RynnVLA-002: A Unified Vision-Language-Action and World Model

Jun Cen[1,2,3,*,†] **Siteng Huang**[1,2,3,*] **Yuqian Yuan**[1,3,*] **Kehan Li**[1,2,*]
Hangjie Yuan[1,2,3] **Chaohui Yu**[1] **Bohan Hou**[1] **Yuming Jiang**[1]
Jiayan Guo[1] **Xin Li**[1,2] **Hao Luo**[1,2] **Fan Wang**[1]
Deli Zhao[1,2] **Hao Chen**[3]
[1]DAMO Academy, Alibaba Group [2]Hupan Lab
[3]Zhejiang University
[*]Equal Contribution [†]Corresponding Author: cenjun.cen@alibaba-inc.com

Abstract

We present **RynnVLA-002 (also named WorldVLA)**, an autoregressive action world model that unifies action and image understanding and generation. Our RynnVLA-002 integrates Vision-Language-Action (VLA) model and world model in one single framework. The world model predicts future images by leveraging both action and image understanding, with the purpose of learning the underlying physics of the environment to improve action generation. Meanwhile, the VLA model generates the subsequent actions based on image observations, aiding in visual understanding and in turn helps visual generation of the world model. We demonstrate that RynnVLA-002 outperforms standalone VLA and world models, highlighting the mutual enhancement between the world model and the VLA model. We evaluate RynnVLA-002 on the LIBERO and RoboTwin 2.0 simulation benchmarks and real-world robot tasks. RynnVLA-002 achieves 97.4% success rate on the LIBERO simulation benchmark without pretraining, while in real-world LeRobot experiments, its integrated world model boosts the overall success rate by 50%. Code and model checkpoints are available at https://github.com/alibaba-damo-academy/RynnVLA-002.

1 Introduction

The Vision-Language-Action (VLA) model has emerged as a promising paradigm for grounding language-conditioned decision making in visual environments, enabling robots to map instructions and observations to actions [9, 39]. These models are constructed by augmenting large-scale pre-trained Multimodal Large Language Models (MLLMs) [51, 42, 79, 2] with either an action head or additional action expert module to generate actions. Recent embodied MLLMs, including RynnBrain and RynnBrain 1.1, further develop spatial reasoning, localization, and planning capabilities [17, 43]. MLLMs contribute robust capabilities in perception and decision making, enabling VLA models to exhibit enhanced generalization across a wide range of robotic tasks [6, 34].

However, standard VLA architectures face three fundamental drawbacks. First, they cannot fully understand actions because actions reside only on the output side, preventing the model from forming an explicit internal representation of action dynamics. Second, they lack imagination: they do not predict how the world might evolve given candidate actions, hindering foresight and counterfactual reasoning. Third, they have no explicit understanding of physics. Without capturing physical dynamics, the model cannot internalize object interactions, contact, or stability. World models directly address these limitations by learning to forecast future observations conditioned on current images and actions, providing agents with action-aware internal states, imagination, and a physics-informed

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

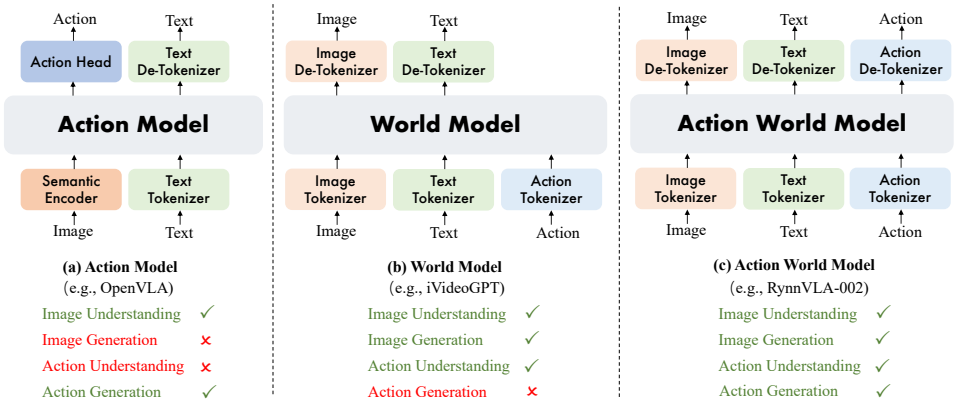


Figure 1: (a) VLA model generates actions based on image understanding; (b) World model generates the image based on image and action understanding; (c) Action World Model unifies both image and action understanding and generation.

representation of environment dynamics. [22, 74]. Despite this advantage, world models are constrained by their inability to directly generate action outputs, resulting in a functional gap that limits their application in scenarios requiring explicit action planning.

To address the constraints inherent in both VLA models and world models, we introduce RynnVLA-002, an autoregressive action world model for unified action and image understanding and generation. As depicted in Fig. 1, RynnVLA-002 employs separate tokenizers to encode images, text, states, and actions. The tokens from different modalities are set to share the same vocabulary so that understanding and generation across these modalities can be unified within a single LLM architecture. The world model component captures the underlying physical dynamics of the environment by generating visual representations based on input actions. This process of action interpretation and environmental physics learning is essential for enabling effective decision making within the VLA model. At the same time, the VLA model embedded within RynnVLA-002 refines the understanding of visual data, thereby improving the precision of image generation performed by the world model. This bidirectional enhancement creates a more robust and comprehensive model capable of understanding and generating both actions and images.

In this work, we explore different action generation mechanisms. Our initial approach involved discretizing actions and unifying them with image and text tokens within a single vocabulary. However, we find that this method struggles with generating coherent action chunks. The primary reason for this is that pretrained multimodal language models have predominantly been exposed to images and text rather than actions, resulting in limited action generalization capabilities. Furthermore, in autoregressive models where subsequent actions are conditioned on preceding ones, error propagation becomes a critical issue, as earlier incorrect predictions influence subsequent actions over time. To alleviate this issue, we proposed an action attention masking strategy that selectively masks prior actions during the generation of current actions. This approach effectively mitigates error accumulation and yields substantial improvements in the task of action chunk generation in simulation.

However, in real-world robot experiments, this discrete design exhibits limited generalization capability and slow inference. We attribute the poor generalization to the high-volume data requirement of discrete autoregressive models [38], which is often unavailable in robotics [78]. The slow inference, meanwhile, stems from the sequential nature of the autoregressive generation process. To address these issues, we evolve our architecture into a hybrid model that retains the original discrete joint modeling while incorporating a continuous Action Transformer head [83]. This new head is significantly smaller than the base LLM, which alleviates overfitting and improves generalization. Furthermore, the Action Transformer's parallel decoding and bidirectional attention mechanism reduce the number of decoding steps, accelerating inference, and generating smoother trajectories.

