STARRY: SPATIAL-TEMPORAL ACTION-CENTRIC WORLD MODELING FOR ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial-temporal prediction and geometric grounding are increasingly critical for high-precision robotic manipulation, yet remain challenging for world action models (WAMs). While these models capture temporal context through future prediction, RGB-only modeling does not explicitly represent the 3D geometry required for precise control. Consequently, visually plausible futures may still fail to expose action-relevant spatial constraints, limiting their ability to improve action generation. We propose **STARRY**, a world action model that aligns future spatial-temporal prediction with action generation through a unified diffusion process, enabling actions to be informed by the predicted evolution of scene appearance, spatial geometry, and end-effector motion. Meanwhile, to make more effective use of the predicted spatial-temporal information for action generation, we introduce **Geometry-Aware Selective Attention Modulation** (GASAM), which converts geometric cues derived from the predicted spatial-temporal variables into token-aligned weights, selectively directing action attention toward regions relevant to future interactions. On RoboTwin 2.0, STARRY achieves **93.82% / 93.30%** average success under Clean and Randomized settings across 50 bimanual tasks. Real-world experiments show that STARRY improves average success from **35.0%** to **67.1%** across four evaluation batches compared with $\pi_{0.5}$. These results demonstrate that spatial-temporal action-centric world modeling improves robotic manipulation by aligning future spatial-temporal prediction with action generation and selectively guiding action attention with predicted geometry.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a prominent paradigm for general-purpose embodied agents by unifying perception, language, and action (Black et al., 2025b;a; Din et al., 2025; Zitkovich et al., 2023). Although large-scale vision-language pretraining provides strong semantic understanding, for tasks such as hanging, handover, and placement, semantic understanding alone is insufficient: Successful actions also require anticipating how object states, end-effector motion, and their interactions evolve over time, as errors in these dynamics can lead to unstable grasps, collisions, or failed placements.

Recent work incorporates world models into policy learning by predicting future observations or latent video states (Lv et al., 2025; Bi et al., 2026; Cai et al., 2026; Zhu et al., 2026; Cen et al., 2025; Team et al., 2025). Nevertheless, RGB-only future prediction describes how a scene may look, but does not explicitly represent the depth needed to reason about 3D interaction geometry. This motivates *spatial-temporal world modeling* that jointly captures scene appearance, spatial geometry, and end-effector motion to represent future robot-object interactions.

However, predicting such spatial-temporal structure does not by itself ensure that the information is effectively used for action generation (Li et al., 2025; Team et al., 2026). In particular, globally fusing spatial information does not explicitly tell action attention which regions are geometrically relevant (Zhen et al., 2025; Zhang et al., 2025). This motivates a mechanism that translates predicted spatial-temporal information into selective guidance for action attention, emphasizing regions relevant to future robot-object interactions.

To address these requirements, we propose **STARRY** (**S**patial-**T**emporal **A**ction-centric **R**epresentation and **R**easoning Polic**Y**), a world action model that aligns future spatial-temporal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

prediction with action generation through a unified diffusion process. By jointly modeling scene appearance, spatial geometry, and end-effector motion alongside actions, STARRY captures future spatial-temporal structure relevant to robot-object interactions. To translate this predictive information into action guidance, we introduce **Geometry-Aware Selective Attention Modulation** (GASAM), which converts geometric cues exposed by the *Geometry Expert* from spatial-temporal and action representations into token-aligned weights. These weights selectively direct action attention toward regions relevant to future interactions while preserving video modeling and semantic understanding.

Our contributions are summarized as follows:

- We propose **STARRY**, a world action model that aligns future spatial-temporal prediction with action generation through a unified diffusion process, jointly capturing scene appearance, spatial geometry, and end-effector motion for robotic manipulation.
- We introduce **Geometry-Aware Selective Attention Modulation** (GASAM), which converts geometric cues derived from predicted spatial-temporal representations into token-aligned weights, selectively guiding attention toward regions relevant to future interactions.
- We validate STARRY through simulation and real-world experiments. It achieves **93.82% / 93.30%** average success on RoboTwin 2.0 under Clean and Randomized settings and improves real-world average success from **35.0%** to **67.1%** over $\pi_{0.5}$. Ablations confirm the benefits of spatial-temporal prediction and geometry-guided action attention.

2 Related work

2.1 Vision-Language-Action policies

Vision-Language-Action (VLA) models unify perception, language, and action, forming a dominant paradigm for embodied manipulation (Din et al., 2025). Representative works such as RT-2 and $\pi_{0.5}$ leverage large-scale vision-language pretraining to learn rich semantic representations, enabling direct mapping from multi-modal inputs to action outputs (Brohan et al., 2023; Zitkovich et al., 2023; ichter et al., 2023; Jiang et al., 2023; Black et al., 2025b;a). However, most VLA policies remain *reactive*, mapping observations directly to actions without explicitly modeling future states, limiting their ability to leverage temporal foresight for action generation. In addition, existing methods are largely 2D-centric and lack explicit geometric reasoning, relying primarily on appearance-based features, which restricts performance in spatially constrained tasks (Qian et al., 2026).

2.2 World models and predictive learning

World models learn environment dynamics to support prediction and decision-making (Li et al., 2025). Recent advances span latent predictive learning (e.g., JEPA), which focuses on learning temporally consistent representations, and large-scale generative models such as DreamDojo and WoW that model future observations from video or interaction data (Bardes et al., 2024; Gao et al., 2026; Chi et al., 2025b; Team et al., 2025). Unified architectures further couple video and action modeling for scalable policy learning. Despite strong predictive capabilities, these models often exhibit a gap between prediction quality and control utility (Shang et al., 2026), as they primarily optimize perceptual fidelity rather than action relevance, and emphasize temporal dynamics with limited modeling of spatial geometry.

2.3 World-model-enhanced policies

Recent work integrates world models into action generation by leveraging predicted futures for policy learning. Representative approaches include foresight-based action prediction (e.g., F1), which formulates action generation as a prediction-guided inverse dynamics problem, and unified video-action modeling frameworks such as Motus that jointly learn latent representations for perception and control (Lv et al., 2025; Bi et al., 2026; Zhu et al., 2025). While these methods improve temporal consistency and multi-modal integration, they provide limited modeling of spatial geometry and lack mechanisms for selectively emphasizing action-relevant regions.

Our work addresses the above gap by learning action-centric future spatial-temporal latent variables and introducing geometry-aware selective modulation for more effective action generation.

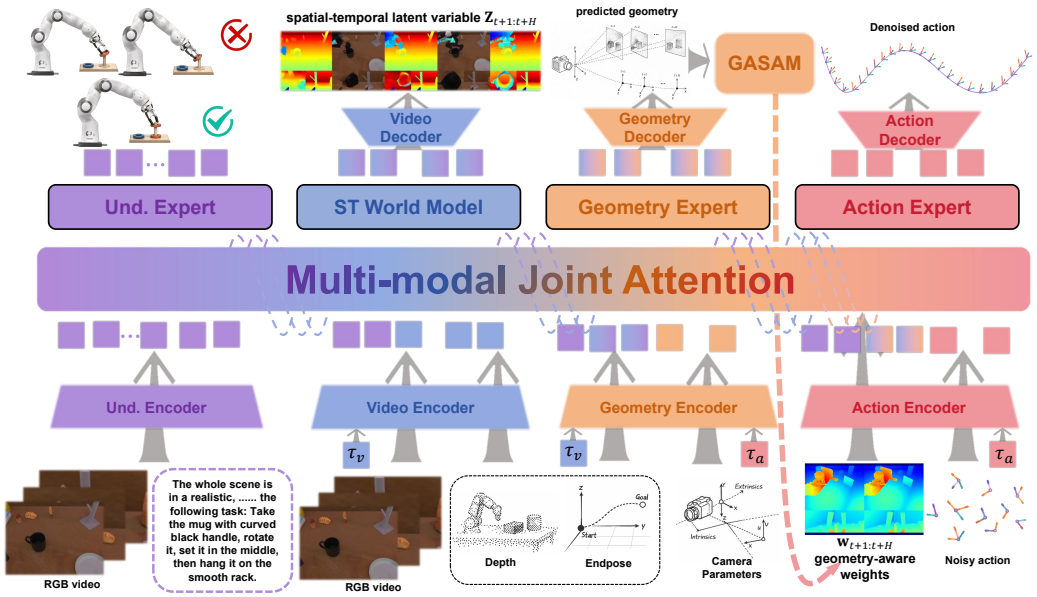

Figure 1: Overview of **STARRY**. The policy contains four modules: *Understanding Expert*, *Spatial-Temporal (ST) World Model*, *Geometry Expert*, and *Action Expert*. The *ST World Model* predicts future spatial-temporal latent variables, while the *Geometry Expert* and GASAM generate geometry-aware weights to selectively modulate the action branch.

3 STARRY: Spatial-Temporal Action-Centric Representation and Reasoning

We propose **STARRY**, a world action model that aligns future spatial-temporal prediction with action generation through a unified diffusion process. We first present the overall architecture (Sec. 3.1), followed by the *ST World Model* (Sec. 3.2), the Geometry-Aware Selective Attention Modulation mechanism (Sec. 3.3), and the model training and data construction (Sec. 3.4).

3.1 Architecture Overview

Problem formulation. We consider language-conditioned embodied manipulation, where the agent observes $\mathbf{o}_t = \{\mathbf{I}_t, \mathbf{D}_t, \mathbf{c}_t, \mathbf{p}_t, \mathbf{l}\}$ at time step t, including multi-view RGB-D observations $\{\mathbf{I}_t, \mathbf{D}_t\}$, camera parameters $\mathbf{c}_t$, current pose $\mathbf{p}_t$, and the language instruction $\mathbf{l}$. The goal is to generate a future action sequence $\mathbf{a}_{t+1:t+H}$ (Zhao et al., 2023). Instead of directly mapping $\mathbf{o}_t$ to actions, STARRY introduces two internal structures: future spatial-temporal latent variables $\mathbf{z}_{t+1:t+H}$ and geometry-aware modulation weights $\mathbf{w}_{t+1:t+H}$. The future spatial-temporal latent variables and actions are jointly modeled as

$$\pi_\theta(\mathbf{a}_{t+1:t+H}, \mathbf{z}_{t+1:t+H} \mid \mathbf{o}_t). \tag{1}$$

Here, $\mathbf{z}_{t+1:t+H}$ captures predictive spatial-temporal structure aligned with action generation, while $\mathbf{w}_{t+1:t+H}$ is generated from predicted future geometry and used to inject action-relevant spatial constraints into the action branch.

Architecture. As shown in Fig. 1, STARRY consists of four modules: the *Understanding Expert*, *Spatial-Temporal (ST) World Model*, *Geometry Expert*, and *Action Expert*. The core specific designs are action-centric future spatial-temporal latent variables combining RGB appearance, depth, and end-effector trajectories and the Geometry Expert–GASAM path that converts predicted geometry into action-attention guidance. Joint denoising and the four-expert architecture provide the supporting framework.

