MUXWAM: A COMPACT WORLD–ACTION MODEL WITH A NEURAL MULTIPLEXER

Anonymous authors
Paper under double-blind review

ABSTRACT

World–Action Models (WAMs) extend action-centric Vision–Language–Action (VLA) policies with explicit foresight by jointly predicting future world states and robot actions. However, dense future-image prediction in WAMs incurs inference overhead, while recursive decoding and re-encoding can compound prediction errors. To address these challenges, we introduce MuxWAM, a compact WAM whose unified multiplexer-inspired world constructor (MUX) maps dense visual patches to a fixed set of K task-conditioned world tokens per view. MUX uses task-indexed routing over shared experts to refine visual summaries, with normalized global fusion completing the construction. The model jointly predicts future world tokens and actions through flow matching, without future-image decoding at inference. We evaluate MuxWAM on LIBERO, LIBERO-Plus, Meta-World, and physical ALOHA manipulation tasks. With only two world tokens per view, MuxWAM achieves 98.8% mean success on LIBERO, 83.76% averaged equally over Meta-World's four difficulty groups, and 78.3% across three physical tasks under normal conditions. These results demonstrate the effectiveness of task-conditioned compact world representations for robot control in simulation and physical manipulation.

1 INTRODUCTION

Vision–Language–Action (VLA) models map visual observations and language instructions to robot actions (Brohan et al., 2023; Zitkovich et al., 2023; Octo Model Team et al., 2024; Kim et al., 2025b; Black et al., 2025b;a; Qu et al., 2025). Yet action outputs alone do not explicitly describe how the world may evolve over an action horizon H. To provide this prospective state, World–Action Models (WAMs) couple future-world prediction with action generation (Hu et al., 2025; Cen et al., 2025b; Zhang et al., 2025; Kim et al., 2026). This raises the question: what must a WAM predict about the world to support control?

When the prediction target is a complete future image, model capacity also goes to texture, illumination, and background rather than only to interaction-relevant state. Moreover, if predicted images are decoded and re-encoded for recursive feedback, codec-cycle residuals can enter later updates. Feature prediction and compact world representations offer an alternative to dense reconstruction (Bardes et al., 2024; Wang et al., 2026a; Tang et al., 2026): for view c, map N dense patches $\boldsymbol{P}_t^c$ to $\boldsymbol{Z}_t^c \in \mathbb{R}^{K\times d_z}$ with $K \ll N$.

However, fixing K specifies the size of the predictive state without determining how visual evidence is processed for different operation tasks m. Token selection, pooling, and merging reduce visual redundancy (Ryoo et al., 2021; Marin et al., 2023; Bolya et al., 2023), while task-specific routers enable task-conditioned feature processing (Liang et al., 2022). This raises our central question: how can the constructor $C_\psi(\boldsymbol{P}_t^c; m, c)$ adapt to different operation tasks while retaining a fixed K-token-per-view interface?

To realize this design, we introduce MuxWAM, a compact world–action model. Its unified *multiplexer-inspired world constructor (MUX)* uses task-indexed expert routing to adapt task-shared visual summaries within the MoE path. Task identity selects router parameters, while token content determines weights over shared experts. Normalized global fusion combines the Linear and refined MoE outputs with coefficients shared across tasks, completing the construction of K world tokens per view. The model then jointly predicts the structured target $\boldsymbol{Y}_t = (\boldsymbol{A}_t, \boldsymbol{Z}_t^+)$, comprising an action chunk and

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.

future world tokens; inference uses neither future images nor a pixel decoder. Our contributions are summarized as follows:

- We propose MuxWAM, a compact world–action framework that jointly predicts robot actions and future world tokens through flow matching, enabling inference without future-image decoding.
- We design MUX as a unified task-conditioned constructor within a fixed K-token-per-view interface. Task IDs select router parameters, token content weights shared experts, and normalized global fusion with task-shared coefficients completes the construction.
- We prove that task-sufficient K-token representations exist when each observed task admits a control-sufficient linear spatial statistic of rank at most K, and bound codec-cycle error for recursive decode–re-encode feedback under Lipschitz assumptions (Propositions 1 and 2).
- Control and perturbation evaluations across LIBERO, LIBERO-Plus, Meta-World, and physical ALOHA tasks support the compact interface. At $K = 2$, the Long routing comparison yields 98.4% success with task-conditioned routing versus 92.8% with shared routing; four-suite diagnostics characterize within-chunk prediction and cross-chunk feedback errors.