In summary, our contributions are as follows:

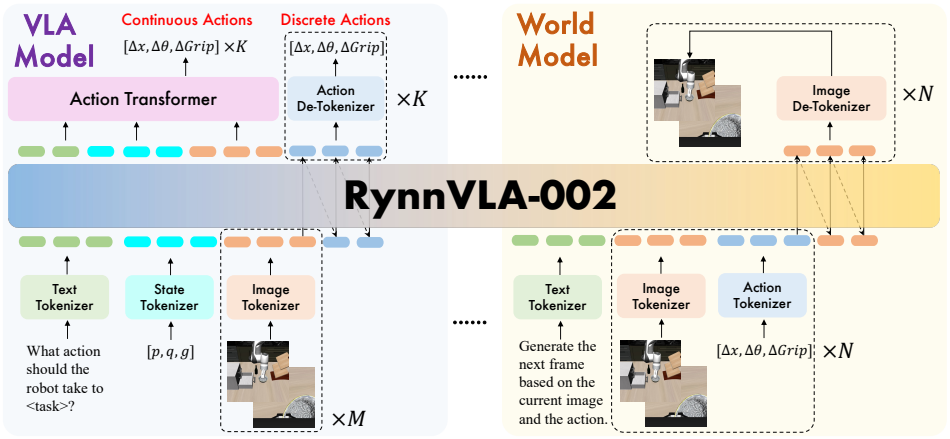


Figure 2: Overview of RynnVLA-002. RynnVLA-002 involves VLA model data and world model data during the training process.

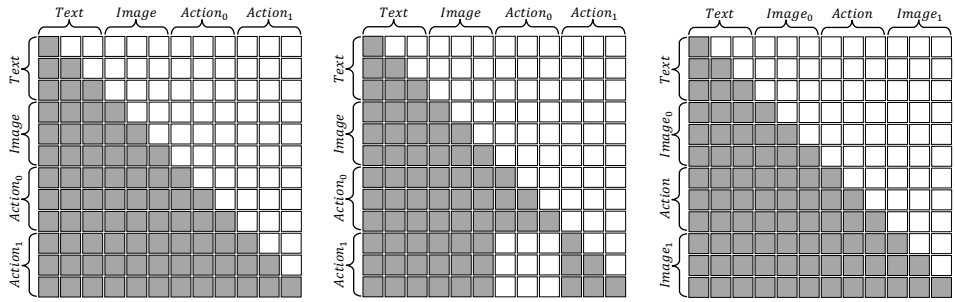
- We propose RynnVLA-002, an action world model that unifies VLA and World Model in a single framework.
- We introduce an action attention masking strategy for the discrete action chunk generation, addressing the challenge of action error accumulation when autoregressively generating action sequences. An additional Action Transformer head is added for stronger generalization ability and smoother trajectory.
- Our experiments demonstrate that RynnVLA-002 outperforms the standalone action and world models, highlighting the mutual enhancement between the world model and VLA model. Additionally, RynnVLA-002 achieves 97.4% success rate on the LIBERO simulation benchmark without pretraining, while in real-world LeRobot experiments, its integrated world model boosts the overall success rate by 50%.

2 Related Work

2.1 Vision-Language-Action Models

VLM-based VLA. Vision-Language Model (VLM)-based VLA models [8, 13, 72, 48, 31, 4, 84, 69] map visual–language inputs to actions. RT-2 [92] first co-trained VLMs on robotic trajectories and web-scale vision-language data, producing actions as discrete tokens. Subsequent works [73, 39, 47, 85, 57] extend this architecture to enhance generalization and representation efficiency. To address precision loss from discrete tokens, LCB [63] introduced a dual-system with a continuous policy head, inspiring variants with different policy head models like diffusion transformers [56], and incorporating diverse training strategies across multiple embodiments [81, 70, 71, 90, 44]. Recent frameworks such as π_0 [7] leverage conditional flow matching [49], the open-source GR00T [5] scales it to complex humanoid control, and $\pi_{0.5}$ [33] further improves generalization by leveraging large-scale multimodal web and cross-embodiment data, enabling direct zero-shot deployment across robot platforms. Complementary work studies fine-grained visual context learning for image retrieval [25] and interleaved multimodal evidence gathering [26].

Visual Generation-based VLA. Beyond static perception, visual-generation approaches model dynamics by predicting future visual states. UniPi [18], DREAMGEN [35] and GeVRM [80] generate future visual observations to guide action generation. Joint frameworks [21, 88, 45] co-generate future frames and corresponding actions, enhancing temporal consistency and policy learning. Others exploit future video prediction as a powerful pretraining objective, including GR-2 [13], VPP [29] and RynnVLA-001 [37]. Collectively, these approaches highlight the potential of predictive visual modeling to bridge perception and action, though challenges remain in visual fidelity, domain transfer, and computational efficiency. Recent unified embodied models also connect physical-world understanding with visual planning and control [91]. Our RynnVLA-002 is built on Chameleon [65],



(a) Default VLA model attention mask (b) Our VLA model attention mask (c) World model attention mask

Figure 3: Attention mask mechanism of (a) default action model, (b) our proposed action model, and (c) world model. Gray entries indicate that the token corresponding to the row is allowed to attend to the token corresponding to the column.

a unified model for image understanding and generation, thus combining the benefits of both VLM and visual-generation-based approaches. Different from prior methods that use visual generation or world models as separate auxiliary modules, RynnVLA-002 unifies action prediction and visual prediction within a single autoregressive model with shared parameters.

2.2 World Models

World models endow embodied AI with internal representations [14, 61, 68] and predictive dynamics of the external world [23, 24, 55, 27], enabling physics-consistent interaction in dynamic environments [76, 52, 22]. Recent advances have realized world models with transformer-based architectures [75, 61, 53]. Notably, Google's Genie framework [10] constructs synthetic interactive environments through large-scale self-supervised video pretraining. Nowadays, such world models are widely utilized to generate varied training data [1], support model-based reinforcement learning algorithms [74], and aid in selecting the most suitable policies from a pool of generated options [46, 3]. In this work, we show that world model and VLA could enhance each other. Unlike approaches that employ world models mainly as external simulators, reward models, or data generators for policy learning, RynnVLA-002 integrates the VLA model and world model into one unified framework, enabling bidirectional enhancement between action generation and image generation.

3 Methods

3.1 Overview

The overall architecture of RynnVLA-002 is shown in Fig. 2. As can be seen, our RynnVLA-002 unifies two foundational models in embodied AI. The first is the VLA model, where a policy π generates an action a_t based on a language goal l, a proprioceptive state s_{t-1}, and an observation history $o_{t-h:t}$:

$$a_t \sim \pi(a_t \mid l, s_{t-1}, o_{t-h:t}). \tag{1}$$

The second is the world model, where the model f predicts the next observation o_t from past observations and actions:

$$\hat{o}_t \sim f(o_t \mid o_{t-h:t-1}, a_{t-h:t-1}). \tag{2}$$

We mix the VLA model data and the world model data to train the RynnVLA-002, an integrated model M_ψ that consolidates the capabilities of action prediction and world modeling. The dual nature of our model is captured by its ability to be queried either as a VLA or as a world model, leveraging a shared group of parameters ψ. During inference, RynnVLA-002 performs only one task per query: when used as a VLA model to generate actions, it does not roll out or use predicted future images; when used as a world model, it predicts future observations conditioned on given actions.

3.2 Data Tokenization

Tokenizers. We initialize the model from Chameleon [65] since it is a unified model for image understanding and generation. Four tokenizers are involved, including an image tokenizer, a text tokenizer, a state tokenizer, and an action tokenizer. The image tokenizer is a VQ-GAN model [19] with additional perceptual losses to specific image regions, *e.g.*, faces and salient objects [20]. The compression ratio of the image tokenizer is 16 and the codebook size is 8192. The image tokenizer generates 256 tokens for 256×256 images and 1024 tokens for 512×512 images. The text tokenizer is a trained BPE tokenizer [62]. The image and text tokenizers are inherited from Chameleon. The state and action tokenizer discretizes each dimension of continuous robot proprioceptive states and actions into one of 256 bins, with bin widths determined by the range of the training data [39, 9]. All image, text, action, and state tokens are in a single token vocabulary with the size of 65536. The continuous actions generated by the Action Transformer are raw actions without tokenization.

