Causal World Modeling for Robot Control

Lin Li[1*], Qihang Zhang[1*†], Yiming Luo[1,3*§], Shuai Yang[1,4§], Ruilin Wang[1,5§], Luyao Zhang[1], Mingrui Yu[1,6§], Zelin Gao[1], Nan Xue[1], Boyu Zhou[7], Xing Zhu[1], Mingyu Ding[8], Yujun Shen[1], Yinghao Xu[2,1‡]

[1]Robbyant, [2]Hong Kong University of Science and Technology, [3] The University of Hong Kong, [4] Zhejiang University, [5] Sun Yat-Sen University, [6] Tsinghua University, [7] Southern University of Science and Technology, [8] University of North Carolina at Chapel Hill

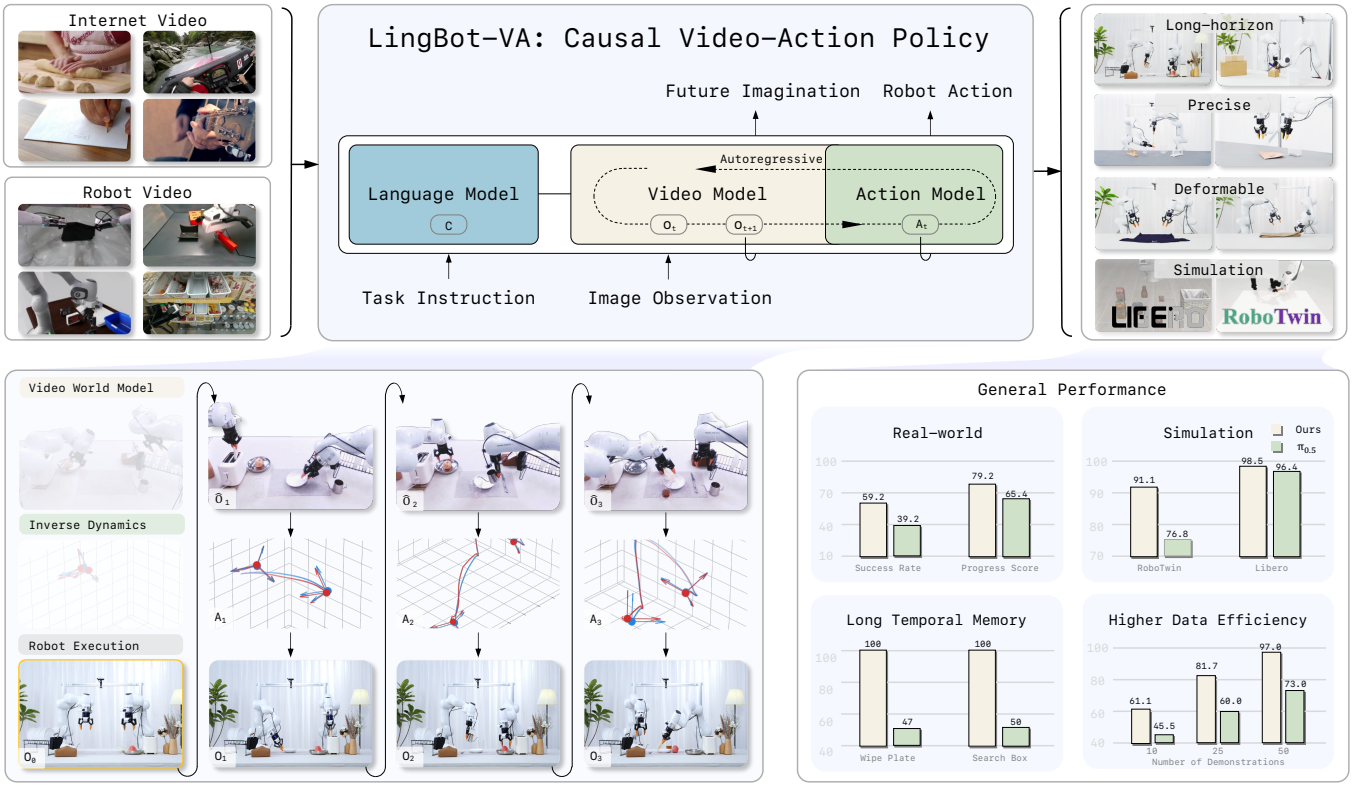

Fig. 1. `Lingbot-VA` **: An Autoregressive World Model for Robotic Manipulation.** (1) **Pretraining:** `Lingbot-VA` is pretrained on diverse in-the-wild videos and robot action data, enabling strong generalization across scenes and objects. (2) **Comprehensive Evaluation:** We conduct extensive experiments on real-world tasks (long-horizon, deformable objects, and precision manipulation) and simulation benchmarks, significantly outperforming state-of-the-art methods including $\pi_{0.5}$. (3) **Versatile Capabilities:** Beyond policy learning, our model supports visual dynamics prediction and inverse dynamics inference from robot videos. (4) **Emergent Properties:** Our causal world modeling approach exhibits long-range temporal memory and strong few-shot adaptation ability. Website: https://technology.robbyant.com/lingbot-va.

***Abstract*—This work highlights that video world modeling, alongside vision-language pre-training, establishes a distinct foundation for robot learning. By capturing environmental dynamics, video world models enable the robot to predict near-future states—a capability essential for effective planning. Inspired by this, we introduce Lingbot-VA, an autoregressive diffusion framework that unifies frame prediction and action inference. Our approach features three key innovations: (1) an autoregressive Mixture-of-Transformers (MoT) that processes visual frames and actions as a single causal sequence; (2) a history integration mechanism using a Key-Value (KV) cache to maintain temporal context from real-world interactions; and (3) a noisy latent augmentation strategy that enables decoding actions directly from intermediate denoised videos for fast inference. Evaluations on simulation and real-world benchmarks demonstrate that Lingbot-VA excels in complex manipulation, including long-horizon, high-precision, and deformable object tasks. Our code and models are publicly available.**

*Equal Contribution †Project Lead ‡Corresponding Author §Work done during internship at Robbyant

I. Introduction

Endowing robots with physical understanding and predictive capabilities is essential for general-purpose manipulation. Currently, Vision-Language-Action (VLA) models [6, 7] attempt to encode these capabilities implicitly by mapping instructions and visual observations directly to actions. However, this end-to-end paradigm typically lacks a mechanism for explicitly modeling physical processes or dynamics. Consequently, in-

stead of acquiring robust physical reasoning, VLA models often rely on superficial pattern matching, rendering them fragile in tasks that require anticipating complex environmental changes.

Second, current chunk-based policies [11, 26, 59] predict from a single frame or a short observation window, treating the task as Markovian and discarding the history needed to disambiguate partially observable states.To address this, recent research has explored video generation models as world simulators that condition action prediction on predicted future frames [15, 58]. While conceptually promising, these methods struggle on real-world manipulation for three reasons. First, existing video models [16, 37, 48] use bidirectional attention that conflicts with the temporal ordering of physical processes, producing temporally inconsistent dynamics and ambiguous action inference. Second, current chunk-based policies [11, 26, 59] predict from a single frame or a short observation window, treating the task as Markovian and discarding the history needed to disambiguate partially observable states. Third, denoising long video latents at inference is prohibitively slow for the high-frequency control loops manipulation demands.

In this paper, we address these issues with Lingbot-VA, an autoregressive (AR) video-action model designed for robotic manipulation. We propose the following designs:

1) ***Interleaved autoregressive generation****:* We propose a Mixture-of-Transformers (MoT) framework that interleaves video and action tokens into a single causal sequence. Modality-specific experts model dynamics and actions under a causal mask: a high-capacity video expert forecasts future visual states given the observation–action history, and a lightweight action expert infers the actions consistent with those predictions. The asymmetric allocation captures complex scene transitions while keeping per-step action decoding cheap.

2) ***Persistent and efficient history integration****:* The causal formulation lets us condition each prediction on the full past observation–action stream rather than on a fixed-length window. At inference, we feed real observed tokens (not predicted ones) into the cache, anchoring the policy in actual interaction history; a Key-Value (KV) cache amortizes the cost across rollout steps.

3) ***Noisy latent augmentation for fast inference****:* We train the action expert to decode from partially denoised video latents, exploiting the insight that control requires high-level semantic structure rather than pixel-perfect detail. This lets us truncate video denoising early at deployment, removing the main inference-time bottleneck while maintaining action precision.

We validate our framework across diverse simulation [9, 32] and real-world scenarios, ranging from long-horizon and high-precision manipulation to deformable object interaction. Experiments demonstrate exceptional learning efficiency, completing tasks with fewer than 100 demonstrations, while outperforming state-of-the-art VLA models by 20% in success rate and 13% in progress rate. These findings show the effectiveness of our proposed video action model for complex real-world manipulation.

II. Related Work

A. Vision-Language-Action Policies

Vision-Language-Action (VLA) policies [14, 20, 22, 25, 34] leverage web-scale data and pre-trained VLMs to outperform task-specific policies [11, 54], with recent work improving deployability through lightweight architectures [42, 43], efficient tokenization [38], and fine-tuning strategies [21, 23]. Two limitations remain. First, these methods map observations directly to actions without explicit dynamics modeling, often leading to trajectory memorization. Second, they condition on a single frame, a short stack of recent frames, or rely on action chunking — none maintains a persistent memory over the full observation history, which is insufficient for partially observable, long-horizon tasks. Lingbot-VA instead uses a causal video world model for explicit dynamics prediction and a KV-cached history for persistent context.

B. Video Models for Robotic Control

Video generative models [16, 24, 37, 48] have recently been adapted to robotics in several forms: as planners predicting future frames or visual chain-of-thought to condition actions [15, 53, 58], as world models for policy learning and simulation [2, 18, 44, 50, 52, 56], and for jointly learning shared video-action features [26, 59]. However, conventional video models in robotics rely on bidirectional attention that sacrifices physical causality and incurs high inference latency. Lingbot-VA addresses both via autoregressive post-training with strict causal attention, combined with Mixture-of-Transformers and noisy latent augmentation for real-time deployment.

The most closely related concurrent work is Motus [4], which shares our Wan video backbone and Mixture-of-Transformers architecture. The key distinction is our *causal* formulation: strict causal attention masking over the interleaved video-action history enables streaming inference, persistent memory via KV cache, and online error correction conditioned on observed rather than only predicted states — all absent in Motus. These design choices yield consistent gains in our experiments, particularly on long-horizon and memory-dependent tasks.

III. Method

A. Problem Statement & Approach Overview

We study robotic manipulation as a sequential decision-making problem under partial observability. At each timestep t, the robot receives a visual observation $o_t \in \mathcal{O}$ and executes an action $a_t \in \mathcal{A}$, which induces a transition in the underlying physical world and produces the next observation o_{t+1}.