2 Related Work

Vision–Language–Action Models. Generalist robot policies map visual observations and language to actions through pretraining and multimodal representations (Reed et al., 2022; Brohan et al., 2023; Zitkovich et al., 2023; Driess et al., 2023; Octo Model Team et al., 2024; Kim et al., 2025b). Action chunking produces action sequences (Zhao et al., 2023), while diffusion and flow matching model action generation (Chi et al., 2023; Black et al., 2025b;a). Spatial grounding and generalist training extend these policies (Qu et al., 2025; NVIDIA et al., 2025). Their action-centric output interface typically does not explicitly predict a future-world trajectory, even when internal features encode dynamics.

World–Action Models. World models learn predictive state representations for control (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025). WAMs couple world prediction with action generation through images, video, features, or structured future states (Hu et al., 2025; Cen et al., 2025b; Zhang et al., 2025; Kim et al., 2026; Jiang et al., 2026; Akbari et al., 2026). Feature-based and compact prediction objectives already offer alternatives to dense pixel targets (Bardes et al., 2024; Guan et al., 2026; Li et al., 2026b; Wang et al., 2026a; Ma et al., 2026; Yu et al., 2026; Qiu et al., 2026; Tang et al., 2026). Within this line of work, MuxWAM studies how to change task-conditioned computation within the world constructor while keeping the K-token-per-view prediction interface fixed.

3 Preliminaries

We ask when a fixed K-token world representation can preserve control-relevant information. At environment step t, the policy observes instruction ℓ, robot state $\boldsymbol{s}_t$, images $X_t = \{\boldsymbol{x}_t^c\}_{c\in\mathcal{C}}$, and a closed-set operation-task index $m \in \{0, \dots, M-1\}$ fixed over the prediction chunk. An encoder E_ϕ maps a view to $\boldsymbol{P}_t^c = E_\phi(\boldsymbol{x}_t^c) \in \mathbb{R}^{N\times d_z}$. Environment time t and horizon H differ from flow time τ and solver count N_{FE}; the token budget K differs from expert count E_{moe} and routing budget q.

Definition 1 (Fixed compact world interface)**.** *A constructor maps each view's patches to*

$$\boldsymbol{Z}_t^c = C_\psi(\boldsymbol{P}_t^c; m, c) \in \mathbb{R}^{K\times d_z}, \qquad K \ll N. \tag{1}$$

For fixed K and d_z, the prediction targets are the action chunk $\boldsymbol{A}_t = \boldsymbol{a}_{t:t+H-1}$ and future world tokens $\boldsymbol{Z}_t^+ = \{\boldsymbol{Z}_{t+1:t+H}^c\}_{c\in\mathcal{C}}$, rather than complete future images. Task or view conditioning changes computation, not the interface shape.

3.1 Task Sufficiency under a Fixed Token Budget

Task sufficiency concerns control-relevant information, not pixel reconstruction.

Definition 2 (Task-sufficient representation)**.** *Let $\boldsymbol{P} \in \mathbb{R}^{N\times d}$ be dense patches with feature width d ($d = d_z$ here), and let the control-relevant target be*

$$\boldsymbol{R}_t = (\boldsymbol{A}_t, \boldsymbol{\Xi}_{t+1:t+H}), \tag{2}$$

where Ξ contains future interaction semantics such as pose, contact, and task progress. A representation $\boldsymbol{Z} = C(\boldsymbol{P}; m) \in \mathbb{R}^{K \times d}$ is task-sufficient under context $(\ell, m, \boldsymbol{s}_t)$ if

$$p(\boldsymbol{R}_t \mid \boldsymbol{P}, \ell, m, \boldsymbol{s}_t) = p(\boldsymbol{R}_t \mid \boldsymbol{Z}, \ell, m, \boldsymbol{s}_t), \tag{3}$$

equivalently $I(\boldsymbol{R}_t; \boldsymbol{P} \mid \boldsymbol{Z}, \ell, m, \boldsymbol{s}_t) = 0$. Semantic collapse means losing predictive information about $\boldsymbol{R}_t$, not failing to reconstruct every pixel.