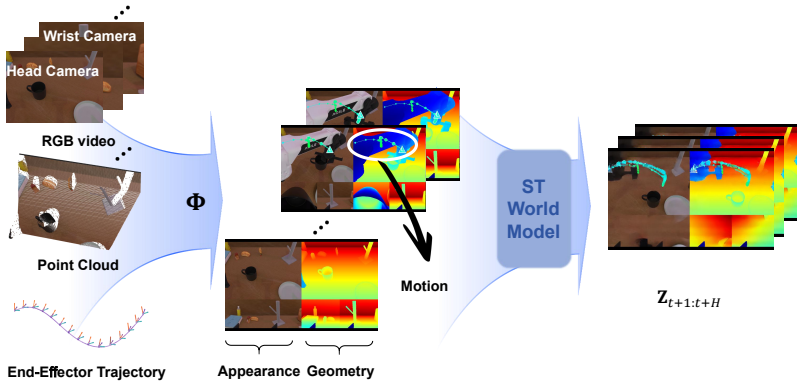

Figure 2: Construction of spatial-temporal inputs for the *ST World Model*. Multi-modal signals, including RGB observations, depth observations, and end-effector trajectories, are unified into a shared representation capturing appearance, geometry, and motion, which is then used for future spatial-temporal prediction.

Concretely, the *Understanding Expert* supplies visual-language conditioning. The *ST World Model* and *Action Expert* jointly denoise future spatial-temporal latent variables and actions through Multi-modal Joint Attention. In parallel, the *Geometry Expert* reads their current intermediate states and GASAM converts its predictions into token-aligned weights applied to action-to-video attention. This enables action-relevant spatial reasoning while preserving video modeling and visual-language understanding.

3.2 ST World Model

Existing world models often emphasize future visual appearance, with limited explicit modeling of the spatial-temporal structure of robot-object interactions. RGB-only future prediction captures appearance but does not explicitly represent depth or end-effector trajectories. We incorporate these complementary cues to model both scene geometry and robot motion.

The *ST World Model* learns future spatial-temporal latent variables $\mathbf{z}_{t+1:t+H}$ that represent multi-view appearance, depth-related spatial structure, and end-effector evolution (Fig. 2). These variables are defined in the Wan-VAE encoding space of spatial-temporal frames composed of RGB views, camera-aligned depth, and projected end-effector trajectories.

Given observations within a temporal window $[t_0, t]$, we construct a unified spatial-temporal representation from multi-view RGB images $\{\mathbf{I}_t^c\}_c$, depth observations $\{\mathbf{D}_t^c\}_c$, and 3D end-effector trajectories $\{\mathbf{e}_\tau^m\}_{\tau \le t,m}$. We first project trajectories into each camera view using camera intrinsics K and extrinsics T: $\mathcal{C}_t^c = \{K_t^c, T_t^c\}$, and then compose appearance, geometry, and motion into a unified representation:

$$\mathbf{u}_\tau^{c,m} = \Pi(K_t^c, T_t^c, \mathbf{e}_\tau^m)\,, \qquad \mathbf{x}_t = \Phi(\{\mathbf{I}_t^c\}_c, \{\mathbf{D}_t^c\}_c, \{\mathbf{u}_\tau^{c,m}\}_{\tau \le t,c,m})\,. \tag{2}$$

Here, $\Pi(\cdot)$ denotes camera projection, and $\Phi(\cdot)$ denotes the spatial-temporal composition function, which arranges the multi-view RGB frames, depth observations, and projected trajectories into a fixed RGB-D layout. The resulting sequence $\mathbf{X}_{t_0:t} = \{\mathbf{x}_{t_0}, \ldots, \mathbf{x}_t\}$ provides a structured description of the observed spatial-temporal history.

VAE encoding followed by tokenization converts $\mathbf{X}_{t_0:t}$ into historical context tokens $\mathbf{v}_{t_0:t}$. These tokens and historical actions condition the prediction of future spatial-temporal latent variables; their clean training values z^0 are encoded from future demonstration frames:

$$\mathbf{z}_{t+1:t+H} = f_\theta^{\mathrm{ST}}(\mathbf{v}_{t_0:t}, \mathbf{a}_{\le t})\,, \qquad z^0 = E_{\mathrm{Wan}}(x_{t+1:t+H}). \tag{3}$$

Here, f_θ^{ST} summarizes the complete denoising process, $\mathbf{z}_{t+1:t+H}$ captures structured spatial-temporal information, including scene evolution, end-effector motion, and geometric constraints, providing effective conditioning for action generation, and E_{Wan} is the Wan VAE encoder. Training corrupts z^0 into z^{τ_v} for joint denoising with actions under historical and visual-language conditioning and inference starts both branches from independent Gaussian noise.

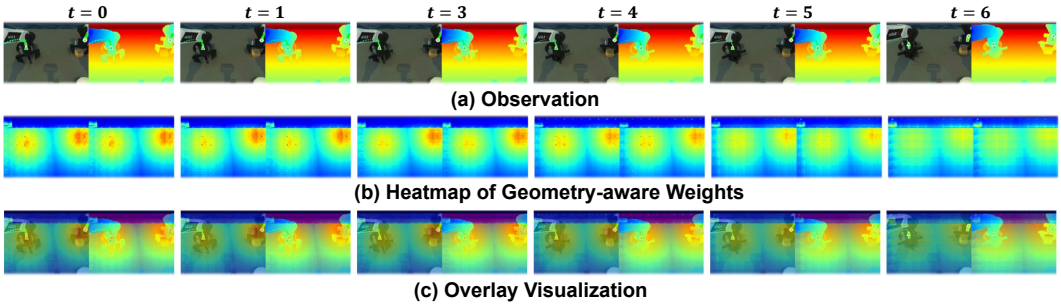

Figure 3: Geometry-aware weight construction and modulation in GASAM. Rows (a)–(c) show observations, geometry-aware weights, and their overlays across selected temporal steps. The constructed weights highlight geometrically action-relevant regions for action-attention modulation.

3.3 Geometry-Aware Selective Attention Modulation

Although the *ST World Model* provides future spatial-temporal latent variables, action generation still needs an explicit mechanism to align 2D visual tokens and 3D physical control. Standard joint attention matches action tokens with 2D visual tokens through query-key dot products in token space, forcing the policy to infer perspective and depth relations implicitly. This may cause large 3D control deviations when visual similarity does not reflect metric distance. To bridge this gap, we propose *Geometry-Aware Selective Attention Modulation* (GASAM), which uses predicted depth and end-effector geometry to lift 2D visual tokens into metric 3D space, thereby explicitly aligning visual observations with physical control.

Current-step representations. At noise levels (τ_v, τ_a), the current future states are $\mathbf{z}^{\tau_v}$ and $\mathbf{a}^{\tau_a}$. The ST and action branches form intermediate representations:

$$h_{\text{st}}^{(\tau_v,\tau_a)} = f_{\text{st}}(z^{\tau_v}, a^{\tau_a}, o_{\le t}, \tau_v, \tau_a), \qquad h_{\text{act}}^{(\tau_v,\tau_a)} = f_{\text{act}}(z^{\tau_v}, a^{\tau_a}, o_{\le t}, \tau_v, \tau_a). \tag{4}$$

Here, $o_{\le t}$ includes encoded history $\mathbf{v}_{t_0:t}$, historical actions, and the visual-language observation $\mathbf{o}_t$. The two f symbols denote ST/Action branch processing up to the intermediate-feature stage: input encoders produce the h features before geometry prediction, GASAM, and the final velocity outputs.

Geometry prediction. Future spatial-temporal latent variables in their noisy states cannot be treated as depth maps or directly unprojected. To obtain stable and interpretable spatial guidance, we introduce a *Geometry Expert*, which reads the current intermediate ST/action representations to expose their implicit geometric content as future metric depth and 3D end-effector positions:

$$(\hat{D}_{t+1:t+H}^{(\tau_v,\tau_a)}, \hat{p}_{t+1:t+H}^{(\tau_v,\tau_a)}) = g_\phi(h_{\text{st}}^{(\tau_v,\tau_a)}, h_{\text{act}}^{(\tau_v,\tau_a)}, \tau_v, \tau_a). \tag{5}$$

Here, g_ϕ is the Geometry Expert, conditioned through same-step features. Its stable, explicit estimates of future geometry complement the implicit modeling in future spatial-temporal latent variables $\mathbf{z}_{t+1:t+H}$ and guide geometry-aware action modulation.

Geometry-aware weight construction and modulation. To make action attention aware of metric spatial relations, GASAM unprojects predicted depth using camera parameters and computes 3D distances between scene points and predicted end-effector positions in a common coordinate frame. These distances are then converted into token-aligned weights:

$$d_{t,j} = \left\| \hat{\mathbf{p}}_t - \hat{\mathbf{P}}_{t,j} \right\|_2, \qquad \mathbf{w}_{t+1:t+H} = \mathcal{T}(\rho(\{d_{t,j}\}_{t,j})). \tag{6}$$

The head view uses the smaller distance to the two arms, while each wrist view uses its corresponding arm. For future step s, the distance-to-weight mapping is

$$\rho(d_{s,j}) = w_{\min} + (w_{\max} - w_{\min}) \exp\left[-\frac{d_{s,j}}{\text{median}_j(d_{s,j}) + 10^{-6}} \right]. \tag{7}$$

The token mapping $\mathcal{T}$ samples each patch-center pixel $j(i)$ and aggregates over the full future horizon:

$$w_i = \sum_{s=t+1}^{t+H} \frac{\exp[\beta \rho(d_{s,j(i)})]}{\sum_{r=t+1}^{t+H} \exp[\beta \rho(d_{r,j(i)})]} \rho(d_{s,j(i)}), \qquad \beta = 20. \tag{8}$$

During denoising, $[w_{\min}, w_{\max}]$ expands linearly, keeping guidance weak at early noisy steps and strengthening it as geometry stabilizes. Neither mapping applies a hard threshold or mask. As shown in Fig. 3, the constructed geometry-aware weights highlight spatial regions that are geometrically relevant to action execution and provide the modulation signal for the action attention branch.

GASAM then uses $\mathbf{w}_{t+1:t+H}$ to modulate action-to-video attention. Let $\mathbf{Q}^a$ denote action queries, and let $\mathbf{K}^v$ and $\mathbf{V}^v$ denote video keys and values. The geometry-modulated action-to-video attention is:

$$\mathrm{Attn}^{a \leftarrow v}_{\mathrm{GASAM}} = \mathrm{Softmax}\left(\frac{\mathbf{Q}^a (\mathbf{K}^v)^\top}{\sqrt{d}} + \lambda \log(\mathbf{w}_{t+1:t+H} + \epsilon) \right) \mathbf{V}^v. \tag{9}$$

Here, the log-weight term acts as a bounded prior that reweights tokens without masking them, preserving access to global semantic context and distant regions. It modifies only the action-to-video logits before softmax normalization over all joint-attention keys. Other logit blocks remain unchanged, and the ST and Understanding Experts retain full context.

