ORDERED ACTION TRANSPORT FOR GENERATIVE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative robot policies, including vision-language-action models (VLAs) and world-action models (WAMs), generate an action chunk by jointly refining a sequence of action tokens, one per future action. These tokens inherit rotary position encoding (RoPE), which relates two tokens by how many steps apart they are. However, existing positional encodings represent action-token relations primarily by their positions in the sequence, overlooking the motion induced by the actions are taken and in what order. We propose **Action-Transport RoPE (AT-RoPE)**, which advances the encoding at each step by a rotation determined by the action rather than by a fixed one. Each current noisy action parameterizes an orthogonal operator, and these operators are composed in execution order, so that the prefix transformations of two tokens differ by the composition of the actions between them. AT-RoPE reduces to RoPE-style rotation when the step is action-independent and to an order-insensitive summed prefix when the operators commute. For backbones with rotary attention, a complementary pathway further conditions query-key matching on the actions being compared. Across four VLA and WAM backbones, AT-RoPE improves success on LIBERO and RoboTwin 2.0 for every backbone, with consistent gains on long-horizon tasks, and improves zero-shot robustness on LIBERO-Pro and LIBERO-Plus. Ablations confirm that ordered composition outperforms prefix-free and summed-prefix alternatives, and rollouts with AT-RoPE are smoother.

1 INTRODUCTION

Recent generative Vision-Language-Action models (VLAs) and World-Action Models (WAMs) jointly refine continuous action chunks through diffusion or flow matching (Black et al., 2024; 2025; Bi et al., 2026; Yuan et al., 2026). At each refinement step, these models jointly update the actions within a chunk, whose positions correspond to successive execution time steps. The quality of a generated chunk thus depends on both the individual actions and their relationships across time. For robotic manipulation, these relationships matter for both task completion and motion smoothness during execution. This motivates studying how action content and intended execution order can jointly structure the representations used during generation.

Existing decoders can learn cross-action dependencies through content-dependent attention and iterative denoising. Standard rotary position encoding (RoPE) supplies index-dependent rotations, while action-dependent relationships are learned through token representations and their interactions (Su et al., 2021). CoPE and CARoPE further show how positional structure can be conditioned on context or token content (Golovneva et al., 2024; Veisi et al., 2025). For action chunks, we consider extending this conditioning to the ordered sequence of preceding actions. This leads us to ask: ***Can composing action-conditioned feature transformations in temporal order help models better exploit dependencies within an action chunk?***

We propose **Action-Transport Rotary Position Encoding (AT-RoPE)**, a rotary-inspired module for *ordered action-conditioned feature transport* (Figure 2). At each refinement step, the current noisy actions parameterize learned orthogonal operators. Through *ordered prefix composition*, these operators are accumulated in temporal order to construct a prefix-dependent transformation for each action position. A *gated token update* incorporates the resulting transformation into the original

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.

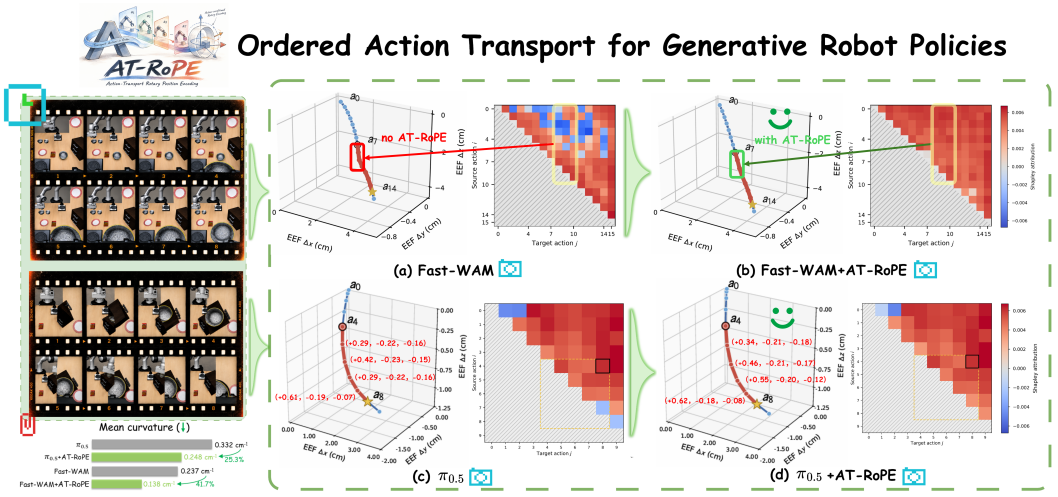

Figure 1: **Selected trajectories and attention-edge attributions.** Fast-WAM and $\pi_{0.5}$ are compared with and without AT-RoPE. Shapley maps report marginal contributions of selected attention connections to reducing target-position prediction loss; positive contributions cover more connections in the displayed comparisons with AT-RoPE.

token through a residual connection, explicitly conditioning its representation on preceding actions. The operator family permits noncommuting transformations, allowing ordered products to distinguish different orderings of the same prefix actions. The prefixes are recomputed at every refinement step, while the decoder continues to refine the entire chunk jointly.

We integrate this ordered token-transport module into both VLA and WAM backbones. In our WAM implementations, we additionally propose *local action-conditioned Q/K modulation* after native RoPE. This supplementary module directly conditions attention comparisons on the individual actions involved, complementing the prefix information supplied through token transport. Learned gates regulate both the token updates and the Q/K modulation, and the integrations retain the backbone's positional encoding, attention mask, and training objective.

We evaluate AT-RoPE on four backbones spanning VLA and WAM policies using LIBERO and RoboTwin 2.0, with additional robustness evaluation on LIBERO-Pro. AT-RoPE improves benchmark-average success over each corresponding baseline on both main benchmarks, with gains on LIBERO-Long across all evaluated backbones. Beyond task success, the analyzed rollouts show lower mean end-effector trajectory curvature with AT-RoPE (Figure 1). A further Fast-WAM case study shows a lower proportion of high-frequency energy in the velocity spectrum (Figure 3). Structural ablations test the contributions of ordered composition and the two interfaces, while prediction-loss-based attention attribution characterizes selected predictive dependencies. Our contributions are:

- **Ordered action transport.** We propose AT-RoPE, which composes action-conditioned feature operators in temporal order, enabling token updates to depend explicitly on preceding actions and their order.
- **Complementary decoding interfaces.** We integrate ordered token transport into both VLA and WAM backbones, with supplementary local post-RoPE Q/K modulation in our WAM implementations to directly condition attention matching on action content.
- **Manipulation performance and execution behavior.** We demonstrate improved benchmark-average success on robotic manipulation tasks. Selected rollout analyses further show lower mean end-effector trajectory curvature and a reduced proportion of high-frequency velocity energy. Real-world executions with AT-RoPE further illustrate sequential object packing, cylinder insertion, and dynamic object manipulation.