Assumption 1 (Low-rank sufficient statistic)**.** *For each observed task m, an r_m-dimensional linear spatial statistic is control-sufficient:*

$$T_m(\boldsymbol{P}) = \boldsymbol{U}_m^\top \boldsymbol{P} \in \mathbb{R}^{r_m \times d}, \qquad \boldsymbol{R}_t \perp\!\!\!\perp \boldsymbol{P} \mid (T_m(\boldsymbol{P}), \ell, m, \boldsymbol{s}_t), \tag{4}$$

where $\boldsymbol{U}_m \in \mathbb{R}^{N \times r_m}$, $\operatorname{rank}(\boldsymbol{U}_m) = r_m$, and $r_m \le K$.

Proposition 1 (Low-rank task sufficiency)**.** *Under Assumption 1, there exist a task-conditioned pooling map $\boldsymbol{M}_m \in \mathbb{R}^{K \times N}$ and readout $\boldsymbol{B}_m \in \mathbb{R}^{r_m \times K}$ such that, for $\boldsymbol{Z}_m = \boldsymbol{M}_m \boldsymbol{P}$,*

$$T_m(\boldsymbol{P}) = \boldsymbol{B}_m \boldsymbol{Z}_m. \tag{5}$$

Consequently, $\boldsymbol{Z}_m$ is task-sufficient, and conditional on $(\ell, m, \boldsymbol{s}_t)$ its Bayes-optimal risk equals that obtainable from all N patches for any bounded task loss.

Under Assumption 1, the condition $K \ge \max_m r_m$ ensures that a task-sufficient compact representation exists for each observed task. This existence result does not establish that trained MUX realizes such a representation, that $K = 2$ suffices for all tasks, or that its tokens avoid semantic collapse. An approximate row-space match, $\|\boldsymbol{U}_m^\top - \boldsymbol{B}_m \boldsymbol{M}_m\|_2 \le \delta_m$, further yields

$$\|T_m(\boldsymbol{P}) - \boldsymbol{B}_m \boldsymbol{Z}_m\|_F \le \delta_m \|\boldsymbol{P}\|_F. \tag{6}$$

The proof is in Appendix A.8.1. Appendix A.8.2 analyzes codec residuals as an additional feedback error term when predicted images are recursively decoded and re-encoded.

The theory gives design conditions; MUX provides a task-conditioned constructor (Section 4.1). The token-budget results (Appendix Table 10) measure control performance; prediction diagnostics (Figure 3) characterize error behavior.

4 Methodology

The fixed interface in Definition 1 sets the shape of the predictive world state. MuxWAM specifies how visual evidence becomes that state and how it conditions joint world–action prediction (Figure 1). We describe the constructor before flow initialization, prediction, and training.

4.1 World Context Encoding and MUX Construction

World context encoding. At control step t, the model receives images from the active views $X_t = \{\boldsymbol{x}_t^c\}_{c \in \mathcal{C}}$, instruction ℓ, robot state $\boldsymbol{s}_t$, and operation-task index m. For each view c, the image encoder produces $\boldsymbol{P}_t^c = E_\phi(\boldsymbol{x}_t^c) \in \mathbb{R}^{N \times d_z}$. From these visual patches, MUX constructs the fixed-interface output of Definition 1: $\boldsymbol{Z}_t^c = C_\psi(\boldsymbol{P}_t^c; m, c) \in \mathbb{R}^{K \times d_z}$ for each view. The current world tokens and language embeddings $L(\ell)$ form the cached context prefix $\boldsymbol{B}_t = [L(\ell); \{\boldsymbol{Z}_t^c\}_{c \in \mathcal{C}}]$. The World–Action Expert attends to this prefix; robot state $\boldsymbol{s}_t$ enters the prediction sequence separately.