VLA Model Data. The overall token sequence of VLA model data is:

$$\texttt{\{text\}}\,\texttt{\{state\}}\,\underbrace{\texttt{\{image-front-wrist\}}}_{\times M}\,\overbrace{\underbrace{\texttt{\{action\}}}_{\times K}}^{\mathcal{L}_{dis_action}}$$

The VLA model generates K actions based on the language instruction, proprioceptive state, and M historical image observations. The text inputs are *"What action should the robot take to* + <*task*>+ *?"*. $\mathcal{L}_{dis_action}$ refers to the cross-entropy loss of discrete action tokens.

World Model Data. World model generates the next image frame given the current image observation and action. The overall token sequence is:

$$\texttt{\{text\}}\,\underbrace{\texttt{\{images-front-wrist\}\{action\}}\overbrace{\texttt{\{images-front-wrist\}}}^{\mathcal{L}_{img}}}_{\times N}$$

All of the training instances for world model share the same text prefix *"Generate the next frame based on the current image and the action."* and there are no other task instructions since the action could totally determine the next state of the world. The overall generation could repeat N times in an autoregressive manner. $\mathcal{L}_{img}$ refers to the cross-entropy loss of discrete image tokens.

Training Objective. We mix the VLA model data and world model data to train our RynnVLA-002. Unless otherwise specified, we use a 1:1 data mix ratio between VLA and world model tasks, and train RynnVLA-002 through one-stage finetuning from Chameleon without additional robot-data pretraining. The overall loss function is $\mathcal{L}_{dis} = \mathcal{L}_{dis_action} + \mathcal{L}_{img}$. In this way, RynnVLA-002 could behave as the VLA model or world model depending on the user queries.

3.3 Action Chunk Generation

Attention Mask for Discrete Action Chunk. Generating multiple actions for execution is critical for efficiency and a higher success rate [40]. However, we find that naively generating consecutive actions in the autoregressive model degrades the performance. Although the foundational MLLM demonstrates robust generalization capabilities across the image and text domains, its capacity to generalize effectively in the action domain is comparatively limited. Consequently, errors originating from earlier actions propagate to subsequent actions under the default causal attention mask, resulting in performance degradation. To address this limitation, we introduce an alternative attention mask tailored for action generation, depicted in Fig. 3 (b). This modified mask ensures that current actions rely solely on textual and visual input, while prohibiting access to prior actions. Such design enables the autoregressive framework to generate multiple actions in isolation, mitigating the error accumulation problem. The world model part adheres to the conventional attention mask, as shown in Fig. 3 (c).

Action Transformer for Continuous Action Chunk. Although our discrete action chunking model performs well in simulation, it rarely succeeds in real-world robot experiments. This discrepancy arises because real-world applications demand significantly higher generalization to cope with dynamic variables like lighting and object positioning—factors not fully captured in simulation. The failure of our discrete model is rooted in two key issues. First, its large autoregressive architecture

Table 1: Evaluation results on LIBERO benchmark. Pretraining means the model is pretrained on the large-scale robot data.

Methods	Pretraining	Action Type	Spatial	Object	Goal	Long	Average
LAPA [77]	✗	Discrete	73.8	74.6	58.8	55.4	65.7
TraceVLA [87]	✓	Discrete	84.6	85.2	75.1	54.1	74.8
OpenVLA [39]	✓	Discrete	84.7	88.4	79.2	53.7	76.5
SpatialVLA [59]	✓	Discrete	88.2	89.9	78.6	55.5	78.1
NORA [32]	✗	Discrete	85.6	89.4	80.0	63.0	79.5
CoT-VLA [82]	✓	Discrete	87.5	91.6	87.6	69.0	83.9
π_0-FAST [6]	✓	Discrete	96.4	96.8	88.6	60.2	85.5
MolmoAct [41]	✓	Discrete	87.0	95.4	87.6	77.2	86.6
FlowVLA [89]	✗	Discrete	93.2	95.0	91.6	72.6	88.1
UniVLA [11]	✓	Discrete	96.5	96.8	95.6	92.0	95.2
Diffusion Policy [16]	✗	Continuous	78.3	92.5	68.3	50.5	72.4
Octo [66]	✓	Continuous	78.9	85.7	84.6	51.1	75.1
MDT [60]	✗	Continuous	78.5	87.5	73.5	64.8	76.1
DiT Policy [28]	✓	Continuous	84.2	96.3	85.4	63.8	82.4
MaIL [36]	✗	Continuous	74.3	90.1	81.8	78.6	83.5
ThinkAct [30]	✓	Continuous	88.3	91.4	87.1	70.9	84.4
π_0 [6]	✓	Continuous	90.0	86.0	95.0	73.0	86.0
$\pi_{0.5}$ [34]	✓	Continuous	<u>98.8</u>	98.2	**98.0**	92.4	96.9
SmolVLA [64]	✗	Continuous	93.0	94.0	91.0	77.0	88.8
OpenVLA-OFT [40]	✓	Continuous	97.6	<u>98.4</u>	<u>97.9</u>	**94.5**	<u>97.1</u>
Seer [67]	✓	Continuous	–	–	–	87.7	–
UVA [46]	✗	Continuous	–	–	–	93.0	–
RynnVLA-002-Discrete	✗	Discrete	94.2	96.8	94.6	87.6	93.3
RynnVLA-002-Continuous	✗	Continuous	**99.0**	**99.8**	96.4	<u>94.4</u>	**97.4**

is prone to severe overfitting when trained on limited real-world dataset, leading to poor generalization. Second, our designed attention mask makes the autoregressive model generate each action in isolation within the same chunk, which cannot ensure trajectory continuity, resulting in severe shaking and non-smooth movements that drastically reduce the success rate.

To overcome these challenges, we propose to augment our architecture with a dedicated Action Transformer to generate continuous action sequences [83]. This module follows the standard ACT-style design: it processes the full context—including language, image, and state tokens—and utilizes learnable action queries to output an entire action chunk in parallel. This design provides two distinct advantages. First, the Action Transformer's more compact architecture is less prone to overfitting on limited data, thereby improving generalization and producing fluid, stable actions. Second, by parallelly generating all actions in a single forward pass, it substantially accelerates the inference speed compared to the autoregressive baselines that generate actions sequentially. We use L1 regression loss $\mathcal{L}_{conti_action}$ to supervise the Action Transformer. The overall loss function is $\mathcal{L} = \mathcal{L}_{dis} + \alpha\mathcal{L}_{conti} = \mathcal{L}_{dis_action} + \mathcal{L}_{img} + \alpha\mathcal{L}_{conti_action}$.

4 Experiments

4.1 Simulation Results

Benchmark. We evaluate our method on the LIBERO benchmark [50]. This benchmark is composed of four distinct suites designed to test a range of robotic capabilities: (1) LIBERO-Spatial focuses on spatial relationships by tasking the robot with placing a bowl in various locations; (2) LIBERO-Object emphasizes object recognition and manipulation with unique objects; (3) LIBERO-Goal tests procedural learning by varying task goals while using a fixed set of objects; (4) LIBERO-Long contains 10 complex long-horizon manipulation tasks.

