JAMB: Joint Action–Motion Diffusion for Bimanual Manipulation

Chuyang Xiao[1,*], Peilin Meng[2,*], and David Held[1,†]

***Abstract*— Coordinated bimanual manipulation is challenging because the motion of either arm can alter the shared 3D scene and thereby affect the other arm. Yet most diffusion policies generate actions without explicitly modeling these future geometric consequences, while predictive variants typically use future state only as auxiliary supervision or fixed conditioning. We address this limitation by proposing JAMB, a diffusion policy that jointly denoises bimanual actions and future 3D point tracks. By allowing action and track hypotheses to evolve together within a shared Transformer, each can inform and refine the other throughout denoising. We further ground multimodal representations in a shared spatiotemporal coordinate system to facilitate geometry-aware interaction during joint denoising. We evaluate JAMB on diverse bimanual manipulation tasks in RoboTwin 2.0 and on a real-world robot, comparing it with action-only policies and alternative future-prediction approaches spanning different state representations and learning objectives. Across 16 simulation tasks, JAMB achieves an average success rate of 83.4%, outperforming the strongest baseline by 23.9 percentage points. On three real-world tasks, it outperforms the action-only and auxiliary geometry prediction methods by 50.0 and 21.2 percentage points, respectively. Beyond these performance gains, JAMB shows stronger generalization to cluttered scenes and out-of-distribution backgrounds than the evaluated baselines. Together, these results demonstrate the effectiveness of our joint action–motion modeling framework for coordinated bimanual manipulation. Our project website is available at `https://jam-bimanual.github.io/`.**

I. INTRODUCTION

Bimanual manipulation requires two robot arms to coordinate their motions while interacting with shared objects and the surrounding environment [1]–[4]. Although geometric reasoning is important for manipulation in general, it becomes particularly critical in bimanual settings, where the two arms are coupled through object contacts and coordinated interactions [5], [6]. The motion of one arm can alter the object pose and the feasible motion of the other, making successful execution depend on their joint effect on the scene. A bimanual policy must therefore reason not only about the immediate commands of each arm, but also about how their coordinated actions are expected to change the scene over time [7]–[9].

Diffusion policies provide a strong framework for bimanual manipulation because they can generate temporally coherent action chunks and model complex action distributions [10]. However, most diffusion policies are trained only to reconstruct demonstrated actions from the current observation. Their objectives do not explicitly connect action generation with predictions of the resulting object motion or scene changes, limiting the policy's ability to coordinate the two arms around their effects on shared objects.

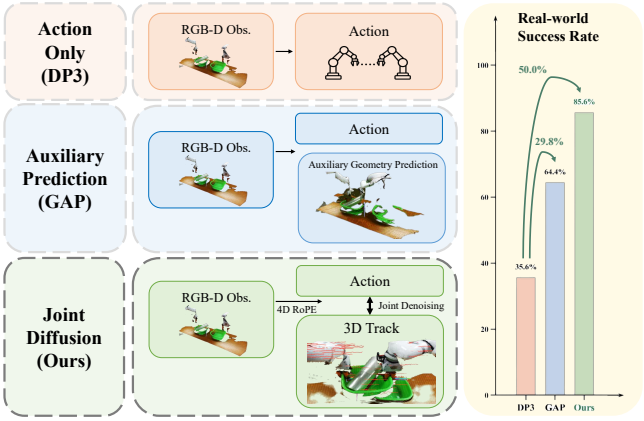

Fig. 1. Paradigm comparison and real-world experiment results. Action-only method denoises robot actions, while auxiliary geometry prediction method augments action prediction with terminal 3D geometry feature prediction. In contrast, our method jointly denoises actions and future 3D tracks, enabling mutual refinement. Our method achieves an average success rate of **85.6%**, compared with 64.4% for GAP and 35.6% for DP3.

Recent approaches address this limitation by predicting future images, video latents, or dense 3D geometry alongside robot actions [8], [9], [11]–[13]. The choice of future representation determines what information is made available to the policy. Pixel- and video-based representations preserve rich visual context, but also model appearance details that may be irrelevant to control. Dense 3D geometry prediction provides rich spatial information, but predicting the complete scene often wastes computational capacity on representing task-irrelevant regions. For manipulation, it is more useful to explicitly model how scene elements move and evolve over time. Point tracks provide such a motion representation by preserving point correspondence and trajectory structure across time. Recent work shows that jointly predicting 2D tracks and actions can improve world–action modeling [14], [15]. Yet image-plane tracks do not directly encode metric depth or 3D displacement, which are important for reasoning about the relative geometry among two arms and manipulated scene objects. Moreover, track tokens possess explicit spatial and temporal relationships that may be difficult to recover from image features alone. These observations motivate us to investigate 3D point tracks with an explicit spatiotemporal positional structure as a future representation for bimanual control.

Our key idea is to couple future 3D motion prediction

[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA.
[2]University of Michigan, Ann Arbor, MI 48109, USA.
*Equal contribution. †Corresponding author.

directly with bimanual action generation. Candidate actions determine how the scene is expected to evolve, while predicted point tracks provide a geometric description of the outcome that those actions should produce. We therefore jointly model bimanual actions and future 3D point tracks within a unified diffusion framework, allowing evolving action and motion predictions to mutually refine each other. To support this interaction, we ground multimodal tokens in a shared coordinate system defined by world-space positions and trajectory time, using 4D rotary positional encoding to incorporate their relative spatial and temporal relationships into attention.

We evaluate the approach on 16 simulated bimanual manipulation tasks in RoboTwin 2.0 [16] and three tasks on a real-world bimanual robot system. JAMB outperforms the strongest baseline by 23.9 percentage points in simulation and achieves consistent improvements in the real world. JAMB also demonstrates stronger generalization to cluttered scenes and out-of-distribution backgrounds than the evaluated baselines. Ablation studies further demonstrate the benefits of joint action–motion denoising over alternative prediction designs and examine how spatiotemporal grounding affect policy performance.

Our contributions are summarized as follows:

- We jointly denoise bimanual actions and 3D point tracks, allowing action and future-motion predictions to mutually refine each other.
- We ground multimodal representations in a shared spatiotemporal coordinate system, enabling geometry-aware interactions during joint action–motion denoising.
- We demonstrate strong performance on simulated and real-world bimanual tasks, along with improved generalization to out-of-distribution scenarios compared with the evaluated baselines.

II. Related Work

A. Bimanual Visuomotor Policy Learning

Learning visuomotor policies from demonstrations has been widely studied through action-chunking and generative policy architectures, including ACT [1], Diffusion Policy [10], and its 3D extension DP3 [17]. Building on these general policy-learning frameworks, recent work has explored different mechanisms for bimanual coordination, including modeling asymmetric acting and stabilizing roles between the two arms [2], [5], transferring pretrained unimanual skills to coordinated dual-arm manipulation [3], [18]–[21], and incorporating spatial and kinematic constraints into policy learning [6]. Complementary work grounds visual and action tokens in a common 3D coordinate frame, using relative spatial attention to guide action generation [22], [23].

These approaches primarily improve coordination at the action level, without explicitly modeling the future scene outcomes induced by those actions. Such outcomes are important for bimanual manipulation because coordinated actions must jointly produce the desired object motion and spatial configuration. Our work models these consequences through future 3D point tracks, jointly predicting them with bimanual actions.

B. Future Prediction for Robot Policy

Future prediction has been incorporated into robot policies through various modalities, including visual observations, geometric representations, and motion representations. Visual world-action models jointly predict future images [8], [11], [12], [24] or videos [25]–[29] alongside robot actions, using future visual prediction during policy learning or as auxiliary supervision. While effective for modeling scene evolution, visual predictions do not explicitly encode metric 3D structure and may devote capacity to appearance details less relevant to control. Another line of work predicts geometric representations such as future depth maps [13] and 3D geometry features [9], providing a more physically grounded description of future scene states. However, dense future prediction can be computationally expensive and may allocate substantial capacity to modeling static regions that are less informative for control; further, predicting future scene states does not explicitly capture how scene motion evolves over time.

Motion representations instead focus directly on scene dynamics. Prior work has modeled motion in image space using optical flow [30] and point tracks, which preserve temporal point correspondence and have been used for action prediction and cross-embodiment transfer [31]–[34]. More recent works further jointly generate point tracks and actions [14], [15]. However, these approaches represent tracks in 2D image space rather than in a shared 3D coordinate frame with robot end-effector motion. Existing methods also incorporate 3D motion into policy learning, using predicted motion to guide robot actions [35]–[40], or using motion supervision and distilled tracker features to support action prediction [41], [42]. However, these methods do not jointly update explicit future motion and action throughout generation. In contrast, we treat future 3D point tracks and bimanual actions as diffusion variables that interact and are jointly refined at each denoising step. Each track query is further paired with its aligned visual patch, while spatiotemporal positional encoding grounds multimodal tokens in a shared 3D coordinate frame.