MUX construction. MUX constructs a task-conditioned world representation within the fixed $K \times d_z$ output interface of Definition 1. It aggregates shared visual evidence, refines learned-query representations according to the task, and closes with normalized global fusion:

$$\boldsymbol{Z} = \mathrm{MUX}_\psi(\boldsymbol{P}; m, c) = \mathcal{F}_\psi(\mathcal{R}_{\psi,m}(\mathcal{A}_\psi(\boldsymbol{P}; c))) \in \mathbb{R}^{K \times d_z}, \tag{7}$$

Here $\mathcal{A}_\psi$ forms the Linear summary $\boldsymbol{V}_L$ and the MoE query-group summaries $\{\boldsymbol{V}_{M,r}\}_{r=1}^2$ in parallel from the same patches. Each has shape $K \times d_z$. The refinement $\mathcal{R}_{\psi,m}$ leaves $\boldsymbol{V}_L$ unchanged and acts on the learned-query summaries; $\mathcal{F}_\psi$ combines their normalized representations.

Shared evidence aggregation. Following visual token selection, pooling, and merging (Ryoo et al., 2021; Marin et al., 2023; Bolya et al., 2023), Linear forms K weighted patch sums using

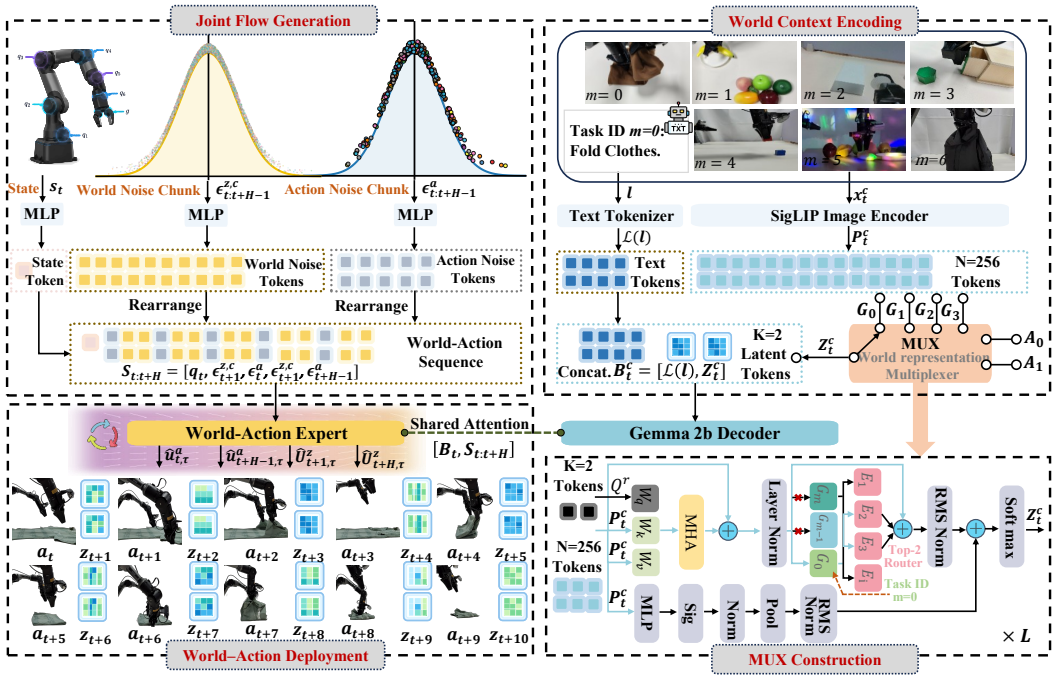


Figure 1: **MuxWAM and its unified MUX constructor.** The context path encodes current images and language, and MUX constructs K world tokens per view. The prediction path initializes future-world and action flows from Gaussian noise and predicts both jointly. Only actions are returned for execution; future images supply training targets only and are absent at inference. Appendix A.1 gives the MUX implementation details.

view-specific scoring heads and normalized sigmoid scores. MoE reads the same patches through two learned-query groups for attention and lightweight pooling (Lee et al., 2019). Queries are learned tokens; keys and values are visual patches. Aggregation parameters are shared across tasks; Appendix A.1 details the groups' internal normalization, projection, residual, and refinement capacities.