Table 2: Evaluation results on real-world LeRobot SO100 robots. Success rate is reported. Task A: Place the block inside the circle. Task B: Place the strawberries into the cup. Sing. = Single-Target, Multi. = Multi-Target, Distr. = w/ Distractors.

Method	Pretrain	Task A			Task B		
		Sing.	Multi.	Distr.	Sing.	Multi.	Distr.
GR00T N1.5 [54]	✓	90.0	60.0	50.0	50.0	50.0	70.0
π_0 [6]	✓	100.0	70.0	50.0	80.0	70.0	40.0
$\pi_{0.5}$ [34]	✓	90.0	60.0	80.0	70.0	70.0	50.0
RynnVLA-002	✗	90.0	90.0	80.0	80.0	80.0	50.0

Table 3: Average success rates (%) on RoboTwin 2.0 under clean and randomized settings.

Setting	OpenVLA [39]	π_0 [6]	UniVLA [11]	RynnVLA-002	X-VLA [86]	$\pi_{0.5}$ [34]
Clean	38.3	48.4	45.8	61.36	72.9	**82.7**
Randomized	26.7	26.4	35.1	60.76	72.8	**76.8**

Dataset and Preprocessing. We first curate the dataset by removing unsuccessful trajectories and filtering out "no-operation" actions, a procedure also adopted by OpenVLA [39]. For world model evaluation, which relies on ground-truth video-action pairs, we partition the cleaned data into a 90% training set and a 10% validation set.

Hyperparameter Settings. The VLA model takes $M = 2$ historical image frames as input. We set the action chunk size $K = 10$ for the longer LIBERO-Long and LIBERO-Spatial tasks and $K = 5$ for the shorter LIBERO-Object and LIBERO-Goal tasks. For the world model, we use a single prediction round ($N = 1$) to maintain computational efficiency. The loss weighting parameter $\alpha = 10$.

Metrics. Our evaluation is twofold. To assess the VLA model, we measure its success rate (SR) across 50 deployment rollouts per task, each initialized in a different state. To assess the world model, we measure its video prediction accuracy on the held-out validation set using four standard metrics: Fréchet Video Distance (FVD), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

Benchmark Results. Tab. 1 shows the evaluation results on LIBERO benchmark. We evaluate the performance of discrete actions and continuous actions separately. Our RynnVLA-002 achieves high success rates of 93.3% with discrete actions and 97.4% with continuous actions, demonstrating the effectiveness of our core design principles: jointly learning the VLA modeling and world modeling, an attention mask mechanism for discrete action generation, and the added continuous Action Transformer. Surprisingly, our RynnVLA-002, without any pretraining, is still on par with strong baseline models pretrained on either LIBERO-90 or massive real-robot datasets [67, 11, 40].

RoboTwin 2.0. To complement LIBERO, we evaluate RynnVLA-002 on RoboTwin 2.0 [15] under clean and domain-randomized conditions. Table 3 reports average success rates. Without robot-data pretraining, RynnVLA-002 reaches 61.36% and 60.76%, respectively, exceeding OpenVLA, π_0, and UniVLA while remaining below X-VLA and $\pi_{0.5}$. These additional experiments broaden our evaluation to bimanual manipulation; we have not extensively tuned the training and evaluation settings for this benchmark.

4.2 Real-World Robot Results

Datasets. We curate a new real-world manipulation dataset collected with a LeRobot SO100 robotic arm [12]. All trajectories are expert demonstrations obtained via human teleoperation. We define two pick and place tasks for evaluation. (1) Place the block inside the circle: Emphasizing basic object detection and grasp execution (248 demonstrations); (2) Place strawberries in the cup: Requiring fine-grained localization and grasp-point prediction (249 demonstrations).

Baselines. We compare with three strong open-source baselines: GR00T N1.5 [54], π_0 [6], and $\pi_{0.5}$ [34]. For these methods, we initialize from the official pretrained checkpoints and finetune

Table 4: Ablation study of VLA model using discrete actions on LIBERO benchmark.

Index	VLA Discrete	World Model	Action Chunking	Our VLA model Attention Mask	Goal	Object	Spatial	Long	Average
1	✓	✗	✗	✗	67.3	82.9	77.8	23.0	62.8
2	✓	✓	✗	✗	73.1	88.0	80.2	27.3	67.2
3	✓	✗	✓	✗	79.6	82.9	36.7	16.9	54.0
4	✓	✗	✓	✓	84.4	90.9	81.8	49.3	76.6
5	✓	✓	✓	✓	85.1	90.9	84.0	52.4	78.1

Table 5: Ablation study of VLA model using continuous actions on LIBERO benchmark.

Index	VLA Continous	World Model	Wrist Camera	Proprioceptive State	Goal	Object	Spatial	Long	Average
1	✓	✗	✗	✗	90.2	92.4	88.4	67.0	84.5
2	✓	✗	✓	✗	91.4	95.4	98.2	81.4	91.6
3	✓	✓	✓	✗	96.0	97.4	99.0	85.8	94.6
4	✓	✓	✓	✓	96.4	99.8	99.0	94.4	97.4

them on the same SO100 dataset used for our model. We adopt the same recipe from the official codebases of these baselines to do finetuning.

Evaluation. As shown in Fig. 4, our evaluation spans three scenarios: (1) Single-target manipulation, with exactly one target object on the desktop; (2) Multi-target manipulation, with multiple target objects present; and (3) Instruction-following with distractors, where both targets and distractors appear. A trial is deemed successful if the robot places at least one target object into its designated location within a predefined time budget. A trial fails if: (1) the time limit is exceeded; (2) the robot accrues more than five consecutive failed grasp attempts on a target; (3) in the instruction-following with distractors setting, the agent attempts to manipulate any distractor objects. Each task is tested for 10 times and we report the success rate.

Results. Tab. 2 shows the results of real-world robot experiments. RynnVLA-002 achieves competitive results with GR00T N1.5 [54], π_0 [6], and $\pi_{0.5}$ [34] without pretraining. $\pi_{0.5}$ improves robustness in distractor settings compared with π_0, while RynnVLA-002 achieves better overall performance. Notably, RynnVLA-002 performs better than the baselines in cluttered environments. For instance, RynnVLA-002 has over 80% success rate on both multi-target tasks and distractor-filled scenarios for the "Place the block" task, surpassing the baselines by 10% to 30%.

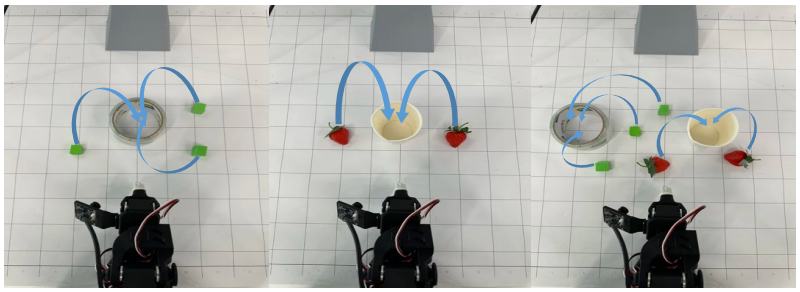
Table 6: Ablation study of VLA model on real-world robots.