III. PROBLEM FORMULATION

We assume a dataset of expert trajectories

$$\mathcal{D} = \{\tau_i\}_{i=1}^{M}, \qquad \tau_i = \{(o_t, s_t, a_t)\}_{t=1}^{T_i}, \tag{1}$$

where o_t denotes the visual observation, s_t denotes the robot proprioceptive state, and $a_t \in \mathbb{R}^{D_a}$ denotes the bimanual robot action. At time step t, the policy predicts an action chunk

$$A_t = [a_t, \ldots, a_{t+H-1}] \in \mathbb{R}^{H \times D_a}. \tag{2}$$

To represent future scene motion, we associate the visual observation with a grid of N patch-aligned 3D query points. Let $q_{i,0} \in \mathbb{R}^3$ denote the current position of query i, and let $q_{i,h}$ denote its position at future step h. We represent

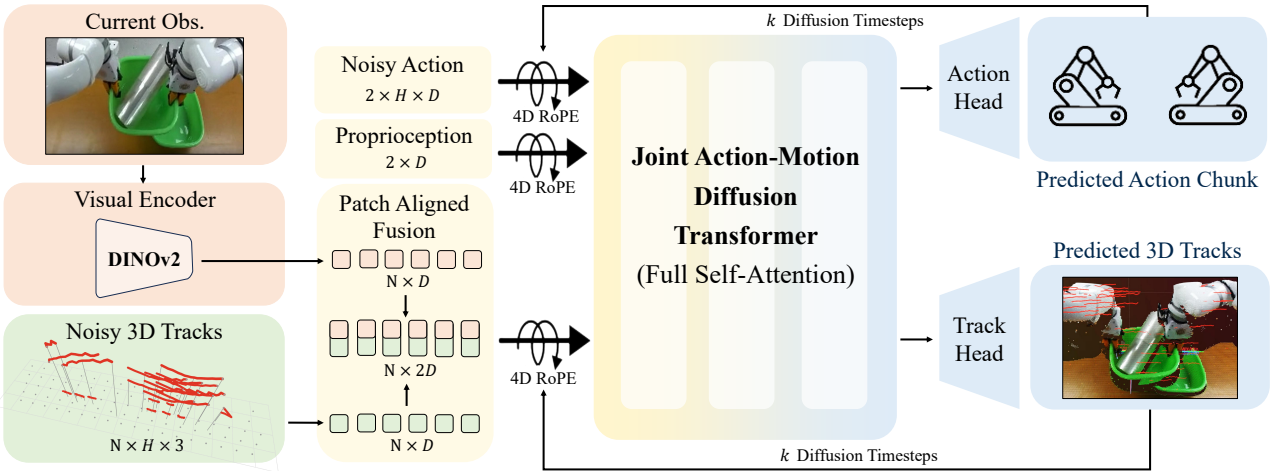

Fig. 2. Overview of JAMB. Patch tokens from a DINOv2 encoder are fused with patch-aligned noisy 3D track queries and placed in a single DiT sequence with the arm-state tokens and the noisy bimanual action chunk. Full self-attention lets actions and future 3D tracks be denoised jointly and refine each other, while 4D rotary positional encoding grounds all tokens in a shared world-space and trajectory-time frame. Only the denoised actions are executed; the denoised 3D tracks act as an internal predictive representation and for visualization.

its future trajectory using displacements from the current position:

$$P_t(i,h) = q_{i,h} - q_{i,0}, \qquad P_t \in \mathbb{R}^{N\times H\times 3}. \tag{3}$$

Each track query corresponds to a visual patch, providing aligned visual and motion representations.

Our objective is to learn the conditional joint distribution

$$p_\theta(A_t, P_t \mid o_t, s_t), \tag{4}$$

so that the action and future-motion predictions can be jointly generated and refined.

IV. Method

A. 3D Track Construction

We construct patch-aligned 3D point tracks as an explicit representation of future scene motion. For each visual patch, we initialize one image-space query at the patch center, yielding a fixed grid of N points distributed over the input image. Each query is then lifted into 3D using the corresponding depth and camera geometry, producing a set of scene points whose motion is tracked over the future horizon.

Let $q_{i,0} \in \mathbb{R}^3$ denote the current 3D position corresponding to query i, and let $q_{i,h}$ denote its position at future step h. We represent the trajectory using displacements from the current position according to Equation 3. Compared with pixel-dense motion representations, the patch-level query grid provides a lower-dimensional description of 3D scene motion while preserving point correspondence over time.

B. Joint Action–Motion Diffusion

We use a Diffusion Transformer (DiT) [43] to jointly model bimanual action sequences and future 3D tracks. A DINOv2 encoder [44] maps the current RGB observation to a set of patch tokens:

$$V = E_{\mathrm{O}}(o_t) = [v_1, \ldots, v_N]. \tag{5}$$

The proprioceptive states of the left and right arms are encoded separately:

$$x_L^S = E_S(s_t^L), \qquad x_R^S = E_S(s_t^R). \tag{6}$$

Let A_t^k and P_t^k denote the noisy action and track variables at diffusion step k. Because each track query corresponds to one visual patch, we concatenate its track embedding with the aligned visual feature and project them into a fused token:

$$z_i^k = E_{\text{fuse}}\left(v_i \| E_P(P_t^k(i,:))\right), \tag{7}$$

where $\|$ denotes feature concatenation. Each step of the noisy action chunk is encoded as:

$$x_j^{A,k} = E_A\big(a_{t+j-1}^k\big), \qquad j = 1, \ldots, H, \tag{8}$$

The complete DiT sequence consists of the fused visual–track tokens, two arm-state tokens, noisy bimanual action tokens, and a diffusion-timestep token:

$$X^k = [z_{1:N}^k; x_L^S; x_R^S; x_{1:H}^{A,k}; x_{\text{diff}}^k]. \tag{9}$$

All tokens interact through full self-attention, allowing the action and future-motion hypotheses to mutually refine each other throughout denoising.

We combine learnable positional embeddings, which encode token identity and sequence role, with 4D rotary positional encoding (RoPE) [45], which encodes relative world-space positions and trajectory time. To establish this shared spatiotemporal frame, fused visual–track tokens are anchored at their current 3D patch positions and state tokens at the current end-effector positions, both with time index zero. Action tokens are assigned future-step indices and spatial positions derived from the noisy actions. Since these positions are unreliable at high noise levels, we interpolate the action tokens' RoPE anchors between the current end-effector positions and the action-derived positions, gradually increasing the interpolation weight assigned to the action-derived positions as denoising progresses.

C. Training and Inference

Before diffusion, we apply per-dimension min–max normalization to map actions to $[-1, 1]$ and normalize 3D track displacements. We add noise to the normalized action and track targets using a shared diffusion timestep:

$$\widetilde{A}_t^k = \alpha_k \widetilde{A}_t^0 + \sigma_k \epsilon_A, \tag{10}$$

$$\widetilde{P}_t^k = \alpha_k \widetilde{P}_t^0 + \sigma_k \epsilon_P, \tag{11}$$

where ϵ_A and ϵ_P are independently sampled standard Gaussian noise. The shared DiT predicts the corresponding targets through separate output projections:

$$(\widehat{Y}_A, \widehat{Y}_P) = D_\theta(X^k), \tag{12}$$

where X^k contains the conditioning tokens and noisy action and track tokens, and Y_A and Y_P denote the diffusion prediction targets defined in the normalized spaces.

Because the patch-aligned query grid contains many nearly static background points, we follow PointWorld [46] and assign each track a motion-dependent weight based on its final displacement in metric space:

$$w_{b,i} = w_{\min} + (1 - w_{\min})\sigma\left(\kappa\left(\left\|P^0_{b,i,H}\right\|_2 - \tau\right)\right), \tag{13}$$

where $\sigma(\cdot)$ denotes the sigmoid function, τ is the motion threshold, κ controls the transition sharpness, and $w_{\min}$ specifies the minimum weight assigned to nearly static points. Here, P^0 denotes the unnormalized track displacement.

The motion-weighted track loss is

$$\mathcal{L}_{\text{track}} = \frac{\sum_{b,i} w_{b,i} \sum_{h=1}^{H} \sum_{d=1}^{3} \left(\widehat{Y}_{P,b,i,h,d} - Y_{P,b,i,h,d}\right)^2}{3 \sum_{b,i} w_{b,i} + \varepsilon}, \tag{14}$$

where b indexes training samples and i indexes patch-aligned track queries. The loss averages over spatial coordinates, sums across the prediction horizon, and takes a motion-weighted average over all track queries. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda_{\text{track}} \mathcal{L}_{\text{track}}, \tag{15}$$

where $\mathcal{L}_{\text{action}}$ is the mean squared error between the predicted and target action outputs, and λ_{track} balances the two objectives.