3.4 Training and Data

Initialization and diffusion objective. We initialize the *ST World Model* from Wan (Ho et al., 2022; Rombach et al., 2022; Peebles & Xie, 2023; Wan et al., 2025) and the *Understanding Expert* from Qwen-VL (Bai et al., 2023). The *Action Expert* uses the ST Transformer architecture with action-specific parameters, the *Geometry Expert* predicts future geometry from the same video and action tokens using a diffusion-aligned Transformer. We jointly model future spatial-temporal latent variables and actions through diffusion velocity prediction (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Chi et al., 2025a). Only in this loss, $\mathbf{x}_{t+1:t+H}$ denotes the clean latent values $\mathbf{z}^0_{t+1:t+H}$, obtained by encoding the spatial-temporal frames in Sec. 3.2 before noising:

$$\mathcal{L}_{\text{obs}} = \mathbb{E}\left[\| \mathbf{v}^o_\theta - (\boldsymbol{\epsilon}_o - \mathbf{x}_{t+1:t+H}) \|^2 \right], \qquad \mathcal{L}_{\text{action}} = \mathbb{E}\left[\| \mathbf{v}^a_\theta - (\boldsymbol{\epsilon}_a - \mathbf{a}_{t+1:t+H}) \|^2 \right]. \tag{10}$$

where $\boldsymbol{\epsilon}_o$ and $\boldsymbol{\epsilon}_a$ denote Gaussian noise, and $\mathbf{v}^o_\theta$ and $\mathbf{v}^a_\theta$ are the predicted velocity fields. Different diffusion time steps (τ_v, τ_a) are used for observations and actions to account for modality differences.

The overall loss is:

$$\mathcal{L}_{\text{diff}} = \lambda_o \mathcal{L}_{\text{obs}} + \lambda_a \mathcal{L}_{\text{action}}. \tag{11}$$

Geometry supervision. The *Geometry Expert* is supervised on depth, end-effector position, and geometry-aware weights:

$$\mathcal{L}_{\text{geo}} = \lambda_d \mathcal{L}_{\text{depth}} + \lambda_p \mathcal{L}_{\text{pose}} + \lambda_w \mathcal{L}_{\text{weight}}, \tag{12}$$

where $\mathcal{L}_{\text{depth}}$ supervises depth prediction, $\mathcal{L}_{\text{pose}}$ constrains the 3D end-effector position, and $\mathcal{L}_{\text{weight}}$ enforces consistency between predicted modulation weights and geometry-derived targets.

The target weights are constructed from ground-truth spatial distances:

$$\mathbf{w}^*_{t+1:t+H} = \mathcal{T}\left(\{ \rho(\|\mathbf{p}_s - \mathbf{P}_{s,j}\|_2) \}_{s,j} \right), \qquad \mathcal{L}_{\text{weight}} = \left\| \mathbf{w}_{t+1:t+H} - \mathbf{w}^*_{t+1:t+H} \right\|_2^2, \tag{13}$$

where $\mathbf{w}^*_{t+1:t+H}$ denotes token-aligned ground-truth weights, providing direct supervision for geometry-aware attention modulation.

The *Geometry Expert* is pretrained independently, then jointly fine-tuned, guiding actions through GASAM without altering the diffusion objective. Clean future actions serve only forward noising and action supervision, ground-truth future depth, poses, and derived weights are $\mathcal{L}_{\text{geo}}$ targets, never GASAM inputs. At inference, ST/action states start from independent Gaussian noise, with geometry predicted online and no ground-truth future actions, depth, or poses provided.

Table 1: Selected RoboTwin 2.0 results under Clean and Randomized settings.

Task	$\pi_{0.5}$		X-VLA		Motus		LingBot-VA		Ours	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Click Alarmclock	97.5%	91%	99%	99%	**100%**	**100%**	99%	**100%**	**100%**	**100%**
Handover Mic	63%	57.5%	0%	0%	78%	63%	94%	96%	**100%**	**99%**
Hanging Mug	10.5%	10%	23%	27%	38%	38%	40%	28%	**69%**	**72%**
Move Can Pot	40%	41%	89%	86%	34%	74%	94%	97%	**100%**	**98%**
Move Pillbottle Pad	58.5%	45%	73%	71%	93%	96%	99%	99%	**100%**	**100%**
Pick Diverse Bottles	43%	37%	58%	36%	90%	91%	89%	82%	**98%**	**96%**
Place Bread Basket	62.5%	60%	81%	71%	91%	94%	97%	95%	**100%**	**99%**
					..... (50 tasks)					
Place Can Basket	40.5%	43.5%	49%	52%	81%	76%	81%	84%	**89%**	**88%**
Place Shoe	74.5%	77%	96%	95%	99%	97%	98%	98%	**100%**	**99%**
Press Stapler	83.5%	76.5%	92%	98%	93%	98%	85%	82%	**100%**	**100%**
Put Bottles Dustbin	48%	44%	74%	77%	81%	79%	87%	91%	**96%**	**93%**
Stack Blocks Three	53%	46%	6%	10%	91%	95%	**99%**	98%	97%	**100%**
Turn Switch	33.5%	30%	40%	61%	84%	78%	44%	45%	**85%**	**89%**
Average	62.86%	60.30%	72.80%	72.84%	88.66%	87.02%	92.93%	91.55%	**93.82%**	**93.30%**

Training pipeline and dataset. We adopt a three-stage training strategy:

Stage 1 (spatial-temporal pretraining): train the *ST World Model* and *Understanding Expert* on egocentric video and multimodal embodied data to learn spatial-temporal dynamics and semantic representations. Web-scale priors are inherited from Wan/Qwen-VL and we perform no additional web-scale pretraining.

Stage 2 (action and geometry learning): introduce the *Geometry Expert* and *Action Expert* for joint training under the diffusion objective, enabling joint learning of future spatial-temporal latent variables and actions under geometric constraints.

Stage 3 (joint finetuning): perform end-to-end finetuning with GASAM, allowing all modules to be optimized jointly for improved spatial-temporal consistency and action generation.

We use egocentric video, geometry-enriched robot data, multi-robot trajectories, and target-robot demonstrations (Grauman et al., 2022; Hoque et al., 2026; Dong et al., 2025; Mandlekar et al., 2022; Khazatsky et al., 2024; Walke et al., 2023; O'Neill et al., 2024). Data provenance and comparison scope are described in Appendix A.3.

4 Experiments

We evaluate STARRY in simulation and real-world settings. Specifically, we test on RoboTwin 2.0 under Clean and Randomized settings (Sec. 4.1), examine physical execution in real-world experiments (Sec. 4.2), and conduct ablations to analyze the effects of spatial-temporal prediction and GASAM (Sec. 4.3).

4.1 Evaluation in simulation environment

RoboTwin 2.0 We first evaluate STARRY on the RoboTwin 2.0 benchmark, which comprises 50 bimanual manipulation tasks with structured domain randomization (Chen et al., 2025). In the main paper, we compare against $\pi_{0.5}$ (Black et al., 2025a), X-VLA (Zheng et al., 2026), Motus (Bi et al., 2026), and LingBot-VA (Li et al., 2026). For each task, we use 50 demonstrations under the Clean setting and 500 demonstrations under the Randomized setting. Demonstrations from all 50 tasks are pooled together and jointly optimized, rather than training each task independently. All models are adapted from their original pretrained checkpoints with a batch size of 256 for 40k downstream steps. Additional baselines, including GO-1 (AgiBot-World-Contributors et al., 2025) and π_0 (Black et al., 2025b), together with full per-task results on all 50 tasks, are provided in Appendix B.6.

STARRY achieves 93.82%/93.30% average success under Clean/Randomized settings, exceeding LingBot-VA (92.93%/91.55%) and Motus (88.66%/87.02%). Notably, these average gains underestimate the advantage on challenging tasks, since many easier tasks are close to saturation for multiple methods. On more challenging tasks, the improvements of STARRY align closely with our design.

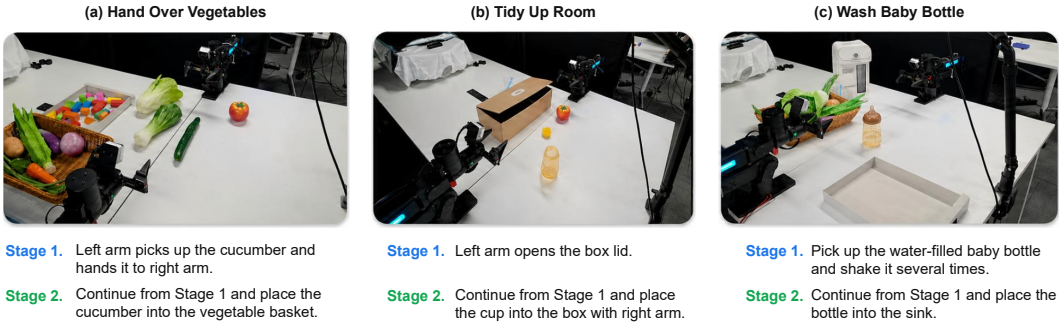

Figure 4: Real-world task settings. We evaluate three representative two-stage bimanual manipulation tasks: *Hand Over Vegetables*, *Tidy Up Room*, and *Wash Baby Bottle*.

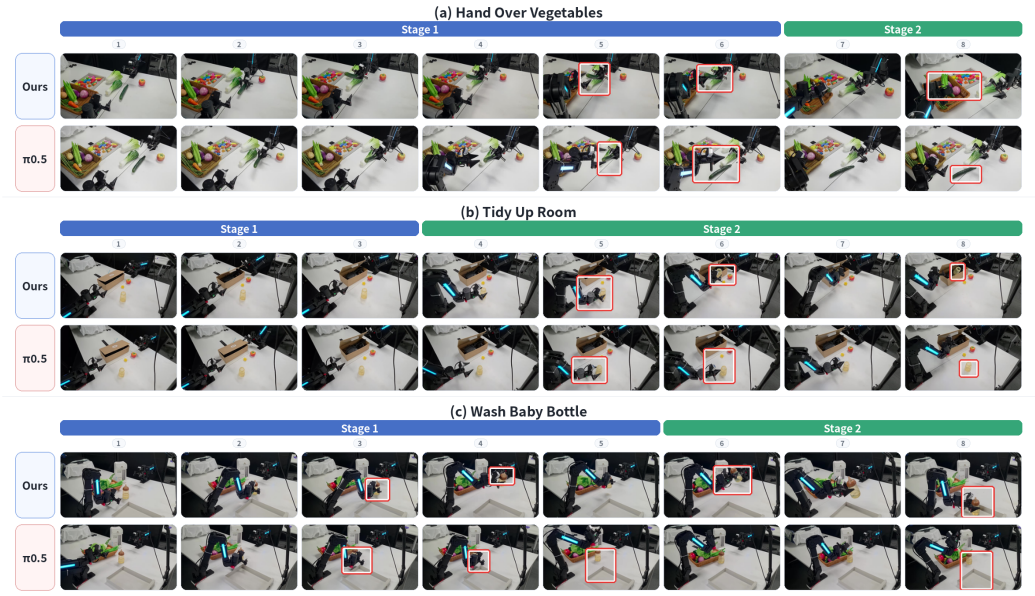

Figure 5: Qualitative comparison in real-world experiments. STARRY and $\pi_{0.5}$ are compared on three bimanual tasks. Red boxes highlight key regions related to success or failure.