Idx	Act. Type	World Mod.	Wrist Cam.	Prop. State	Sing. Tgt.	Multi. Tgt.
1	Disc.	✓	✓	✓	0	0
2	Conti.	✓	✓	✗	0	0
3	Conti.	✓	✗	✓	0	0
4	Conti.	✗	✓	✓	30.0	10.0
5	Conti.	✓	✓	✓	80.0	80.0

4.3 Ablation Study

World Model Benefits the VLA Model. On the LIBERO simulation benchmark, incorporating world model data during training consistently improves performance. Specifically, in Tab. 4, the success rate for discrete actions increases from 62.8% (Line 1) to 67.8% (Line 2), and from 76.6% (Line 4) to 78.1% (Line 5). A similar trend is observed for continuous actions in Tab. 5, where the success rate increases from 91.6% (Line 2) to 94.6% (Line 3). In real-world robot experiments, the benefit of world model data is even more pronounced.

As shown in Line 4 of Tab. 6, the model trained without world model data achieves a very low success rate on real-world robot tasks, which is below 30%. In contrast, augmenting VLA training with world model significantly boosts performance to over 80%. Fig. 5 shows that the model trained without world model data moves directly toward the target location without successfully grasping the cheese or bottle, while the model jointly trained with world model keeps retrying to grasp the target object when encountering failures (see the Appendix for additional real-world visualization).



(a) (b) (c)

Figure 4: Real-world robot settings. (a) Place the block inside the circle. (b) Place the strawberries into the cup. (c) Task with distractors.

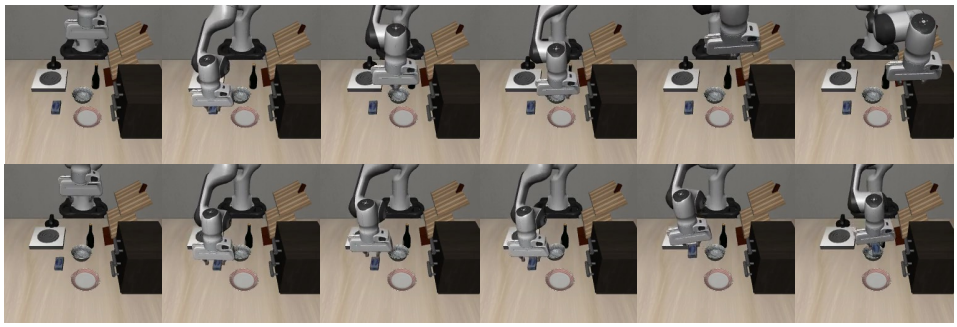


Figure 5: VLA model visualization on LIBERO. Task: put the cream cheese in the bowl. Top: w/o world model. Bottom: w/ world model.

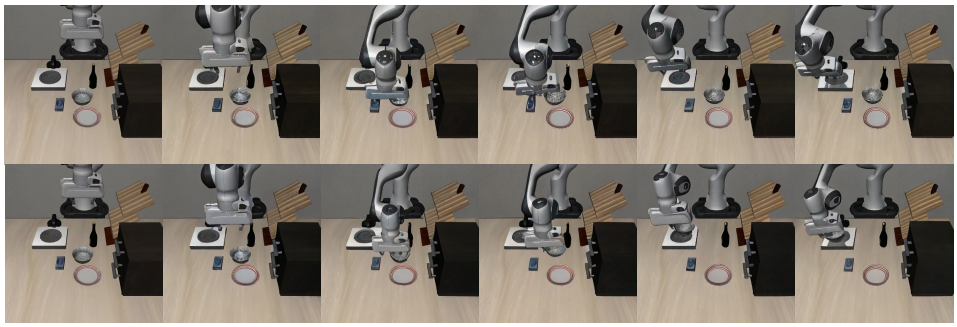


Figure 6: World model visualizationon LIBERO. Task: put the bowl on the stove. Top: w/o VLA model. Bottom: w/ VLA model.

This behavior suggests that the world model data helps the VLA model focus more on the manipulated objects—because the world model's training objective requires accurate prediction of object motion, thereby reinforcing attention to object interaction dynamics.

VLA Model Enhances the World Model. As shown in Tab. 7, the model trained on a mixture of VLA and world model data achieves generation results that are comparable to or better than those of the world model trained solely on world model data. Furthermore, the example in Fig. 6 demonstrates that the world model augmented with VLA data successfully generates the complete manipulation sequence, including grasping the bowl and placing the bowl on the stove, whereas the standalone world model fails at grasping the bowl. This highlights how the image understanding capabilities inherited from the VLA model strengthen the world model's generation performance.

Attention Mask for Discrete Action Chunk Generation. Line 3 (54.0%) v.s. Line 1 (62.8%) in Tab. 4 shows that generating multiple actions with default causal attention mask significantly

Table 7: Ablation study of world model on LIBERO validation set.

Goal	FVD↓	PSNR↑	SSIM↑	LPIPS↓
World Model	370.0	**22.25**	77.84	19.70
Action World Model	**336.8**	22.13	**78.13**	**19.43**
Object	FVD↓	PSNR↑	SSIM↑	LPIPS↓
World Model	1141.6	20.31	59.59	27.30
Action World Model	**877.2**	**22.18**	**65.03**	**22.60**
Spatial	FVD↓	PSNR↑	SSIM↑	LPIPS↓
World Model	405.4	22.32	79.15	20.28
Action World Model	**373.1**	**23.88**	**82.41**	**16.33**
Long	FVD↓	PSNR↑	SSIM↑	LPIPS↓
World Model	557.73	18.24	69.16	31.60
Action World Model	**427.86**	**19.36**	**72.19**	**27.78**

degrades the performance due to the error accumulation problem of the autoregressive action generation process. Line 4 (76.6%) illustrates that our designed attention mask mitigates this problem since the later actions cannot see the former actions.

Ablation Study of Wrist Camera and Proprioceptive State. As shown in Line 1 of Tab. 5, our model could achieve reasonable performance without the wrist camera or proprioceptive state on the LIBERO simulation benchmark. Incorporating these two sources of information brings in further performance gains (see Line 2 and Line 4 in Tab. 5). In real-world experiments (Line 2 and Line 3 in Tab. 6), the robot consistently fails when the wrist camera or proprioceptive state is absent. On one hand, the wrist camera provides crucial visual feedback on the relative pose between the gripper and the object, especially when the robot is outside the field of view of the front camera. On the other hand, we find proprioceptive state is essential for accurately timing gripper closure and object lifting during manipulation. Therefore, both these two types of information are significant for real-world robots.

Discrete Actions Accelerate the Convergence. We retain discrete actions during training alongside the continuous Action Transformer, as we find that this hybrid approach not only speeds up the convergence of VLA training but also improves the ultimate success rate (see the Appendix for detailed analysis).

5 Conclusion

In this work, we propose RynnVLA-002, a unified framework that integrates Vision-Language-Action models and world models within a single architecture, and demonstrate that these two components can mutually enhance each other when jointly trained. By unifying action understanding and generation with visual understanding and prediction, RynnVLA-002 bridges decision-making and environment dynamics modeling in embodied AI. Our results show that this unified design leads to consistent performance improvements across both simulation benchmarks and real-world robotic tasks. Furthermore, we believe this work helps lay the groundwork for a unified foundation for multi-modal understanding and generation that spans text, vision, and action.