At inference time, normalized actions and tracks are initialized independently from Gaussian noise:

$$\widetilde{A}_t^K \sim \mathcal{N}(0, I), \qquad \widetilde{P}_t^K \sim \mathcal{N}(0, I), \tag{16}$$

and updated jointly using the same denoising schedule. After the final denoising step, both outputs are de-normalized using their respective training-set statistics. Only the predicted actions are executed, while the generated tracks are retained for analysis and visualization.

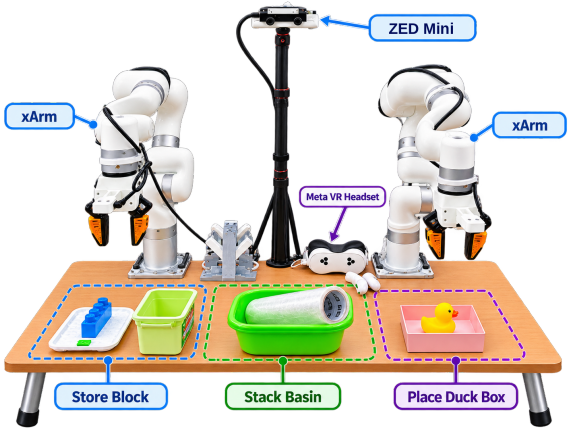

Fig. 3. Real-world bimanual setup with two xArm manipulators, a fixed ZED Mini stereo camera, and a Meta VR headset for teleoperation. We evaluate on three tasks covering complementary forms of bimanual coordination.

V. Experiments

Our experiments investigate three main questions: **Q1:** Does predicting future 3D point tracks improve bimanual policy performance? **Q2:** How do joint action–motion diffusion and spatiotemporal grounding affect policy performance? **Q3:** Does our method generalize better to unseen backgrounds and object appearances?

A. Experimental Setup

a) Simulation benchmark: We evaluate on RoboTwin 2.0 [16], following the two bimanual task categories used in prior work: 8 Sync-bimanual tasks, and 8 Seq-coordinate tasks. Sync-bimanual tasks require simultaneous coordination, and Seq-coordinate tasks require sequential interaction between the two arms. Table I and II report the complete per-task results for the two categories.

b) Real-world evaluation: We evaluate on three real-world bimanual manipulation tasks—Store Block, Stack Basin, and Place Duck Box—using two xArm manipulators and a fixed ZED Mini stereo camera mounted above the workspace. Demonstrations are collected via teleoperation using a Meta VR headset. The policy uses the same observation and action modalities as in simulation. The three tasks cover complementary forms of coordination, including sequential transfer, coordinated object transport, and simultaneous dual-arm interaction.

c) Observations and track supervision: All policies receive a single fixed RGB view together with robot proprioception. In simulation, we construct ground-truth 3D tracks by projecting each current visual patch center onto the corresponding scene geometry and tracking the associated 3D point over the future horizon in the simulator. For real-world training, we first obtain 2D point tracks using CoTracker3 [47]. We estimate metric depth from rectified stereo RGB with FoundationStereo [48] and use the estimated depth together with calibrated camera parameters to lift the tracks into 3D in a common world coordinate frame. At inference time, future observations and 3D tracks are not

TABLE I

COMPARISON ON SYNC-BIMANUAL TASKS (8 TASKS). TASKS REQUIRING SYNCHRONIZED AND COORDINATED OPERATION OF BOTH ARMS. WE REPORT TASK SUCCESS RATE (%) AS MEAN ± STANDARD DEVIATION ACROSS THREE EVALUATION SEEDS. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Method	Avg.↑ (%)	Pick Diverse Bottles	Pick Dual Bottles	Place Bread Skillet	Place Bread Basket	Place Burger Fries	Place Cans Plastic Box	Place Dual Shoes	Put Bottles Dustbin
DP [10]	45.8	38.0 ±3.3	50.0 ±8.6	38.7 ±10.9	39.3 ±0.9	97.3 ±0.9	68.0 ±2.8	12.0 ±4.3	23.3 ±2.5
DP3 [17]	58.1	67.3 ±5.7	78.0 ±3.3	49.3 ±4.1	32.7 ±5.0	85.3 ±5.2	73.3 ±5.7	0.0 ±0.0	78.7 ±5.0
DICP [8]	57.5	23.3 ±9.4	43.3 ±3.4	45.3 ±7.5	32.0 ±7.1	96.7 ±0.9	95.3 ±2.5	56.0 ±7.1	68.0 ±1.6
GAP [9]	45.3	21.3 ±6.2	76.0 ±2.8	21.3 ±5.0	16.0 ±1.6	84.0 ±4.3	40.0 ±6.5	38.7 ±7.4	65.3 ±5.1
ATM [32]	63.2	50.7 ±6.8	78.7 ±1.9	50.7 ±3.8	58.0 ±3.3	96.7 ±1.9	94.0 ±4.3	14.7 ±5.2	62.0 ±2.8
JAMB	**85.2**	**85.3** ±0.9	**99.3** ±0.9	**72.7** ±5.1	**77.3** ±1.9	**98.0** ±1.6	**100.0** ±0.0	**62.0** ±5.9	**86.7** ±0.9

TABLE II

COMPARISON ON SEQ-COORDINATE TASKS (8 TASKS). TASKS REQUIRING MULTI-STEP MANIPULATION AND LONG-HORIZON REASONING. WE REPORT TASK SUCCESS RATE (%) AS MEAN ± STANDARD DEVIATION ACROSS THREE EVALUATION SEEDS. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Method	Avg.↑ (%)	Handover Block	Hang Mug	Place Object Basket	Scan Object	Stack Blocks Two	Stack Blocks Three	Stack Bowls Two	Stack Bowls Three
DP [10]	39.0	64.0 ±6.5	20.7 ±5.2	34.7 ±6.6	20.7 ±5.2	10.7 ±1.9	0.0 ±0.0	87.3 ±1.9	74.0 ±4.3
DP3 [17]	51.9	95.3 ±1.9	22.0 ±1.6	52.0 ±4.3	42.7 ±5.2	38.0 ±2.8	2.0 ±1.6	92.7 ±2.5	70.7 ±1.9
DICP [8]	61.5	84.0 ±3.3	37.3 ±4.7	62.0 ±4.3	27.3 ±6.6	74.0 ±1.6	38.0 ±0.0	**94.7** ±3.4	75.0 ±5.0
GAP [9]	51.1	73.3 ±1.9	24.7 ±9.0	52.7 ±4.7	45.0 ±3.0	41.3 ±0.9	0.7 ±0.9	92.0 ±2.8	**78.7** ±5.7
ATM [32]	40.8	56.7 ±2.5	24.7 ±1.9	40.0 ±7.1	28.7 ±5.2	24.0 ±1.6	8.0 ±1.6	76.7 ±5.0	67.3 ±3.4
JAMB	**81.6**	**96.7** ±0.9	**54.0** ±3.3	**70.7** ±5.7	**72.0** ±2.8	**91.3** ±2.5	**96.0** ±1.6	94.0 ±4.3	78.0 ±4.3

available to the policy. We estimate the current depth map using FastFoundationStereo [49] and use it to recover the 3D positions required by spatiotemporal RoPE.

d) Compared methods: We compare against baselines spanning both action-only policy learning and different forms of future prediction. **Action-only policies** include DP [10] and DP3 [17], which directly predict action chunks from 2D and 3D observations, respectively. **Future-state prediction methods** augment action learning with predictions of future scene states: DICP [8] predicts future visual representations, while GAP [9] predicts dense geometric features at the endpoint alongside actions. **Trajectory-guided policies** are represented by ATM [32], which first predicts future 2D point tracks and then conditions action prediction on the predicted tracks. In contrast, our method jointly denoises bimanual actions and full-horizon 3D point tracks, allowing the two predictions to refine each other throughout generation. All methods are trained and evaluated with matched demonstrations, observation modalities, action representations, and initial-state distributions whenever applicable. For real-world evaluation, we compare against DP3 [17] and GAP [9].

e) Evaluation protocol: In simulation, each policy is trained on 100 expert demonstrations per task and evaluated with three independent seeds, each over 50 rollouts with matched initial states. We report the mean and standard deviation of task success rate across seeds. In the real world, each policy is trained on 50 demonstrations per task and evaluated over 30 rollouts per method–task pair.

f) Implementation details: For all experiments, the policy receives a single head-camera RGB observation with an observation horizon of $H_o = 1$. We use an action horizon of $H_a = 16$ and a track horizon of $H_p = 16$. Images are resized to 238×322 and encoded with DINOv2 [44] using a patch size of 14, yielding a 17×23 patch grid and $N = 391$ track queries, one per visual patch. The DiT consists of 4 Transformer blocks with hidden dimension 1024 and 8 attention heads. We train with AdamW using a learning rate of 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 10^{-6}, cosine decay with 500 warm-up steps, batch size 64, and 200 epochs. We use 100 diffusion timesteps during training and 10 DDIM denoising steps at inference. Real-world experiments use the same model and optimization settings, with regions outside the tabletop workspace cropped from the visual observations.