Vision-Language-Action (VLA) Policies. Most existing VLA policies learn a direct, reactive mapping from observation history to actions:

$$a_t \sim \pi_\theta(\cdot \mid o_t), \tag{1}$$

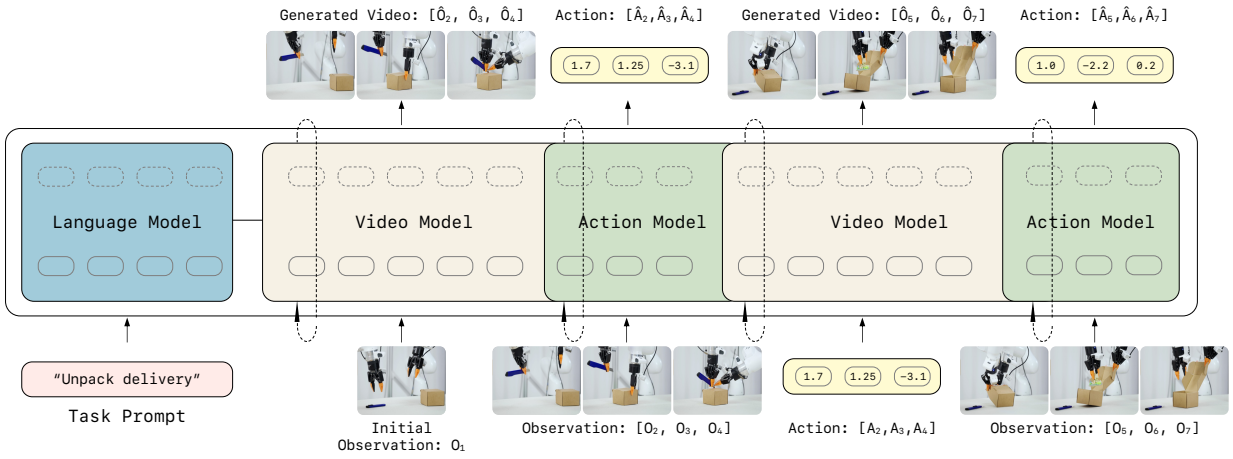

Fig. 2. **Framework overview**: `Lingbot-VA` is conditioned by autoregressive diffusion for unified video-action world modeling. A dual-stream Mixture-of-Transformers (MoT) interleaves video and action tokens. At each step, the video stream (initialized from Wan2.2-5B) predicts future latent visual states via flow matching; the action stream then decodes actions via inverse dynamics conditioned on the predicted transitions.

through imitation learning on robot demonstration data. While this end-to-end approach has shown impressive results, it suffers from a fundamental coupling problem: the model must simultaneously learn visual scene understanding, physical dynamics, and motor control from a single supervision signal of paired observations and actions. This entanglement leads to poor sample efficiency and limited generalization, as quantitatively confirmed by our ablation (table III): removing the explicit video prediction stage collapses success from 92.93% to 48.31%.

Our Approach. Unlike VLA policies that directly learn action distributions, we adopt a world modeling perspective: instead of learning $\pi(a_t \mid o_t)$, we predict how the visual world will evolve, then infer actions based on these predictions. Our approach operates in two stages:

$$\begin{aligned} &\text{(Stage 1) Visual dynamics prediction:} \quad o_{t+1} \sim p_\theta(\cdot \mid o_{\le t}), \\ &\text{(Stage 2) Inverse dynamics:} \quad a_t \sim g_\psi(\cdot \mid o_t, o_{t+1}). \end{aligned} \tag{2}$$

Stage 1 learns to predict future visual observations given observation history. Stage 2 uses an inverse dynamics model to decode actions from desired visual transitions. This decomposition enables Stage 1 to leverage large-scale video data for learning physical priors, while Stage 2 only requires robot demonstrations to ground visual predictions in executable actions. Figure 2 illustrates the details of our framework.

B. Autoregressive Video-Action World Modeling

To leverage rich visual dynamics priors from video data for robot manipulation, we propose a unified video-action world modeling framework that jointly models visual observations and robot actions within a single autoregressive process. Unlike prior approaches that either decouple video prediction from action inference [10, 19] or rely on bidirectional diffusion within segments [59], our method unifies video and action within a single *causal autoregressive* framework, enabling persistent memory through KV cache and seamless integration of real-time observations.

World Dynamics with Autoregressive Modeling. Recent world models for robotics often adopt bidirectional video generation approaches [3, 15, 17, 27], which face fundamental limitations for robot control. The bidirectional attention within each chunk violates causality, preventing seamless integration with real-time observations during execution. They also lack persistent memory across chunks, as each chunk is generated independently without access to the full history, leading to temporal inconsistencies and drift over long horizons [8].

Physical interactions admit a natural causal ordering: future states depend only on the past, motivating an autoregressive treatment of world modeling for control. This causal autoregressive formulation underpins the three properties claimed above—persistent memory via KV cache, causal consistency for streaming inference, and efficient chunk-wise parallelism—without restating them at each layer of abstraction.

We formalize this as an autoregressive process: at each step, the world model predicts the next chunk of K video frames using conditional flow matching [31]:

$$o_{t+1:t+K} \sim p_\theta(\cdot \mid o_{\le t}), \tag{3}$$

where tokens within each chunk are generated in parallel via bidirectional attention, while maintaining causal structure across chunks. This chunk-wise formulation balances generation efficiency with autoregressive flexibility for closed-loop correction.

Video-Action State Encoding. Operating directly on pixel-level video observations is computationally prohibitive due to the high dimensionality and redundancy of raw visual data. We leverage a causal video VAE [48] to compress visual observations into compact latent tokens $z_t = E(o_t \mid o_{<t}) \in \mathbb{R}^{N \times 48}$, where N is the number of spatial tokens after passing into video VAE. By conditioning on previous latent states, the encoder maintains temporal coherence while processing observations sequentially, naturally aligning with our autoregressive world modeling framework. To align robot actions with visual tokens, we project action vectors to token

embeddings $a_t \in \mathbb{R}^D$ via a lightweight MLP $\phi(\cdot)$ where D is the dimension of the video token after patchfication, enabling unified interleaving of visual and action tokens.

Latent Video State Transition. While standard video generation models predict future frames based solely on visual history, robotic manipulation requires accounting for the embodiment's physical state and interaction with the environment. During deployment, the robot's state evolves through continuous interaction: each action modifies the embodiment's configuration (e.g., gripper position, joint angles), which in turn influences how the scene evolves.

In many manipulation settings, actions encode absolute pose information (e.g., end-effector poses in world coordinates), so the action history $a_{<t}$ effectively captures the trajectory of the embodiment's configuration. Conditioning on action history thus provides knowledge of how the robot has moved and interacted with objects. We extend our autoregressive formulation to condition on both observation and action histories:

$$z_{t+1:t+K} \sim p_\theta(\cdot \mid z_{\leq t}, a_{<t}), \tag{4}$$

where z_t is the latent visual state and a_t is the action token.
Inverse Dynamics for Action Decoding. Once the world model predicts future visual states, we leverage these predictions to decode actions. Rather than directly predicting actions from current observations, we employ an inverse dynamics model that infers actions by conditioning on desired future observations, enabling the policy to reason about what action leads to a desired visual outcome.

However, simply conditioning on the current and next states (z_t, z_{t+1}) is insufficient for accurate action prediction. The action history $a_{<t}$ encodes the embodiment's state trajectory for determining feasible actions, while the observation history $z_{<t}$ provides temporal context for multi-step interactions (e.g., whether an object was previously grasped). We therefore formulate inverse dynamics as:

$$a_{t:t+K-1} \sim g_\psi(\cdot \mid \hat{z}_{t+1:t+K}, z_{\leq t}, a_{<t}), \tag{5}$$

where the inverse dynamics model g_ψ takes as input the predicted chunk of visual states $\hat{z}_{t+1:t+K}$ inferred by Eq. 4, observation history $z_{\leq t}$, and action history $a_{<t}$. Here $\hat{z}_{t+1:t+K}$ are the *denoised* latents predicted by the dynamics model in Eq. 4, while $\tilde{z}_{\leq t}$ (defined in Eq. 6 below) is the *noise-augmented* version of the observation history $z_{\leq t}$ used during training.

C. Lingbot-VA: Unified Architecture & Training

Architecture. To jointly model video and action, we utilize an asymmetric dual-stream transformer architecture that performs conditional flow matching for autoregressive prediction. The architecture consists of a video stream (initialized from Wan2.2-5B [48] with dimension d_v) and a parallel action stream with equal depth but significantly smaller width ($d_a \ll d_v$). This asymmetry leverages the lower inherent complexity of action distributions compared to visual dynamics, optimizing parameters while maintaining high expressive capacity.

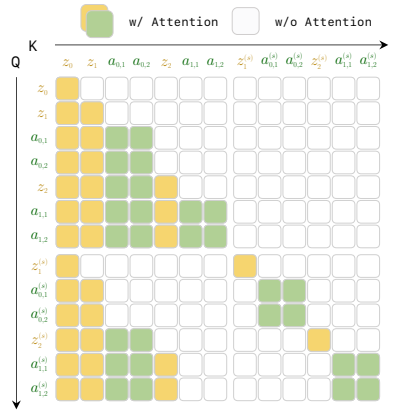

Fig. 3. **Teacher Forcing Attention Mask**: Causal attention mask for unified video-action pretraining. Yellow cells denote attention between video tokens (z), green cells denote attention involving action tokens (a); white cells are masked out. Each token only attends to preceding clean tokens and to its own noisy counterpart ($z^{(s)}$, $a^{(s)}$).

Video Sparsification. To handle temporal redundancy and mismatched sampling rates, we temporally downsample video frames by a factor of $\tau = 4$. While video frames are sparsified for efficiency, actions are maintained at a higher frequency. We interleave each latent video frame $\mathbf{z}_t$ with its τ corresponding consecutive actions $\{\mathbf{a}_{t,1}, \ldots, \mathbf{a}_{t,\tau}\}$ to form a unified temporal sequence: $[\mathbf{z}_t, \mathbf{a}_{t,1}, \ldots, \mathbf{a}_{t,\tau}, \mathbf{z}_{t+1}, \ldots]$. This allows the model to generate τK actions for every K video frames, enabling high-frequency robotic control without the computational overhead of dense video generation.

Mixture-of-Transformer Block. To facilitate cross-modal interaction while preserving modality-specific features, we employ a Mixture-of-Transformers (MoT) architecture [4, 13, 28]. At each layer, video and action streams independently compute their queries, keys, and values using separate projection matrices to maintain distinct feature spaces. To enable fusion, action tokens are linearly projected to the video dimension to participate in joint self-attention. A residual connection then projects these features back to the original action-specific dimension. This design allows bidirectional influence between modalities while preventing interference between their respective representations. Finally, the action stream output is mapped to low-dimensional control vectors via a linear projection head.

Action Network Initialization. Proper initialization of the action stream is critical for training stability and convergence. We find that training the action network from scratch leads to unstable optimization, as the initial divergence between action and video token distributions disrupts the joint attention mechanism. To address this, we initialize the action network by interpolating pretrained video weights to match the action dimensions. We then apply a scaling factor $\alpha = \sqrt{d_v/d_a}$ to preserve output variance, where d_v and d_a are the video and action dimensions, respectively. This strategy ensures action tokens begin with a distribution comparable to video tokens, stabilizing early-stage training and accelerating convergence.

Teacher Forcing for Unified Video-Action Training. We formulate both visual dynamics prediction (Eq. 4) and inverse dynamics (Eq. 5) as autoregressive modeling problems con-

ditioned on historical observations and actions. This unified approach enables a training strategy analogous to next-token prediction in natural language processing. By treating the interleaved video-action sequence as a single stream, we train the model to predict each token conditioned on its predecessors using teacher forcing. During training, causal dependency is strictly enforced via attention masking, ensuring each token only attends to temporally prior context. This formulation is particularly effective for robotics; while teacher forcing in pure generative tasks can lead to distribution mismatch, robot policies receive real-world observations during deployment that align with the training regime. This unified objective offers two primary advantages: 1) it enables the simultaneous end-to-end learning of world dynamics and action inference, and 2) it allows for the parallel optimization of both components across all timesteps in a single forward pass.

Noisy History Augmentation. The primary bottleneck during inference is the computational cost of generating video tokens, which significantly outnumber action tokens and require multiple denoising steps. To mitigate this, we introduce a noise augmentation strategy during training that enables partial denoising at inference. We hypothesize that the inverse dynamics model can extract sufficient action-relevant information from partially noisy video states without requiring high-fidelity, fully denoised visual representations. During training, we augment the video history $\mathbf{z}_{\leq t}$ with noise using the flow matching interpolation scheme:

$$\tilde{z}_{\leq t} = \begin{cases} (1 - s_{\text{aug}})\epsilon + s_{\text{aug}} z_{\leq t}, & \text{with probability } p = 0.5 \\ z_{\leq t}, & \text{with probability } 1 - p = 0.5 \end{cases} \tag{6}$$

where $s_{\text{aug}} \in [0.5, 1], \epsilon \sim \mathcal{N}(0, I)$. This augmentation trains the action decoder to extract control signals directly from partially noisy latent representations. At inference, this enables a significant computational speedup: rather than integrating video tokens across the full flow duration, we terminate the process prematurely. This strategy reduces the required video denoising steps while maintaining high-fidelity action prediction, effectively decoupling visual synthesis quality from control performance.