Task-conditioned refinement. For each query group $r \in \{1, 2\}$, task index m acts as the select input of an M-to-1 multiplexer, choosing $R_{m,r}$ from router bank $\{R_{j,r}\}_{j=0}^{M-1}$ with shared experts (Liang et al., 2022). For slot $\boldsymbol{v} = (\boldsymbol{V}_{M,r})_k$, set $\boldsymbol{u} = \mathrm{LN}(\boldsymbol{v})$ and $\delta_j(m) = \mathbf{1}\{j = m\}$. The selected router maps slot content to expert weights, with $T = \mathrm{Top}_q(\boldsymbol{\rho})$:

$$\boldsymbol{\rho} = \tau_r^{-1} \sum_{j=0}^{M-1} \delta_j(m) R_{j,r}(\boldsymbol{u}), \quad g_e = \frac{\mathbf{1}\{e \in T\} \exp(\rho_e)}{\sum_{i \in T} \exp(\rho_i)}, \quad (\boldsymbol{U}_{M,r})_k = \boldsymbol{v} + \sum_{e=1}^{E_{\mathrm{moe}}} g_e f_{r,e}(\boldsymbol{u}). \tag{8}$$

All $E_{\mathrm{moe}} = 4$ expert responses are computed before the Top-q weights gate the residual sum. Each query group has its own expert bank, shared across tasks, views, frames, and slots. Linear bypasses refinement, $\boldsymbol{U}_L = \boldsymbol{V}_L$; routing budget $q = 2$ is distinct from world-token budget K.

Normalized global fusion. Token-wise RMS normalization aligns scales before fusion forms the final K world tokens. Set $\overline{\boldsymbol{U}}_L = \mathrm{RMS}(\boldsymbol{U}_L)$ and $\overline{\boldsymbol{U}}_M = \sum_{r=1}^{2} \eta_r \, \mathrm{RMS}(\boldsymbol{U}_{M,r})$, where the learned global weights satisfy $\sum_r \eta_r = 1$. The constructor output is

$$\boldsymbol{Z} = \pi_L \overline{\boldsymbol{U}}_L + \pi_M \overline{\boldsymbol{U}}_M, \qquad \pi_L + \pi_M = 1. \tag{9}$$

The fusion coefficients are shared across tasks, samples, views, frames, and slots; task conditioning acts through the internal routers. Appendix A.1 gives the normalized weight expressions.

4.2 Joint Flow Initialization and World–Action Prediction

Conditioned on the current compact context prefix $\boldsymbol{B}_t$, MuxWAM jointly predicts the action chunk $\boldsymbol{A}_t$ and future world tokens $\boldsymbol{Z}_t^+$ defined in Definition 1. They form one structured target $\boldsymbol{Y}_t = (\boldsymbol{A}_t, \boldsymbol{Z}_t^+)$, with all H future positions generated together within a chunk.

Joint generation. Independent Gaussian blocks initialize future-world groups and actions, with samples $\boldsymbol{\varepsilon}^{z,c}_{t+1:t+H}$ and $\boldsymbol{\varepsilon}^{a}_{t:t+H-1}$, respectively (Figure 1). Each future-world group contains K tokens per active camera view. Robot state $\boldsymbol{s}_t$, noisy world tokens, and noisy actions receive separate projections to the World–Action Expert width; modality-specific MLPs incorporate sinusoidal flow-time embeddings for world and action tokens. At flow time τ, the world-first suffix places each future-world group before its paired action:

$$\boldsymbol{S}_\tau = [\boldsymbol{s}_t; \widetilde{\boldsymbol{Z}}_{t+1,\tau}, \widetilde{\boldsymbol{a}}_{t,\tau}; \ldots; \widetilde{\boldsymbol{Z}}_{t+H,\tau}, \widetilde{\boldsymbol{a}}_{t+H-1,\tau}], \qquad \widehat{\boldsymbol{u}}_\tau = \mathcal{F}_\theta(\boldsymbol{B}_t, \boldsymbol{S}_\tau, \tau; \mathcal{M}), \tag{10}$$

where all entries are projected embeddings and each future-world group contains the active camera views. The World–Action Expert jointly predicts future-world and action velocities from $\boldsymbol{B}_t$ and $\boldsymbol{S}_\tau$. The block-causal mask $\mathcal{M}$ lets each action read its paired future-world group, while preventing that group from reading its paired action or later groups. Modality-specific heads predict action velocities $\widehat{\boldsymbol{u}}^a_{t+h-1,\tau}$ and world-token velocities $\widehat{\boldsymbol{U}}^{z,c}_{t+h,\tau}$ for $h = 1, \ldots, H$ (Black et al., 2025b). Starting from independent Gaussian noise $\widetilde{\boldsymbol{Y}}_{\tau_0}$, the joint world–action state follows $1 = \tau_0 > \cdots > \tau_{N_{\mathrm{FE}}} = 0$ through Euler updates:

$$\widetilde{\boldsymbol{Y}}_{\tau_{n+1}} = \widetilde{\boldsymbol{Y}}_{\tau_n} + (\tau_{n+1} - \tau_n)\mathcal{F}_\theta(\boldsymbol{B}_t, \boldsymbol{S}_{\tau_n}, \tau_n; \mathcal{M}). \tag{11}$$

Each solver step refines the entire chunk in flow time τ; it does not advance environment time or generate the H future positions one by one.

MuxWAM optimization. During training, observed future images supply the world-token targets. Current and future images share the encoder and MUX, with gradients through both context and target construction and no pixel reconstruction objective. Together with the ground-truth action chunk, these targets form $\boldsymbol{Y}_t$. Following flow matching (Lipman et al., 2023), independent Gaussian noises for actions and each view share one flow time τ:

$$\widetilde{\boldsymbol{Y}}_\tau = (1 - \tau)\boldsymbol{Y}_t + \tau\boldsymbol{\varepsilon}, \qquad \boldsymbol{u}_\tau = \boldsymbol{\varepsilon} - \boldsymbol{Y}_t. \tag{12}$$

Interpolation and target velocities act component-wise. The prefix, world-first suffix, and mask above condition the predicted field $\widehat{\boldsymbol{u}}_\tau$ on current observations while aligning each action with its paired future world. Let $\mathcal{L}_{\mathrm{act}}$ and $\mathcal{L}^c_{\mathrm{world}}$ average absolute velocity errors over Hd_a action coordinates and HKd_z view-c token coordinates, where d_a is the action dimension. The joint objective is

$$\mathcal{L} = \lambda_A(\mathcal{L}_{\mathrm{act}} + \lambda_b\mathcal{R}_{\mathrm{bal}}) + \sum_{c\in\mathcal{C}} \lambda_c\mathcal{L}^c_{\mathrm{world}}. \tag{13}$$

Deployment. The joint sampler produces a predicted action chunk $\widehat{\boldsymbol{A}}_t = \hat{\boldsymbol{a}}_{t:t+H-1}$ and future world tokens $\widehat{\boldsymbol{Z}}_t^+ = \{\widehat{\boldsymbol{Z}}^c_{t+1:t+H}\}_{c\in\mathcal{C}}$. Only actions are executed; future tokens remain internal predictions. Standard replanning uses newly observed images to rebuild the context. Predicted tokens already occupy the fixed K-token interface and are not passed through MUX again. This sampling path operates directly on world tokens, requiring neither future images nor a pixel decoder at deployment.