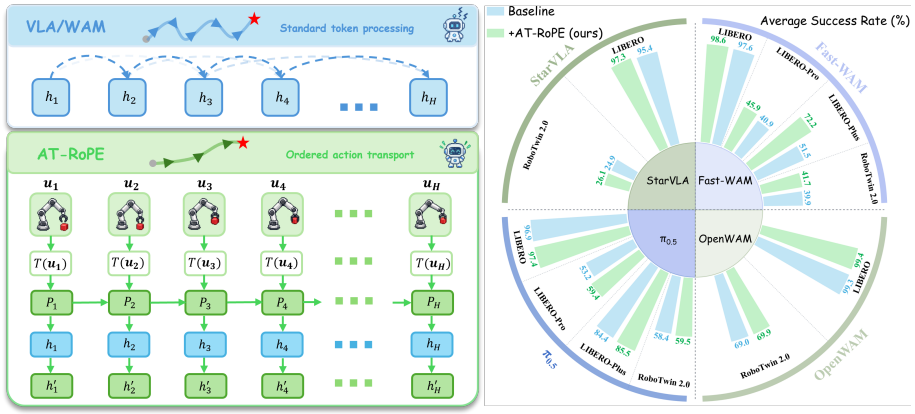

Figure 2: **Ordered action transport and performance overview. Left, top:** content-dependent interactions among native action tokens. **Left, bottom:** current noisy actions parameterize feature operators composed in temporal order within the chunk; the resulting prefixes condition updates to the original token representations before decoding. **Right:** success-rate comparisons for the illustrated backbones and evaluation settings.

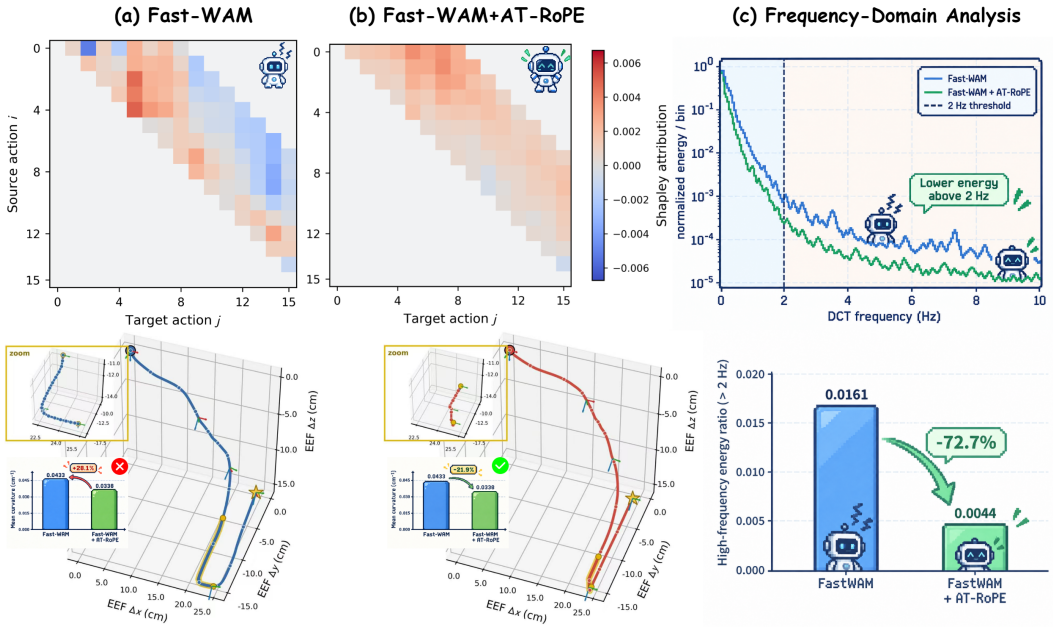

Figure 3: **Candidate-action dependencies and execution dynamics.** A Fast-WAM case study supplements the comparisons in Figure 1. Attention-edge Shapley maps quantify the predictive contributions of selected preceding-action connections. End-effector trajectories and local enlargements characterize path geometry, while signed-velocity DCT spectra describe temporal variation.

2 Analysis and Motivation

Predictive utility of candidate-action context. We examine whether current noisy candidate actions provide useful context for joint action prediction and whether AT-RoPE improves the model's use of this context. We consider generative VLA and WAM policies that jointly refine an action chunk $\mathbf{u}_{1:H}$ at refinement time τ, where H is the chunk length and i indexes intended execution order. These action variables are perturbed demonstration actions during training and the current noisy iterate during inference.

Players and coalitions. For a target action token at position j in the current noisy action chunk, we select up to eight preceding action tokens, indexed by $\mathcal{N}_j \subseteq \{1, \ldots, j-1\}$. Each selected source token is treated as a *player*, whose contribution to predicting the target action is evaluated through its designated K/V attention connections to token j. A *coalition* $S \subseteq \mathcal{N}_j$ specifies which of these source tokens can contribute through the tested connections. Concretely, target token j can attend to the keys and values of source tokens in S through these connections, whereas the corresponding connections from tokens in $\mathcal{N}_j \setminus S$ are masked.

Value function and attribution. Let $\mathcal{L}_j(S)$ denote the native target-position prediction loss under coalition S. We define the coalition value as $v_j(S) = -\mathcal{L}_j(S)$, so that lower prediction loss corresponds to higher value. The Shapley contribution of player i is

$$\phi_{i \to j} = \mathbb{E}_\pi[v_j(\mathcal{S}_i^\pi \cup \{i\}) - v_j(\mathcal{S}_i^\pi)], \tag{1}$$

where π is a uniformly random permutation of $\mathcal{N}_j$, and $\mathcal{S}_i^\pi$ contains the players preceding i in that permutation. The permutation is used only to compute marginal contributions; candidate execution positions remain unchanged. Thus, $\phi_{i \to j} > 0$ indicates that retaining the selected information route $i \to j$ reduces the target-position prediction loss on average, corresponding to a positive marginal predictive contribution.

Figure 1 shows selected comparisons, with a further Fast-WAM case study in Figure 3. The attribution measures the tested attention routes under the specified masking intervention. In AT-RoPE, action information can also enter target representations through ordered token transport, which remains active during this intervention. The maps therefore characterize selected attention-edge contributions, rather than all cross-action information flow. Trajectory geometry provides a separate descriptive view of execution; lower curvature alone does not establish better task performance.

3 METHODOLOGY

3.1 ORDERED ACTION TRANSPORT

We first construct composable per-action operators, then accumulate them in chunk order to condition later token updates on preceding actions.

Action-conditioned operators. A coefficient network produces $\mathbf{c}_i = f(\mathbf{u}_i) \in \mathbb{R}^K$ using layer normalization and an affine projection with a fixed output scale, where K is the number of generators. Selected feature channels are partitioned into blocks of width p. For block b, the skew-symmetric generator is

$$\mathbf{A}_i^{(b)} = \sum_{m=1}^{K} c_{im} \frac{\mathbf{B}_m^{(b)} - (\mathbf{B}_m^{(b)})^\top}{\sqrt{p}}, \tag{2}$$

where $\mathbf{B}_m^{(b)} \in \mathbb{R}^{p \times p}$ is learned, c_{im} is the m-th coefficient, and m indexes the generators. Assembling these blocks with zeros on untouched channels gives $\mathbf{A}_i \in \mathbb{R}^{D \times D}$. Its per-action operator is $\mathbf{U}_i = \exp(\mathbf{A}_i)$, where $\exp$ denotes the matrix exponential, evaluated blockwise without materializing the full matrix. The generator parameters are shared across sequence positions.