Training Objective. We jointly optimize both video and action using flow matching with the noisy history augmentation described above. For video tokens z_t, the dynamics loss supervises velocity field prediction conditioned on (potentially noisy) history:

$$\mathcal{L}_{\text{dyn}} = \mathbb{E}_{t,s,z_{t+1},\epsilon}\left[\|v_\theta(z_{t+1}^{(s)}, s, \tilde{z}_{\leq t}, a_{<t}|c) - \dot{z}_{t+1}^{(s)}\|^2\right], \tag{7}$$

where $s \in [0, 1]$ is flow time, $z_{t+1}^{(s)} = (1 - s)\epsilon + s z_{t+1}$ with $\epsilon \sim \mathcal{N}(0, I)$, $\dot{z}_{t+1}^{(s)} = z_{t+1} - \epsilon$, $\tilde{z}_{\leq t}$ is the augmented history of Eq. 6, and c is the language instruction. For action tokens a_t, the inverse dynamics loss conditions on current and next observations:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{t,s,a_t,\epsilon}\left[\|v_\psi(a_t^{(s)}, s, \tilde{z}_{\leq t+1}, a_{<t}|c) - \dot{a}_t^{(s)}\|^2\right], \tag{8}$$

where $a_t^{(s)} = (1 - s)\epsilon + s a_t$ with $\epsilon \sim \mathcal{N}(0, I)$, $\tilde{z}_{\leq t+1} = (\tilde{z}_{\leq t}, \tilde{z}_{t+1})$ denotes the noise-augmented context up through frame $t{+}1$ (with $\tilde{z}_{t+1}$ obtained by the same Eq. 6 scheme), and c is the language instruction. The complete objective is $\mathcal{L} = \mathcal{L}_{\text{dyn}} + \lambda \mathcal{L}_{\text{inv}}$.

KV Cache for Efficient Autoregressive Inference. Our autoregressive formulation naturally enables KV cache acceleration during inference. By caching the key and value pairs of previously processed observation and action tokens, we eliminate redundant computations across the temporal sequence. Consequently, each autoregressive step only requires attention computation for newly generated tokens, drastically reducing latency during long-horizon manipulation tasks.

IV. Implementation Details

A. Dataset Curation and Preprocessing

We curate a large-scale training corpus by aggregating data from seven diverse sources spanning multiple embodiments, environments, and task categories, totaling 16k hours (see App. A for details). All datasets are preprocessed for consistency in format and annotation. We define a Universal Action Interface to unify disparate robot embodiments. Each arm is represented by its end-effector pose—comprising a position and a quaternion (7-D)—and its joint angles. Joint angles for each arm are padded to a 7-D representation where necessary. Including a 1D gripper action per arm, the total dual-arm action space is $\mathbf{a}_t \in \mathbb{R}^{30}$.

B. Implementation & Training Details

We utilize Wan2.2-5B as the backbone for the video stream, while the action stream employs a shared-depth architecture with a reduced hidden dimension, resulting in a total model size of 5.3B parameters. Both streams incorporate RoPE positional encoding and are integrated via the MoT architecture. For visual tokenization, we adopt the Wan2.2 causal VAE. The action encoder and decoder consist of single-layer MLPs, with actions normalized using per-dimension quantile statistics derived from the training set. Task instructions are processed through a frozen T5 text encoder and integrated via cross-attention. During training, noise augmentation is applied with probability $p = 0.5$ and $s_{\text{aug}} \sim \text{Uniform}[0.5, 1.0]$.

For pre-training, we employ the AdamW optimizer with a peak learning rate of 1×10^{-4}, weight decay of 0.01, and a cosine annealing schedule with linear warmup. The model demonstrates high sample efficiency during post-training, where as few as 50 demonstrations are sufficient for effective deployment when fine-tuned for 3k steps at a reduced learning rate of 1×10^{-5}. At inference, we use an Euler solver with separate integration settings: 3 steps for video tokens (integrating to $s = 0.6$) and 10 steps for action tokens (integrating to $s = 1.0$). Details are provided in App. A.

V. Results

A. Real-world Deployment

Experimental Setup. We deploy `Lingbot-VA` on a physical platform and evaluate six tasks across three categories:

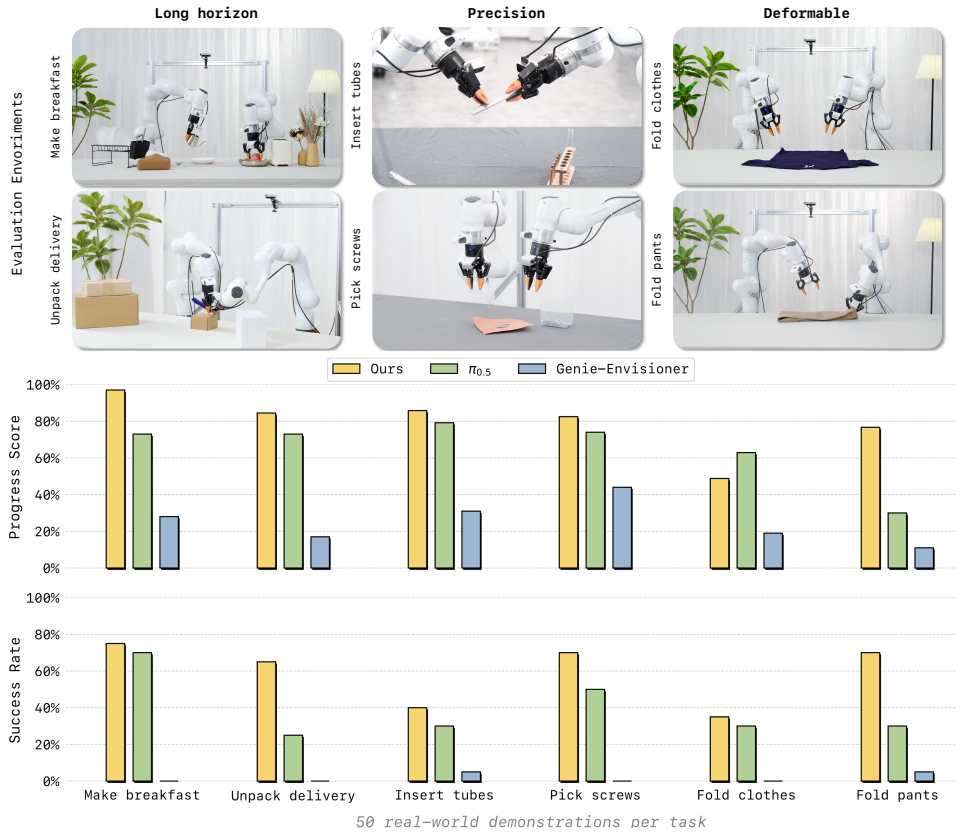

Fig. 4. **Real-world deployment results.** We evaluate `Lingbot-VA` on six manipulation tasks across three categories: long-horizon tasks (Make Breakfast, Pick Screws), precision tasks (Insert Tubes, Unpack Delivery), and deformable & articulated object manipulation (Fold Clothes, Fold Pants). Our method achieves state-of-the-art performance on both metrics.

long-horizon (Make Breakfast, Unpack Delivery), *precision* (Insert Tubes, Pick Screws), and *deformable* (Fold Clothes, Fold Pants). Each task is trained from only 50 real-world demonstrations (500 fine-tuning steps, lr 1×10^{-4}, sequence length 150K). **All baselines ($\pi_{0.5}$ and Genie-Envisioner) are fine-tuned with the same demonstrations, each is evaluated over the same 20 trials with shared randomized object initial poses to ensure a paired comparison. Per-trial scoring rubrics and per-task ±std values are provided in App. A. Metrics.** We report two complementary metrics. *Success Rate* (SR) is a binary indicator that all rubric stages of a task were completed in a single trial. *Progress Score* (PS) measures the fraction of stages completed *strictly before the first failure*. The per-stage rubric is in App. A; detailed task procedures are in Fig. S3. **Results.** As shown in fig. 4, `Lingbot-VA` consistently achieves state-of-the-art performance across all six tasks and both evaluation metrics (success rate and progress score), substantially outperforming strong baseline $\pi_{0.5}$ [20] and Genie-Envisioner [29].

We highlight several key observations that validate our design choices: (1) The superior performance on *long-horizon tasks* demonstrates that our video-action world model possesses strong temporal memory capabilities. By jointly modeling video and action sequences, the model effectively maintains task context over extended horizons, enabling coherent multi-step reasoning without losing track of intermediate goals. (2) The strong results on *precision tasks* validate the effectiveness of our unified latent space design. By aligning video and action representations within a shared embedding space, our model achieves tighter coupling between visual perception and motor control, resulting in more accurate and fine-grained action predictions. (3) The robust performance on *deformable objects* highlights the value of video generation as implicit guidance. The generated video futures provide rich predictive signals about object dynamics and state transitions, which inform the action model to produce more physically plausible manipulation trajectories for challenging non-rigid materials.

B. Simulation Evaluation

Experimental Setup. We evaluate `Lingbot-VA` on two widely-used simulation benchmarks: RoboTwin 2.0 [9] and LIBERO [32], covering diverse manipulation tasks across different robot embodiments. 1) In RoboTwin 2.0, we adopt a multi-task training setup [4] where all models are trained on 2,500 demonstrations collected in clean scenes (50 per task) plus 25,000 demonstrations from heavily randomized scenes (500 per task). We downsample the original 50 Hz video to 12.5 Hz while maintaining the action frequency at 50Hz. The model is trained for 50K steps with a learning rate of 1×10^{-5}. To facilitate a clearer comparison of performance, we categorize the 50 RoboTwin tasks according to their horizons (e.g., *Place Dual Shoes* has two steps, and

TABLE I

EVALUATION ON ROBOTWIN 2.0 SIMULATION (EASY VS HARD, 50 TASKS). ROBOTWIN 2.0 IS A CHALLENGING BIMANUAL MANIPULATION BENCHMARK REQUIRING COORDINATED DUAL-ARM CONTROL. EASY USES FIXED INITIAL CONFIGURATIONS WHILE HARD INVOLVES RANDOMIZED OBJECT POSES AND SCENE LAYOUTS. * X-VLA AND MOTUS NUMBERS ARE COPIED VERBATIM FROM THE MOTUS PAPER [4]. $\pi_{0.5}$ HERE IS REPRODUCED FROM THE OFFICIAL JAX IMPLEMENTATION (RATHER THAN THE PYTORCH PORT USED BY MOTUS); THE JAX VERSION YIELDS HIGHER SUCCESS RATES DUE TO NUMERICAL-PRECISION DIFFERENCES WIDELY REPORTED IN COMMUNITY GITHUB ISSUES. π_0 AND OUR METHOD USE IDENTICAL TRAINING DATA AND SPLIT AS THE OTHER BASELINES. IMPROVEMENTS IN PARENTHESES INDICATE GAINS OVER THE SECOND-BEST METHOD (UNDERLINED).