B. Simulation Policy Results

Tables I and II report per-task success rates across the two RoboTwin task categories. JAMB achieves an average success rate of 85.2% on Sync-bimanual tasks, which require simultaneous coordination, and 81.6% on Seq-coordinate tasks, which require sequential interaction between the two arms. Across all 16 tasks, JAMB achieves an overall success rate of 83.4%, outperforming the strongest baseline by 23.9 percentage points. Fig. 4 shows that 3D point tracks predicted by JAMB closely match ground truth in simulation.

The results show a consistent benefit from explicitly modeling future motion. Compared with action-only policies such as DP [10] and DP3 [17], JAMB provides the policy with an additional representation of how the observed scene is expected to evolve.

We further compare against methods that incorporate future information. DICP [8] predicts future visual observations, while GAP [9] predicts future 3D geometry at the prediction horizon. In contrast, JAMB preserves the

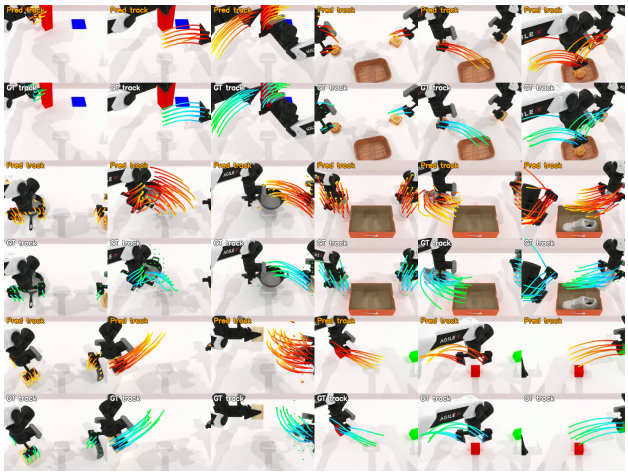

Fig. 4. Qualitative visualization of future 3D point tracks predicted by our method and the corresponding ground-truth tracks on RoboTwin. Red–yellow tracks denote predictions, while green–cyan tracks denote ground truth.

motion between the current and future states through temporally aligned 3D point tracks. Its stronger performance on coordination-intensive and multi-stage tasks suggests that this intermediate motion structure provides useful guidance that is not captured by either future visual observation or geometric representation alone.

C. Real-World Bimanual Evaluation

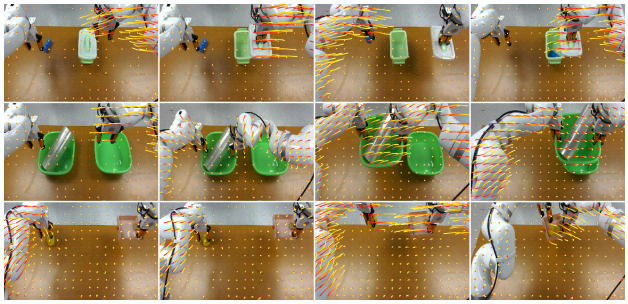

Fig. 5. Qualitative visualization of future 3D point tracks predicted by our method on three real-world tasks: Store Block (top), Stack Basin (middle), and Place Duck Box (bottom). Predicted future 3D point tracks are color-coded by time, progressing from red to yellow.

Table III reports the real-world evaluation. JAMB achieves the best performance on all three tasks, averaging 85.6% and outperforming GAP [9] by 21.2 percentage points. The gap is small on *Stack Basin*, where static future geometry already provides a strong cue, but grows substantially on *Place Duck Box* (80.0% vs. 33.3%), suggesting that temporally structured tracks are more useful when success depends on coordinated motion throughout execution. JAMB also consistently outperforms DP3 [17] without directly encoding dense point clouds, instead incorporating 3D geometry through positional encoding and explicit future-motion prediction. Fig. 5 shows that JAMB predicts coherent task-relevant future motion in real-world manipulation.

TABLE III

REAL-WORLD TASK SUCCESS RATE (%). EACH METHOD IS EVALUATED OVER 30 TRIALS PER TASK. BEST RESULTS ARE SHOWN IN BOLD.

Method	Avg.↑	Store Block	Stack Basin	Place Duck Box
DP3 [17]	35.6	23.3	76.7	6.7
GAP [9]	64.4	73.3	86.7	33.3
JAMB	**85.6**	**83.3**	**93.3**	**80.0**

D. Ablation Study

We conduct controlled ablations on eight representative RoboTwin tasks, including four Sync-bimanual tasks and four Seq-coordinate tasks. All variants use the same visual encoder, action representation, training data, and DiT capacity whenever possible.

a) Future-track modeling: We first study how future-track prediction should interact with action generation. Results can be found in Table IV. *Action-only* removes track prediction entirely, while *Track regression* retains visual–track fusion and shared self-attention but predicts clean tracks without maintaining an evolving track diffusion state. This variant raises average success from 61.4% to 73.9%, supporting the value of future-motion supervision. *Track-conditioned action diffusion* first predicts future tracks and uses this fixed prediction to condition action denoising, reaching 76.3%. The full model achieves 81.7% by jointly denoising actions and tracks. Its advantage over these alternatives is evident on both synchronized and sequential tasks, suggesting that joint refinement benefits both simultaneous and multi-stage coordination. Removing 4D RoPE reduces average policy success from 81.7% to 74.1%, suggesting that explicit spatiotemporal grounding across modalities benefits policy performance.

TABLE IV

ABLATION STUDY ON EIGHT ROBOTWIN TASKS. WE REPORT AVERAGE SUCCESS RATES (%) OVER THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN BOLD.

Variant	Avg.↑	Sync.	Seq.
Action-only	61.4	57.5	65.3
Track regression	73.9	68.4	79.3
Track-conditioned action diffusion	76.3	72.8	79.8
w/o 4D RoPE	74.1	65.7	82.5
Full model	**81.7**	**74.3**	**89.0**

b) Track Prediction and Policy Performance: We further examine the relationship between future-track accuracy and manipulation success. Table V reports average displacement error (ADE) and final displacement error (FDE) for moving patches on the validation sets of the eight ablation tasks. The full model achieves the lowest ADE/FDE of 7.6/10.0 mm and the highest average success rate of 81.7%. Compared with track-conditioned action diffusion, joint denoising reduces ADE/FDE from 9.9/12.5 mm and improves success by 5.4 percentage points. Removing 4D

TABLE V

FUTURE 3D TRACK PREDICTION ON THE VALIDATION SETS OF THE EIGHT ROBOTWIN TASKS. WE REPORT ADE AND FDE AVERAGED ACROSS TASKS. BEST RESULTS ARE SHOWN IN **BOLD**.

Variant	ADE (mm) ↓	FDE (mm) ↓
Track regression	11.1	14.8
Track-conditioned action diffusion	9.9	12.5
w/o 4D RoPE	13.2	17.2
Full model	**7.6**	**10.0**

RoPE increases prediction errors and reduces policy success, supporting the benefits of shared spatiotemporal grounding.

c) Future representations and generalization to harder visual conditions: We evaluate policy generalization by training all methods on the *Easy* setting and directly testing on the corresponding *Hard* setting across the eight ablation tasks, without fine-tuning. Hard scenes introduce unseen object textures and cluttered backgrounds. As shown in Table VI, JAMB achieves the highest average success rate of 17.9%, compared with 4.3% for the strongest baseline GAP. We further examine alternative future representations within our framework in Table VII, by performing ablations to our policy architecture. Our full-horizon 3D track formulation achieves higher success than full-horizon 2D tracks (9.3%) and endpoint 3D geometry (10.3%). These results complement the baseline comparisons and support the effectiveness of our 3D track-based formulation for generalization to harder visual conditions.

TABLE VI

EASY-TO-HARD GENERALIZATION ON EIGHT ROBOTWIN TASKS WITHOUT FINE-TUNING. WE REPORT AVERAGE SUCCESS RATES (%) OVER THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN **BOLD**.

Method	DP3 [17]	DICP [8]	GAP [9]	ATM [32]	**JAMB**
Avg. (%)	1.7	3.9	4.3	0.0	**17.9**

VI. LIMITATIONS AND FUTURE WORK

Our method currently uses a single camera view and a fixed grid of track queries, while real-world track supervision depends on the accuracy of point tracking and depth estimation. Future work will explore efficient multi-view fusion, adaptive query selection, and robustness to noisy supervision. Incorporating force or contact information is another direction for future work.