The construction builds on matrix-exponential sequence representations and learned rotational encodings, using the current action variables to parameterize decoder-feature transformations.

Since $\mathbf{A}_i^\top = -\mathbf{A}_i$, its exponential satisfies

$$\mathbf{U}_i^\top \mathbf{U}_i = \exp(-\mathbf{A}_i) \exp(\mathbf{A}_i) = \mathbf{I}_D.$$

Consequently, $\|\mathbf{U}_i \mathbf{x}\|_2 = \|\mathbf{x}\|_2$ for any $\mathbf{x} \in \mathbb{R}^D$. Products of these operators therefore preserve feature norms before residual mixing, allowing action-conditioned transformations to accumulate without amplifying or attenuating the feature norm. With multiple generators and block width $p > 2$, the operator family can represent noncommuting rotations. For a learned pair with a nonzero commutator, reversing their order changes the composition:

$$\mathbf{U}_i \mathbf{U}_j - \mathbf{U}_j \mathbf{U}_i = [\mathbf{U}_i, \mathbf{U}_j] \neq \mathbf{0},$$

where $[\mathbf{U}_i, \mathbf{U}_j]$ denotes the commutator. By contrast, summed-generator aggregation satisfies $\exp(\mathbf{A}_i + \mathbf{A}_j) = \exp(\mathbf{A}_j + \mathbf{A}_i)$ and cannot distinguish this reordering. Ordered composition can thus retain distinctions in candidate execution order that are lost by summation. This is an expressive capacity of learned feature transformations, not a prescribed model of physical pose transitions.

Ordered prefix composition. Starting from the identity $\mathbf{P}_0 = \mathbf{I}_D$, we evolve

$$\mathbf{P}_i = \mathbf{P}_{i-1}\mathbf{U}_i, \tag{3}$$

where $\mathbf{I}_D$ is the $D \times D$ identity matrix and $\mathbf{P}_i$ contains the first i action-conditioned operators. The recurrence creates a direct route from preceding actions to later token transformations; useful dependencies are learned through the native prediction objective. The adjacent prefixes $\mathbf{P}_{i-1}$ and $\mathbf{P}_i$ exclude and fully include the current increment, respectively. We associate the token with a centered transform,

$$\mathbf{M}_i = \mathbf{P}_{i-1} \exp\left(\tfrac{1}{2}\mathbf{A}_i\right), \tag{4}$$

which also satisfies $\mathbf{M}_i = \mathbf{P}_i \exp(-\frac{1}{2}\mathbf{A}_i)$. Traversing half of the increment from either adjacent prefix gives the same operator, placing the token symmetrically on its own learned feature-transform segment.

Gated token update. To incorporate this prefix context without replacing the native token, we apply a residual update:

$$\widehat{\mathbf{h}}_i = \mathbf{h}_i + \beta_i\big(\mathbf{M}_i\mathbf{h}_i - \mathbf{h}_i\big), \tag{5}$$

where $\beta_i = s_h\sigma\big(b_h(\mathbf{h}_i) + b_a(\mathbf{u}_i) + b_{\tau(\mathbf{e}_\tau)}\big)$ controls the contribution of the transported features; where σ is the sigmoid, b_h is a layer-normalized scalar projection, and b_a and b_τ are scalar affine maps. The scale $s_h \in [0, s_{\max}]$ controls the residual amplitude, with fixed bound $s_{\max} > 0$. Integrations that omit a conditioning signal set its term to zero. The updated tokens enter the native decoder with its original attention mask. Unlike the underlying operators, the residual mixture is not required to preserve norms.

Ordered transport provides prefix-conditioned content, but its effect on attention remains mediated by the query and key projections. For backbones with rotary attention, we introduce a separate action-conditioned interface after these projections. This directly controls local matching without another prefix scan.

Let $\mathbf{q}_i^{(\ell)}, \mathbf{k}_i^{(\ell)} \in \mathbb{R}^d$ be the query and key after native projection and normalization at layer ℓ, where d is the attention-head width. Let $\mathbf{R}_i \in \mathbb{R}^{d\times d}$ be the native RoPE operator. We instantiate the generator construction in Eq. (2) at width d with parameters separate from the token branch. Exponentiating the resulting generator gives the local operator $\mathbf{U}_i^{\mathrm{qk}} \in \mathbb{R}^{d\times d}$.

Post-RoPE modulation. We form the local residual operator $\mathbf{D}_i^{(\ell)} = \mathbf{I}_d + \alpha_i^{(\ell)}(\mathbf{U}_i^{\mathrm{qk}} - \mathbf{I}_d)$, where $\mathbf{I}_d$ is the $d \times d$ identity matrix and $\alpha_i^{(\ell)}$ is a scalar gate. For either the query or the key, denoted by $\mathbf{z}_i^{(\ell)}$, we apply

$$\widehat{\mathbf{z}}_i^{(\ell)} = \mathbf{D}_i^{(\ell)}\mathbf{R}_i\mathbf{z}_i^{(\ell)}. \tag{6}$$

The gate $\alpha_i^{(\ell)} = \lambda_\ell\sigma\big(g_a(\mathbf{u}_i) + g_\tau(\mathbf{e}_\tau)\big)$ conditions the correction on the local action and refinement time, with scalar affine maps g_a and g_τ. The layer scale is bounded by the parameterization $\lambda_\ell = \lambda_{\max}\sigma(\rho_\ell)$, where ρ_ℓ is learned and $\lambda_{\max} > 0$ is a fixed upper bound. The query and key share both the local operator and the gate.

Action-conditioned matching. Substituting Eq. (6) into the attention score gives the unmasked action-to-action logit

$$s_{ij}^{(\ell)} = \frac{1}{\sqrt{d}}\,(\mathbf{q}_i^{(\ell)})^\top\mathbf{R}_i^\top(\mathbf{D}_i^{(\ell)})^\top\mathbf{D}_j^{(\ell)}\mathbf{R}_j\mathbf{k}_j^{(\ell)}, \tag{7}$$

where i and j index the query and key action positions. The inserted factor $(\mathbf{D}_i^{(\ell)})^\top\mathbf{D}_j^{(\ell)}$ introduces direct dependence on the actions being compared, while prefix dependence remains in the transported content. The Q/K module does not directly modify values, masks, or the softmax operation.