	X-VLA* [57]		π_0 [6]		$\pi_{0.5}$ [20]		Motus [4]		`Lingbot-VA` (Ours)	
Metric	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Average $_{\text{Horizon = 1}}$	81.6	82.5	66.5	61.6	85.1	80.2	<u>91.0</u>	<u>90.6</u>	**93.8** (+2.8)	**93.1** (+2.5)
Average $_{\text{Horizon = 2}}$	59.3	55.9	66.1	54.7	79.3	73.0	<u>85.2</u>	<u>80.9</u>	**88.8** (+3.6)	**86.2** (+5.3)
Average $_{\text{Horizon = 3}}$	61.2	66.0	61.6	50.2	78.6	67.4	<u>85.0</u>	<u>84.2</u>	**90.4** (+5.4)	**94.2** (+10.0)
Average $_{\text{50 Tasks}}$	72.9	72.8	65.9	58.4	82.7	76.8	<u>88.7</u>	<u>87.0</u>	**92.0** (+3.3)	**91.1** (+4.1)

TABLE II

EVALUATION ON LIBERO BENCHMARKS. LIBERO TESTS MANIPULATION ACROSS FOUR TASK SUITES: SPATIAL, OBJECT, GOAL, AND LONG-HORIZON. OUR METHOD ACHIEVES NEW STATE-OF-THE-ART ON LIBERO-OBJECT (99.6%), LIBERO-LONG (98.5%), LIBERO-SPATIAL (98.5%), AND OVERALL AVERAGE (98.5%). BASELINE RESULTS ARE ADOPTED FROM [57]; $\pi_{0.5}$ NUMBERS REPRODUCED FROM THE OFFICIAL JAX IMPLEMENTATION.

Methods	**LIBERO**				
	Spatial	Object	Goal	Long	Avg
Octo [45]	78.9	85.7	84.6	51.1	75.1
Seer [46]	-	-	-	87.7	-
SpatialVLA [39]	88.2	89.9	78.6	55.5	78.1
GR00T-N1 [5]	94.4	97.6	93.0	90.6	93.9
π_0 [6]	96.8	98.8	95.8	85.2	94.1
π_0+FAST [38]	96.4	96.8	88.6	60.2	85.5
OpenVLA-OFT [23]	97.6	98.4	97.9	94.5	97.1
$\pi_{0.5}$ [20]	**98.8**	98.2	**98.0**	92.4	96.9
X-VLA [57]	98.2	98.6	97.8	97.6	98.1
`Lingbot-VA` (Ours)	98.5 ± 0.3	$\mathbf{99.6 \pm 0.3}$	97.2 ± 0.2	$\mathbf{98.5 \pm 0.5}$	**98.5**

Stack Blocks Three has three steps). The detailed horizons are listed in table I. 2) In LIBERO, we train our model on four LIBERO suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Each suite contains 10 tasks with 50 demonstrations per task (500 total). Following OpenVLA [22], we filter unsuccessful demonstrations before training. The model is finetuned for 4K steps with a learning rate of 1×10^{-5} and a sequence length of 1×10^5. Specifically, we report the average success rate over three random seeds, with each seed comprising 500 evaluation trials (totally $3 \times 500 = 1500$) for every task suite.

Results. RoboTwin 2.0 is a challenging bimanual manipulation benchmark featuring over 50 tasks that require coordinated dual-arm control. We evaluate under both *Easy* (fixed initial configurations) and *Hard* (varied object poses and scene layouts) settings. As shown in table I, `Lingbot-VA` achieves an average success rate of 92.0% (Easy) and 91.1% (Hard), substantially outperforming prior methods including π_0, $\pi_{0.5}$, X-VLA, and Motus. Notably, the improvement becomes more pronounced for longer-horizon tasks: at Horizon = 3, our method achieves gains of +5.4% (Easy) and +10.0% (Hard) over the second-best approach. This suggests that our autoregressive mechanism effectively maintains long-range temporal memory, enabling more robust performance as task complexity increases. We further evaluate on LIBERO benchmark (table II). On LIBERO, we obtain an average success rate of 98.5%, with particularly strong performance on LIBERO-Long (98.5%). These results establish new state-of-the-art performance in average success rates among foundational VLAs, demonstrating the effectiveness of our video-action world model for generalist robot control.

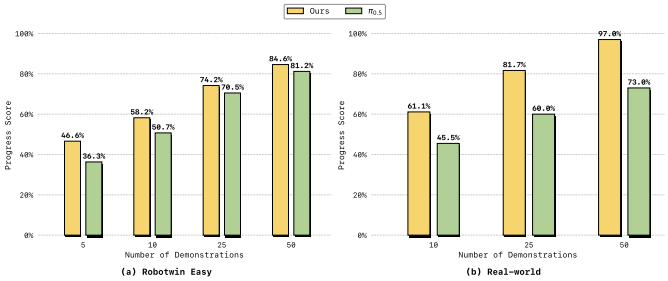

Fig. 5. **Sample efficiency comparison**. `Lingbot-VA` consistently outperforms $\pi_{0.5}$ across various data regimes on the "Make Breakfast" task, demonstrating superior data efficiency in the post-training stage.

C. Analysis

Sample Efficiency. We investigate data efficiency by comparing `Lingbot-VA` against $\pi_{0.5}$ in low-data regimes on both real-world ("Make Breakfast" long-horizon task) and simulation (RoboTwin 2.0 Easy) settings. As shown in fig. 5, our method consistently outperforms $\pi_{0.5}$ across all data regimes; in the lowest-data regime (10 demonstrations), `Lingbot-VA`

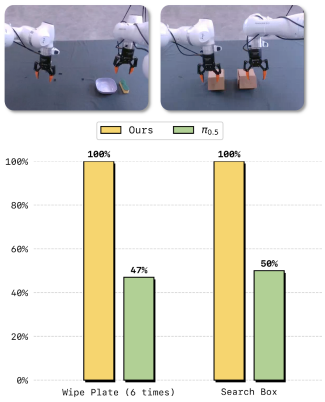

Fig. 6. **Temporal memory evaluation.** Success rates on two memory tasks (Wipe Plate and Search Box). `Lingbot-VA` significantly outperforms $\pi_{0.5}$ on both tasks, demonstrating superior temporal state tracking ability.

TABLE III
ABLATION STUDIES ON ROBOTWIN 2.0 (50 TASKS).

Variant	Avg. Success (%)
`Lingbot-VA` (Baseline, 25 denoise steps)	**92.93**
w/o MoT (shared weights)	92.15
w/o video prediction (action-only)	48.31
w/o causal (bidirectional)	81.46
Denoise 15 steps (noisy-latent aug.)	92.91

achieves 15.6% higher progress score than $\pi_{0.5}$ on Make Breakfast and 10.3% higher on RoboTwin 2.0 Easy. We attribute this to the visual-dynamics priors transferred from the pretrained video backbone, which serve as implicit regularization that VLA models like $\pi_{0.5}$—lacking explicit dynamics modeling—cannot leverage.

Temporal Memory. We design the following tasks that explicitly require maintaining state information across time to evaluate our model's temporal memory capabilities, as shown in Figure S4: 1) *Wipe Plate*—the robot must wipe a plate exactly six times, requiring it to count and remember repeated actions; 2) *Search Box*—Two boxes (left and right) are in the scene, with only one containing a block. The robot opens them sequentially from right to left. In data collection, the block is equally likely to be in either box; at test time, it is always in the left box. Without memory, after finding the right box empty, the model has a 50% chance of re-opening it. With memory, it proceeds to search the left box. As shown in fig. 6, `Lingbot-VA` substantially outperforms $\pi_{0.5}$ on both memory tasks. We attribute this to the autoregressive nature of our world model: during training, teacher forcing conditions predictions on full history; during inference, KV-cache naturally preserves all historical information for persistent memory.

D. Ablation Studies

We conduct ablations for several key design choices as shown in III. Numbers are average success rate (%).

Mixture-of-Transformers. As shown in Appendix Fig. S1, MoT improves training stability (lower gradient norm) and accelerates convergence compared to a shared-weights baseline. Quantitatively, replacing MoT with a shared-weights transformer drops the average success rate from 92.93% to 92.15% (table III).

Explicit video prediction. Removing the video-prediction stage and using video only as an auxiliary training signal (*w/o video prediction*) collapses success to 48.31%. This validates our *decoupled* formulation: explicitly forecasting next-frame latents lets the action stream inherit dynamics priors from the pretrained video backbone, and turns a hard implicit policy-learning problem into a tractable conditional inference.

Causal formulation. A bidirectional variant drops to 81.46%. Qualitatively, bidirectional models also produce videos with temporally inconsistent artifacts that violate causal physics (fig. S1 in Appendix). This confirms that the causal formulation is a critical design choice.

Noisy-latent augmentation. Reducing video denoising from 25 to 15 steps yields 92.91% (vs. 92.93% baseline), confirming that the augmentation effectively decouples action quality from full visual fidelity. We further reduce to 5 steps in real-world deployment to maximize control rate.

Inference latency. On a single RTX 5880 Ada GPU, each closed-loop step of `Lingbot-VA` takes ≈ 0.5 s, comprising 5 video-denoising steps for 4 video latent frames and 10 action-denoising steps for 16 action frames. With our temporal sparsification (τ=4), this corresponds to an effective control rate of ≈ 32 Hz on the 16 predicted actions in each cycle.

VI. CONCLUSION

We present `Lingbot-VA`, an autoregressive diffusion framework that unifies video dynamics prediction and action inference for robotic manipulation. By interleaving video and action tokens within a Mixture-of-Transformers architecture, our model captures the causal structure of physical interactions while enabling closed-loop control through continuous integration of real-world observations. Extensive evaluation demonstrates strong performance across simulation benchmarks (92.0% on RoboTwin 2.0, 98.5% on LIBERO) and real-world deployment, achieving over 20% improvement on challenging tasks compared to $\pi_{0.5}$ with only 50 demonstrations for adaptation. These results suggest that autoregressive video-action world modeling provides a principled foundation for learning generalizable manipulation policies, offering a compelling alternative to reactive VLA paradigms.

Limitations. Our framework has three known limitations. (1) History-aware policies can overfit to dataset dynamics patterns and may degrade under severe out-of-distribution conditions (e.g., human disturbances reverting task progress). (2) Although noisy-latent augmentation reduces denoising steps, the video backbone still imposes higher latency than VLA policies without video predictions. (3) Pretraining requires substantial compute (see App. A), which may limit reproducibility in resource-constrained settings.

Future directions include developing more efficient video compression schemes to reduce computational overhead and incorporating multi-modal sensory inputs (tactile, force, audio) for more robust manipulation in tasks with complex contact dynamics.

References

[1] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[2] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *Int. Conf. Learn. Represent.*, 2025.

[3] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[4] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[5] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Jim Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, 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, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[8] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Adv. Neural Inform. Process. Syst.*, 2024.

[9] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, 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.

[10] Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *Int. Conf. Comput. Vis.*, 2025.

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

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

[13] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[14] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.

[15] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Adv. Neural Inform. Process. Syst.*, 2023.

[16] Google DeepMind. Veo: A text-to-video generation system. *Google DeepMind Technical Report*, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

[17] 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. In *Adv. Neural Inform. Process. Syst.*, 2024.

[18] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.

[19] 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. In *Int. Conf. Mach. Learn.*, 2025.

[20] Physical Intelligence et al. $\pi_{0.5}$: A generalist robot policy with flow matching and world models. In *Conference on Robot Learning (CoRL)*, 2025. URL https://arxiv.org/abs/2504.16054.

[21] Dong Jing, Gang Wang, Jiaqi Liu, Weiliang Tang, Zelong Sun, Yunchao Yao, Zhenyu Wei, Yunhui Liu, Zhiwu Lu, and Mingyu Ding. Mixture of horizons in action chunking. *arXiv preprint arXiv:2511.19433*, 2025.

[22] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov,

Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

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

[24] Kuaishou. Kling ai. https://klingai.kuaishou.com/, 2024.

[25] Jiacheng Li, Mengzhou Sun, Bowen Zhang, Zhe Zhao, Xiu Liu, et al. Gr-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025.

[26] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems*, 2025.

[27] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Conference on Robot Learning (CoRL)*, 2024.