For example, on Handover Mic, it reaches 100%/99% versus Motus's 78%/63%; on Hanging Mug, it improves the strongest baseline by 29/34 points. These tasks involve bimanual coordination and precise geometric alignment. Further analysis of non-saturated tasks is provided in Appendix B.3.

4.2 Real-world experiments

To further evaluate the effectiveness of STARRY in real robotic settings, we conduct real-world experiments on representative bimanual manipulation tasks and compare against $\pi_{0.5}$. We consider three tasks, namely *Hand Over Vegetables*, *Tidy Up Room*, and *Wash Baby Bottle*, as shown in Fig. 4. To better characterize policy capability, each task is evaluated in two stages: Stage 1 focuses on the key subskill, while Stage 2 requires completing the full multi-step task built upon Stage 1. All real-world experiments are conducted on the ARX R5 bimanual robot platform from ARX. For each task, we collect 50 real-robot demonstrations for training and evaluate the same checkpoint in four batches of 20 rollouts per method–task–stage setting (80 outcomes per setting), reporting mean success and sample SD across batches.

Table 2: Success rates (%) in real-world experiments: mean ± sample SD across four evaluation batches.

Method	Hand Over Vegetables		Tidy Up Room		Wash Baby Bottle	
	Stage 1	Stage 2	Stage 1	Stage 2	Stage 1	Stage 2
$\pi_{0.5}$	53.8 ± 9.5	30.0 ± 9.1	48.8 ± 11.1	26.3 ± 11.1	32.5 ± 11.9	18.8 ± 10.3
Ours	$\mathbf{80.0 \pm 7.1}$	$\mathbf{67.5 \pm 6.5}$	$\mathbf{72.5 \pm 6.5}$	$\mathbf{60.0 \pm 7.1}$	$\mathbf{67.5 \pm 6.5}$	$\mathbf{55.0 \pm 4.1}$

Table 3: Ablations on RoboTwin 2.0 (success, %). In (a,b), −/+ denotes without/with GASAM. Panels (b,c) use Randomized settings. Panel (c) changes only inference-time strength on the same Full-ST checkpoint, without retraining, $\lambda = 1$ is the default.

(a) Representation / GASAM

	Clean		Randomized	
Representation	−	+	−	+
Action-only	63.42	72.50	64.96	75.88
RGB	86.64	87.86	85.80	86.96
Full ST	90.40	**93.82**	88.82	**93.30**

(b) Depth / trajectory

	GASAM	
Representation	−	+
RGB	85.80	86.96
RGB + trajectory	86.52	88.06
RGB + depth	87.52	90.88
Full ST	88.82	**93.30**

(c) Strength

λ	Success
0.5	89.08
1	**93.30**
2	88.17

In Table 2, STARRY averages 67.1% across the six stages versus 35.0% for $\pi_{0.5}$, outperforming it on every task and stage. The larger 35.8-point gain in the more challenging Stage 2 suggests benefits beyond short-horizon stabilization, reflecting stronger spatial-temporal modeling for multi-step execution. Qualitative comparisons in Fig. 5 support this observation. In *Hand Over Vegetables* (a), $\pi_{0.5}$ often fails to establish the spatial-temporal configuration for bimanual handover, causing grasping or transfer errors, whereas STARRY completes the sequence more reliably. In *Tidy Up Room* (b), $\pi_{0.5}$ may open the box but misalign the object with its opening during placement, while STARRY better preserves the target-container spatial relationship. In *Wash Baby Bottle* (c), irregular bottle geometry makes grasping, shaking, and placement sensitive to local contact and temporal coordination, STARRY maintains a stable grasp, executes shaking, and completes placement. These results are consistent with our design: spatial-temporal world modeling captures future scene evolution and end-effector motion, while geometric cues from the predicted representations selectively guide action attention toward future interaction regions, supporting coordinated motion, object alignment, and multi-step execution in real-world bimanual manipulation.

4.3 Ablation study

Table 3(a) studies representation and GASAM under matched initialization, data, and training budgets. RGB prediction substantially improves over action-only denoising; Full ST improves further, and GASAM benefits every representation. Action-only with GASAM retains auxiliary geometry prediction without generating or supervising future spatial-temporal latent variables. Panel (b) separates the depth and trajectory cues: each improves RGB prediction, while their combination in Full ST performs best. Panel (c) varies only inference-time modulation on the same checkpoint: both weaker and stronger guidance underperform the default, indicating sensitivity to the geometric prior without retraining. Detailed ablation analyses, task-level results, and robustness evaluations are provided in Appendix B.1, B.4 and B.6.

5 Conclusion

We presented **STARRY**, a world action model that jointly predicts spatial-temporal futures and actions, using GASAM to inject predicted geometry into action attention. On RoboTwin 2.0, STARRY achieves **93.82% / 93.30%** average success under Clean and Randomized settings. In real-world experiments, it improves average success from **35.0%** to **67.1%** across four evaluation batches compared with $\pi_{0.5}$, demonstrating improved execution reliability in physical manipulation. These results demonstrate the effectiveness of action-centric spatial-temporal world modeling for spatially and temporally demanding robotic manipulation.

AI USE STATEMENT

Generative AI tools were used for manuscript polishing, assistance with code writing, and LaTeX formatting. The authors take responsibility for all AI-assisted text and code and for the final content of this work.

ETHICS STATEMENT

This work studies robotic manipulation in simulation and physical experiments. Potential benefits and risks, including unsafe robot behavior under distribution shift, are discussed in Appendix D. Physical deployment requires appropriate safety constraints, monitoring, and human oversight. The scope and limitations of the evaluation are discussed in Appendix C.

REPRODUCIBILITY STATEMENT

The architecture, training objectives, and data pipeline are described in Secs. 3.1–3.4. Appendix A provides representation and GASAM implementation details, training provenance, inference profiling, hyperparameters, compute resources, and pseudocode. Evaluation protocols are described in Secs. 4.1–4.3, with additional experiments and complete per-task results in Appendix B.

REFERENCES

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

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. In *Transactions on Machine Learning Research (TMLR)*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, Proceedings of Machine Learning Research, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

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.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025a.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world-omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025b.

Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Vision language action models in robotic manipulation: A systematic review. *arXiv preprint arXiv:2507.10672*, 2025.

Zibin Dong, Fei Ni, Yifu Yuan, Yinchuan Li, and Jianye Hao. Embodiedmae: A unified 3d multi-modal representation for robot manipulation. *arXiv preprint arXiv:2505.10105*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems (NeurIPS)*, 35: 8633–8646, 2022.

Ryan Hoque, Peide Huang, David Yoon, Mouli sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 4218–4237, 2026.

brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 14–18 Dec 2023.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14975–15022. PMLR, 23–29 Jul 2023.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems (RSS)*, Delft, Netherlands, July 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Xinqing Li, Xin He, Le Zhang, Min Wu, Xiaoli Li, and Yun Liu. A comprehensive survey on world models for embodied ai. *arXiv preprint arXiv:2510.16732*, 2025.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. Geopredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13529–13539, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *AI Open*, 7:208–226, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025.

HY Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, et al. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 1723–1736. PMLR, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Learning 4d embodied world models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5337–5347. IEEE, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

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 *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. In *International Conference on Learning Representations*, volume 2026, pp. 62486–62502, 2026.

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 *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 2165–2183. PMLR, 2023.

A METHOD, IMPLEMENTATION, AND DEPLOYMENT

A.1 FUTURE SPATIAL-TEMPORAL LATENT VARIABLES AND DENOISING INTERFACES

The composition operator Φ acts before VAE encoding and has no learned parameters. Camera-aligned depth is linearly normalized and rendered with the JET colormap; projected end-effector trajectories are overlaid on the RGB and depth-derived views. Writing $h/l/r$ for head/left-wrist/right-wrist cameras and overbars for trajectory overlays, the fixed layout is

$$\mathbf{x}_s = \mathrm{Resize}_{384\times640}\begin{pmatrix} \overline{\mathbf{I}}_s^h & \overline{\mathbf{D}}_s^h \\ \mathrm{Resize}_{1/2}(\overline{\mathbf{I}}_s^l) \mid \mathrm{Resize}_{1/2}(\overline{\mathbf{I}}_s^r) & \mathrm{Resize}_{1/2}(\overline{\mathbf{D}}_s^l) \mid \mathrm{Resize}_{1/2}(\overline{\mathbf{D}}_s^r) \end{pmatrix}. \tag{14}$$

The vertical bar denotes horizontal concatenation. Historical and future spatial-temporal frames use the same layout. This construction organizes appearance, depth, and motion in image space; it does not concatenate raw depth channels to RGB. For physical calibration, an end-effector-mounted camera is moved to measured global points. Joint encoders and robot kinematics recover the global-to-base transform and 6-DoF end-effector trajectory, enabling distance-map construction without visual tracking or extra trajectory annotation.

Historical spatial-temporal frames $\mathbf{X}_{t_0:t}$ pass through the Wan VAE, patch embedding, and flattening to form historical context tokens $\mathbf{v}_{t_0:t}$. Together with historical actions and visual-language information, these tokens condition prediction. Future demonstration frames are encoded by the same Wan VAE to obtain the clean values z^0 of the future spatial-temporal latent variables (Eq. 3). Historical tokens supply observed context, whereas the clean future values provide denoising supervision; neither quantity is a current-step intermediate representation.

During training, forward noising corrupts z^0 into the current noisy states z^{τ_v}, while clean actions are separately corrupted into noisy action states. The ST and action branches jointly denoise these states under the shared conditioning. At inference, both branches start from independent Gaussian noise, with no future demonstration frames or future trajectories supplied. Iterative joint denoising produces the future spatial-temporal latent variables $\mathbf{z}_{t+1:t+H}$ and the corresponding action chunk. Thus, z^0, z^{τ_v}, and $\mathbf{z}_{t+1:t+H}$ describe clean training values, current noisy states, and predictions after complete denoising, respectively, rather than three different representation types. Equation 3 summarizes the complete prediction process; the functions in Eq. 4 instead produce intermediate representations within one denoising step. The Geometry Expert reads these intermediate representations, as detailed in Appendix A.2.

Table 4: Frames, historical tokens, and future spatial-temporal latent variables. Clean values from demonstrations provide training supervision.

Variable	Role
$\mathbf{x}_s$, $\mathbf{X}_{t_0:t}$	Pre-VAE spatial-temporal frame and historical frame sequence.
$\mathbf{v}_{t_0:t}$	Historical context tokens after VAE encoding, patch embedding, and flattening.
$\mathbf{z}^0$	Clean values of future spatial-temporal latent variables encoded from a demonstration.
$\mathbf{z}^{\tau_v}$	Current noisy states of the future spatial-temporal latent variables.
$\mathbf{z}_{t+1:t+H}$	Future spatial-temporal latent variables after the complete denoising process.
$\mathbf{a}^0$, $\mathbf{a}^{\tau_a}$	Clean future action target and current noisy action state.
$\mathbf{h}_{\mathrm{st}}$, $\mathbf{h}_{\mathrm{act}}$	Intermediate current-step features before geometry prediction and final velocity outputs.
$\hat{\mathbf{D}}$, $\hat{\mathbf{p}}$	Explicit future depth and end-effector predictions used for GASAM.