VII. CONCLUSION

We presented JAMB, a diffusion policy that jointly generates robot actions and future 3D tracks in a shared self-attention sequence. By jointly denoising actions and tracks, the model allows future-motion and action hypotheses to refine one another rather than treating future prediction as an auxiliary objective or fixed condition. Experiments on RoboTwin and real-world bimanual manipulation show consistent improvements over action-only and future-aware baselines, with particularly strong gains on coordination-intensive and multi-stage tasks. Our ablations further highlight the importance of joint action–motion generation and spatiotemporal positional encoding. Together, these results support future 3D tracks as a practical intermediate representation for coordinated robot action generation.

TABLE VII

EASY-TO-HARD GENERALIZATION WITH ALTERNATIVE FUTURE REPRESENTATIONS ON EIGHT ROBOTWIN TASKS. WE REPORT AVERAGE SUCCESS RATES (%) OVER THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN **BOLD**.

Future prediction target	Avg.↑	Sync.	Seq.
2D tracks (full horizon)	9.3	7.5	11.2
3D geometry (endpoint)	10.3	14.2	6.5
3D tracks (full horizon, ours)	**17.9**	**21.3**	**14.5**

REFERENCES

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[2] J. Grannen, Y. Wu, B. Vu, and D. Sadigh, "Stabilize to act: Learning to coordinate for bimanual manipulation," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 229. PMLR, 2023, pp. 563–576.

[3] G. Lu, T. Yu, H. Deng, S. S. Chen, Y. Tang, and Z. Wang, "AnyBimanual: Transferring unimanual policy for general bimanual manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 13 662–13 672.

[4] X. Wang, P. Ding, J. Xu, D. Wang, and Z. Fan, "CUBic: Coordinated unified bimanual perception and control framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 20 856–20 866.

[5] I.-C. A. Liu, S. He, D. Seita, and G. S. Sukhatme, "VoxAct-B: Voxel-based acting and stabilizing policy for bimanual manipulation," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 2025, pp. 4354–4370.

[6] Q. Lv, H. Li, X. Deng, R. Shao, Y. Li, J. Hao, L. Gao, M. Y. Wang, and L. Nie, "Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 17 394–17 404.

[7] H. Chen, J. Xu, L. Sheng, T. Ji, S. Liu, Y. Li, and K. Driggs-Campbell, "Learning coordinated bimanual manipulation policies using state diffusion and inverse dynamics models," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2025, pp. 5644–5651.

[8] H. Xu, J. Ding, J. Xu, R. Wang, J. Chen, J. Mai, Y. Fu, B. Ghanem, F. Xu, and M. Elhoseiny, "Diffusion-based imaginative coordination for bimanual manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 11 469–11 479.

[9] C. Xu, H. Li, S. Cheng, J. Hu, H. Fan, Z. Feng, and S. Liu, "Action-geometry prediction with 3d geometric prior for bimanual manipulation," *arXiv preprint arXiv:2602.23814*, 2026.

[10] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2023.

[11] Y. Guo, Y. Hu, J. Zhang, Y.-J. Wang, X. Chen, C. Lu, and J. Chen, "Prediction with action: Visual policy learning via joint denoising process," *arXiv preprint arXiv:2411.18179*, 2024.

[12] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta, "Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets," in *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[13] J. Han, S. Jeon, J. Jung, R. Zurbrügg, H. An, T. Portela, M. Hutter, M. Pollefeys, S. Kim, and S. Hong, "Geometric action model for robot policy learning," 2026. [Online]. Available: https://arxiv.org/abs/2606.17046

[14] J. Guan, W. Zhao, Y. Pei, Z. Chen, A. Solin, and J. Kannala, "Point tracking improves world action models," *arXiv preprint arXiv:2605.23856*, 2026.

[15] Z. Cao, H. Ji, K. Zhang, K. Ge, L. Fei-Fei, J. Wu, and H. Huang, "Tract: Bridging robot control and visual prediction with visual tracks," 2026. [Online]. Available: https://arxiv.org/abs/2608.24101

[16] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li *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.

[17] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[18] H. Im, E. Jeong, A. Kolobov, J. Fu, and Y. Lee, "Twinvla: Data-efficient bimanual manipulation with twin single-arm vision-language-action models," in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 61 969–61 995.

[19] S. Bajamahal, L. Y. Chen, T. Lin, Z. Ma, J. Malik, and K. Goldberg, "Monoduo: Using one robot arm to learn bimanual policies," *arXiv preprint arXiv:2605.29298*, 2026.

[20] M. Song, X. Deng, J. Wei, D. Jiang, L. Nie, and W. Guan, "Energyaction: Unimanual to bimanual composition with energy-based models," *arXiv preprint arXiv:2603.20236*, 2026.

[21] G. Franzese, L. de Souza Rosa, T. Verburg, L. Peternel, and J. Kober, "Interactive imitation learning of bimanual movement primitives," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 5, pp. 4006–4018, 2023.

[22] N. Gkanatsios, J. Xu, M. Bronars, A. Mousavian, T.-W. Ke, and K. Fragkiadaki, "3d flowmatch actor: Unified 3d policy for single- and dual-arm manipulation," *arXiv preprint arXiv:2508.11002*, 2025.

[23] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[24] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *arXiv e-prints*, pp. arXiv–2302, 2023.

[25] S. Li, Y. Gao, D. Sadigh, and S. Song, "Unified video action model," 2025. [Online]. Available: https://arxiv.org/abs/2503.00200

[26] T. Ma, J. Zheng, Z. Wang, C. Jiang, A. Cui, J. Liang, and S. Yang, "Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control," *arXiv preprint arXiv:2603.10448*, 2026.

[27] M. Team, C. Xiang, F. Bao, H. Liu, H. Tan, H. Bi, J. Li, J. Liu, J. Pang, K. Jing, L. Liu, M. Cai, R. Cui, R. Zhao, R. Wang, S. Huang, Y. Feng, Y. Rong, Z. Wang, and J. Zhu, "Motubrain: An advanced world action model for robot control," 2026. [Online]. Available: https://arxiv.org/abs/2604.27792

[28] T. Yuan, Z. Dong, Y. Liu, and H. Zhao, "Fast-wam: Do world action models need test-time future imagination?" *arXiv preprint arXiv:2603.16666*, 2026.

[29] J. Zhou, Q. Zhang, G. Xu, C. Fan, Y. Zhao, R. Wang, Y. Luo, S. Yang, X. Zhu, Y. Shen *et al.*, "Zero-wam: In-context world-action modeling from human videos for open-ended task generalization," *arXiv preprint arXiv:2608.26103*, 2026.

[30] K. Ranasinghe, H. Zhou, Y. Fang, L. Yang, L. Xue, R. Xu, C. Xiong, S. Savarese, M. S. Ryoo, and J. C. Niebles, "Future optical flow prediction improves robot control video generation," 2026. [Online]. Available: https://arxiv.org/abs/2601.10781

[31] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg, "Amplify: Actionless motion priors for robot learning from videos," *arXiv preprint arXiv:2506.14198*, 2025.

[32] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," *arXiv preprint arXiv:2401.00025*, 2023.

[33] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2405.01527

[34] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," 2024. [Online]. Available: https://arxiv.org/abs/2407.15208

[35] X. Wang, C. Wang, Y. Xu, M. Ye, F.-C. Zhang, J. Tian, X. Zhan, L. Zhu, C. Lu, and L. Yang, "Lamp: Learning vision-language-action policies with 3d scene flow as latent motion prior," 2026. [Online]. Available: https://arxiv.org/abs/2603.25399

[36] A. Hung, B. P. Duisterhof, and J. Ichnowski, "3pointr: 3d point tracks for learning manipulation from unconstrained human videos," 2026. [Online]. Available: https://arxiv.org/abs/2603.08485

[37] B. Lin, Y. Xu, X. Zhan, H. Fang, J. Tian, F.-C. Zhang, Y.-L. Li, C. Lu, and L. Yang, "Chronoflow-policy: Unifying past-current-future interaction flow in visuomotor policy learning," 2026. [Online]. Available: https://arxiv.org/abs/2606.31493

[38] S. Lee, Y. Jung, I. Chun, Y.-C. Lee, Z. Cai, H. Huang, A. Talreja, T. Dao, Y. Liang, J.-B. Huang *et al.*, "Tracegen: World modeling in 3d trace space enables learning from cross-embodiment videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 20 721–20 731.

[39] M. Tong, H. Jiang, Q. Feng, L. Liu, and J. Gu, "Pointaction: 3d points as universal action representations for robot control," *arXiv preprint arXiv:2606.03943*, 2026.

[40] S. Lin, J. Chen, H. Xu, Z. Li, G. Wang, Y. Jing, S. Xu, R. Zhao, B. Sheil, L.-P. Chau *et al.*, "Roboflow4d: A lightweight flow world model toward real-time flow-guided robotic manipulation," *arXiv preprint arXiv:2605.17522*, 2026.

[41] B. Li, X. Yin, M. Lin, Y. Zhang, and D. Xu, "Egowam: World action models beyond pixels with in-the-wild egocentric human data," in *Robot World Models*, 2026.