[28] Weixin Liang, LILI YU, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *Transactions on Machine Learning Research*, 2025.

[29] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation, 2025. URL https://arxiv.org/abs/2508.05635.

[30] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *Int. Conf. Learn. Represent.*, 2025.

[31] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Int. Conf. Learn. Represent.*, 2023.

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

[33] Fangchen Liu, Chuanyu Li, Yihua Qin, Austin Shaw, Jing Xu, Pieter Abbeel, and Rui Chen. Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface. *arXiv preprint arXiv:2504.06156*, 2025.

[34] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: A diffusion foundation model for bimanual manipulation. In *Int. Conf. Learn. Represent.*, 2025.

[35] Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Benjamin Burchfiel, and Shuran Song. Maniwav: Learning robot manipulation from in-the-wild audio-visual data. *arXiv preprint arXiv:2406.19464*, 2024.

[36] Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[37] OpenAI. Video generation models as world simulators. *OpenAI Technical Report*, 2024. URL https://openai.com/index/video-generation-models-as-world-simulators/.

[38] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems*, 2025.

[39] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*, 2025.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[41] Omar Rayyan, John Abanes, Mahmoud Hafez, Anthony Tzes, and Fares Abu-Dakka. Mv-umi: A scalable multi-view interface for cross-embodiment learning. *arXiv preprint arXiv:2509.18757*, 2025.

[42] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. In *Conference on Robot Learning (CoRL)*, 2025.

[43] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[44] Gemini Robotics Team, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Fangchen Liu, Anirudha Majumdar, et al. Evaluating gemini robotics policies in a veo world simulator. *arXiv preprint arXiv:2512.10675*, 2025.

[45] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

[46] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Seer: Predictive inverse dynamics models are scalable learners for robotic manipulation. In *Int. Conf. Learn. Represent.*, 2025.

[47] Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, Yaping Li, Ping Wang, Junhao Cai, Jia Zeng, Hao Dong, et al. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025.

[48] WanTeam. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[49] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Shichao Fan, Xinhua Wang, Fei Liao, Zhen Zhao, Guangyu Li, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems*, 2025.

[50] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning (CoRL)*, 2022.

[51] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, Zhaoye Long, Yue Wang, Chong Liu, Dihan Wang, Ziqiang Ni, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

[52] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent. In *Int. Conf. Mach. Learn.*, 2025.

[53] 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.

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

[55] Zhaxizhuoma, Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, Haoming Song, Delin Qu, Dong Wang, Zhigang Wang, Nieqing Cao, et al. Fastumi: A scalable and hardware-independent universal manipulation interface. *arXiv preprint arXiv:2409.19499*, 2024.

[56] 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. In *Int. Conf. Mach. Learn.*, 2024.

[57] 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.

[58] Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025.

[59] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems*, 2025.

APPENDIX

ADDITIONAL IMPLEMENTATION DETAILS

Training Data Composition. We aggregate data from six sources spanning diverse embodiments, environments, and task categories:

- **Agibot** [1]: Large-scale dataset with diverse manipulation tasks from mobile manipulators.
- **RoboMind** [49]: Multi-embodiment manipulation demonstrations.
- **InternData-A1** [47]: Large-scale simulation dataset for sim-to-real transfer.
- **OXE** [36]: Multi-embodiment dataset; we use the OpenVLA subset.
- **UMI Data** [12, 30, 33, 35, 41, 55]: Human demonstration dataset collected via universal manipulation interface[1], excluding DexUMI.
- **RoboCOIN** [51]: Cross-embodiment bimanual robotics data.

In total, our training corpus comprises approximately 16K hours of robot manipulation data across diverse tasks and environments, including our own self-collected demonstrations.

Architecture. We use Wan2.2-5B as the backbone for the video stream, with hidden dimension $d_v = 3072$ and 30 transformer layers. The action stream shares the same depth but uses a reduced hidden dimension $d_a = 768$ ($4\times$ smaller), resulting in approximately 350M additional parameters and a total model size of 5.3B parameters. Both streams employ RoPE positional encoding and are connected via the MoT architecture described in §III-C. We adopt the Wan2.2 causal VAE for tokenization with a $4 \times 16 \times 16$ (temporal $\times$ height $\times$ width) compression ratio, combined with a patchify operation that further reduces spatial dimensions by 2. The encoded views are concatenated along the width dimension, resulting in a total of $N = 192$ spatial tokens per frame. The action encoder ϕ and decoder are implemented as single-layer MLPs with hidden dimension 256. We normalize actions using per-dimension quantile normalization statistics computed from the training set. Task instructions are encoded using a frozen T5 text encoder [40] and injected via cross-attention. During training, chunk size K is randomly sampled from $[1, 4]$.

Pre-Training. We pretrain `Lingbot-VA` on the curated dataset for 1.4T tokens. We use the AdamW optimizer with peak learning rate 1×10^{-4}, weight decay 0.01, and cosine annealing schedule with linear warmup. Training is conducted in bfloat16 mixed precision with gradient clipping at 2.0. We apply classifier-free guidance with text dropout rate 0.1. The loss weight λ for inverse dynamics is set to 1. The dataset is sampled uniformly across all sources to ensure balanced learning. We monitor convergence using flow matching loss on the validation set. We use uniform SNR sampler for video model. For both video and action model, we use a uniform SNR sampler.

[1]https://umi-data.github.io/

Post-Training. While the pretrained model exhibits zero-shot generalization to seen embodiments, adapting to novel robot platforms requires a small amount of task-specific data. We find that post-training with as few as 50 demonstrations is sufficient for effective deployment. We use a reduced learning rate of 1×10^{-5} and train for 3K steps, which yields robust performance. Alternatively, a higher learning rate of 1×10^{-4} with 1K steps also produces reasonable results, though slightly inferior, offering a faster adaptation option when computational resources are limited.

Inference. For inference, we use Euler solver with 3 steps for video tokens (integrating to $s = 0.6$) and 10 steps for action tokens (integrating to $s = 1.0$). Video CFG scale is set to 5.0, while action CFG scale is set to 1.0. During training, noise augmentation is applied with probability $p = 0.5$ and $s_{\text{aug}} \sim \text{Uniform}[0.5, 1.0]$. Following LLM practices, we pack multiple episodes into long sequences (up to 10K tokens) with attention masks.

ABLATION STUDIES

The main-text Ablation Studies (Sec. V-D, Table III) cover the three most consequential design choices: explicit video prediction, causal masking, and noisy-latent augmentation. We provide additional ablations below.

Pretrained `Lingbot-VA` v.s. WAN. To validate the design choices in our video-action architecture, we conduct a controlled ablation study comparing our pretrained `Lingbot-VA` model with WAN (Wan2.2-5B) as the initialization baseline for fine-tuning on RoboTwin tasks. Both models are fine-tuned on the same RoboTwin dataset using identical post-training procedures (50 task-specific demonstrations, learning rate 1×10^{-5}, 3K steps).

As shown in table S1, our pretrained `Lingbot-VA` model substantially outperforms WAN fine-tuning across both Easy and Hard settings. Specifically, `Lingbot-VA` achieves an average success rate of 92.10% (Easy) and 91.12% (Hard), while WAN fine-tuning yields significantly lower performance. This performance gap highlights the effectiveness of our joint video-action pretraining strategy, which endows the model with rich visual-motor priors that facilitate fast adaptation to complex bimanual manipulation tasks.

Action Network Initialization. Proper initialization of the action stream is critical for training stability and convergence. We compare our curated initialization strategy (Section III-C) with naive random initialization.

As shown in fig. S2, random initialization from scratch exhibits volatile training dynamics with significantly slower convergence. This instability arises because action tokens' output distribution initially diverges dramatically from the video distribution, disrupting the joint attention mechanism in our unified architecture. In contrast, our curated initialization strategy—where action network weights are initialized by interpolating pretrained video weights with a scaling factor $\alpha = \sqrt{d_v/d_a}$—produces smooth convergence and substantially lower loss.

TABLE S1

ADDITIONAL ABLATIONS ON ROBOTWIN 2.0 (EASY). DEPLOYMENT MODE (ASYNC VS. SYNC) AND PRETRAINING (OURS VS. WAN). THE MAIN-TEXT ABLATION TABLE COVERS MOT, VIDEO PREDICTION, CAUSAL FORMULATION, AND DENOISING STEPS.

Ablation	**Setting**	**Easy**$_{\text{all}}$	**Easy** $_{\text{Horizon = 1}}$	**Easy** $_{\text{Horizon = 2}}$	**Easy** $_{\text{Horizon = 3}}$
Baseline	`Lingbot-VA` (Ours)	92.0	93.8	88.8	90.4
Deployment	FDM-grounded Async	90.4	92.5	87.7	85.6
	Naive Async	74.3	83.3	70.3	32.9
Pretrain	WAN	80.6	84.9	76.3	67.6

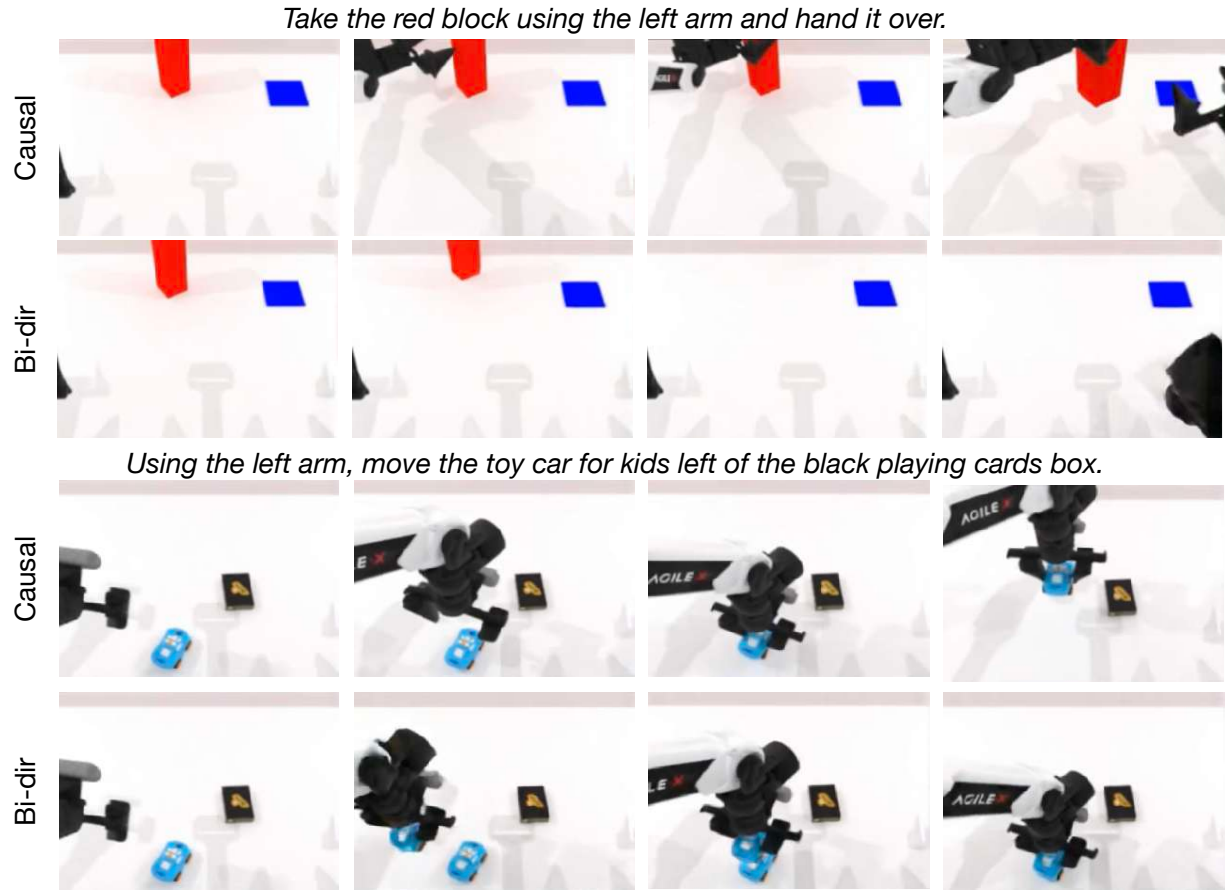

Fig. S1. **Causal vs. bidirectional video quality.** Bidirectional models produce videos with temporally inconsistent artifacts that violate physical causality, whereas our causal formulation generates videos respecting correct temporal logic.