The Wan VAE decoder maps the generated future spatial-temporal latent variables back to spatial-temporal frames for visualization. Appearance, depth, and trajectory content in the constructed frames supplies spatial-temporal information, but this does not imply disentangled or directly readable metric coordinates in the latent variables. In particular, noisy states are not depth maps and cannot be directly unprojected; explicit metric depth and end-effector positions are predicted by the Geometry Expert from current-step intermediate representations.

Decoding the future spatial-temporal latent variables yields eight spatial-temporal frames at 384×640. The corresponding action chunk contains sixteen actions, with two action steps per frame. Both streams cover the same physical horizon despite their different lengths, and only the action chunk is executed. Executing the real-robot action chunk takes 1.6 s at 10 Hz; the interval between model updates additionally includes synchronous inference.

A.2 Geometry prediction and attention modulation

The Geometry Expert reads the current states of the future spatial-temporal latent variables and actions through their intermediate representations. At each step, the input encoders process the current noisy states, observed history, and noise levels to form the intermediate features in Eq. 4. The Geometry Expert reads these features and predicts future depth and end-effector positions. GASAM then constructs token-aligned weights and applies its log-weight bias to the action-to-video block before the joint-attention softmax. The output heads subsequently predict ST/action velocity fields that update both noisy states. This order repeats at the next step: geometry does not require a completed future video, and intermediate features are not interchangeable with the final velocity outputs.

Clean future actions are used for forward noising and action supervision. Future depth, end-effector positions, and geometry-derived weights supervise the geometry branch and modulation signal; they are not separate oracle inputs to action generation. At inference, ST/action states begin from independent Gaussian noise and all future geometry is predicted online. Thus, future trajectories used to construct training frames are never assumed available at deployment.

Scene points unprojected from predicted depth and predicted end effectors use a common coordinate frame. The head-camera view takes the smaller distance to the two arms, while each wrist view uses its corresponding arm. Equations 7 and 8 specify median-normalized distance weighting, patch-center sampling, and softmax-weighted aggregation over the full future horizon with $\beta = 20$. The bounds expand linearly from $[0.99, 1.01]$ to $[0.9, 1.1]$ during denoising, keeping early guidance close to neutral. Main experiments use $\lambda = 1$ and $\epsilon = 10^{-6}$.

The prior is continuous and bounded, with no hard threshold or visibility mask. Distant tokens remain accessible and can receive stronger weights when the predicted trajectory approaches them later in the horizon; the ST and Understanding Experts retain global context. At the final bounds, geometry alone contributes at most an approximately 1.22-fold relative preference. Figure 3 shows geometry-derived weights, not complete attention probabilities. GASAM changes Put Bottles Dustbin from 91% to 93% and Dump Bin Bigbin from 94% to 97%, while Click Alarmclock remains at 100%. These outcomes suggest that geometry-aware modulation preserves global context without impairing distant-goal reasoning.

A.3 Training provenance and comparison scope

All compared methods start from their original pretrained checkpoints and undergo the same RoboTwin 2.0 adaptation and evaluation protocol: fifty tasks, fifty Clean or five hundred Randomized demonstrations per task, batch size 256, and 40K downstream optimization steps. The reported 40K steps therefore refer to shared benchmark adaptation. Each method retains its original pretraining, and the comparison evaluates strong pretrained systems under the same downstream data and optimization budget.

Table 5 summarizes the pretraining and embodied-training resources. Motus and LingBot-VA are themselves strongly pretrained models using large-scale embodied and multimodal data. Motus reports approximately 1.01M samples and 18K GPU hours, while LingBot-VA reports approximately 16K robot hours and 1.4T tokens. STARRY inherits foundation-model priors from Wan and Qwen-VL and performs no additional web-scale continued pretraining; its egocentric training scale is comparable to Motus. Its method-specific embodied training follows Sec. 3.4, using Ego4D/EgoDex, DROID/DROID-3D, BridgeData/OXE, and target-robot data. DROID-3D augments existing DROID trajectories with geometry rather than introducing a separate collection of raw demonstrations.

The controlled ablations in Table 3 directly assess whether STARRY's gains can be explained solely by inherited pretraining. All within-STARRY variants share the same initialization, backbone, pretraining resources, downstream demonstrations, optimization budget, and evaluation protocol,

Table 5: Pretraining and embodied-training resources of the compared methods. Reported scale units are retained without conversion into a common data or compute budget; all methods subsequently follow the shared RoboTwin adaptation protocol.

Method	Training resources	Reported scale
STARRY	Ego4D/EgoDex; DROID/DROID-3D; BridgeData/OXE; target robot	Egocentric scale comparable to Motus; no additional web pretraining
Motus (Bi et al., 2026)	EgoDex; multi-robot and internal data	$\sim$1.01M samples; $\sim$18K GPU hours
LingBot-VA (Li et al., 2026)	OXE; multi-robot and internal data	$\sim$16K robot hours; 1.4T tokens
X-VLA (Zheng et al., 2026)	DROID; RoboMind; AgiBot	290K episodes; seven configurations
$\pi_{0.5}$ (Black et al., 2025a)	Cross-embodiment; semantic and web data	$\sim$400 target-robot hours; 97.6% other data

while varying the designated prediction content or modulation component. In particular, the RGB-only variant without GASAM retains future RGB prediction but removes the depth/trajectory content and geometry-aware modulation. It achieves 86.64%/85.80% Clean/Randomized success, below both Motus (88.66%/87.02%) and LingBot-VA (92.93%/91.55%). Replacing RGB-only prediction with the proposed future spatial-temporal latent variables and enabling GASAM raises success to 93.82%/93.30%, gains of 7.18/7.50 percentage points. Thus, the same pretrained foundation falls below both baselines without the proposed components and exceeds them with the complete model. These controlled results demonstrate that inherited pretraining alone does not explain STARRY's advantage.

The two components also provide separately measurable gains. With GASAM retained, replacing RGB-only prediction with Full ST improves Clean/Randomized success from 87.86%/86.96% to 93.82%/93.30%, gains of 5.96/6.34 points. With Full ST fixed, adding GASAM improves success from 90.40%/88.82% to 93.82%/93.30%, gains of 3.42/4.48 points. Removing either Full ST modeling or GASAM places the corresponding variant below LingBot-VA in both settings. Together, these comparisons establish substantial contributions from the proposed future spatial-temporal latent variables and geometry-aware modulation under fixed pretraining and downstream training resources.

For completeness, the Action-only variant with GASAM removes future spatial-temporal latent-variable generation and its denoising supervision, but retains auxiliary future-geometry prediction from historical context and intermediate action features. Its comparison with Action-only without GASAM measures the contribution of auxiliary geometry guidance without future latent-variable generation.

A.4 Inference efficiency

Table 6 summarizes the computational cost of the complete inference pipeline. Although the Geometry Expert and GASAM predict geometry and modulate attention at every denoising step, comparison with the variant without geometry guidance shows that this path introduces only limited additional latency. Together with the gains observed in the ablation study, these results demonstrate that the proposed geometry guidance improves action generation without substantially increasing inference cost.

Table 6: Inference profiling on one A100-SXM4-80GB with BF16, batch size 1, 384×640 inputs, and ten denoising steps. Latency is measured over ten runs after two warm-ups; $\pm$ denotes run-to-run SD.

Component or measurement	Value
Language processing	552.6 ms
VAE processing	532.8 ms
Geometry module	113.8 ms
Denoising	1600.9 ms
End-to-end query	2800.1 ± 35.6 ms
Without full geometry-guidance path	2658.5 ms
Geometry-guidance overhead	141.6 ms (5.1%)
Peak memory	41.58 GiB

A.5 HYPERPARAMETERS

We summarize the key numerical hyperparameters used in STARRY in Table 7. The model uses branch-specific diffusion timesteps for video and action denoising, while the Geometry Expert predicts depth and multicamera XYZ end-effector positions to provide GASAM-based action modulation.

Table 7: Key implementation hyperparameters of STARRY.

Component	Hyperparameter	Value
Policy	State / action dim.	14 / 14
	Input resolution	384×640
	Predicted video frames	8
	Action chunk size	16
	Inference denoising steps	10
	Observation / action loss weight	1.0 / 2.0
ST World Model	Hidden size / FFN dim.	3072 / 14336
	Layers / heads	30 / 24
	Input / output dim.	48 / 48
	Frequency embedding dim.	256
	Text length	512
	Layer norm epsilon	1×10^{-6}
Action Expert	Hidden size	1024
	Layers / heads	30 / 16
	FFN multiplier / norm epsilon	$4 / 1 \times 10^{-6}$
Understanding Expert	Hidden size	512
	FFN multiplier / norm epsilon	$4 / 1 \times 10^{-5}$
Geometry Expert	Video / action token dim.	3072 / 1024
	Hidden size / camera dim.	512 / 256
	Number of cameras	3
	Depth / pose / target loss weight	1.0 / 1.0 / 1.0
GASAM	λ / β / ϵ	$1 / 20 / 10^{-6}$
	Initial weight bounds	$[0.99, 1.01]$
	Final weight bounds	$[0.9, 1.1]$
Training	Optimizer	AdamW
	Training steps	40k
	Learning rate	$1\text{–}5 \times 10^{-5}$
	Weight decay	0.01

A.6 COMPUTE RESOURCES

For fine-tuning and evaluation, we use GPU workers equipped with 8 NVIDIA Tesla A100-80G GPUs. Each worker provides 112 vCPUs, 1960 GiB memory, 80 Gbit/s network bandwidth, and 32 Gbit/s

disk bandwidth. A typical fine-tuning run with batch size 256 for 40k steps takes approximately one week on this configuration. STARRY's method-specific embodied training uses the same GPU-worker configuration, with the number of nodes scaled according to the training stage and data volume. The main reported experiments, including RoboTwin 2.0 evaluation and real-world policy evaluation, are conducted with this A100-based configuration.

A.7 PSEUDOCODE OF STARRY

Algorithm 1 summarizes the training procedure of STARRY. For each training batch, we first construct spatial-temporal inputs from multi-view RGB-D observations and projected end-effector trajectories, and encode them into video tokens. The *Understanding Expert* provides semantic grounding tokens, while the *ST World Model* and *Action Expert* jointly denoise future spatial-temporal latent variables and action sequences under branch-specific diffusion timesteps.

At the sampled noise levels, the *Geometry Expert* predicts future depth and end-effector positions from the current intermediate ST/action representations , from which GASAM computes token-aligned geometry-aware weights. These weights are injected only into the action attention branch, so geometric priors guide action generation without perturbing video modeling or visual-language understanding.