Algorithm 1 AT-RoPE: VLA

Require: Observation o, initial $u^{(0)}$;
N Euler updates; mode flags b_c, b_f
Ensure: Normalized action chunk $u^{(N)}$

$(Z, m, \mathcal{K}) \leftarrow \textsc{Prefill}(o)$
$\kappa \leftarrow \textsc{ContextCoeff}(Z, m; b_c)$
for $n = 0, \dots, N-1$ **do**
  $\tau_n \leftarrow 1 - n/N$
  $h \leftarrow \textsc{ActionProj}(u^{(n)})$
  $(A_{1:H}, \beta_{1:H}) \leftarrow \textsc{TokenParams}(h, u^{(n)}, \kappa)$
  $P_0 \leftarrow I_D$
  for $i = 1, \dots, H$ **do**
    $M_i \leftarrow P_{i-1} \exp(A_i/2)$
    $\hat{h}_i \leftarrow h_i + \beta_i(M_i h_i - h_i)$
    $P_i \leftarrow P_{i-1} \exp(A_i)$
  end for
  $F \leftarrow \textsc{FutureQueries}(P_{1:H}; b_f)$
  $e_{\tau_n} \leftarrow \textsc{TimeMLP}(\tau_n)$
  $z \leftarrow \textsc{CachedSuffix}([F; \hat{h}_{1:H}], e_{\tau_n}, \mathcal{K}, m)$
  $u^{(n+1)} \leftarrow u^{(n)} - \textsc{ActionHead}(z_{\text{act}})/N$
end for
return $u^{(N)}$

Algorithm 2 AT-RoPE: WAM

Require: Observation o, initial $u^{(0)}$;
context $\mathcal{C}$, schedule $\mathcal{T} = \{(\tau_n, \Delta_n)\}_{n=0}^{N-1}$
Ensure: Normalized action chunk $u^{(N)}$

$(h^v, e^v) \leftarrow \textsc{VideoEmbed}(\text{VAE}(o), 0, \mathcal{C})$
for $\ell = 1, \dots, L$ **do**
  $(h^v, \mathcal{K}_\ell) \leftarrow \textsc{VideoCache}(h^v, e^v, \ell)$
end for
for $n = 0, \dots, N-1$ **do**
  $(h, e_{\tau_n}) \leftarrow \textsc{Embed}(u^{(n)}, \tau_n, \mathcal{C})$
  $h \leftarrow \textsc{Prefix}(h, u^{(n)}, e_{\tau_n})$
  for $\ell = 1, \dots, L$ **do**
    $(Q, K, V) \leftarrow \textsc{ActionQKV}(h, \ell)$
    $(\bar{Q}, \bar{K}) \leftarrow \textsc{RoPE}(Q, K)$
    $D^{(\ell)}_{1:H} \leftarrow \textsc{LocalParams}(u^{(n)}, e_{\tau_n}, \ell)$
    $[\hat{q}_i, \hat{k}_i] \leftarrow D^{(\ell)}_i[\bar{q}_i, \bar{k}_i] \quad \forall i$
    $O \leftarrow \textsc{CachedAttn}(\hat{Q}, \hat{K}, V, \mathcal{K}_\ell)$
    $h \leftarrow \textsc{ActionPost}(h, O, \ell)$
  end for
  $u^{(n+1)} \leftarrow \textsc{Step}(u^{(n)}, \textsc{Head}(h), \tau_n, \Delta_n)$
end for
return $u^{(N)}$

Because the residual operators need not be orthogonal, the correction can affect magnitude as well as direction, rather than only phase.

The factor $(\mathbf{D}_i^{(\ell)})^\top \mathbf{D}_j^{(\ell)}$ introduces direct dependence on the actions being compared, while prefix dependence remains in the transported content. The Q/K module does not directly modify values, masks, or the softmax operation. Because the residual operators need not be orthogonal, the correction can affect magnitude as well as direction, rather than only phase. The corresponding VLA and WAM inference procedures are summarized in Algorithm 1-2.

4 Experiments

4.1 Experimental Setups

Baselines. We evaluate AT-RoPE on four backbones spanning two action-generation paradigms: StarVLA-PIv3 and $\pi_{0.5}$ for VLA, and Fast-WAM and OpenWAM for WAM. For VLA backbones, AT-RoPE is integrated into the action expert. For WAM backbones, it is introduced into both the video- and action-generation branches while retaining the original world-modeling branch and joint video-action objective.

Evaluation benchmarks. We evaluate AT-RoPE on robotic manipulation benchmarks spanning two categories: (1) **standard manipulation benchmarks**, including LIBERO Liu et al. (2023) and RoboTwin 2.0 Chen et al. (2025), which evaluate in-distribution task execution and action generation performance; and (2) **robustness-oriented benchmarks**, including LIBERO-Pro Zhou et al. (2025b), which evaluates performance under targeted perturbations and distribution shifts. Additional evaluations on LIBERO-Plus Fei et al. (2026) are provided in Appendix C.

4.2 Evaluation Results on LIBERO and LIBERO-Pro

Performance across VLA and WAM backbones. Tables 1 and 2 examine standard-task performance and perturbation robustness, respectively. AT-RoPE improves average success on standard LIBERO for all four evaluated backbones, covering both VLA and WAM policies. Since each

paired comparison retains the native training objective and configuration, the improvements support explicit action-conditioned structure as a complement to both paradigms, rather than an alternative source of world-modeling supervision.

Table 1: **Performance comparison on LIBERO.** Success rates (%). The best and second-best results within each paradigm are shown in bold and underline, respectively.

VLA						WAM					
Method	**Spat.**	**Obj.**	**Goal**	**Long**	**Avg.**	**Method**	**Spat.**	**Obj.**	**Goal**	**Long**	**Avg.**
StarVLA (2026)	97.8	98.6	96.2	93.8	96.6	Motus (2026)	96.8	<u>99.8</u>	96.6	97.6	97.7
GR00T-N1.6 (2025)	97.7	98.5	97.5	94.4	97.0	ImageWAM (2026)	97.2	99.2	98.8	98.4	98.4
OpenVLA-OFT (2025)	97.6	98.4	97.9	94.5	97.1	LingBot-VA (2026)	98.5	99.6	97.2	<u>98.5</u>	98.5
X-VLA (2026)	98.2	98.6	97.8	**97.6**	98.1	DiT4DiT (2026)	–	–	–	–	98.6
ABot-M0 (2026)	<u>98.8</u>	**99.8**	<u>99.0</u>	96.6	<u>98.6</u>	Being-H0.7 (2026b)	–	–	–	–	99.2
Being-H0.5 (2026a)	**99.2**	<u>99.6</u>	**99.4**	<u>97.4</u>	**98.9**	ABot-M0.5 (2026)	**100.0**	<u>99.8</u>	<u>99.4</u>	98.4	**99.4**
$\pi_{0.5}$ (2025)	<u>98.8</u>	98.2	98.0	92.4	96.9	OpenWAM-α (2026)	<u>99.6</u>	99.6	**99.8**	98.2	<u>99.3</u>
StarVLA-π (2026)	98.2	99.2	95.6	88.4	95.4	Fast-WAM (2026)	98.2	**100.0**	97.0	95.2	97.6
With AT-RoPE (ours)											
$\pi_{0.5}$	<u>98.8</u>	99.4	97.2	94.2	97.4	Fast-WAM	98.2	**100.0**	99.0	97.2	98.6
StarVLA-π	98.2	98.6	98.4	93.8	97.3	OpenWAM-α	<u>99.6</u>	99.4	**99.8**	**98.6**	**99.4**

Long-horizon execution and perturbation robustness. Table 1 reveals a selective benefit: AT-RoPE improves long-horizon execution across all backbones while leaving Spatial performance unchanged. This pattern aligns with its intervention on action-sequence structure. Rather than transforming each token solely from its own action condition, ordered transport incorporates the preceding candidate actions into its transformation, providing explicit sequence context that may help coordinate successive movements.