[42] C. Wang, X. Wang, B. Lin, J. Tian, F. Zhang, C. Lu, and L. Yang, "Track4action: Distilling world-centric 3d tracker into vision-language-action policies," 2026. [Online]. Available: https://arxiv.org/abs/2608.03727

[43] W. Peebles and S. Xie, "Scalable diffusion models with transformers," *arXiv preprint arXiv:2212.09748*, 2022.

[44] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[45] Anonymous Authors, "An Empirical Study on What Matters for Viewpoint-Generalizable Policies in Visual Imitation Learning," 2026, anonymous manuscript.

[46] W. Huang, Y.-W. Chao, A. Mousavian, M.-Y. Liu, D. Fox, K. Mo, and L. Fei-Fei, "Pointworld: Scaling 3d world models for in-the-wild robotic manipulation," *arXiv preprint arXiv:2601.03782*, 2026.

[47] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker3: Simpler and better point tracking by pseudo-labelling real videos," *arXiv preprint arXiv:2410.11831*, 2024.

[48] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2025, pp. 5249–5260.

[49] B. Wen, S. Dewan, and S. Birchfield, "Fast-foundationstereo: Real-time zero-shot stereo matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026, pp. 7513–7524.

APPENDIX

A. Additional Experimental Results

We additionally evaluate JAMB on the Dominant-select task category from the RoboTwin 2.0 benchmark. Table VIII complements our main bimanual evaluation by demonstrating JAMB's effectiveness on tasks that require selecting the appropriate arm.

B. Future Track Prediction Metrics

We evaluate future 3D track prediction using average displacement error (ADE) and final displacement error (FDE). Both metrics are computed over moving patches, defined as queries whose ground-truth endpoint displacement satisfies $\|P_t(i, H_p)\|_2 > 0.01\,\mathrm{m}$. For each validation sample, we reindex these queries as $i = 1, \ldots, N_{\mathrm{mov}}$ and define

$$\mathrm{ADE} = \frac{1}{N_{\mathrm{mov}} H_p} \sum_{i=1}^{N_{\mathrm{mov}}} \sum_{h=1}^{H_p} \left\| \widehat{P}_t(i,h) - P_t(i,h) \right\|_2, \quad (17)$$

$$\mathrm{FDE} = \frac{1}{N_{\mathrm{mov}}} \sum_{i=1}^{N_{\mathrm{mov}}} \left\| \widehat{P}_t(i,H_p) - P_t(i,H_p) \right\|_2, \quad (18)$$

where N_{mov} is the number of selected moving queries, H_p is the prediction horizon, and $\widehat{P}_t(i,h)$ and $P_t(i,h)$ denote the predicted and ground-truth 3D displacements relative to the query's anchor position, respectively. ADE averages the Euclidean displacement error across the full prediction horizon, whereas FDE measures the error at the final step. Both metrics use de-normalized displacements in meters, converted to millimeters for reporting. Lower values indicate better prediction accuracy.

C. Ablation Settings and Per-Task Results

We construct ablation variants from the full model to examine future-track supervision, action–track coupling, visual–track fusion, and spatiotemporal grounding. Unless otherwise specified, we retain the remaining model components and training settings.

a) Ablation settings: The variants are implemented as follows:

- **Action-only.** This variant retains the action-generation backbone but removes the track-prediction branch and all 3D track supervision.
- **Track regression.** We replace track diffusion with deterministic regression from fixed track queries. Each query remains concatenated with its corresponding visual feature and participates in shared self-attention with action tokens. The regression head predicts clean tracks at every action-denoising step, but these predictions are not fed back as an evolving track diffusion state. We retain the prediction from the final denoiser forward pass.
- **Track-conditioned action diffusion.** This variant uses two-stage prediction. In the first stage, noisy 3D track tokens are concatenated with their corresponding patch-level visual features and denoised to predict future 3D tracks. In the second stage, the predicted tracks are concatenated with the corresponding visual features and participate in self-attention with action tokens during action denoising. The track predictions remain fixed throughout the second stage.
- **w/o Visual–Track Concatenation.** We remove the concatenation of each track representation with its corresponding patch-level visual feature. Instead, track and action tokens interact with separate visual tokens through cross-attention. The model must therefore learn visual–track associations through attention rather than receiving explicitly paired features.
- **w/o 4D RoPE.** We remove the rotary encoding of spatial coordinates and trajectory time while retaining learnable positional embeddings for token identity and sequence role. All other components of joint action–track denoising remain unchanged.

b) Evaluation protocol: Table IX reports success rates on the eight RoboTwin tasks used for ablation. For each variant and task, we evaluate the trained policy using three evaluation seeds, with 50 rollouts per seed, and report the mean success rate across seeds. Table X reports average displacement error (ADE) and final displacement error (FDE) for moving patches on the corresponding validation sets. Both metrics are computed using de-normalized 3D displacements, reported in millimeters, and averaged across tasks.

c) Track prediction and policy performance: The full model achieves the lowest ADE/FDE of $7.6/10.0$ mm and the highest average policy success rate of 81.7%. Compared with track-conditioned action diffusion, the full model reduces ADE/FDE from $9.9/12.5$ mm and improves success by 5.4 percentage points. Removing 4D RoPE increases ADE/FDE to $13.2/17.2$ mm and reduces success to 74.1%, supporting the benefits of shared spatiotemporal grounding for both prediction and control. Removing visual–track concatenation also increases ADE/FDE to $11.9/15.3$ mm and reduces success to 70.0%. This comparison supports the effectiveness of the full model's patch-aligned fusion design relative to the separate-token cross-attention variant.

However, lower track error does not consistently translate into higher policy success. Track regression predicts more accurate tracks than the variant without 4D RoPE ($11.1/14.8$ mm vs. $13.2/17.2$ mm), yet their success rates are similar (73.9% vs. 74.1%). Likewise, the variant without visual–track concatenation has lower track errors than the variant without 4D RoPE, but a lower success rate (70.0% vs. 74.1%). These results suggest that track accuracy alone does not explain policy performance; the way motion representations interact with visual features and action generation also matters. The full model combines accurate future-track prediction with effective action generation, although these comparisons do not isolate a causal effect of track accuracy on manipulation success.

D. Per-Task Easy-to-Hard Generalization Results

Table XI reports per-task results when transferring policies trained on Easy scenes to Hard scenes with unseen textures and clutter, without fine-tuning. Each policy is evaluated

TABLE VIII

COMPARISON ON DOMINANT-SELECT TASKS (16 TASKS). TASKS REQUIRING APPROPRIATE ARM SELECTION. WE REPORT TASK SUCCESS RATE (%) AS MEAN ± STANDARD DEVIATION ACROSS THREE EVALUATION SEEDS. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Method	Avg.↑ (%)	Beat Block Hammer	Place Shoe	Move Can Pot	Open Laptop	Open Microwave	Place A2B Left	Place A2B Right
DP [10]	41.7	**92.7** ±5.0	46.0 ±7.1	70.0 ±15.6	64.0 ±4.3	0.0 ±0.0	17.3 ±4.7	17.3 ±2.5
DP3 [17]	64.4	88.0 ±1.6	62.0 ±2.8	**92.7** ±1.9	39.3 ±1.9	22.0 ±1.6	**53.3** ±0.9	**50.7** ±5.7
DICP [8]	48.3	86.0 ±3.3	72.0 ±7.1	74.7 ±3.4	80.0 ±4.3	16.0 ±4.3	8.0 ±1.6	10.0 ±1.6
GAP [9]	51.7	39.3 ±1.9	60.7 ±2.5	66.0 ±8.8	81.3 ±2.5	36.0 ±8.0	14.0 ±1.6	11.3 ±3.4
ATM [32]	47.5	84.7 ±2.5	40.7 ±5.7	91.3 ±3.8	68.7 ±3.8	1.3 ±1.9	24.7 ±2.5	14.7 ±0.9
JAMB	**73.5**	86.7 ±6.2	**96.0** ±1.6	90.0 ±2.8	**92.7** ±0.9	**83.3** ±2.5	23.3 ±4.1	24.0 ±2.8

Turn Switch	Place Fan	Place Container Plate	Press Stapler	Place Phone Stand	Rotate QR Code	Place Empty Cup	Place Object Stand	Shake Bottle
17.3 ±3.4	4.7 ±4.1	51.3 ±3.4	20.0 ±1.6	38.0 ±4.9	24.0 ±1.6	61.3 ±4.1	48.7 ±1.9	94.7 ±1.9
36.7 ±6.6	50.7 ±5.0	92.0 ±1.6	67.3 ±2.5	54.0 ±4.9	**69.3** ±2.5	75.3 ±8.2	**76.7** ±3.4	**100.0** ±0.0
25.3 ±3.8	40.0 ±4.3	86.7 ±4.7	39.3 ±0.9	11.3 ±5.7	50.7 ±1.9	52.0 ±3.3	36.0 ±4.3	84.0 ±1.6
34.7 ±1.9	45.3 ±4.7	78.0 ±9.1	68.0 ±3.3	30.0 ±8.6	62.0 ±4.3	61.3 ±6.6	39.3 ±4.1	99.3 ±0.9
23.3 ±2.5	4.0 ±0.0	69.3 ±8.1	31.3 ±6.2	58.0 ±4.9	48.0 ±4.3	64.7 ±8.1	40.7 ±6.6	94.0 ±4.3
38.7 ±1.9	**62.7** ±1.9	**98.0** ±1.6	**80.7** ±7.7	**90.0** ±2.8	62.0 ±2.8	**94.7** ±0.9	54.7 ±8.4	98.7 ±0.9