Algorithm 1 STARRY training with spatial-temporal denoising and GASAM

Require: Training dataset $\mathcal{B} = \{(\mathbf{o}_{t_0:t+H}, \mathbf{a}_{t+1:t+H})\}$ with associated historical actions and camera calibration
Require: ST World Model f_θ^{ST}, Action Expert f_θ^a, Geometry Expert g_ϕ, Understanding Expert h_ψ
Ensure: Optimized parameters θ, ϕ, ψ

1: **for** each training iteration **do**
2: Sample a mini-batch from $\mathcal{B}$
3: Project end-effector trajectories into camera views as $\mathbf{u}_\tau^{c,m} = \Pi(K_t^c, T_t^c, \mathbf{e}_\tau^m)$, and construct historical and future spatial-temporal inputs:

$$\mathbf{x}_t = \Phi(\{\mathbf{I}_t^c\}_c, \{\mathbf{D}_t^c\}_c, \{\mathbf{u}_\tau^{c,m}\}_{\tau \le t, c, m})$$

4: Encode history as $\mathbf{v}_{t_0:t}$ and future spatial-temporal frames as clean future spatial-temporal latent variables z^0 (Eq. 3); obtain understanding tokens $\mathbf{s} = h_\psi(\mathbf{o}_t)$
5: Sample branch-specific diffusion timesteps (τ_v, τ_a) and corrupt z^0 and clean future actions with independent Gaussian noise to obtain z^{τ_v} and a^{τ_a}; compute their intermediate features using the observed conditioning and noise levels (Eq. 4)
6: Predict future geometry with the Geometry Expert:

$$(\hat{D}_{t+1:t+H}^{(\tau_v,\tau_a)}, \hat{p}_{t+1:t+H}^{(\tau_v,\tau_a)}) = g_\phi(h_{\mathrm{st}}^{(\tau_v,\tau_a)}, h_{\mathrm{act}}^{(\tau_v,\tau_a)}, \tau_v, \tau_a)$$

7: Compute geometry-aware weights $\mathbf{w}_{t+1:t+H}$ from distances between predicted end-effector positions and scene points unprojected from predicted depth in a common coordinate frame, aggregating over the future horizon using Eqs. 7 and 8
8: Apply GASAM only to action-to-video attention:

$$\mathrm{Attn}_{\mathrm{GASAM}}^{a \leftarrow v} = \mathrm{Softmax}\left(\frac{\mathbf{Q}^a (\mathbf{K}^v)^\top}{\sqrt{d}} + \lambda \log(\mathbf{w}_{t+1:t+H} + \epsilon) \right) \mathbf{V}^v$$

9: Compute joint-attention features from the current noisy states and observed conditioning, with GASAM applied to the action-to-video logits
10: Predict velocity fields $\mathbf{v}_\theta^o$ and $\mathbf{v}_\theta^a$ for the future spatial-temporal latent variables and actions using the ST and Action output heads, respectively
11: Compute diffusion loss $\mathcal{L}_{\mathrm{diff}} = \lambda_o \mathcal{L}_{\mathrm{obs}} + \lambda_a \mathcal{L}_{\mathrm{action}}$
12: Compute geometry loss $\mathcal{L}_{\mathrm{geo}} = \lambda_d \mathcal{L}_{\mathrm{depth}} + \lambda_p \mathcal{L}_{\mathrm{pose}} + \lambda_w \mathcal{L}_{\mathrm{weight}}$
13: Update parameters using $\mathcal{L} = \mathcal{L}_{\mathrm{diff}} + \mathcal{L}_{\mathrm{geo}}$
14: **end for**

During inference, STARRY follows the same denoising structure but removes all supervision terms.

Starting from independent Gaussian noise for the future spatial-temporal latent variables and actions , the model iteratively predicts future geometry, constructs GASAM weights, and

updates the future spatial-temporal latent variables and actions through joint denoising . The final action sequence $\mathbf{a}_{t+1:t+H}$ is executed by the robot, while the predicted future spatial-temporal latent variables and geometry serve as internal guidance for action generation.

B Supplementary experiments and analysis

B.1 Robustness and embodiment adaptation

To assess robustness to imperfect geometric observations in practical deployment, we replace simulator ground-truth depth with Depth Anything V3 (Lin et al., 2025) estimates while keeping the other training and evaluation settings unchanged. Table 8 shows a decrease of only 2.04 percentage points, suggesting that STARRY retains effective geometric guidance despite depth-estimation errors. Our physical robot also uses Depth Anything V3 to estimate depth from RGB, providing further evaluation under practical sensing conditions. We additionally examine guidance strength by varying inference-time λ on the same Full-ST checkpoint without retraining. As shown in Table 3(c), both weaker and stronger modulation underperform the default setting, suggesting that an appropriate strength balances predicted geometric guidance with the original attention.

To evaluate whether a model trained on one robot embodiment can be efficiently adapted to another, we start from the STARRY checkpoint already trained on Aloha-agilex and further fine-tune it on simulated ARX-X5. Adaptation uses ten Clean and ten Randomized demonstrations per task for 10K steps. The results in Table 8 support the reuse of the learned representations across embodiments, demonstrating effective adaptation of the trained Aloha-agilex model to ARX-X5 with limited target data and fine-tuning.

Table 8: Robustness to estimated depth and cross-embodiment adaptation on RoboTwin 2.0. All results report Randomized success. The depth comparison uses Aloha-agilex. ARX-X5 adaptation starts from the checkpoint already trained on Aloha-agilex and uses twenty target demonstrations per task and 10K additional fine-tuning steps.

Experiment	Configuration	Success (%)
Depth source	Simulator ground-truth depth	93.30
	Depth Anything V3 estimates	91.26
Embodiment	Aloha-agilex (source)	93.30
	ARX-X5 (adapted)	85.00

B.2 Repeated physical evaluation

Each method uses a fixed trained checkpoint across four evaluation batches, each containing twenty trials per task-stage setting. This gives eighty outcomes for each success criterion. Stage 1 measures the key subskill and Stage 2 full completion. Table 9 reports per-setting means, sample SDs across batches, and Wilson 95% intervals. The intervals use eighty outcomes per setting.

Table 9: Repeated physical evaluation. Mean rates, sample SDs, and Wilson 95% intervals are percentages. Each interval pools eighty outcomes; SDs are across four evaluation batches.

Task and stage	Method	Mean ± SD	Wilson 95% CI
Hand Over Vegetables, S1	$\pi_{0.5}$	53.8 ± 9.5	$[42.9, 64.3]$
	STARRY	80.0 ± 7.1	$[70.0, 87.3]$
Hand Over Vegetables, S2	$\pi_{0.5}$	30.0 ± 9.1	$[21.1, 40.8]$
	STARRY	67.5 ± 6.5	$[56.6, 76.8]$
Tidy Up Room, S1	$\pi_{0.5}$	48.8 ± 11.1	$[38.1, 59.5]$
	STARRY	72.5 ± 6.5	$[61.9, 81.1]$
Tidy Up Room, S2	$\pi_{0.5}$	26.3 ± 11.1	$[17.9, 36.8]$
	STARRY	60.0 ± 7.1	$[49.0, 70.0]$
Wash Baby Bottle, S1	$\pi_{0.5}$	32.5 ± 11.9	$[23.2, 43.4]$
	STARRY	67.5 ± 6.5	$[56.6, 76.8]$
Wash Baby Bottle, S2	$\pi_{0.5}$	18.8 ± 10.3	$[11.7, 28.7]$
	STARRY	55.0 ± 4.1	$[44.1, 65.4]$

For the macro-average, we first average the six task-stage rates within each batch, then compute the mean and sample SD over four batch averages. STARRY obtains $67.1 \pm 3.1\%$, versus $35.0 \pm 5.6\%$ for $\pi_{0.5}$; the paired batch gain is 32.1 ± 4.0 percentage points. STARRY wins all 24 task-stage-by-batch comparisons, and the pooled Wilson confidence intervals are non-overlapping between the two methods in all six task-stage settings.

B.3 Non-saturated tasks analysis and failure analysis

We select all Randomized tasks on which both Motus and LingBot-VA score at most 90%, independently of STARRY's scores. Table 10 includes the complete ten-task subset. STARRY averages 83.9%, compared with 76.5% and 75.1%, and exceeds the stronger per-task baseline on eight tasks. Handover Block and Put Object Cabinet remain regressions, with 48% versus 78% and 77% versus 87%, respectively. Across all fifty Randomized tasks, the win/tie/loss counts are 32/8/10 against LingBot-VA and 43/5/2 against Motus. The complete results below retain both gains and regressions.

Table 10: Complete non-saturated Randomized subset, selected using only the two baseline scores. Success rates are percentages.

Task	Motus	LingBot-VA	STARRY
Handover Block	73	78	48
Hanging Mug	38	28	72
Move Stapler Pad	85	79	93
Place Bread Skillet	83	90	95
Place Can Basket	76	84	88
Place Dual Shoes	87	89	94
Place Object Basket	87	88	91
Put Object Cabinet	71	87	77
Stack Bowls Three	87	83	92
Turn Switch	78	45	89
Average	76.5	75.1	83.9

STARRY underperforms LingBot-VA on Handover Block, Open Microwave, and Blocks Ranking Size, which impose strict terminal requirements beyond core manipulation: stable final placement, sufficient door opening, and fine-grained size ordering, respectively. Trajectory overlays may interfere with the subtle visual cues required at these terminal states. Nevertheless, Table 3(b) shows that adding trajectory information consistently improves performance both with and without GASAM. With GASAM, adding trajectories improves Randomized success by 1.10 percentage points over RGB-only prediction and by 2.42 points over RGB + depth prediction. These results support the overall benefit of trajectory modeling despite its potential interference with fine-grained terminal-state discrimination in particular tasks.

B.4 Further ablation analysis

Table 3(a) examines the contributions of future prediction and geometry-aware modulation. Without GASAM, moving from Action-only denoising to RGB prediction improves success from 63.42% to 86.64% under Clean and from 64.96% to 85.80% under Randomized settings, supporting the value of future visual context for action generation. Full ST further improves these results to 90.40% and 88.82%, respectively. This suggests that explicitly modeling spatial geometry and end-effector motion provides useful information beyond future appearance, particularly for interactions requiring contact, alignment, and coordinated movement. Enabling GASAM further improves all representation variants, with Full ST reaching 93.82% under Clean and 93.30% under Randomized settings. These results support the complementary roles of spatial-temporal world modeling and geometry-aware modulation: the former captures future scene structure and motion, while the latter selectively guides action attention toward regions relevant to future interactions.

Table 3(b) further shows that the benefit of modulation depends on the prediction content. Adding depth to RGB prediction improves Randomized success by 1.72 points without GASAM and 3.92 points with GASAM. Across RGB, RGB + trajectory, RGB + depth, and Full ST, enabling GASAM yields gains of 1.16, 1.54, 3.36, and 4.48 points, respectively. The larger gains with depth-bearing representations are consistent with explicit scene structure providing useful information for geometric guidance. Trajectory information also improves both RGB and RGB + depth prediction, supporting its complementary role in describing end-effector motion relative to the scene. Full ST combines these spatial and temporal cues and achieves the strongest performance with and without GASAM.