LIBERO-Pro further distinguishes task proficiency from robustness: the ranking of Fast-WAM and $\pi_{0.5}$ reverses under perturbations (Table 2), showing that stronger standard-task performance does not necessarily imply better generalization. Both backbones benefit from AT-RoPE without adaptation, suggesting that action-conditioned structure remains useful when evaluation conditions change. Its effect nevertheless depends on the underlying policy and perturbation type, as illustrated by the object-related regression for $\pi_{0.5}$.

Robustness on LIBERO-Pro. LIBERO-Pro evaluates robustness under targeted perturbations covering position, object, language, task, and environment variations.

AT-RoPE improves the reported average performance for both backbones. For $\pi_{0.5}$, the average success rate increases from 53.2% to 59.4%, with particularly large gains under position/swap and task perturbations, although object robustness decreases from 96.0% to 87.0%. For Fast-WAM, the average increases from 40.9% to 45.9%, with improvements under all five reported perturbation types. These results indicate that the benefits of AT-RoPE extend beyond the standard LIBERO setting, while the magnitude of the gain depends on the backbone and perturbation type.

Table 2: **LIBERO-Pro robustness.**

	$\pi_{0.5}$		**Fast-WAM**	
Perturb.	**Base**	**Ours**	**Base**	**Ours**
Pos./Swap	20.8	**28.7**	0.4	**4.0**
Object	**96.0**	87.0	74.5	**80.0**
Lang./Sem.	95.8	**96.0**	95.1	**97.7**
Task	0.8	**26.4**	4.5	**4.8**
Env.	52.8	**58.8**	30.1	**43.1**
Avg.	53.2	**59.4**	40.9	**45.9**

4.3 Evaluation Results on RoboTwin 2.0

Clean2Random evaluations. Table 3 examines whether the improvements extend to bimanual manipulation and randomized scenes using policies trained only on clean demonstrations. AT-RoPE improves both clean- and randomized-scene success for every paired backbone. The benefit is therefore not obtained by trading performance in one condition for the other. It also appears at markedly

different baseline performance levels: OpenWAM-α remains the strongest listed WAM after integration, reaching 69.9% average success, while the lower-performing StarVLA and Fast-WAM policies also improve. This cross-backbone pattern supports the applicability of the same action-conditioning design to the evaluated bimanual policies.

Table 3: **RoboTwin 2.0 clean-to-random evaluation.** Success rates (%). The lower block reports AT-RoPE applied to the named backbones. Bold and underline denote the best and second-best displayed values within each paradigm.

VLA				WAM			
Method	**Clean**	**Rand.**	**Avg.**	**Method**	**Clean**	**Rand.**	**Avg.**
StarVLA	46.5	3.2	24.9	AHA-WAM	64.3	3.2	33.8
GR00T-N1.7	43.6	20.7	32.2	Fast-WAM	77.8	1.9	39.9
X-VLA	68.0	20.9	44.5	X-WAM	70.0	25.8	47.9
Spatial Forcing	**77.2**	26.7	52.0	4D-WAM	81.5	41.8	61.7
$\pi_{0.5}$	70.7	<u>46.0</u>	<u>58.4</u>	OpenWAM-α	<u>89.4</u>	<u>48.7</u>	<u>69.0</u>
With AT-RoPE (ours)							
StarVLA	47.8	4.4	26.1	Fast-WAM	80.1	3.3	41.7
$\pi_{0.5}$	<u>72.1</u>	**46.8**	**59.5**	OpenWAM-α	**90.5**	**49.4**	**69.9**

The clean and randomized columns nevertheless capture distinct aspects of performance. Fast-WAM with AT-RoPE reaches 80.1% success in clean scenes but only 3.3% in randomized scenes, so its average improvement should not be interpreted as resolving clean-to-random generalization. The paired gains show that explicit action structure can improve an existing policy under both conditions, while the remaining gap identifies robustness to scene variation as an important unresolved challenge in this evaluation.

4.4 Ablation Study

All prefix variants retain local Q/K modulation. The token transform is $M_i = P_{i-1}\exp(A_i/2)$, with A_i defined in the method. Local sets $P_{i-1} = I$; Sum-prefix uses $P_{i-1} = \exp(\sum_{k=1}^{i-1} A_k)$; and AT-RoPE uses $P_{i-1} = \exp(A_1)\cdots\exp(A_{i-1})$.

Table 4: **Ablations of prefix composition and the two pathways.** Success rates (%).

(a) Prefix composition				(b) Two pathways				
Prefix	Clean	Rand.	Avg.	Token	Q/K	Clean	Rand.	Avg.
Local	78.7	2.3	40.5	×	×	77.8	1.9	39.9
Sum-prefix	79.1	2.7	40.9	✓	×	79.3	2.7	41.0
				×	✓	78.5	2.4	40.5
AT-RoPE (full)	**80.1**	**3.3**	**41.7**	✓	✓	**80.1**	**3.3**	**41.7**

Effect of prefix composition. Table 4 (a) separates the benefit of prefix information from that of its composition rule. Sum-prefix improves over Local, but does not recover the full result. AT-RoPE improves both clean and randomized success, supporting ordered composition over the tested prefix summary. The backbone remains sequence-aware in all variants; the comparison concerns the added composition rule, not the presence or absence of positional information.

Contribution of the two pathways. Table 4 (b) separates ordered token transport from local post-RoPE Q/K modulation. Token-only provides the larger standalone gain, but neither individual pathway recovers the full result. Their combination improves both clean and randomized success over either single-pathway variant. This comparison is consistent with their distinct intervention points: token transport organizes the representation entering the decoder, whereas local Q/K modulation directly adjusts its subsequent attention matching.

Stability across retraining runs. To examine whether the gains in Table 4 (a) are robust to training variation rather than driven by a particular optimization trajectory, Figure 4 reports mean success rates and ± 1 sample standard deviation over five independent retraining runs per variant. AT-RoPE retains the highest mean performance under both clean and randomized evaluation, reaching 41.66% average success compared with 40.87% for Sum-prefix and 40.53% for Local.

