	0%	15%	7%	20%	0%	34%	0%	29%
#038	0%	0%	0%	0%	7%	11%	0%	1%	0%	0%
#040	0%	9%	0%	16%	0%	0%	7%	41%	7%	27%
#041	0%	2%	0%	12%	0%	3%	0%	5%	0%	0%
#042	13%	38%	80%	91%	27%	40%	60%	73%	60%	71%
#043	0%	2%	0%	0%	0%	25%	7%	45%	0%	48%
#044	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
#045	0%	0%	0%	4%	0%	3%	0%	1%	0%	3%
#046	7%	27%	0%	24%	13%	51%	7%	59%	33%	71%
#047	0%	9%	0%	15%	0%	12%	0%	14%	0%	6%
#048	0%	19%	0%	28%	7%	40%	7%	49%	0%	52%
#049	0%	7%	0%	5%	0%	5%	0%	2%	0%	5%
#050	0%	0%	0%	0%	0%	0%	0%	15%	0%	15%
#051	0%	0%	0%	0%	0%	3%	0%	17%	0%	10%
#053	0%	0%	0%	2%	0%	0%	0%	0%	20%	28%
#054	0%	10%	40%	63%	47%	53%	20%	37%	0%	33%
#055	0%	2%	0%	7%	0%	7%	0%	7%	0%	2%

Table S6. Real-world evaluation on GM-100 [29] benchmark (Galaxea R1Pro, Part II).

Tasks	WALL-OSS		GR00T N1.6		$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	SR	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**	**SR**	**PS**
#056	0%	8%	7%	32%	0%	22%	0%	52%	47%	68%
#057	0%	0%	0%	4%	0%	0%	0%	38%	0%	3%
#058	0%	8%	0%	15%	7%	15%	20%	30%	20%	33%
#060	0%	15%	7%	47%	7%	53%	20%	58%	20%	68%
#061	47%	57%	60%	63%	47%	57%	87%	93%	93%	97%
#062	0%	5%	0%	2%	0%	3%	0%	0%	0%	1%
#063	0%	7%	0%	23%	0%	23%	7%	32%	7%	22%
#064	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
#065	0%	17%	53%	70%	53%	53%	73%	80%	67%	80%
#066	0%	13%	0%	13%	0%	30%	0%	15%	0%	17%
#067	0%	0%	13%	13%	0%	0%	47%	48%	47%	47%
#068	7%	10%	20%	30%	40%	53%	13%	23%	0%	10%
#069	0%	3%	0%	8%	13%	40%	0%	33%	13%	45%
#070	7%	7%	73%	73%	60%	60%	73%	73%	87%	87%
#071	67%	73%	7%	20%	27%	30%	33%	65%	33%	63%
#072	0%	9%	0%	7%	0%	0%	0%	2%	0%	0%
#073	0%	3%	7%	41%	0%	12%	7%	12%	7%	40%
#074	0%	17%	0%	1%	7%	17%	20%	52%	0%	2%
#075	13%	13%	0%	11%	20%	33%	20%	37%	0%	16%
#076	20%	50%	0%	50%	0%	43%	13%	57%	13%	47%
#077	7%	7%	87%	96%	33%	38%	47%	53%	80%	82%
#078	80%	80%	80%	80%	93%	93%	87%	87%	80%	80%
#079	7%	9%	67%	81%	60%	73%	0%	9%	0%	9%
#080	0%	0%	0%	0%	0%	0%	0%	23%	40%	57%
#081	0%	17%	0%	22%	0%	22%	0%	50%	33%	62%
#082	0%	2%	0%	5%	20%	32%	47%	67%	0%	48%
#083	7%	12%	0%	22%	0%	22%	27%	57%	27%	57%
#084	0%	23%	13%	40%	7%	37%	20%	50%	13%	60%
#085	93%	93%	60%	60%	67%	71%	93%	93%	87%	93%
#086	0%	7%	0%	23%	7%	23%	13%	47%	40%	53%
#087	7%	18%	0%	2%	20%	54%	40%	67%	27%	37%
#088	0%	4%	0%	3%	0%	19%	53%	88%	73%	93%
#089	0%	0%	0%	0%	0%	1%	7%	10%	0%	3%
#090	0%	7%	0%	3%	0%	15%	0%	15%	20%	35%
#091	0%	0%	40%	70%	13%	20%	13%	23%	93%	93%
#092	7%	20%	27%	63%	0%	12%	47%	73%	0%	32%
#093	27%	27%	47%	47%	27%	27%	67%	67%	73%	73%
#094	0%	31%	0%	29%	0%	23%	0%	16%	0%	8%
#095	0%	22%	60%	67%	7%	22%	13%	29%	73%	87%
#096	0%	0%	0%	3%	0%	0%	0%	0%	0%	0%
#097	0%	0%	33%	58%	27%	27%	27%	29%	20%	33%
#098	0%	0%	0%	0%	27%	45%	0%	3%	7%	7%
#099	0%	3%	0%	0%	0%	0%	0%	3%	0%	10%
#100	0%	11%	20%	38%	53%	69%	27%	33%	60%	60%
#102	0%	2%	7%	16%	0%	29%	0%	33%	0%	18%
#103	0%	13%	0%	17%	20%	50%	33%	67%	33%	63%
#104	0%	2%	40%	51%	0%	0%	0%	2%	13%	20%
#105	0%	3%	0%	3%	0%	23%	0%	28%	0%	7%
#106	0%	1%	0%	5%	0%	2%	0%	25%	0%	24%
#107	53%	57%	7%	12%	7%	12%	47%	48%	0%	0%