B.5 Visualization of future spatial-temporal latent variables

To further illustrate what is captured by the learned future spatial-temporal latent variables, we visualize decoded future predictions in both simulation and real-world settings. These visualizations are not used as additional supervision during evaluation, but serve to qualitatively examine whether the predicted future spatial-temporal latent variables preserve temporally coherent scene evolution, end-effector motion, and spatial geometry. As shown below, STARRY predicts not only future appearance changes, but also structured depth and geometry patterns that are closely aligned with manipulation-relevant regions.

Fig. 6 shows representative visualizations on RoboTwin 2.0. The decoded predictions maintain coherent future scene evolution across diverse simulated tasks, including changes in object positions, manipulator configurations, and depth structure. This supports our claim that the ST World Model learns future spatial-temporal latent variables rather than only predicting visually plausible RGB frames.

Fig. 7 further visualizes future spatial-temporal latent variables in real-world tasks. Despite real-world visual noise, lighting variation, and cluttered backgrounds, the predictions remain temporally coherent and preserve task-relevant spatial layouts. These results indicate that the learned future spatial-temporal latent variables provide useful predictive guidance for physical manipulation, complementing the quantitative real-world results reported in the main paper.

B.6 Complete RoboTwin 2.0 results

Table 11 reports the complete task-level results on all 50 RoboTwin 2.0 tasks under Clean and Randomized settings. STARRY achieves the best average success rates, reaching **93.82% / 93.30%**. Overall averages are computed from unrounded per-task success rates and may therefore differ slightly from the means of the rounded values displayed in the table.

Table 12 provides the complete task-level ablation results. Full spatial-temporal modeling with GASAM achieves the strongest performance, improving from 88.82% to 93.30% under Randomized and from 90.40% to 93.82% under Clean compared with disabling GASAM. These results further validate the complementary effects of spatial-temporal modeling and geometry-aware action modulation.

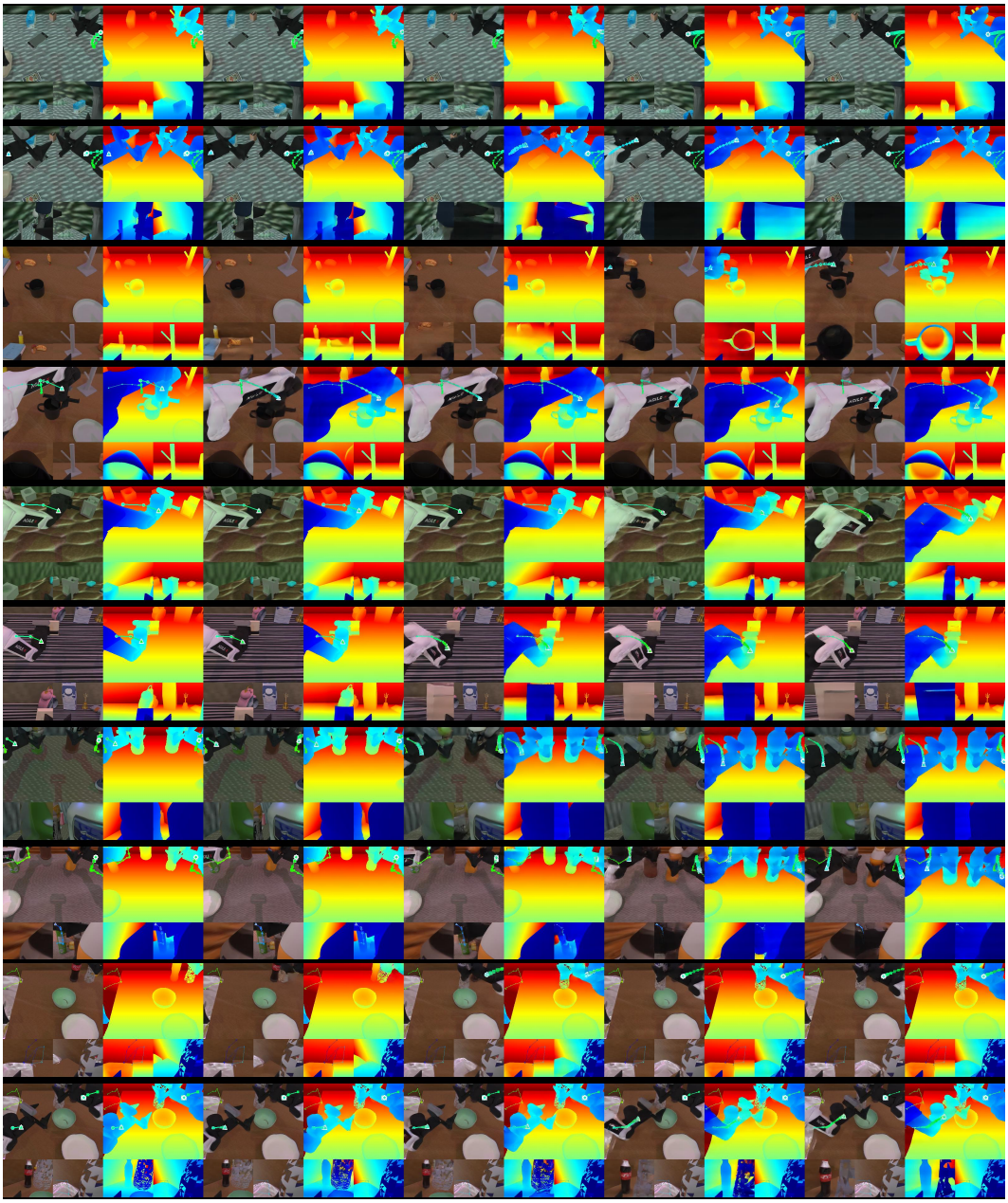

Figure 6: Visualization of future spatial-temporal latent variables on RoboTwin 2.0. Each row corresponds to a manipulation rollout, and columns show consecutive future predictions decoded from the future spatial-temporal latent variables. The predicted future spatial-temporal latent variables preserve object layouts, end-effector motion trends, and depth-related spatial structure across future steps, indicating that the ST World Model captures more than appearance-level temporal consistency.

Table 11: Complete task-level results on RoboTwin 2.0 under Clean and Randomized settings. We report success rates over all 50 tasks for each method. Bold numbers indicate the best result for each task and setting; ties are all bolded.

Simulation Task	GO-1		π_0		$\pi_{0.5}$		X-VLA		Motus		LingBot-VA		Ours	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	49%	62%	99%	95%	89.5%	91%	**100%**	**99%**	89%	93%	90%	94%	98%	97%
Beat Block Hammer	6%	10%	79%	84%	79.5%	71.5%	92%	88%	95%	88%	**96%**	**98%**	92%	89%
Blocks Ranking Rgb	7%	3%	80%	63%	67.5%	60%	83%	83%	99%	97%	99%	**98%**	**100%**	**98%**
Blocks Ranking Size	2%	2%	14%	5%	28.5%	20%	67%	74%	75%	63%	**94%**	**96%**	71%	74%
Click Alarmclock	95%	90%	77%	68%	97.5%	91%	99%	99%	**100%**	**100%**	99%	**100%**	**100%**	**100%**
Click Bell	98%	95%	71%	48%	87%	71%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Dump Bin Bigbin	57%	45%	88%	83%	61%	69.5%	79%	77%	**95%**	91%	89%	96%	93%	**97%**
Grab Roller	99%	99%	98%	94%	95%	94.5%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Handover Block	9%	12%	47%	31%	42%	38%	73%	37%	86%	73%	**99%**	**78%**	56%	48%
Handover Mic	12%	8%	97%	97%	63%	57.5%	0%	0%	78%	63%	94%	96%	**100%**	**99%**
Hanging Mug	0%	0%	14%	11%	10.5%	10%	23%	27%	38%	38%	40%	28%	**69%**	**72%**
Lift Pot	92%	92%	80%	72%	48%	42.5%	99%	**100%**	96%	99%	**100%**	99%	**100%**	**100%**
Move Can Pot	16%	4%	68%	48%	40%	41%	89%	86%	34%	74%	94%	97%	**100%**	**98%**
Move Pillbottle Pad	9%	11%	67%	46%	58.5%	45%	73%	71%	93%	96%	99%	99%	**100%**	**100%**
Move Playingcard Away	37%	24%	74%	65%	77.5%	75.5%	93%	98%	**100%**	96%	**100%**	99%	**100%**	**100%**
Move Stapler Pad	3%	4%	41%	24%	36%	30%	78%	73%	83%	85%	91%	79%	**93%**	**93%**
Open Laptop	65%	60%	71%	81%	54.5%	65.5%	93%	**100%**	**95%**	91%	92%	94%	93%	91%
Open Microwave	12%	14%	4%	32%	34.5%	57%	79%	71%	**95%**	**91%**	82%	86%	58%	61%
Pick Diverse Bottles	61%	56%	69%	31%	43%	37%	58%	36%	90%	91%	89%	82%	**98%**	**96%**
Pick Dual Bottles	81%	74%	59%	37%	51.5%	34.5%	47%	36%	96%	90%	**100%**	99%	**100%**	**100%**
Place A2b Left	33%	36%	43%	47%	74.5%	71%	48%	49%	85%	79%	**97%**	**93%**	96%	91%
Place A2b Right	31%	22%	39%	34%	74.5%	70.5%	36%	36%	90.5%	87%	**97%**	95%	94%	**96%**
Place Bread Basket	47%	52%	62%	46%	62.5%	60%	81%	71%	91%	94%	97%	95%	**100%**	**99%**
Place Bread Skillet	2%	1%	66%	49%	61.5%	56%	77%	67%	86%	83%	95%	90%	**96%**	**95%**
Place Burger Fries	88%	92%	81%	76%	80%	78.5%	94%	94%	98%	98%	97%	95%	**99%**	**99%**
Place Can Basket	29%	37%	55%	46%	40.5%	43.5%	49%	52%	81%	76%	81%	84%	**89%**	**88%**
Place Cans Plasticbox	68%	77%	63%	45%	67%	65.5%	97%	98%	98%	94%	**100%**	99%	**100%**	**100%**
Place Container Plate	73%	70%	97%	92%	85%	86.5%	97%	95%	98%	99%	99%	97%	**100%**	**100%**
Place Dual Shoes	6%	10%	59%	51%	43.5%	41%	79%	88%	93%	87%	94%	89%	**96%**	**94%**
Place Empty Cup	44%	39%	91%	85%	87.5%	92.5%	**100%**	98%	99%	98%	**100%**	**100%**	**100%**	**100%**
Place Fan	1%	0%	66%	71%	56%	60.5%	80%	75%	91%	87%	**99%**	**93%**	95%	92%
Place Mouse Pad	15%	10%	20%	20%	40.5%	32.5%	70%	70%	66%	68%	93%	96%	**96%**	**98%**
Place Object Basket	48%	49%	67%	70%	61.5%	56%	44%	39%	81%	87%	**91%**	88%	89%	**91%**
Place Object Scale	26%	27%	57%	52%	63%	64.5%	52%	74%	88%	85%	**96%**	**95%**	89%	88%
Place Object Stand	56%	63%	82%	68%	82.5%	75%	86%	88%	98%	97%	99%	96%	**100%**	**100%**
Place Phone Stand	30%	37%	49%	53%	65%	67%	88%	87%	87%	86%	**97%**	**97%**	91%	**97%**
Place Shoe	15%	13%	76%	76%	74.5%	77%	96%	95%	99%	97%	98%	98%	**100%**	**99%**
Press Stapler	66%	51%	44%	37%	83.5%	76.5%	92%	98%	93%	98%	85%	82%	**100%**	**100%**
Put Bottles Dustbin	7%	4%	65%	56%	48%	44%	74%	77%	81%	79%	87%	91%	**96%**	**93%**
Put Object Cabinet	60%	43%	73%	60%	52%	47%	46%	48%	**88%**	71%	85%	**87%**	84%	77%
Rotate Qrcode	22%	9%	74%	70%	68%	71.5%	34%	33%	89%	73%	**96%**	**91%**	94%	84%
Scan Object	1%	2%	55%	42%	57%	51.5%	14%	36%	67%	66%	**96%**	91%	93%	**96%**
Shake Bottle Horizontally	97%	92%	98%	92%	97.5%	99.5%	**100%**	**100%**	**100%**	98%	**100%**	99%	**100%**	99%
Shake Bottle	97%	93%	94%	91%	95%	98.5%	99%	**100%**	**100%**	97%	**100%**	97%	**100%**	97%
Stack Blocks Three	1%	1%	72%	52%	53%	46%	6%	10%	91%	95%	**99%**	98%	97%	**100%**
Stack Blocks Two	12%	22%	93%	79%	72.5%	78%	92%	87%	**100%**	98%	**100%**	98%	**100%**	**100%**
Stack Bowls Three	4%	7%	77%	75%	55%	53%	76%	86%	79%	87%	86%	83%	**91%**	**92%**
Stack Bowls Two	51%	45%	94%	95%	86.5%	81%	96%	93%	98%	98%	94%	98%	**100%**	**99%**
Stamp Seal	19%	13%	46%	33%	57.5%	39%	76%	82%	93%	92%	96%	97%	**100%**	**99%**
Turn Switch	34%	30%	41%	42%	33.5%	30%	40%	61%	84%	78%	44%	45%	**85%**	**89%**
Average (%)	37.8%	36.24%	65.92%	58.4%	62.86%	60.3%	72.8%	72.84%	88.66%	87.02%	92.93%	91.55%	**93.82%**	**93.3%**