REAL-WORLD EVALUATION DETAILS

The detailed procedures of the real-world tasks are summarized in Fig. S3. We present detailed evaluation results for all real-world manipulation tasks. Each task is evaluated with 20 trials for both our method and the baseline ($\pi_{0.5}$). To ensure fair comparison, we adopt an alternating evaluation protocol: one trial with $\pi_{0.5}$, followed by one trial with our method, and so on.

For each trial, we record the success status of every intermediate step. If a step requires a retry to succeed, we assign a score of 0.5; if it fails, the score is 0; if it succeeds on the first attempt, the score is 1. A trial is marked as successful only if all steps are completed (i.e., the total score equals the maximum possible score).

We report two metrics:

- **Progress Score (PS)**: The average score across all trials divided by the maximum possible score, expressed as a percentage: $\text{PS} = \frac{\text{Average Progress}}{\text{Max Steps}} \times 100\%$.
- **Success Rate (SR)**: The number of successful trials divided by the total number of trials, expressed as a percentage: $\text{SR} = \frac{\text{\# Successful Trials}}{N} \times 100\%$.

We evaluate on six diverse real-world tasks: **Make Breakfast** (10 steps: preparing a complete breakfast including toasting bread, pouring water, and plating), **Pick Screws** (5 steps: picking up paper, pouring screws, and inserting three screws), **Fold Clothes** (6 steps: folding a shirt including sleeves and

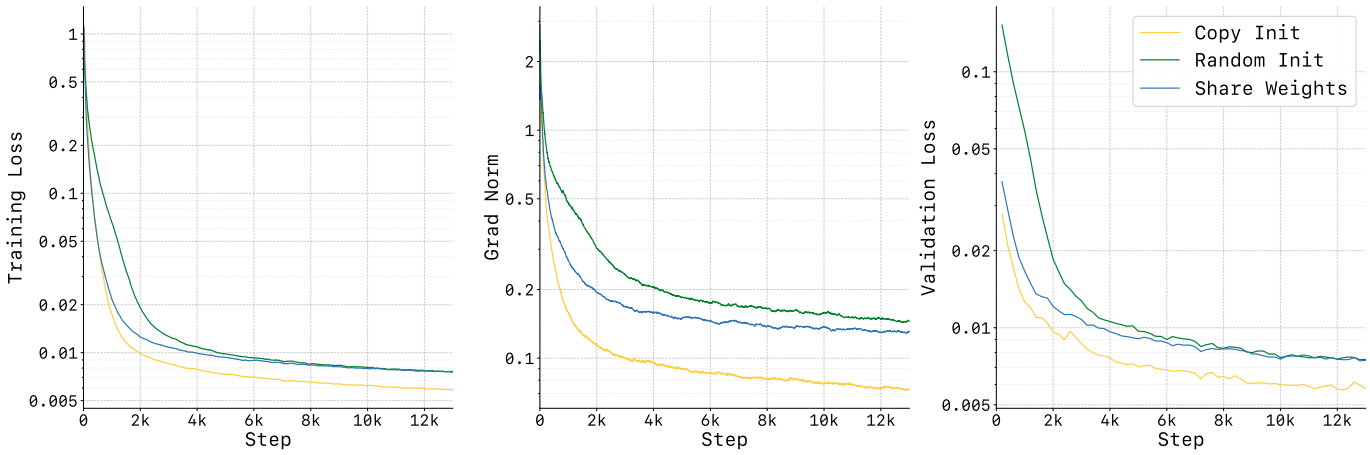

Fig. S2. **Training dynamic comparison between different action network initialization streategy**: Random initialization leads to unstable optimization (high gradient norms) and slow convergence. Although re-using video network weights stabilizes training, the resulting performance is not optimal. Our approach, which initializes by copying pretrained video weights with proper scaling, proves to be the most effective, ensuring smooth training dynamics and faster convergence.

smoothing), **Unpack Delivery** (5 steps: opening a package using a utility knife), **Insert Tubes** (2 categories: grasping and inserting 3 tubes), and **Fold Pants** (3 steps: folding pants and placing them). These tasks span long-horizon sequential manipulation, precision control, and deformable object handling. The following tables present per-trial results for each task.

Fig. S3. **Detailed task progressions and key execution steps of the six real-world tasks.** Each task involves a sequence of manipulation primitives, with scoring criteria detailed in Tables S3 through S5.

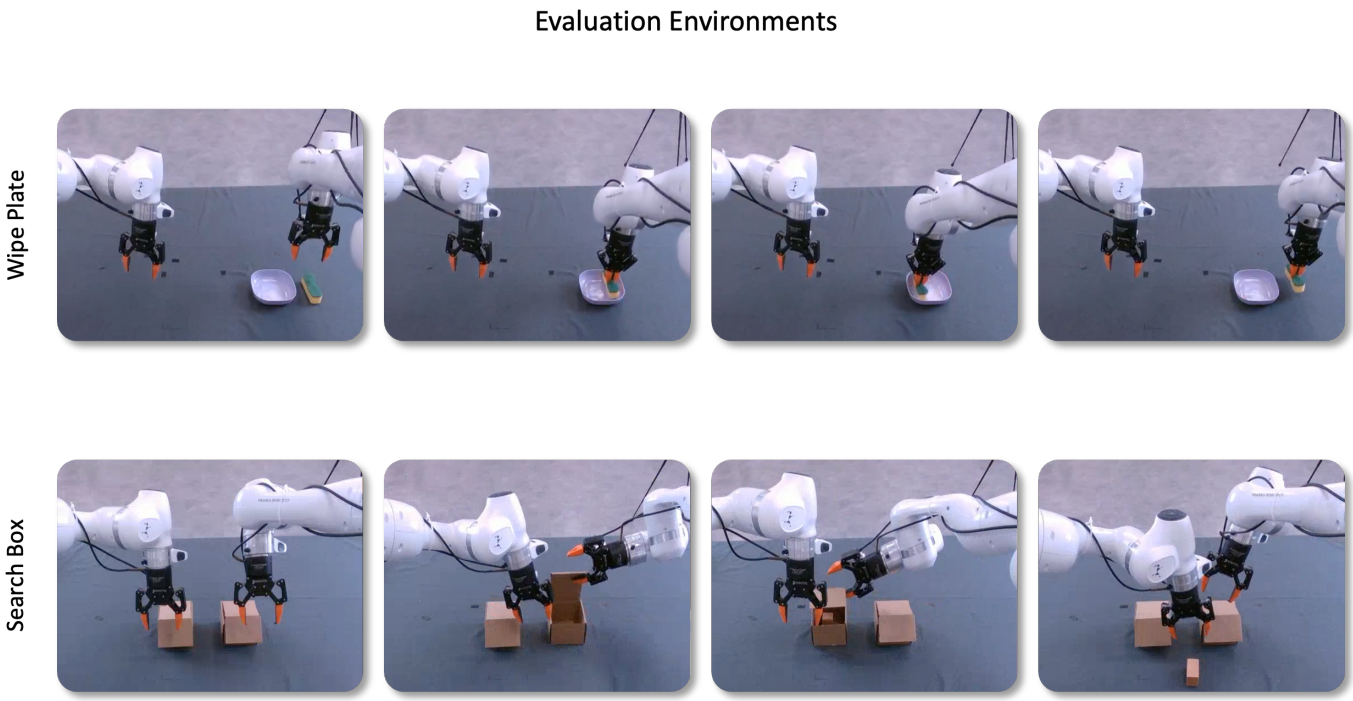

Fig. S4. **Temporal memory evaluation.** Task process demonstrations.

TABLE S2

Evaluation on RoboTwin 2.0 Simulation (Easy vs Hard, 50 tasks). ROBOTWIN 2.0 IS A CHALLENGING BIMANUAL MANIPULATION BENCHMARK REQUIRING COORDINATED DUAL-ARM CONTROL. EASY USES FIXED INITIAL CONFIGURATIONS WHILE HARD INVOLVES RANDOMIZED OBJECT POSES AND SCENE LAYOUTS.

Simulation Task	**Horizon**	**Ours**		π_0 [6]		$\pi_{0.5}$ [6]		X-VLA [57]		Motus [4]	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	97%	96%	99%	95%	**100%**	**99%**	**100%**	**99%**	89%	93%
Beat Block Hammer	1	92%	**97%**	79%	84%	**96%**	93%	92%	88%	95%	88%
Blocks Ranking RGB	3	98%	**99%**	80%	63%	92%	85%	83%	83%	**99%**	97%
Blocks Ranking Size	3	**92%**	**93%**	14%	5%	49%	26%	67%	74%	75%	63%
Click Alarmclock	1	**100%**	**100%**	77%	68%	98%	89%	99%	99%	**100%**	**100%**
Click Bell	1	**100%**	**100%**	71%	48%	**99%**	66%	**100%**	**100%**	**100%**	**100%**
Dump Bin Bigbin	1	93%	**97%**	88%	83%	92%	**97%**	79%	77%	**95%**	91%
Grab Roller	1	**100%**	**100%**	98%	94%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Handover Block	2	**100%**	91%	47%	31%	66%	57%	73%	37%	86%	73%
Handover Mic	2	87%	**91%**	**97%**	**97%**	**98%**	**97%**	0%	0%	78%	63%
Hanging Mug	2	**27%**	**21%**	14%	11%	18%	17%	23%	27%	**38%**	**38%**
Lift Pot	1	**100%**	**99%**	80%	72%	96%	85%	**99%**	**100%**	96%	**99%**
Move Can Pot	1	92%	**94%**	**68%**	48%	51%	55%	89%	86%	34%	74%
Move Pillbottle Pad	1	**99%**	**98%**	67%	46%	84%	61%	73%	71%	93%	**96%**
Move Playingcard Away	1	**99%**	**99%**	74%	65%	96%	84%	93%	**98%**	**100%**	96%
Move Stapler Pad	1	74%	**70%**	41%	24%	56%	42%	78%	73%	**83%**	**85%**
Open Laptop	1	**97%**	**100%**	71%	81%	90%	96%	93%	**100%**	95%	91%
Open Microwave	1	75%	85%	4%	32%	34%	**77%**	**79%**	71%	**95%**	**91%**
Pick Diverse Bottles	2	**93%**	88%	69%	31%	81%	71%	58%	36%	90%	**91%**
Pick Dual Bottles	2	**100%**	**94%**	59%	37%	93%	63%	47%	36%	96%	90%
Place A2B Left	1	**97%**	94%	43%	47%	87%	82%	48%	49%	82%	79%
Place A2B Right	1	**97%**	93%	39%	34%	87%	84%	36%	36%	90%	**87%**
Place Bread Basket	1	**95%**	90%	62%	46%	77%	64%	81%	71%	91%	**94%**
Place Bread Skillet	2	**96%**	90%	66%	49%	**85%**	66%	77%	67%	86%	**83%**
Place Burger Fries	2	92%	**98%**	81%	76%	**94%**	87%	**94%**	**94%**	**98%**	**98%**
Place Can Basket	2	**90%**	**88%**	55%	46%	**62%**	**62%**	49%	52%	**81%**	76%
Place Cans Plasticbox	2	**99%**	96%	63%	45%	94%	84%	**97%**	**98%**	**98%**	94%
Place Container Plate	1	**97%**	95%	**97%**	92%	**99%**	95%	**97%**	95%	98%	**99%**
Place Dual Shoes	2	**96%**	88%	59%	51%	75%	75%	79%	**88%**	93%	87%
Place Empty Cup	1	**100%**	**100%**	91%	85%	**100%**	99%	**100%**	98%	99%	98%
Place Fan	1	91%	**92%**	66%	71%	**87%**	85%	80%	75%	**91%**	87%
Place Mouse Pad	1	**96%**	**94%**	20%	20%	**60%**	39%	**70%**	**70%**	66%	68%
Place Object Basket	2	**83%**	84%	67%	70%	80%	76%	44%	39%	81%	**87%**
Place Object Scale	1	**99%**	**94%**	57%	52%	**86%**	80%	52%	74%	88%	85%
Place Object Stand	1	**99%**	**97%**	82%	68%	91%	85%	86%	88%	**98%**	**97%**
Place Phone Stand	1	**97%**	**97%**	49%	53%	81%	81%	**88%**	87%	87%	86%
Place Shoe	1	**99%**	**96%**	76%	76%	92%	93%	**96%**	95%	**99%**	**97%**
Press Stapler	1	81%	87%	44%	37%	87%	83%	**92%**	**98%**	93%	**98%**
Put Bottles Dustbin	3	**77%**	**100%**	65%	56%	**84%**	79%	74%	77%	81%	79%
Put Object Cabinet	2	79%	78%	73%	60%	**80%**	**79%**	46%	48%	**88%**	71%
Rotate QRcode	1	**96%**	**92%**	74%	70%	89%	87%	34%	33%	**89%**	73%
Scan Object	2	**95%**	**88%**	55%	42%	72%	65%	14%	36%	67%	66%
Shake Bottle Horizontally	1	**100%**	**99%**	98%	92%	**99%**	**99%**	**100%**	**100%**	**100%**	98%
Shake Bottle	1	**100%**	98%	94%	91%	**99%**	97%	**99%**	**100%**	**100%**	**97%**
Stack Blocks Three	3	**100%**	**97%**	72%	52%	91%	76%	6%	10%	91%	**95%**
Stack Blocks Two	2	**100%**	**100%**	93%	79%	97%	**100%**	92%	87%	**100%**	98%
Stack Bowls Three	3	**85%**	**82%**	**77%**	75%	**77%**	71%	76%	**86%**	79%	**87%**
Stack Bowls Two	2	95%	**98%**	94%	95%	95%	96%	96%	93%	**98%**	**98%**
Stamp Seal	1	**98%**	**96%**	46%	33%	79%	55%	76%	82%	93%	92%
Turn Switch	1	55%	43%	41%	42%	**62%**	54%	40%	61%	**84%**	**78%**
Average (%)	–	**91.99**	**91.11**	65.92	58.40	82.74	76.76	72.80	72.84	88.66	87.02