TABLE IX

PER-TASK ABLATION RESULTS ON EIGHT ROBOTWIN TASKS. WE REPORT PER-TASK AVERAGE SUCCESS RATE (%) ACROSS THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Variant	Avg.↑ (%)	Handover Block	Stack Blocks Two	Stack Blocks Three	Scan Object	Place Bread Skillet	Place Dual Shoes	Pick Diverse Bottles	Place Bread Basket
Action-only	61.4	67.3	71.3	59.3	63.3	48.0	43.3	74.0	64.7
Track regression	73.9	94.7	83.3	80.0	59.3	70.0	60.7	77.0	66.0
Track-conditioned action diffusion	76.3	95.0	85.3	84.0	54.7	**75.0**	59.0	**87.0**	70.0
w/o Visual–Track Concatenation	70.0	81.3	78.0	70.0	60.0	68.7	**67.3**	74.7	60.0
w/o 4D RoPE	74.1	86.7	90.7	86.7	66.0	67.3	62.0	61.3	72.0
Full model	**81.7**	**96.7**	**91.3**	**96.0**	**72.0**	72.7	62.0	85.3	**77.3**

TABLE X

FUTURE 3D TRACK PREDICTION ON THE VALIDATION SETS OF THE EIGHT ROBOTWIN TASKS. ADE AND FDE ARE COMPUTED OVER MOVING PATCHES AND AVERAGED ACROSS TASKS. BEST RESULTS ARE SHOWN IN **BOLD**.

Variant	ADE (mm) ↓	FDE (mm) ↓
Track regression	11.1	14.8
Track-conditioned action diffusion	9.9	12.5
w/o Visual–Track Concatenation	11.9	15.3
w/o 4D RoPE	13.2	17.2
Full model	**7.6**	**10.0**

using three evaluation seeds with 50 rollouts per seed, and we report the mean and standard deviation across these seeds. JAMB achieves the highest success rate on seven of the eight tasks, with an average of 17.9% compared with 4.3% for the strongest baseline, GAP. However, success rates remain low overall, and JAMB does not succeed on Place Bread Skillet, highlighting the remaining challenges of generalization under these visual changes.

E. Additional Future Prediction Targets

We further examine whether predicting richer future observations provides additional benefits beyond 3D tracks. Starting from the full JAMB model, we additionally predict either future RGB observations or future 3D geometry features, while retaining the same DiT backbone configuration and evaluation protocol.

For future-image prediction, we encode RGB observations using a frozen variational autoencoder (VAE) from Stable Diffusion XL (SDXL) and supervise the predicted VAE latents. For endpoint geometry prediction, we extract target features using a frozen pretrained Pi3 encoder and apply supervision directly in feature space. Although these features can be decoded into a point map, the loss is computed on the features rather than the decoded geometry. In both variants, the additional latent or feature tokens are incorporated into the DiT sequence, grounded by 4D RoPE, and jointly denoised with actions and tracks.

Table XII compares these variants on the same eight RoboTwin tasks used for ablation. Adding future-image or geometry-feature prediction reduces average success from 81.7% to 79.4% and 74.2%, respectively. Under the evaluated configuration, these additional prediction targets provide no further performance benefit over joint action–track prediction alone.

F. Alternative Future Representations

We compare different future prediction targets within our framework on the same eight RoboTwin tasks. The full-horizon 2D track variant replaces 3D displacement targets with 2D pixel displacements over the same prediction horizon, with the prediction head adjusted accordingly. We

TABLE XI

PER-TASK EASY-TO-HARD GENERALIZATION ON EIGHT ROBOTWIN TASKS. ALL METHODS ARE TRAINED ON *Easy* AND EVALUATED ON *Hard* WITHOUT FINE-TUNING. WE REPORT TASK SUCCESS RATE (%) AS MEAN ± STANDARD DEVIATION ACROSS THREE EVALUATION SEEDS. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Method	Avg.↑ (%)	Handover Block	Stack Blocks Two	Stack Blocks Three	Scan Object	Place Bread Skillet	Place Dual Shoes	Pick Diverse Bottles	Place Bread Basket
DP3 [17]	1.7	4.7 ±2.5	1.3 ±0.9	0.0 ±0.0	2.0 ±1.6	0.0 ±0.0	0.0 ±0.0	2.7 ±1.9	2.7 ±2.5
DICP [8]	3.9	0.7 ±0.9	0.7 ±0.9	0.0 ±0.0	4.7 ±2.5	**8.7** ±3.4	3.3 ±2.5	4.7 ±3.4	8.0 ±6.5
GAP [9]	4.3	6.0 ±1.6	0.0 ±0.0	0.0 ±0.0	3.3 ±2.5	2.0 ±2.8	8.0 ±1.6	9.3 ±1.9	6.0 ±1.6
ATM [32]	0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0
JAMB	**17.9**	**12.0** ±1.6	**10.7** ±3.8	**22.0** ±3.3	**13.3** ±2.5	0.0 ±0.0	**12.7** ±3.4	**47.3** ±6.6	**25.3** ±5.2

TABLE XII

EFFECT OF ADDITIONAL FUTURE PREDICTION TARGETS ON EIGHT ROBOTWIN TASKS. WE REPORT AVERAGE SUCCESS RATES (%) OVER THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN **BOLD**.

Future prediction target	Avg.↑	Sync.	Seq.
3D tracks	**81.7**	**74.3**	**89.0**
3D tracks + future image	79.4	71.8	87.0
3D tracks + future 3D geometry	74.2	64.5	83.9

TABLE XIII

COMPARISON OF ALTERNATIVE FUTURE REPRESENTATIONS ON EIGHT ROBOTWIN TASKS. WE REPORT AVERAGE SUCCESS RATES (%) OVER THREE EVALUATION SEEDS, WITH 50 ROLLOUTS PER SEED PER TASK. BEST RESULTS ARE SHOWN IN **BOLD** AND SECOND-BEST RESULTS ARE UNDERLINED.

Future prediction target	Avg.↑	Sync.	Seq.
2D tracks (full horizon)	76.1	67.5	84.7
2D tracks + 4D RoPE (full horizon)	79.5	70.2	88.8
3D geometry (endpoint)	**82.8**	**76.7**	88.9
3D point displacements (endpoint)	81.2	73.0	**89.4**
3D tracks (full horizon, ours)	81.7	74.3	89.0

evaluate this variant both with and without 4D RoPE. When enabled, 4D RoPE retains the same depth-derived world-space anchors as the full model, even though the prediction targets are in image space.

The endpoint 3D displacement variant predicts only the displacement at the final horizon step, whereas the full model predicts displacements at all future steps. Both variants retain patch-aligned visual–track fusion. For the endpoint geometry variant, we replace the track prediction target with Pi3 features extracted by a frozen pretrained encoder. Patch-level noisy geometry features are concatenated with their corresponding visual features, and supervision is applied directly in feature space rather than to the decoded point map. All variants jointly denoise their respective future representations with actions.

As shown in Table XIII, adding 4D RoPE to full-horizon 2D tracks improves average success from 76.1% to 79.5%, supporting the value of spatiotemporal grounding across modalities even when motion is predicted in image space. Endpoint 3D displacements and full-horizon 3D tracks achieve comparable success rates of 81.2% and 81.7%, respectively, while endpoint 3D geometry achieves 82.8%. Performance across these three 3D prediction targets falls within 1.6 percentage points, demonstrating that our joint prediction architecture accommodates different future representations with comparable manipulation success. Our track-based formulation achieves comparable performance using depth-derived spatial anchors, without the additional Pi3 feature extraction used by the geometry variant.

We further compare these representations under an Easy-to-Hard visual shift, where all variants are trained on Easy scenes and evaluated on Hard scenes without fine-tuning. Despite their similar in-distribution performance, full-horizon 3D tracks achieve higher Easy-to-Hard success (17.9%) than endpoint displacements (12.2%) and endpoint geometry (10.3%), as shown in Table XIV. These results suggest that full-horizon 3D tracks support better generalization under the evaluated visual shift. One possible explanation is that endpoint geometry features encode surrounding scene structure and appearance, potentially making them more sensitive to changes in background and clutter. Endpoint displacements focus on point motion but supervise only the net change at the final step. Full-horizon tracks additionally supervise intermediate motion, which may encourage the model to learn task-relevant motion patterns that transfer across visual conditions. These explanations remain hypotheses rather than mechanisms isolated by the current experiments.