Table S7. Simulation evaluation on RoboTwin 2.0 [8] benchmark, under "clean" and "randomized" settings.

Simulation Tasks	$\pi_{0.5}$		Ours *w/o* depth		Ours *w/* depth	
	Clean	**Rand.**	**Clean**	**Rand.**	**Clean**	**Rand.**
Adjust Bottle	100%	99%	100%	100%	100%	100%
Beat Block Hammer	96%	93%	87%	91%	92%	89%
Blocks Ranking Rgb	92%	85%	92%	91%	92%	91%
Blocks Ranking Size	49%	26%	66%	73%	76%	70%
Click Alarmclock	98%	89%	93%	26%	97%	43%
Click Bell	99%	66%	32%	19%	43%	36%
Dump Bin Bigbin	92%	97%	97%	92%	97%	97%
Grab Roller	100%	100%	100%	99%	100%	100%
Handover Block	66%	57%	80%	83%	83%	95%
Handover Mic	98%	97%	94%	98%	94%	99%
Hanging Mug	18%	17%	32%	27%	34%	53%
Lift Pot	96%	85%	100%	99%	100%	100%
Move Can Pot	51%	55%	79%	84%	89%	87%
Move Pillbottle Pad	84%	61%	93%	94%	92%	90%
Move Playingcard Away	96%	84%	96%	99%	98%	100%
Move Stapler Pad	56%	42%	74%	49%	74%	48%
Open Laptop	90%	96%	96%	96%	98%	96%
Open Microwave	34%	77%	91%	75%	91%	92%
Pick Diverse Bottles	81%	71%	79%	86%	88%	85%
Pick Dual Bottles	93%	63%	82%	95%	99%	90%
Place A2b Left	87%	82%	86%	83%	89%	85%
Place A2b Right	87%	84%	74%	77%	80%	80%
Place Bread Basket	77%	64%	92%	93%	95%	93%
Place Bread Skillet	85%	66%	90%	89%	90%	92%
Place Burger Fries	94%	87%	95%	96%	98%	94%
Place Can Basket	62%	62%	68%	78%	75%	72%
Place Cans Plasticbox	94%	84%	97%	100%	100%	98%
Place Container Plate	99%	95%	99%	99%	99%	100%
Place Dual Shoes	75%	75%	80%	83%	87%	86%
Place Empty Cup	100%	99%	100%	100%	100%	100%
Place Fan	87%	85%	91%	79%	92%	87%
Place Mouse Pad	60%	39%	82%	78%	86%	79%
Place Object Basket	80%	76%	90%	91%	90%	88%
Place Object Scale	86%	80%	84%	90%	90%	88%
Place Object Stand	91%	85%	97%	93%	93%	88%
Place Phone Stand	81%	81%	92%	93%	90%	87%
Place Shoe	92%	93%	99%	94%	99%	99%
Press Stapler	87%	83%	90%	88%	86%	93%
Put Bottles Dustbin	84%	79%	88%	92%	92%	93%
Put Object Cabinet	80%	79%	92%	86%	85%	88%
Rotate Qrcode	89%	87%	93%	84%	86%	82%
Scan Object	72%	65%	91%	97%	92%	96%
Shake Bottle Horizontally	99%	99%	100%	100%	99%	98%
Shake Bottle	99%	97%	99%	100%	100%	99%
Stack Blocks Three	91%	76%	92%	99%	96%	95%
Stack Blocks Two	97%	100%	100%	100%	100%	99%
Stack Bowls Three	77%	71%	72%	83%	71%	77%
Stack Bowls Two	95%	96%	92%	95%	90%	97%
Stamp Seal	79%	55%	76%	86%	74%	77%
Turn Switch	62%	54%	61%	65%	67%	63%