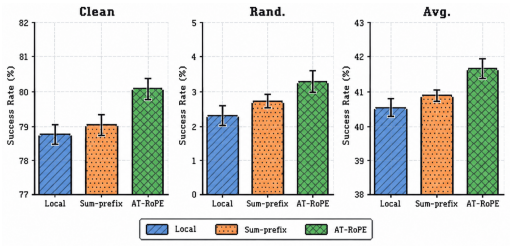

Figure 4: **Stability across retraining runs.** Mean success rates with ± 1 sample standard deviation over five retraining runs.

4.5 Real-World Experiments

We evaluate AT-RoPE on three real-world tasks using a Franka Research 3 (FR3) arm with an XHand dexterous hand and an AgileX PiPER arm. We compare $\pi_{0.5}$ with and without AT-RoPE on *object packing* and *cylinder insertion*, and Fast-WAM with and without AT-RoPE on *dynamic object manipulation*. We report task success over 20 trials per method and task. Object packing tests a shared policy's ability to sequentially place four objects with different shapes and appearances into one basket. Cylinder insertion requires aligning three differently sized cylinders with their matching holes before insertion. Dynamic object manipulation tests coordinated approach and grasp timing for a banana moving on a conveyor belt.

Success across tasks. Figure 5 shows that AT-RoPE increases object-packing success from 65% to 80% and dynamic-object-manipulation success from 20% to 75%. The packing result supports more reliable execution of a shared grasp-and-place policy across multiple target objects, while the dynamic-task result indicates improved success when grasping must be coordinated with target motion.

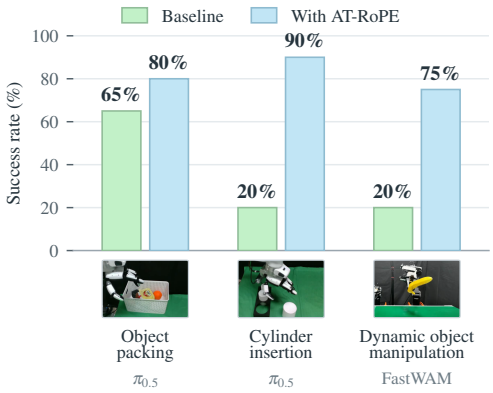

Figure 5: **Real-world task success rates.**

Precision-sensitive cylinder insertion. The largest improvement occurs on cylinder insertion, where success rises from 20% to 90%, a gain of 70 percentage points. Compared with placing objects into a basket, insertion imposes tighter positional and angular alignment requirements: the cylinder must be centered over the matching opening and remain aligned during descent. The recorded AT-RoPE rollouts qualitatively exhibit smoother approach-to-insertion motion and more precise final alignment, consistent with the higher completion rate on this task. These observations suggest that coordinating successive actions is particularly useful when small deviations can interrupt insertion. Selected execution sequences appear in Figure 6 in Appendix D, together with hardware configurations and detailed task descriptions.

5 Conclusion

In this work, we investigate how candidate-action context shapes generative robot policies and reveal that preceding candidate actions provide predictive context for later action generation. Motivated by this finding, we introduce AT-RoPE, which explicitly models such dependencies by composing action-conditioned transformations in execution order. Across VLA and WAM backbones, AT-RoPE consistently improves benchmark-average success, zero-shot robustness, and real-world manipulation performance.

AI USE STATEMENT

Generative AI tools, including Grammarly and ChatGPT, were used solely for grammar checking and language polishing. All technical content, experimental design, data analysis, and conclusions were generated and verified exclusively by the human authors. The use of AI tools does not affect the originality or authorship of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the formulation and implementation of AT-RoPE are described in Section 3, with the inference procedures for the VLA and WAM instantiations summarized in Algorithms 1 and 2, respectively. The experimental backbones, benchmarks, and evaluation settings are described in Section 4.1.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. URL https://arxiv.org/abs/2503.06669.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. URL https://arxiv.org/abs/2409.16283.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. 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, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025.

Jun Cen, Siteng Huang, Yuqian Yuan, Hangjie Yuan, Chaohui Yu, Yuming Jiang, Jiayan Guo, Kehan Li, Hao Luo, Fan Wang, Xin Li, Deli Zhao, and Hao Chen. RynnVLA-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025. URL https://arxiv.org/abs/2511.17502.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative

video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.

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.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023. URL https://arxiv.org/abs/2302.00111.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026.

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025. URL https://arxiv.org/abs/2507.12898.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. URL https://arxiv.org/abs/2503.20020.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*, 2024. URL https://arxiv.org/abs/2405.18719.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024. URL https://arxiv.org/abs/2412.14803.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng,

Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. URL https://arxiv.org/abs/2505.12705.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2601.16163.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.016.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025. URL https://arxiv.org/abs/2503.00200.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. URL https://arxiv.org/abs/2508.00795.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. URL https://arxiv.org/abs/2410.07864.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-H0.5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026a.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026b.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Jim Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loïc Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025. URL https://arxiv.org/abs/2512.15692.

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

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Ali Veisi, Delaram Fartoot, and Hamidreza Amirzadeh. Context-aware rotary position embedding. *arXiv preprint arXiv:2507.23083*, 2025. URL https://arxiv.org/abs/2507.23083.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. OpenWAM: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025. URL https://arxiv.org/abs/2502.05855.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025. URL https://arxiv.org/abs/2510.27607.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. URL https://arxiv.org/abs/2312.13139.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. ABot-M0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Jim Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025. URL https://arxiv.org/abs/2507.04447.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. ImageWAM: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1702–1713, 2025. doi: 10.1109/CVPR52734.2025.00166. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhao_CoT-VLA_Visual_Chain-of-Thought_Reasoning_for_Vision-Language-Action_Models_CVPR_2025_paper.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3952–3971. PMLR, 2025. URL https://proceedings.mlr.press/v305/zheng25a.html.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2Goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025a. URL https://arxiv.org/abs/2512.23541.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024. URL https://arxiv.org/abs/2404.12377.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025b.

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.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Related Work

Vision-Language-Action Models. Current Vision-Language-Action (VLA) models have demonstrated strong control and generalization capabilities in robotic manipulation Zitkovich et al. (2023); Kim et al. (2024); Black et al. (2024; 2025); NVIDIA et al. (2025); AgiBot-World-Contributors et al. (2025); Shukor et al. (2025); Gemini Robotics Team et al. (2025); Liu et al. (2024); Wen et al. (2025). These methods typically adapt pretrained vision-language models and action heads using action trajectories from expert demonstrations, enabling the policy to generate robot actions conditioned on visual observations and language instructions. Beyond autoregressive action prediction, an increasing number of recent methods adopt diffusion- or flow-matching-based action heads, where noisy action tokens or action trajectories are progressively denoised or updated through velocity prediction to produce continuous action chunks Black et al. (2024; 2025); Liu et al. (2024); Wen et al. (2025). These models are already able to learn temporal dependencies among actions through action self-attention, cross-attention, and the denoising process. However, such dependencies are primarily learned implicitly through content representations, while actions themselves are less commonly used as structured conditions that explicitly determine how action representations are transformed and matched.