G. Additional Implementation Details

1) Architecture and Optimization: For real-world experiments, the policy uses a single head-camera RGB view with an observation horizon of $H_o = 1$. We crop regions outside the tabletop workspace, resulting in input images of 168×322 pixels (height $\times$ width). DINOv2 [44] encodes these images with a patch size of 14, yielding a 12×23 patch grid and $N = 276$ track queries, one per visual patch. Both the action and track prediction horizons are set to $H_a = H_p = 16$.

JAMB uses a DiT with 4 Transformer blocks, a hidden dimension of 1024, and 8 attention heads. We optimize the model with AdamW using a learning rate of 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 10^{-6}, and a batch size of 64. The learning-rate schedule consists of 500 warm-up steps followed by cosine decay.

TABLE XIV

PER-TASK EASY-TO-HARD GENERALIZATION WITH ALTERNATIVE FUTURE REPRESENTATIONS ON EIGHT ROBOTWIN TASKS. WE REPORT TASK SUCCESS RATE (%) AS MEAN ± STANDARD DEVIATION ACROSS THREE EVALUATION SEEDS. BEST RESULTS ARE SHOWN IN **BOLD**.

Future prediction target	Avg.↑ (%)	Handover Block	Stack Blocks Two	Stack Blocks Three	Scan Object	Place Bread Skillet	Place Dual Shoes	Pick Diverse Bottles	Place Bread Basket
2D tracks (full horizon)	9.3	2.7 ±2.5	16.7 ±3.4	**22.0** ±4.3	3.3 ±0.9	0.0 ±0.0	5.3 ±2.5	16.0 ±1.6	8.7 ±2.5
2D tracks + 4D RoPE (full horizon)	8.8	0.0 ±0.0	12.7 ±3.4	3.3 ±3.4	8.7 ±5.2	0.0 ±0.0	6.7 ±1.9	31.3 ±8.2	7.3 ±3.4
3D geometry (endpoint)	10.3	2.7 ±0.9	11.3 ±5.0	1.3 ±1.9	10.7 ±2.5	0.0 ±0.0	10.7 ±4.7	28.7 ±5.2	17.3 ±5.2
3D point displacements (endpoint)	12.2	7.3 ±1.9	**26.7** ±8.2	5.3 ±1.9	**13.3** ±3.8	0.0 ±0.0	2.7 ±0.9	31.3 ±2.5	11.3 ±2.5
3D tracks (full horizon, ours)	**17.9**	**12.0** ±1.6	10.7 ±3.8	**22.0** ±3.3	**13.3** ±2.5	0.0 ±0.0	**12.7** ±3.4	**47.3** ±6.6	**25.3** ±5.2

2) Diffusion and Loss Hyperparameters: We use clean-sample prediction (`prediction_type=sample`) for both action and track diffusion, supervising the outputs $\widehat{Y}_A$ and $\widehat{Y}_P$ with normalized clean targets:

$$Y_A = \widetilde{A}_t^0, \qquad Y_P = \widetilde{P}_t^0. \tag{19}$$

Both branches use the same cosine noise schedule (`squaredcos_cap_v2`) with $K = 100$ training diffusion timesteps and 10 DDIM denoising steps at inference.

We set $\lambda_{\text{track}} = 10^{-4}$, $w_{\min} = 0$, and $\tau = 0.01\,\text{m}$, with sigmoid steepness $\kappa = 5/\tau = 500\,\text{m}^{-1}$. The track loss sums squared errors over the prediction horizon and three spatial coordinates, weighted by $w_{b,i}$ for query i in batch element b, and divides by $3\max(\sum_{b,i} w_{b,i}, 10^{-6})$. It is therefore not averaged over the prediction horizon.

3) Track Validity and Invalid-Query Supervision: For simulator-generated track supervision, a patch query is valid if its center has a positive, finite depth value sampled by bilinear interpolation, and its segmentation ID belongs to a tracked rigid body. Tracked bodies include objects and robot arm links. Queries with invalid depth or segmentation IDs outside this set are assigned zero displacement targets.

For each valid query, we transform its anchor-frame 3D position into the owning body's local frame and compute future positions using that body's ground-truth poses. Validity is determined at the anchor frame and propagated across the prediction horizon; future visibility and occlusion are not re-evaluated. For prediction steps beyond the episode boundary, we reuse the final recorded pose.

We normalize 3D track displacements as $\widetilde{P}_t^0 = P_t^0/\sigma_{\text{rms}}$, where σ_{rms} is the root mean square of all valid displacement components across samples and prediction steps in each training dataset. This scalar scale is shared across the three spatial dimensions, with no mean subtraction, so zero displacement targets remain zero after normalization. Invalid queries are excluded from scale estimation but included in the movement-weighted track loss without validity masking. Under the specified loss hyperparameters, they receive a weight of $\text{sigmoid}(-5) \approx 0.0067$ and therefore contribute weak zero-displacement supervision.

4) Training Duration and Checkpoint Selection: DP and DP3 are trained for 600, 3000 epochs, respectively. JAMB, DICP, and GAP are each trained for 200 epochs. For ATM, the Track Transformer is trained for 100 epochs, while the ATM Diffusion Policy is trained for 200 epochs. For every method, we use the checkpoint from the final training epoch for evaluation.

5) 4D-RoPE Implementation: We apply four-dimensional rotary positional encoding (4D RoPE) to incorporate relative spatial and temporal relationships into multimodal attention. Each grounded token is associated with a coordinate $\mathbf{c} = (x, y, z, u)$, where (x, y, z) denotes its position in a common world frame, measured in meters, and u denotes its physical timestep index rather than the diffusion timestep.

a) Axis allocation and coordinate scaling: Each attention head has dimension 128, which we divide into four groups of 32 dimensions for the x, y, z, and temporal axes. RoPE is applied independently to each group of the query and key vectors. For an axis coordinate c, the rotation angle for the j-th pair of feature dimensions is

$$\phi_j(c) = \frac{c}{s}\, b^{-2j/d}, \qquad j = 0, \ldots, d/2 - 1, \tag{20}$$

where $d = 32$ is the dimension allocated to each axis. We use a frequency base of $b = 100$ for all four axes, a spatial scale of $s_{\text{xyz}} = 0.01\,\text{m}$, and a temporal scale of $s_u = 16$. These rotations encode relative coordinate offsets in query–key dot products.

b) Token coordinates: Visual patches and their associated track queries are anchored at the corresponding current 3D patch centers and assigned temporal index $u = 0$. The fused visual–track tokens inherit these coordinates. Each arm-state token is anchored at that arm's current end-effector position, also with $u = 0$. Action tokens are assigned temporal indices $u = h$ for $h = 1, \ldots, H_a$ and spatial coordinates computed from the current noisy action sequence, as described below. Special tokens, including the CLS token when present and the diffusion-timestep token, are excluded from rotary encoding.

Each track token represents the complete future trajectory of one query point rather than an individual future timestep. Its temporal coordinate therefore identifies the current reference frame from which the trajectory is predicted. The ordered future displacements are encoded within the track embedding, while temporally indexed action tokens provide timestep-specific interactions through attention.

c) Noise-dependent action grounding: To obtain spatial anchors for action tokens, we combine the current end-effector position with a position derived from the noisy action at the current diffusion step. Let $\mathbf{g}_a$ denote the current world-frame end-effector position of arm a, and let $\widetilde{\mathbf{a}}_{h,a}^k$ denote its

normalized noisy action at future step h and diffusion step k. We define

$$\mathbf{r}_{h,a}^{k} = \Phi_a\big(\widetilde{\mathbf{a}}_{h,a}^{k}\big)\,, \tag{21}$$

where Φ_a applies the inverse action normalization and extracts the absolute end-effector position of arm a. Both arms' positions are expressed in the same world frame. The spatial anchor is

$$\mathbf{x}_{h,a}^{k} = (1-\rho_k)\mathbf{g}_a + \rho_k \mathbf{r}_{h,a}^{k}, \qquad \rho_k = 1 - \frac{k}{K}, \tag{22}$$

where $k \in \{0, \ldots, K-1\}$ denotes the scheduler timestep on the training diffusion grid, rather than the DDIM iteration index. At inference, we use the scheduler-selected timesteps to compute ρ_k.

At high noise levels, the anchor remains close to the current end-effector position, reducing the influence of noisy action coordinates on rotary encoding. As denoising progresses, the anchor increasingly follows the position derived from the evolving action sequence. We recompute these anchors at every denoising step and use the same construction during training and inference. The anchors are computed from the current noisy actions, without using clean ground-truth future actions.