Acknowledgments and Disclosure of Funding

This work received no specific funding. The authors declare no competing interests.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 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] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models, 2024. URL `https://arxiv.org/abs/2412.03572`.

[4] Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint, 2024. URL `https://github.com/Stanford-ILIAD/openvla-mini`.

[5] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*, 2024.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[10] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[11] 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.

[12] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024.

[13] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[14] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

[15] 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. URL `https://arxiv.org/abs/2506.18088`.

[16] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[17] Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, et al. RynnBrain: Open embodied foundation models. *arXiv preprint arXiv:2602.14979*, 2026. URL https://arxiv.org/abs/2602.14979.

[18] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[20] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

[21] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

[22] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[23] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[24] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[25] Bohan Hou, Haoqiang Lin, Xuemeng Song, Haokun Wen, Meng Liu, Yupeng Hu, and Xiangyu Zhao. FiRE: Enhancing MLLMs with fine-grained context learning for complex image retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 803–812. Association for Computing Machinery, 2025. doi: 10.1145/3726302.3729979. URL https://doi.org/10.1145/3726302.3729979.

[26] Bohan Hou, Jiuning Gu, Jiayan Guo, Ronghao Dang, Sicong Leng, Xin Li, Xuemeng Song, and Jianfei Yang. InterLV-Search: Benchmarking interleaved multimodal agentic search. *arXiv preprint arXiv:2605.07510*, 2026. URL https://arxiv.org/abs/2605.07510.

[27] Bohan Hou, Gen Li, Jindou Jia, Tuo An, Xinying Guo, Sicong Leng, Haoran Geng, Yanjie Ze, Tatsuya Harada, Philip Torr, Oier Mees, Marc Pollefeys, Zhuang Liu, Jiajun Wu, Pieter Abbeel, Jitendra Malik, Yilun Du, and Jianfei Yang. World model for robot learning: A comprehensive survey. *The International Journal of Robotics Research*, 2026. doi: 10.1177/02783649261488980. URL https://doi.org/10.1177/02783649261488980.

[28] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.

[29] 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.

[30] 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.

[31] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[32] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[33] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0. 5: a vision-language-action model with open-world generalization, 2025. *URL https://arxiv.org/abs/2504.16054*, 1(2):3, 2025.

[34] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[35] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv e-prints*, pages arXiv–2505, 2025.

[36] Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. Mail: Improving imitation learning with selective state space models. In *8th Annual Conference on Robot Learning*, 2024.

[37] Yuming Jiang, Siteng Huang, Shengke Xue, Yaxi Zhao, Jun Cen, Sicong Leng, Kehan Li, Jiayan Guo, Kexiang Wang, Mingxiu Chen, et al. Rynnvla-001: Using human demonstrations to improve robot manipulation. *arXiv preprint arXiv:2509.15212*, 2025.

[38] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[39] 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.

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

[41] 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.

[42] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[43] Kehan Li, Bohan Hou, Minghao Zhu, Tianyi Zhang, Zesen Cheng, Zhikai Wang, Sicong Leng, Xin Li, Xiao Lin, Biying Yao, et al. RynnBrain 1.1: Towards more capable and generalizable embodied foundation model. *arXiv preprint arXiv:2607.17977*, 2026. URL `https://arxiv.org/abs/2607.17977`.

[44] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[45] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[46] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *arxiv*, 2025.

[47] Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024.

[48] 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.

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

[50] 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.

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

[52] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. Genrl: Multimodal-foundation world models for generalization in embodied agents. *Advances in neural information processing systems*, 37:27529–27555, 2024.

[53] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[54] NVIDIA, Nikita Cherniadev Johan Bjorck andFernando 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.

[55] Masashi Okada and Tadahiro Taniguchi. Dreamingv2: Reinforcement learning with discrete world models without reconstruction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 985–991. IEEE, 2022.

[56] 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.

[57] 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.

[58] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

[59] 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.

[60] 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.

[61] Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

[62] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[63] Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: Latent codes as bridges in hierarchical robot control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8539–8546. IEEE, 2024.

[64] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[65] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[66] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[67] 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.

[68] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024.

[69] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. *arXiv preprint arXiv:2507.01016*, 2025.

[70] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024.

[71] 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.

[72] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

[73] 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.

[74] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2025.

[75] Weijia Wu, Zhuang Li, Yefei He, Mike Zheng Shou, Chunhua Shen, Lele Cheng, Yan Li, Tingting Gao, and Di Zhang. Paragraph-to-image generation with information-enriched diffusion model. *International Journal of Computer Vision*, pages 1–22, 2025.

[76] Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

[77] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[78] Yifan Ye, Yankai Fu, Yaoxu Lv, Bohan Hou, Jun Cen, Lingdong Kong, Duo Zheng, Tianxing Chen, Jiaming Liu, Ziang Cao, et al. Data pyramid for embodied manipulation: A survey. *arXiv preprint arXiv:2607.24744*, 2026. URL `https://arxiv.org/abs/2607.24744`.

[79] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[80] Hongyin Zhang, Pengxiang Ding, Shangke Lyu, Ying Peng, and Donglin Wang. Gevrm: Goal-expressive video generation model for robust visual manipulation. *arXiv preprint arXiv:2502.09268*, 2025.

[81] Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024.

[82] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[83] 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.

[84] Wei Zhao, Pengxiang Ding, Min Zhang, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. *arXiv preprint arXiv:2502.13508*, 2025.

[85] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[86] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. URL `https://arxiv.org/abs/2510.10274`.

[87] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[88] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

[89] Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Wenxuan Song, Jiayi Chen, and Haoang Li. Flowvla: Thinking in motion with a visual chain of thought. *arXiv preprint arXiv:2508.18269*, 2025.

[90] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025.

[91] Minghao Zhu, Bohan Hou, Kun Xiang, Zixuan Wang, Yinzhou Tang, Yifei Yang, Zhikai Wang, Tianyi Zhang, Kehan Li, Haoyu Zhao, Siteng Huang, Leigang Qu, Sicong Leng, Dongchi Huang, Mingxiu Chen, Shijian Lu, Yue Wang, Xiaodan Liang, Zhenyu Zeng, Deli Zhao, Chengju Liu, Jianfei Yang, and Xin Li. RynnUnify: Towards unified embodied brain. *arXiv preprint arXiv:2610.32965*, 2026. URL `https://arxiv.org/abs/2610.32965`.

[92] 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.

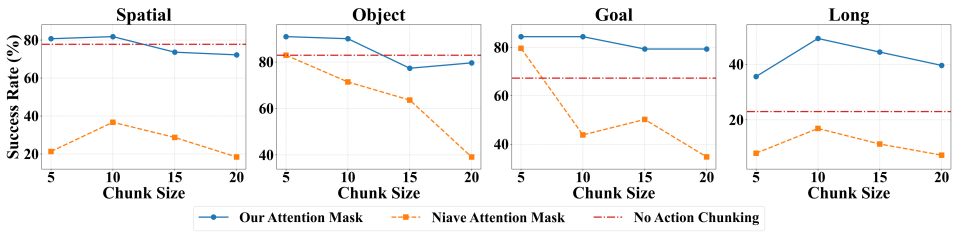


Figure 7: Ablation study of chunk length with discrete actions.