World-Action Models. More recently, a growing body of work has explored World-Action Models (WAMs), which leverage dynamic knowledge learned through video generation and world modeling for robotic manipulation Du et al. (2023); Wu et al. (2023); Zhou et al. (2024); Bharadhwaj et al. (2024); Feng et al. (2025); Cheang et al. (2024); Jang et al. (2025); Zhao et al. (2025); Zheng et al. (2025); Zhang et al. (2025). These methods typically model environment changes induced by actions through future visual prediction, joint video-action modeling, or latent world representation learning, and further use such dynamic representations to support action prediction Pai et al. (2025); Liang et al. (2025); Zhu et al. (2025); Li et al. (2026); Ye et al. (2026); Bi et al. (2026); Won et al. (2025); Cen et al. (2025); Zhou et al. (2025a); Kim et al. (2026); Liao et al. (2025); Hu et al. (2024); Li et al. (2025); Yuan et al. (2026). Some WAMs also adopt diffusion- or flow-matching-based objectives to generate continuous action chunks, allowing action dependencies to emerge progressively during generation and denoising Zhu et al. (2025); Pai et al. (2025); Li et al. (2026); Bi et al. (2026); Won et al. (2025); Kim et al. (2026); Li et al. (2025). Therefore, although VLAs and WAMs obtain their representations from different sources, both paradigms still rely primarily on content representations to learn action dependencies during action generation. AT-RoPE is complementary to these approaches and can be integrated into both VLA and WAM action-generation backbones, introducing an explicit action-conditioned relation pathway without changing the original training objective.

Rotary Position Embedding. Rotary Position Embedding (RoPE) encodes positional information by applying position-dependent rotations to query and key representations, thereby inducing relative positional relationships directly within self-attention Su et al. (2021). With Transformer-based and diffusion-based architectures increasingly adopted for continuous robot control, positional encoding also plays an important role in organizing action tokens within temporally ordered action chunks Black et al. (2024); Liu et al. (2024); Wen et al. (2025). However, positional proximity alone does not fully characterize the relationship between robotic actions. The same relative displacement between two action tokens may correspond to substantially different state transitions depending on their action content and the preceding execution history. Standard RoPE determines its rotations from token positions and therefore provides a content-independent notion of relative order, without explicitly representing such action-dependent transitions. In contrast, AT-RoPE augments the native positional structure with action-conditioned transformations constructed from the current action content and candidate execution order. By composing these transformations sequentially, preceding candidate actions explicitly participate in organizing subsequent token representations, while a complementary local modulation pathway further conditions post-RoPE query-key matching on the actions being compared. Thus, AT-RoPE retains the positional inductive bias of RoPE while introducing an explicit mechanism for modeling action-dependent relational structure.

B Implementation Details

Implementation details. All models are initialized from their officially released pretrained checkpoints and trained on eight NVIDIA A100 GPUs. Apart from the added AT-RoPE modules, we retain each backbone's data-processing pipeline, optimizer, learning-rate schedule, training steps, and inference configuration. Each paired baseline and AT-RoPE comparison therefore follows the same native training and evaluation configuration.

B.1 Operator Computation and Interfaces

This section specifies the routines used in Algorithms 1 and 2, distinguishing ordered token transport from local post-RoPE modulation and clarifying their conditioning inputs and inference scope.

Operator computation. The coefficient network applies layer normalization, an affine projection, and a fixed output scale to the current noisy action. The resulting coefficients combine learned skew-symmetric generators according to Eq. (2), with generator parameters shared across action positions. Matrix exponentials and prefix products are evaluated blockwise over the selected feature channels, without materializing full-width matrices; untouched channels use identity transformations. The token and Q/K branches instantiate this construction at token width D and attention-head width d, respectively, using separate generator parameters. Orthogonality applies to the exponentiated operators and their products, but is not imposed on the gated residual mappings.

Token interface. TokenParams returns the token generators $A_{1:H}$ and residual gates $\beta_{1:H}$. With additional context conditioning disabled, the coefficients follow the action-only map $c_i = f(u_i)$ and determine A_i through Eq. (2). Prefix implements Eqs. (3)-(5): starting from $P_0 = I_D$, it constructs the centered transform $M_i = P_{i-1}\exp(A_i/2)$, applies the gated token update, and accumulates $P_i = P_{i-1}\exp(A_i)$. The scan follows candidate execution order $i = 1, \dots, H$. At each refinement step, the generators and prefixes are recomputed from the current noisy action chunk, with the prefix initialized again to the identity. This execution-order scan does not replace the native decoder attention mask.

Conditioning terms not supplied by an integration are set to zero. In Algorithm 1, refinement time is supplied to the native decoder through TimeMLP, rather than to TokenParams; the explicit $b_\tau(e_\tau)$ term in the token gate of Eq. (5) is therefore omitted in this interface. Algorithm 2 additionally supplies the refinement-time embedding to Prefix and LocalParams.

Attention interface. ActionQKV denotes the native action-query, key, and value projections, including normalization preceding RoPE. Native RoPE is applied separately, producing $\bar{q}_i^{(\ell)} = R_i q_i^{(\ell)}$ and $\bar{k}_i^{(\ell)} = R_i k_i^{(\ell)}$. LocalParams returns the residual operator $D_i^{(\ell)} = I_d + \alpha_i^{(\ell)}(U_i^{qk} - I_d)$, which is applied to both the query and key at the same action position according to Eq. (6). The query and key share the local operator and its gate. This interface performs no additional prefix scan: the correction is conditioned on the local noisy action and refinement time, while ordered-prefix information enters through the transported token content.

CachedAttn combines the corrected action queries and keys, action values, and the layer-specific visual K/V cache $\mathcal{K}_\ell$ under the native attention mask. Equation (7) gives the resulting action-to-action logit. ActionPost denotes the remaining native operations of the action layer. The local Q/K insertion does not directly transform values or modify the mask or softmax operation.

Optional VLA routines. The binary flags b_c and b_f in Algorithm 1 distinguish observation-derived coefficient conditioning from prefix-derived query-token construction. We use the conventions ContextCoeff$(Z, m; 0) = \varnothing$ and FutureQueries$(P_{1:H}; 0) = [\,]$. An empty context disables additional coefficient conditioning, whereas an empty query sequence leaves only the transported action tokens in the decoder suffix. Thus, $b_c = b_f = 0$ selects the core action-only token-transport path. When additional query tokens are present, the suffix is $[F; \hat{h}_{1:H}]$; z_{act} denotes only the decoder outputs at action-token positions and excludes the additional query-token outputs.

Table 5: **Zero-shot robustness on LIBERO-Plus.** Success rates (%). Bold and underline denote the best and second-best values among the listed methods within each paradigm, respectively.