TABLE S3
DETAILED EVALUATION RESULTS FOR MAKE BREAKFAST TASK (10 STEPS, MAX SCORE 10).

Trial	Succ.	Grasp Plate	Grasp Bread	Grasp Fork	Place Bread	Press Toaster	Grasp Cup	Grasp Kettle	Pour	Grasp Apple	Serve	Prog.
Ours												
1	0	1	1	1	1	1	1	1	0	1	1	9
2	1	1	1	1	1	1	1	1	1	1	1	10
3	1	1	1	1	1	1	1	1	1	1	1	10
4	1	1	1	1	1	1	1	1	1	1	1	10
5	1	1	1	1	1	1	1	1	1	1	1	10
6	1	1	1	1	1	1	1	1	1	1	1	10
7	0	1	1	1	1	1	1	1	1	1	0	9
8	1	1	1	1	1	1	1	1	1	0.5	1	9.5
9	0	1	1	1	0	1	1	1	1	1	1	9
10	1	1	1	1	1	1	1	1	1	1	1	10
11	0	1	1	1	1	1	1	1	0	1	1	9
12	1	1	1	1	1	1	1	1	1	1	1	10
13	0	1	1	1	0	1	1	1	1	1	1	9
14	1	1	1	1	1	1	1	1	1	1	1	10
15	1	1	1	1	1	1	1	1	1	1	1	10
16	1	1	1	1	1	1	1	1	1	1	1	10
17	1	1	1	1	1	1	1	1	1	0.5	1	9.5
18	1	1	1	1	1	1	1	1	1	1	1	10
19	1	1	1	1	1	1	1	1	1	1	1	10
20	1	1	1	1	1	1	1	1	1	1	1	10
Avg	**0.75**	**1.00**	**1.00**	**1.00**	**0.90**	**1.00**	**1.00**	**1.00**	**0.90**	**0.95**	**0.95**	**9.70**
Progress Score	**97.0%** (= 9.70/10 × 100%)											
Success Rate	**75.0%** (= 15/20 × 100%)											
$\pi_{0.5}$												
1	0	1	1	1	1	0	0	0	0	0	0	4
2	1	1	1	1	1	1	1	1	0	1	1	9
3	1	1	1	1	1	1	1	1	0	1	1	9
4	1	1	1	1	1	0	0	1	1	1	1	8
5	0	1	1	1	0	1	0	1	1	0	1	7
6	1	1	1	1	1	1	1	1	0	1	1	9
7	1	1	1	0	1	1	1	1	0	1	1	8
8	0	1	1	1	1	1	1	1	1	1	0	9
9	0	1	1	0	1	0	1	1	0	0	0	5
10	0	1	1	1	1	0	0	0	0	0	0	4
11	1	1	1	1	1	0	0	1	1	1	1	8
12	1	1	1	1	0	0	0	1	1	1	1	7
13	1	1	1	0	1	0	0	1	1	1	1	7
14	1	1	1	1	0	1	0	1	1	1	1	8
15	1	1	1	1	1	0	0	1	1	1	1	8
16	1	1	1	1	0	1	0	1	1	1	1	8
17	1	1	1	1	1	1	0	1	1	1	1	9
18	1	1	1	1	1	1	1	0	0	1	1	8
19	1	1	1	0	1	0	0	1	0	1	1	6
20	0	1	1	1	0	1	0	0	0	1	0	5
Avg	**0.70**	**1.00**	**1.00**	**0.80**	**0.75**	**0.55**	**0.35**	**0.80**	**0.50**	**0.80**	**0.75**	**7.30**
Progress Score	**73.0%** (= 7.30/10 × 100%)											
Success Rate	**70.0%** (= 14/20 × 100%)											
GenieEnvisioner												
1	0	1	1	0	1	0	0	0	0	0	0	3
2	0	1	1	0	1	1	0	0	0	0	0	4
3	0	1	1	1	1	0	1	1	0	0	0	6
4	0	1	1	0	0.5	1	0	0	0	0	0	3.5
5	0	1	1	0	0.5	1	0	0	0	0	0	3.5
6	0	0	1	0	0	0	0	0	0	0	0	1
7	0	1	1	0	1	1	0	0	0	0	0	4
8	0	1	0	0	0	0	0	0	0	0	0	1
9	0	1	1	0	1	0	0	0	0	0	0	3
10	0	1	1	0	1	0	1	0	0	0	0	4
11	0	1	1	0	0	0	0	0	0	0	0	2
12	0	1	0	0	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	1	0	0	0	0	0	0	2
15	0	0	1	0	0	0	0	0	0	0	0	1
16	0	1	1	0	1	1	0	0	0	0	0	4
17	0	1	1	0	0	1	0	0	0	0	0	3
18	0	1	1	0	1	0	1	1	0	0.5	0	5.5
19	0	1	1	0	1	0	0	0	0	0	0	3
20	0	1	0	0	0	0	0	0	0	0	0	1
Avg	**0.00**	**0.80**	**0.80**	**0.05**	**0.55**	**0.30**	**0.15**	**0.10**	**0.00**	**0.025**	**0.00**	**2.775**
Progress Score	**27.75%** (= 2.775/10 × 100%)											
Success Rate	**0.0%** (= 0/20 × 100%)											

TABLE S4
DETAILED EVALUATION RESULTS FOR PICK SCREWS TASK (5 STEPS, MAX SCORE 5).

Trial	Success	Grab Paper	Pour Screws	Screw 1	Screw 2	Screw 3	Progress
Ours							
1	1	1	1	0.5	0.5	1	4
2	0	1	0	1	1	0.5	3.5
3	1	1	1	1	0.5	0.5	4
4	1	1	1	1	1	1	5
5	1	1	1	1	1	0.5	4.5
6	1	1	1	0.5	1	0.5	4
7	1	1	1	1	1	1	5
8	1	1	1	1	1	1	5
9	0	1	0	0	0	0	1
10	1	1	1	1	1	1	5
11	1	1	1	1	1	1	5
12	0	1	0	1	1	1	4
13	0	1	0	1	1	1	4
14	1	1	1	1	0.5	0.5	4
15	0	1	0	0	0.5	1	2.5
16	1	1	1	1	1	1	5
17	1	1	1	1	1	0.5	4.5
18	1	1	1	0.5	0.5	1	4
19	0	1	1	0	1	0.5	3.5
20	1	1	1	1	1	1	5
Avg	**0.70**	**1.00**	**0.75**	**0.78**	**0.83**	**0.78**	**4.13**
Progress Score			**82.5%** (= 4.13/5 × 100%)				
Success Rate			**70.0%** (= 14/20 × 100%)				
$\pi_{0.5}$							
1	0	1	0	1	1	1	4
2	0	0.5	0	1	1	0.5	3
3	1	1	1	1	1	1	5
4	0	1	0	1	0.5	0.5	3
5	1	1	1	0.5	1	0.5	4
6	1	1	1	1	0.5	1	4.5
7	1	1	1	1	1	1	5
8	1	1	1	0.5	0.5	1	4
9	0	1	0	0.5	0.5	0	2
10	1	1	1	1	1	1	5
11	1	1	1	0.5	1	1	4.5
12	0	1	0	1	1	0.5	3.5
13	0	1	1	1	0	0.5	3.5
14	1	1	1	0.5	1	1	4.5
15	0	1	1	1	1	0	4
16	1	1	1	1	1	1	5
17	0	0	0	1	0.5	0.5	2
18	0	0	0	0	0	0	0
19	0	1	0	1	0.5	0.5	3
20	1	1	1	1	1	0.5	4.5
Avg	**0.50**	**0.88**	**0.60**	**0.83**	**0.75**	**0.65**	**3.70**
Progress Score			**74.0%** (= 3.70/5 × 100%)				
Success Rate			**50.0%** (= 10/20 × 100%)				
GenieEnvisioner							
1	0	1	1	1	0	0	3
2	0	1	1	0	0	0	2
3	0	1	0	0	0	0	1
4	0	1	1	0	0	0	2
5	0	1	1	0	0	0	2
6	0	1	0	0	0	0	1
7	0	1	1	0	0	0	2
8	0	1	0	0	0	0	1
9	0	1	0.5	0	0	0	1.5
10	0	1	1	1	1	0	4
11	0	1	1	1	1	0	4
12	0	1	1	1	1	0	4
13	0	1	1	0	0	0	2
14	0	1	1	0	0	1	3
15	0	1	1	0	0	0	2
16	0	1	0	0	0	0	1
17	0	1	1	1	0	0	3
18	0	0	0	0	0	0	0
19	0	1	1	0	0	0	2
20	0	1	1	0	1	0	3
Avg	**0.0**	**0.95**	**0.725**	**0.25**	**0.2**	**0.05**	**2.175**
Progress Score			**43.5%** (= 2.175/5 × 100%)				
Success Rate			**0.0%** (= 0/20 × 100%)				

TABLE S5
DETAILED EVALUATION RESULTS FOR FOLD CLOTHES TASK (6 STEPS, MAX SCORE 6).