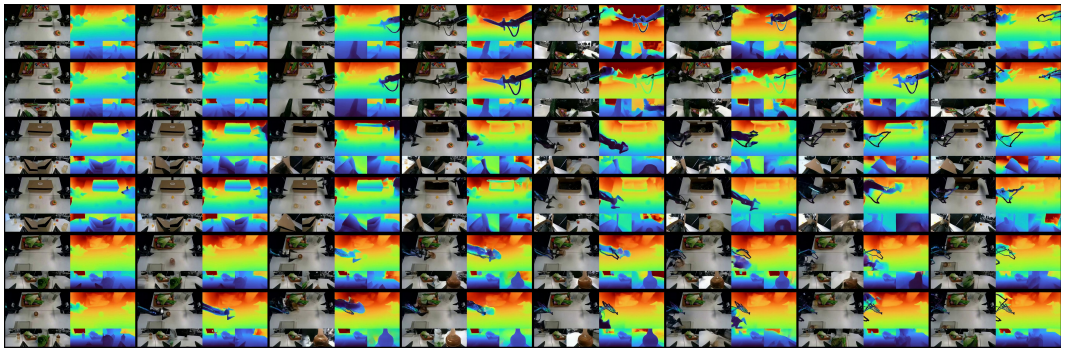

Figure 7: Visualization of future spatial-temporal latent variables in real-world manipulation. The decoded predictions show temporally consistent RGB-D structure under physical robot execution, including future end-effector motion and task-relevant spatial changes.

Table 12: Task-level ablation on RoboTwin 2.0. **Act.**: Action-Only Denoising. **App.**: Appearance-Only Prediction. **ST**: Full Spatial-Temporal Modeling. **Dis./En.**: GASAM disabled/enabled.

Task	Randomized						Clean					
	Act.		App.		ST		Act.		App.		ST	
	Dis.	En.	Dis.	En.	Dis.	En.	Dis.	En.	Dis.	En.	Dis.	En.
Adjust Bottle	91%	90%	91%	100%	100%	97%	96%	98%	99%	97%	97%	98%
Beat Block Hammer	59%	77%	62%	88%	90%	89%	49%	64%	75%	88%	88%	92%
Blocks Ranking Rgb	11%	23%	95%	94%	95%	98%	13%	12%	91%	96%	98%	100%
Blocks Ranking Size	9%	60%	66%	66%	71%	74%	6%	61%	70%	64%	74%	71%
Click Alarmclock	100%	98%	100%	100%	100%	100%	100%	100%	99%	100%	100%	100%
Click Bell	100%	99%	100%	97%	99%	100%	100%	100%	100%	100%	100%	100%
Dump Bin Bigbin	63%	87%	89%	85%	94%	97%	69%	90%	94%	89%	94%	93%
Grab Roller	96%	100%	99%	100%	100%	100%	99%	98%	100%	100%	100%	100%
Handover Block	8%	28%	32%	32%	28%	48%	15%	13%	21%	31%	24%	56%
Handover Mic	78%	86%	69%	97%	99%	99%	75%	99%	95%	100%	99%	100%
Hanging Mug	11%	8%	53%	50%	51%	72%	6%	10%	43%	42%	68%	69%
Lift Pot	86%	97%	98%	100%	100%	100%	85%	90%	100%	100%	100%	100%
Move Can Pot	66%	85%	92%	69%	95%	98%	54%	61%	78%	91%	96%	100%
Move Pillbottle Pad	89%	95%	97%	98%	100%	100%	85%	72%	97%	100%	99%	100%
Move Playingcard Away	95%	96%	100%	100%	97%	100%	95%	96%	98%	98%	86%	100%
Move Stapler Pad	32%	80%	60%	70%	75%	93%	31%	71%	73%	87%	87%	93%
Open Laptop	90%	94%	94%	87%	87%	91%	89%	94%	94%	93%	90%	93%
Open Microwave	39%	10%	76%	20%	19%	61%	28%	43%	68%	29%	34%	58%
Pick Diverse Bottles	78%	17%	89%	92%	95%	96%	90%	38%	80%	90%	95%	98%
Pick Dual Bottles	89%	15%	95%	100%	99%	100%	96%	21%	99%	99%	100%	100%
Place A2b Left	75%	87%	89%	82%	84%	91%	71%	83%	79%	91%	90%	96%
Place A2b Right	71%	84%	91%	85%	92%	96%	69%	76%	93%	88%	95%	94%
Place Bread Basket	74%	86%	88%	96%	93%	99%	78%	74%	94%	96%	95%	100%
Place Bread Skillet	65%	87%	81%	93%	88%	95%	70%	71%	87%	88%	93%	96%
Place Burger Fries	99%	99%	89%	93%	93%	99%	95%	97%	100%	92%	96%	99%
Place Can Basket	24%	65%	79%	72%	78%	88%	20%	44%	88%	86%	84%	89%
Place Cans Plasticbox	87%	95%	95%	91%	100%	100%	82%	65%	99%	99%	100%	100%
Place Container Plate	94%	100%	98%	97%	100%	100%	93%	98%	99%	100%	100%	100%
Place Dual Shoes	94%	92%	72%	98%	97%	94%	58%	85%	80%	75%	94%	96%
Place Empty Cup	96%	100%	99%	99%	99%	100%	94%	95%	98%	100%	100%	100%
Place Fan	63%	86%	88%	89%	87%	92%	60%	87%	89%	85%	84%	95%
Place Mouse Pad	35%	80%	82%	83%	86%	98%	39%	70%	77%	79%	94%	96%
Place Object Basket	17%	51%	87%	89%	91%	91%	23%	57%	90%	84%	91%	89%
Place Object Scale	61%	86%	84%	90%	87%	88%	64%	83%	87%	90%	88%	89%
Place Object Stand	98%	97%	99%	100%	99%	100%	93%	97%	96%	98%	100%	100%
Place Phone Stand	67%	81%	87%	90%	93%	97%	67%	73%	84%	88%	93%	91%
Place Shoe	91%	97%	100%	99%	95%	99%	88%	96%	100%	99%	99%	100%
Press Stapler	91%	98%	100%	96%	95%	100%	84%	99%	99%	99%	97%	100%
Put Bottles Dustbin	7%	31%	73%	89%	91%	93%	5%	30%	82%	86%	92%	96%
Put Object Cabinet	19%	32%	46%	74%	78%	77%	7%	31%	41%	71%	75%	84%
Rotate Qrcode	57%	71%	80%	77%	79%	84%	72%	78%	83%	70%	84%	94%
Scan Object	52%	81%	87%	87%	96%	96%	58%	70%	85%	87%	90%	93%
Shake Bottle	97%	96%	98%	94%	97%	97%	99%	100%	99%	98%	96%	100%
Shake Bottle Horizontally	94%	97%	97%	97%	96%	99%	99%	100%	96%	98%	97%	100%
Stack Blocks Three	28%	94%	95%	96%	99%	100%	23%	90%	91%	96%	99%	97%
Stack Blocks Two	73%	87%	100%	100%	98%	100%	65%	55%	94%	100%	100%	100%
Stack Bowls Three	20%	42%	87%	86%	85%	92%	17%	37%	88%	90%	92%	91%
Stack Bowls Two	74%	98%	92%	99%	93%	99%	66%	95%	97%	96%	98%	100%
Stamp Seal	85%	91%	95%	99%	98%	99%	80%	93%	99%	99%	98%	100%
Turn Switch	50%	58%	75%	63%	80%	89%	51%	65%	64%	71%	77%	85%
Average (%)	64.96%	75.88%	85.80%	86.96%	88.82%	93.30%	63.42%	72.50%	86.64%	87.86%	90.40%	93.82%

C LIMITATIONS

Although STARRY improves spatial-temporal action generation in both simulation and real-world experiments, several limitations remain. First, real-world evaluation is conducted on a limited number of bimanual manipulation tasks, and broader validation across more robot embodiments and open-ended environments is needed. Second, highly ambiguous observations may reduce the reliability of the geometric predictions used by GASAM, although the Depth Anything V3 ablation shows limited degradation under estimated depth (Table 8). Finally, STARRY builds on large video and vision-language backbones, which increases training cost compared with smaller reactive policies.

D IMPACT STATEMENT

STARRY aims to improve robotic manipulation by enhancing spatial-temporal prediction and geometry-aware action generation. This may benefit applications such as assistive robotics, manufacturing, and household automation by improving execution reliability in spatially demanding tasks. Potential risks include unsafe robot behaviors under distribution shift or inappropriate deployment in unstructured environments. Therefore, STARRY should be viewed as a research-stage system and deployed only with appropriate physical safety constraints, monitoring, and human oversight.