Method	Camera	Robot	Lang.	Light	BG	Noise	Layout	Avg.
				VLA				
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
OpenVLA-OFT	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
StarVLA	52.5	49.8	**88.5**	95.7	95.7	73.0	76.9	74.1
ABot-M0	60.4	67.9	86.4	96.2	91.6	86.4	82.6	80.5
$\pi_{0.5}$	**78.4**	73.6	80.8	96.2	94.1	**89.0**	84.5	84.4
ACoT-VLA	<u>72.6</u>	**82.6**	<u>87.5</u>	**97.7**	<u>96.5</u>	87.8	**88.1**	**86.6**
$\pi_{0.5}$ (AT-RoPE)	69.9	<u>73.8</u>	84.4	<u>97.0</u>	**98.6**	<u>88.1</u>	<u>86.4</u>	<u>85.5</u>
				WAM				
Being-H0.7	**82.0**	59.0	82.8	<u>97.8</u>	**90.0**	<u>93.5</u>	**88.5**	82.1
Cosmos-Policy	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
ImageWAM	<u>80.8</u>	50.3	**91.4**	**98.1**	85.5	**93.8**	80.5	<u>83.1</u>
ABot-M0.5	70.5	**87.4**	<u>88.6</u>	94.0	<u>89.7</u>	75.5	<u>85.2</u>	**83.4**
OpenWAM-α	33.8	<u>76.1</u>	88.0	97.0	87.1	39.8	77.5	69.2
FastWAM	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
FastWAM (AT-RoPE)	54.6	61.7	86.1	91.3	55.7	78.0	78.8	72.2

Visual cache and inference scope. Algorithm 1 prefills the observation prefix before action refinement and reuses its cache during suffix decoding. Algorithm 2 similarly constructs layer-specific visual K/V caches from the observation and context before entering the action-refinement loop. The listed visual-cache inputs do not include the current noisy action chunk, so the caches are reused while action-dependent token transport and local Q/K corrections are recomputed during refinement. Algorithm 2 describes this cached action-inference interface, rather than the complete joint video-action training graph. Both algorithms retain their specified native action-update procedures and return a normalized action chunk.

C Additional Experiments

We further evaluate AT-RoPE on LIBERO-Plus to assess zero-shot robustness under diverse distribution shifts. The benchmark includes perturbations in camera viewpoint, robot configuration, language, lighting, background, observation noise, and scene layout. Table 5 reports results for both VLA and WAM methods.

AT-RoPE improves the reported average performance of both evaluated backbones over their corresponding baselines. The improvement is modest for $\pi_{0.5}$ and substantially larger for FastWAM, with gains observed across several perturbation types. These results provide additional evidence that AT-RoPE remains effective under distribution shifts beyond the standard evaluation setting.

D Real-World Experimental Details

D.1 Robot and Camera Configurations

We use two robot configurations. All cameras in both configurations are Intel RealSense D435 units.

Franka Research 3 with XHand. The first configuration combines a Franka Research 3 (FR3) arm with an XHand dexterous hand. Its observation setup comprises two third-person cameras and one wrist-mounted camera.

AgileX PiPER. The second configuration uses an AgileX PiPER arm. Its observation setup comprises one third-person camera and one wrist-mounted camera.

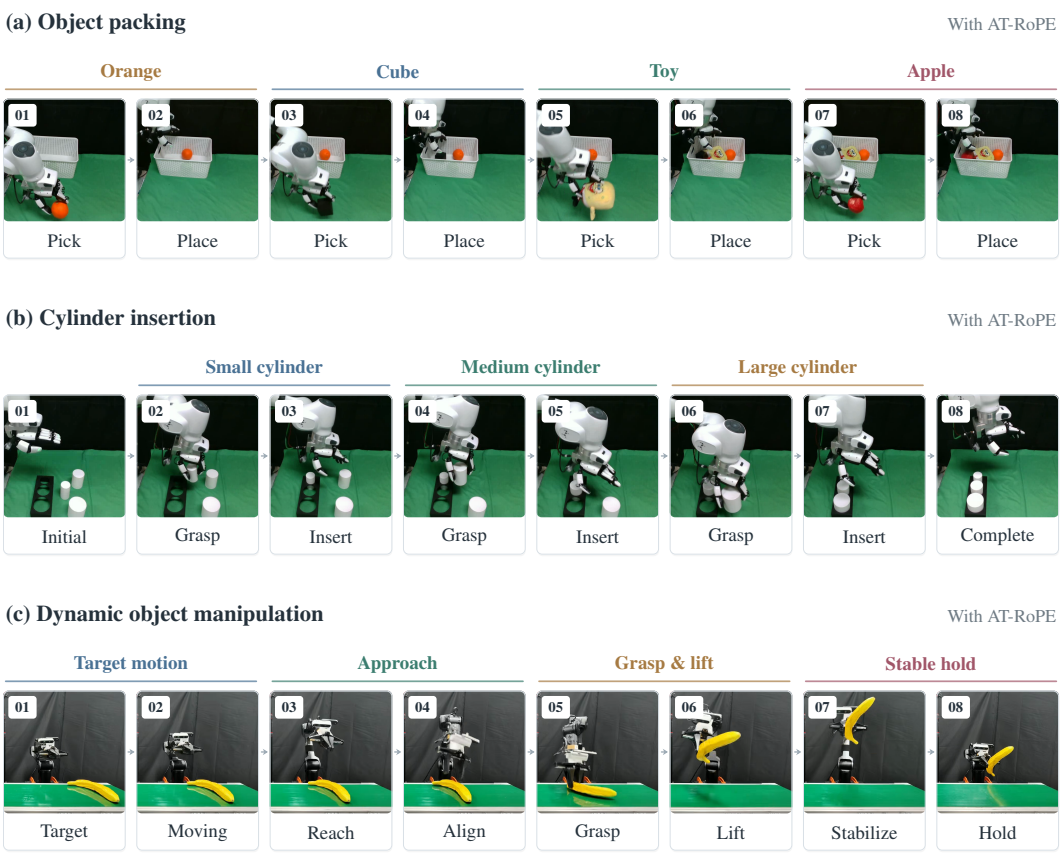

Figure 6: **Selected real-world executions with AT-RoPE.** (a) *Object packing*: one policy sequentially picks and places four objects into the same basket. (b) *Cylinder insertion*: three differently sized cylinders are aligned with and inserted into their matching holes. (c) *Dynamic object manipulation*: the robot approaches, grasps, lifts, and holds a target moving on a conveyor belt. Frames progress from left to right within each row, with a fixed spatial crop for each task. Numbers denote frame order; intervals between selected frames vary.

D.2 Task Descriptions

Object packing. A single policy sequentially transfers an orange, a cube, a toy, and an apple into the same basket. The objective is to place all four objects inside. Figure 6(a) combines the object-specific recordings in execution order: each pair shows pickup and placement, with the accumulated basket contents retained across successive transfers.

Cylinder insertion. The robot inserts three differently sized cylinders into corresponding holes. Figure 6(b) shows the robot handling the small, medium, and large cylinders in order, transporting each to its matching opening before lowering and releasing it. This couples size matching with alignment during insertion.

Dynamic object manipulation. The robot grasps and lifts a banana moving on a conveyor belt. Figure 6(c) shows the target approaching from the right, gripper approach and closure, and lifting followed by stable holding. The changing target position requires the approach and grasp timing to be coordinated throughout acquisition.