Trial	Success	Fold Half	Left Sleeve	Right Sleeve	Fold Again	Flatten	Place	Progress
Ours								
1	1	1	1	1	1	1	1	6
2	0	1	1	0	0	0	0	2
3	1	1	1	1	1	0.5	0.5	5
4	1	1	1	1	1	1	1	6
5	0	0.5	0	0	0	0	0	0.5
6	0	0	0	0	0	0	0	0
7	0	0.5	0	0	0	0	0	0.5
8	1	1	1	1	1	0.5	0.5	5
9	1	1	1	1	1	1	1	6
10	1	1	1	1	1	1	1	6
11	0	0.5	0	0	0	0	0	0.5
12	0	0.5	0	0	0	0	0	0.5
13	0	0.5	0	0	0	0	0	0.5
14	0	1	1	1	1	1	0	5
15	0	0.5	0	0	0	0	0	0.5
16	0	1	1	1	1	0.5	0	4.5
17	0	0	0	0	0	0	0	0
18	0	1	1	1	0.5	0	0	3.5
19	0	0.5	0	0	0	0	0	0.5
20	1	1	1	1	1	1	1	6
Avg	**0.35**	**0.73**	**0.55**	**0.50**	**0.48**	**0.38**	**0.30**	**2.93**
Progress Score				**48.8%** (= 2.93/6 × 100%)				
Success Rate				**35.0%** (= 7/20 × 100%)				
$\pi_{0.5}$								
1	1	1	1	1	1	1	1	6
2	1	1	1	1	1	1	1	6
3	1	1	1	1	0.5	1	0.5	5
4	0	1	1	1	0.5	1	0	4.5
5	0	0.5	1	1	0.5	1	0	4
6	0	1	1	0.5	1	1	0	4.5
7	0	0.5	0	0	0	0	0	0.5
8	0	0.5	0	0	0	0	0	0.5
9	0	1	1	1	1	1	0	5
10	0	0.5	1	1	0.5	0	0	3
11	0	0.5	0	0	0	0	0	0.5
12	0	1	1	1	1	1	0	5
13	0	1	1	0.5	0	0	0	2.5
14	1	1	1	1	1	1	1	6
15	1	1	1	1	0.5	1	1	5.5
16	0	1	1	1	0.5	0	0	3.5
17	1	1	1	1	1	1	1	6
18	0	1	1	1	0.5	1	0	4.5
19	0	0	0	0	0	0	0	0
20	0	1	1	1	0	0	0	3
Avg	**0.30**	**0.83**	**0.80**	**0.75**	**0.53**	**0.60**	**0.28**	**3.78**
Progress Score				**62.9%** (= 3.78/6 × 100%)				
Success Rate				**30.0%** (= 6/20 × 100%)				
GenieEnvisioner								
1	0	0.5	0	0	0	0	0	0.5
2	0	0	0	0	0	0	0	0
3	0	1	1	1	0	0	0	3
4	0	0.5	0	0	0	0	0	0.5
5	0	0	0	0	0	0	0	0
6	0	0.5	0	0	0	0	0	0.5
7	0	0.5	0	0	0	0	0	0.5
8	0	1	1	0	0	0	0	2
9	0	0.5	0	0	0	0	0	0.5
10	0	1	1	0	0	0	0	2
11	0	0.5	0	0	0	0	0	0.5
12	0	0.5	0	0	0	0	0	0.5
13	0	1	1	1	0	0	0	3
14	0	1	1	0	0	0	0	2
15	0	0.5	0	0	0	0	0	0.5
16	0	1	1	0	0	0	0	2
17	0	1	1	0	0	0	0	2
18	0	1	0	0	0	0	0	1
19	0	1	1	0	0	0	0	2
20	0	0	0	0	0	0	0	0
Avg	**0.00**	**0.65**	**0.40**	**0.10**	**0.00**	**0.00**	**0.00**	**1.15**
Progress Score				**19.2%** (= 1.15/6 × 100%)				
Success Rate				**0.0%** (= 0/20 × 100%)				

TABLE S6
DETAILED EVALUATION RESULTS FOR UNPACK DELIVERY TASK (5 STEPS, MAX SCORE 5).

Trial	Success	Grab Knife	Push Blade	Handover	Cut Seal	Open Lid	Progress
Ours							
1	1	1	1	1	1	1	5
2	1	1	1	1	1	1	5
3	1	1	1	1	1	1	5
4	0	1	0	0	0	0	1
5	0	1	1	1	0.5	0	3.5
6	0	1	1	1	0	0	3
7	1	1	1	1	1	1	5
8	0	1	1	1	0	0	3
9	1	1	1	1	1	1	5
10	1	1	1	1	1	1	5
11	1	1	1	1	1	1	5
12	1	1	1	1	1	1	5
13	0	1	1	1	0	0	3
14	1	1	1	1	1	1	5
15	1	1	1	1	1	1	5
16	0	1	1	1	0	0	3
17	1	1	1	1	0.5	1	4.5
18	1	1	1	1	1	1	5
19	0	1	1	1	0.5	0	3.5
20	1	1	1	1	1	1	5
Avg	**0.65**	**1.00**	**0.95**	**0.95**	**0.68**	**0.65**	**4.23**
Progress Score			**84.5%** (= 4.23/5 × 100%)				
Success Rate			**65.0%** (= 13/20 × 100%)				
$\pi_{0.5}$							
1	0	1	1	0.5	0.5	0	3
2	0	1	1	1	0	0	3
3	0	1	1	1	0.5	0	3.5
4	0	1	1	1	0.5	0	3.5
5	0	1	1	1	0.5	0	3.5
6	0	1	1	1	0	0	3
7	0	1	1	1	0	0	3
8	1	1	1	1	0.5	1	4.5
9	0	1	1	1	0.5	0	3.5
10	1	1	1	1	1	1	5
11	0	1	1	1	0.5	0	3.5
12	0	1	1	1	0.5	0	3.5
13	1	1	1	1	1	1	5
14	0	1	1	1	0.5	0	3.5
15	1	1	1	1	0.5	1	4.5
16	0	1	1	1	0	0	3
17	0	1	1	1	0	0	3
18	0	1	1	1	0	0	3
19	0	1	1	1	0.5	0	3.5
20	1	1	1	1	1	1	5
Avg	**0.25**	**1.00**	**1.00**	**0.98**	**0.43**	**0.25**	**3.65**
Progress Score			**73.0%** (= 3.65/5 × 100%)				
Success Rate			**25.0%** (= 5/20 × 100%)				
GenieEnvisioner							
1	0	1	0	0	0	0	1
2	0	1	0	0	0	0	1
3	0	1	1	0	0	0	2
4	0	1	0	0	0	0	1
5	0	1	1	1	0	0	3
6	0	1	1	0	0	0	2
7	0	0	0	0	0	0	0
8	0	1	0	0	0	0	1
9	0	1	0	0	0	0	1
10	0	0	0	0	0	0	0
11	0	1	0	0	0	0	1
12	0	1	0	0	0	0	1
13	0	1	0	0	0	0	1
14	0	1	0	0	0	0	1
15	0	1	0	0	0	0	1
16	0	0	0	0	0	0	0
17	0	1	0	0	0	0	1
18	0	1	0	0	0	0	1
19	0	0	0	0	0	0	0
20	0	1	0	0	0	0	1
Avg	**0.00**	**0.80**	**0.15**	**0.05**	**0.00**	**0.00**	**1.0**
Progress Score			**20.0%** (= 1.0/5 × 100%)				
Success Rate			**0.0%** (= 0/20 × 100%)				

TABLE S7
DETAILED EVALUATION RESULTS FOR INSERT TUBES TASK (2 CATEGORIES: GRASP AND INSERT, MAX SCORE 6).

Trial	Success	Grasp (3)	Insert (3)	Progress
Ours				
1	0	3	2	5
2	1	3	3	6
3	0	3	2	5
4	0	2	2	4
5	1	3	3	6
6	1	3	3	6
7	0	3	2	5
8	1	3	3	6
9	0	3	2	5
10	0	3	2	5
11	1	3	3	6
12	0	3	2	5
13	1	3	3	6
14	0	3	2	5
15	0	3	1	4
16	1	3	3	6
17	0	2	2	4
18	0	2	2	4
19	1	3	3	6
20	0	2	2	4
Avg	**0.40**	**2.80**	**2.35**	**5.15**
Progress Score		**85.8%** (= 5.15/6 × 100%)		
Success Rate		**40.0%** (= 8/20 × 100%)		
$\pi_{0.5}$				
1	0	3	1	4
2	0	3	1	4
3	0	3	1	4
4	0	3	1	4
5	0	3	1	4
6	0	3	1	4
7	0	3	1	4
8	0	3	2	5
9	1	3	3	6
10	1	3	3	6
11	1	3	3	6
12	0	2	1	3
13	0	3	2	5
14	0	2	2	4
15	1	3	3	6
16	1	3	3	6
17	0	3	2	5
18	0	2	2	4
19	1	3	3	6
20	0	3	2	5
Avg	**0.30**	**2.85**	**1.90**	**4.75**
Progress Score		**79.2%** (= 4.75/6 × 100%)		
Success Rate		**30.0%** (= 6/20 × 100%)		
GenieEnvisioner				
1	0	1	0	1
2	0	2	1	3
3	0	3	2	5
4	0	2	0	2
5	0	0	0	0
6	0	1	0	1
7	0	1	0	1
8	0	1	0	1
9	0	1	1	2
10	0	1	0	1
11	0	0	0	0
12	1	3	3	6
13	0	1	0	1
14	0	2	0	2
15	0	2	0	2
16	0	2	1	3
17	0	1	0	1
18	0	2	0	2
19	0	1	0	1
20	0	2	0	2
Avg	**0.05**	**1.45**	**0.40**	**1.85**
Progress Score		**30.8%** (= 1.85/6 × 100%)		
Success Rate		**5.0%** (= 1/20 × 100%)		

TABLE S8
DETAILED EVALUATION RESULTS FOR FOLD PANTS TASK (3 STEPS, MAX SCORE 3).

Trial	Success	Fold 1	Fold 2	Place	Progress
Ours					
1	1	1	1	1	3
2	1	1	1	1	3
3	1	1	1	1	3
4	0	0	0	0	0
5	0	1	0	0	1
6	0	1	0	0	1
7	1	1	1	1	3
8	1	1	1	1	3
9	1	1	1	1	3
10	1	1	1	1	3
11	1	1	1	1	3
12	1	1	1	1	3
13	1	1	1	1	3
14	1	1	1	1	3
15	1	1	1	1	3
16	1	1	1	1	3
17	1	1	1	1	3
18	0	1	0	0	1
19	0	1	0	0	1
20	0	0	0	0	0
Avg	**0.70**	**0.90**	**0.70**	**0.70**	**2.30**
Progress Score		**76.7%** (= 2.30/3 × 100%)			
Success Rate		**70.0%** (= 14/20 × 100%)			
$\pi_{0.5}$					
1	1	1	1	1	3
2	1	1	1	1	3
3	1	1	1	1	3
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	1	1	1	1	3
12	1	1	1	1	3
13	1	1	1	1	3
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
Avg	**0.30**	**0.30**	**0.30**	**0.30**	**0.90**
Progress Score		**30.0%** (= 0.90/3 × 100%)			
Success Rate		**30.0%** (= 6/20 × 100%)			
GenieEnvisioner					
1	0	0	0	0	0
2	1	1	1	1	3
3	0	1	0	0	1
4	0	1	0	0	1
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	1	0	0	1
15	0	0	0	0	0
16	0	0	0	0	0
17	0	1	0	0	1
18	0	0	0	0	0
19	0	1	0	0	1
20	0	1	0	0	1
Avg	**0.05**	**0.35**	**0.05**	**0.05**	**0.45**
Progress Score		**15.0%** (= 0.45/3 × 100%)			
Success Rate		**5.0%** (= 1/20 × 100%)			