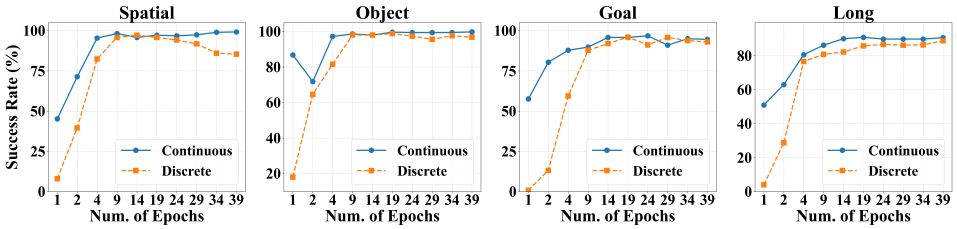


Figure 8: Performance comparison between discrete action and continuous action.

A Ablation Study of Chunk Length

Proper action chunk size is significant for manipulation performance. The small chunk size cannot guarantee the smoothness of the trajectory. However, if the chunk size is too large, the robot's ability to timely adjust its policy becomes constrained, leading to a decrease in overall performance. As demonstrated in Fig. 7, The optimal action chunk size is 10 for LIBERO-Spatial and LIBERO-Object and 5 for LIBERO-Goal and LIBERO-Object.

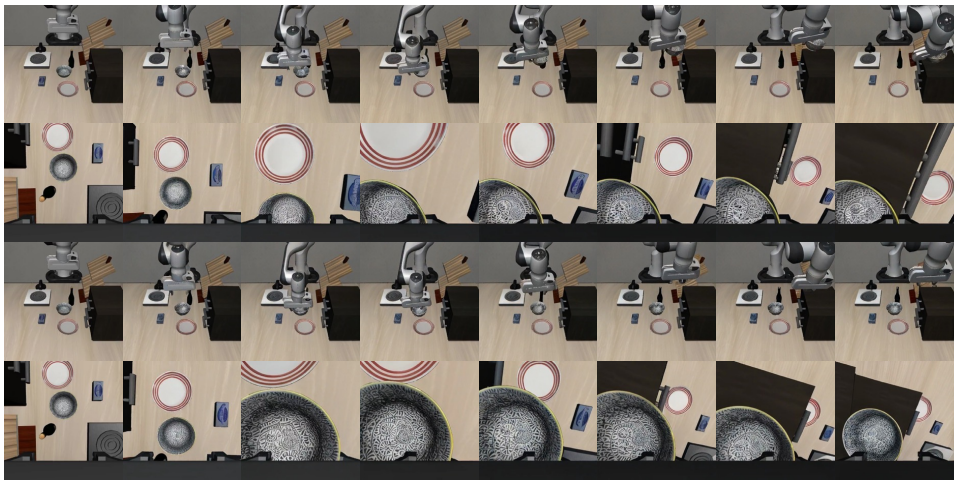
The grasping success rate decreases rapidly with larger action chunks if a vanilla causal attention mask is used. This degradation arises because later actions become overly dependent on preceding ones rather than being grounded in visual input which is a distinct modality. The model's action generalization capability is limited because this modality was not included in its pretraining phase. Consequently, errors tend to accumulate as the sequence of generated actions increases. The proposed attention masking mechanism ensures that each action is generated independently and solely determined by the visual input, thereby mitigating the issue of error propagation within the action sequence. As illustrated in Fig. 7, the model incorporating the proposed attention mask demonstrates superior performance compared to the vanilla attention mask, particularly for the predictions of large action chunks. This highlights the efficacy of the introduced attention mask design.

B Discrete Action and Continuous Action

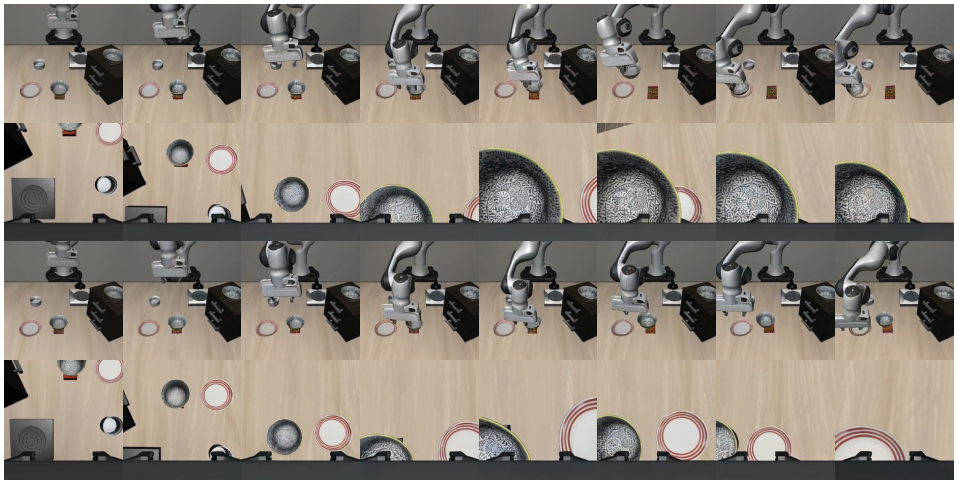
Our model supports both discrete and continuous action generation, and the experimental results reveal a clear advantage for the latter. Concretely, in the LIBERO simulation benchmark, continuous actions result in significantly faster convergence (Fig. 8). Although the final performances of these two models in simulation are comparable, their performance gap becomes considerably significant in real-robot experiments. As shown in Table. 5, our real-world experiments demonstrate a much more substantial improvement when using continuous actions over discrete ones.

C World Model Pretraining for VLA Model

Our RynnVLA-002 unifies VLA model and world model into a single training stage. We further investigate the possibility of cold starting a VLA model with world model pretraining. This form of pretraining necessitates that the model develop an understanding of visual inputs, actions, and the underlying physical dynamics governing state transitions. We pretrain the model using the same data source as the VLA model. As presented in Table 8, employing the world model for pretraining leads to notable improvements in grasping performance. These findings highlight the potential of



(a) Task: put the bowl on top of the cabinet. Top: our action world model. Bottom: world model.



(b) Task: pick up the black bowl and place it on the plate. Top: our action world model. Bottom: world model.

Figure 9: World model visualization.

leveraging world model pretraining in robotic applications, particularly in enhancing task-specific performance through prior exposure to general world knowledge.

Table 8: Ablation study of world model pretraining.

	Goal	Object	Spatial	Long
w/o World Model Pretrain	67.3	82.9	77.8	23.0
w/ World Model Pretrain	**73.1**	**84.0**	**79.8**	**30.2**

D World Model Visualization

Fig. 9 compares video generations from our action world model with a baseline world model trained without VLA data. The baseline world model fails to predict a successful grasp of the bowl from the front camera perspective in both two examples. In contrast, our action world model consistently generates correct videos depicting a successful grasp. Notably, the baseline world model also exhibits a significant inconsistency: as seen in Fig. 9 (a), the front camera shows a failed grasp while

the wrist camera shows a successful one. This highlights a critical disalignment between the model's predictions for different viewpoints.

Table 9: Ablation study of VLA model on efficiency and action chunking on LIBERO benchmark. Frequency is measured in Hz.

Index	Action Type	Action Chunking	Wrist Camera	History Length	Chunk Size = 5			Chunk Size = 10		
					Frequency	Goal	Object	Frequency	Spatial	Long
1	Discrete	✗	✗	0	2.50	60.0	71.0	2.50	77.0	10.4
2	Discrete	✓	✗	0	3.69	83.2	90.0	3.69	83.6	46.0
3	Discrete	✗	✗	1	1.88	74.6	72.8	1.88	78.6	17.0
4	Discrete	✓	✗	1	3.24	92.8	91.6	3.24	86.2	64.0
5	Discrete	✗	✓	1	1.25	82.6	89.6	1.25	96.6	61.2
6	Discrete	✓	✓	1	2.74	92.8	97.8	2.74	96.8	80.8
7	continuous	✓	✗	0	24.94	80.0	89.6	48.20	84.0	48.6
8	continuous	✓	✗	1	14.97	90.2	92.4	28.30	88.4	67.0
9	continuous	✓	✓	1	7.75	91.4	99.4	15.78	98.2	81.4

E Efficiency Analysis

Efficiency Analysis. As shown in Tab. 9, incorporating additional input images, such as images from the wrist camera or historical frames, improves performance but reduces speed. For discrete actions, action chunking yields both higher inference speed and better performance compared to generating a single action per inference step. This is because KV-cache could be used with action chunking [58], while generating one action per time needs the whole model inference every time. Continuous action generation is significantly faster owing to its parallel generation nature, and frequency scale almost linearly with chunk size as generating additional actions incurs negligible extra time.

F Discrete Actions Accelerate the Convergence

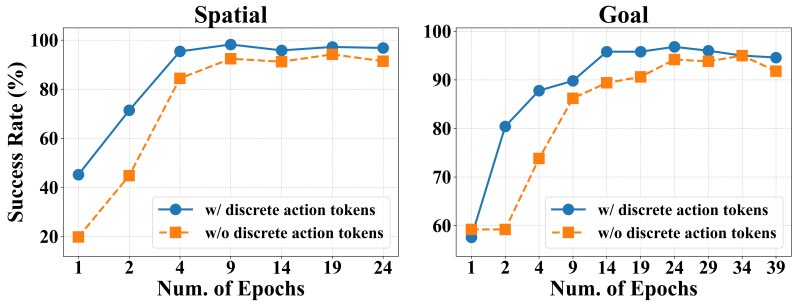


Figure 10: Discrete action tokens accelerate the convergence of continuous action generation.

As shown in Fig. 10, models trained with discrete action tokens achieve a substantially higher success rate than those trained without them, with the advantage being most pronounced during the initial stages of training.

G Real-World VLA Visualization

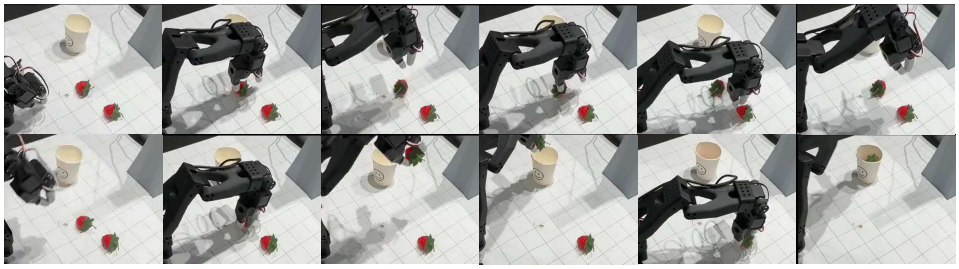


Figure 11: VLA model visualization on real-world scenario. Task: Place strawberries in the cup. Top: w/o world model. Bottom: w/ world model.

Fig. 11 illustrates a real-world task of placing strawberries into a cup: without world model data, the robot approaches the strawberry but fails to achieve a precise grasp for both two strawberries and thus cannot complete the placement; with world-model training, the robot can precisely localize and grasp the target, repeatedly grasp two strawberries, and ultimately lift them and place them into the cup successfully.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions: a unified VLA/world-model framework, an action attention masking strategy with a continuous Action Transformer, and evaluation on LIBERO and real-world SO100 robot tasks, with additional RoboTwin 2.0 evaluation. These claims are supported by the method description in Section 3 and the benchmark and ablation results in Section 4, including Tables 1–7.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses several practical limitations and failure modes, including the limited generalization and non-smooth trajectories of the discrete action model in real-world settings in Section 3.3, and the dependence on wrist-camera and proprioceptive-state inputs in Section 4.3. The experimental scope is also explicitly bounded to LIBERO and RoboTwin 2.0 simulation and LeRobot SO100 real-world tasks in Section 4.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present theoretical results, theorems, or formal proofs. The contributions are empirical and architectural, focusing on model design and robot evaluation.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [No]

   Justification: Sections 3–4 describe the architecture, training objectives, LIBERO evaluation, and SO100 setup. RoboTwin 2.0 clean and randomized success rates are reported, but the complete benchmark-specific training and evaluation recipe is not provided.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [No]

   Justification: The public repository linked in the abstract provides the implementation, model checkpoints, and training and evaluation instructions for LIBERO and LeRobot. Full release and documentation of the newly collected real-world demonstration data remain incomplete.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: Sections 3–4 provide the LIBERO and SO100 settings. Complete training and evaluation configurations for the additional RoboTwin 2.0 results are not specified.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: The paper reports success rates over fixed evaluation protocols, including 50 deployment rollouts per LIBERO task and 10 trials per real-world task, but does not report error bars or confidence intervals. We omit them due to the high cost of repeated robot training and deployment, and report controlled ablations under the same setting instead.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: The current submission does not fully specify the hardware type, memory, or total compute needed to reproduce every experiment. We will add detailed compute-resource information, including GPU configuration and approximate training time, in the final version.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The experiments are conducted on simulation benchmarks and controlled tabletop manipulation tasks using a LeRobot SO100 arm, without collecting personally identifiable information or deploying the system in safety-critical environments.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: RynnVLA-002 may positively advance general-purpose embodied AI by improving robot manipulation, dynamics understanding, and data-efficient policy learning. Potential negative impacts include misuse of more capable robot policies in unsafe physical settings; mitigation requires controlled deployment, hardware safety constraints, and careful evaluation before real-world use.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The current submission does not release a high-risk pretrained model, scraped dataset, or generative system intended for unrestricted deployment. The experiments are limited to controlled simulation and tabletop robot manipulation settings; any future release should include usage guidelines and safety notices for physical robot deployment.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: The paper cites the existing models, datasets, and benchmarks used, including Chameleon, LIBERO, LeRobot, GR00T, and π_0/$\pi_{0.5}$. However, the current draft does not explicitly list all asset licenses and terms of use, which should be added to the appendix or release documentation.

    Guidelines:

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

13. **New assets**

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

    Answer: [No]

    Justification: The public RynnVLA-002 implementation is linked in the abstract. The paper also introduces a newly collected real-world LeRobot SO100 manipulation dataset and describes the tasks and number of demonstrations in Section 4.2. Full documentation, licensing, and release instructions for this dataset are not yet provided.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject experiments. The real-world trajectories are expert robot teleoperation demonstrations and do not collect personal information, behavioral study data, or participant responses.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve human-subject experiments or crowdsourcing studies requiring IRB approval. Expert teleoperation is used only to collect robot demonstration trajectories and does not involve personal data collection.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLM-like multimodal transformers are a core component of the method: RynnVLA-002 is initialized from Chameleon, and Sections 2–3 describe how language, image, action, and state tokens are unified in a shared autoregressive model. The paper does not rely on LLMs for non-standard data annotation or evaluation beyond the model architecture itself.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.