5 Experiments

5.1 Experimental Setup

In this section, we conduct extensive experiments to address the following research questions: (i) Can a fixed compact world interface support control across manipulation tasks and distribution shifts? (ii) How does task-conditioned construction affect control under a fixed token budget? (iii) How do future-token errors behave within prediction chunks and during cross-chunk latent feedback?

Benchmarks. We evaluate three simulation settings: LIBERO (Liu et al., 2023) tests multi-task manipulation, Meta-World (Yu et al., 2020; McLean et al., 2025) tests control across four difficulty

Table 1: Robustness on **LIBERO-Plus**: success rates (%) with declines from Original (↓, percentage points) on the right; darker red denotes larger declines. **Bold** marks the best generalization scores in each perturbation and Avg. Avg. equally averages MuxWAM's seven category scores; its in-domain category scores also equally average the four suites (Appendix Table 9).

Method	Original	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
Trained on LIBERO-Plus (In-domain Evaluation)									
OpenVLA-OFT (Kim et al., 2025a)	97.1	92.8 ↓4.3	30.3 ↓66.8	85.8 ↓11.3	94.9 ↓2.2	93.9 ↓3.2	89.3 ↓7.8	77.6 ↓19.5	79.6
$\pi_{0.5}$ (Black et al., 2025a)	96.9	67.2 ↓29.7	42.4 ↓54.5	59.4 ↓37.5	75.8 ↓21.1	74.9 ↓22.0	72.6 ↓24.3	64.5 ↓32.4	65.3
MuxWAM (Ours)	98.8	96.3 ↓2.5	56.7 ↓42.1	95.6 ↓3.2	97.0 ↓1.8	96.9 ↓1.9	92.3 ↓6.5	85.5 ↓13.3	88.6
Trained on LIBERO (Generalization)									
OpenVLA (Kim et al., 2025b)	76.5	0.8 ↓75.7	3.5 ↓73.0	23.0 ↓53.5	8.1 ↓68.4	34.8 ↓41.7	15.2 ↓61.3	28.5 ↓48.0	15.6
OpenVLA-OFT (Kim et al., 2025a)	97.1	56.4 ↓40.7	31.9 ↓65.2	79.5 ↓17.6	88.7 ↓8.4	93.3 ↓3.8	75.8 ↓21.3	74.2 ↓22.9	69.6
WorldVLA (Cen et al., 2025b)	79.1	0.1 ↓79.0	27.9 ↓51.2	41.6 ↓37.5	43.7 ↓35.4	17.1 ↓62.0	10.9 ↓68.2	38.0 ↓41.1	25.0
Fast-WAM (Yuan et al., 2026b)	97.6	16.4 ↓81.2	44.5 ↓53.1	68.9 ↓28.7	78.2 ↓19.4	53.7 ↓43.9	37.7 ↓59.9	60.7 ↓36.9	50.0
SG-WAM (Zhao et al., 2026)	98.5	58.6 ↓39.9	48.9 ↓49.6	81.4 ↓17.1	89.8 ↓8.7	86.1 ↓12.4	80.7 ↓17.8	74.2 ↓24.3	73.0
DreamWAM (Yuan et al., 2026a)	98.9	53.8 ↓45.1	63.6 ↓35.3	94.8 ↓4.1	**96.7** ↓2.2	71.6 ↓27.3	67.2 ↓31.8	80.7 ↓18.2	75.5
VLA-JEPA (Sun et al., 2026)	97.2	63.3 ↓33.9	**67.1** ↓30.1	85.4 ↓11.8	95.6 ↓1.6	93.6 ↓3.6	66.3 ↓30.9	**85.1** ↓12.1	79.5
UniVLA (Bu et al., 2025)	95.2	1.8 ↓93.4	46.2 ↓49.0	69.6 ↓25.6	69.0 ↓26.2	81.0 ↓14.2	21.2 ↓74.0	31.9 ↓63.3	42.9
ResVLA (Zhong et al., 2026)	96.6	53.2 ↓43.4	57.5 ↓39.1	88.2 ↓8.4	94.5 ↓2.1	**96.3** ↓0.3	**81.8** ↓14.8	78.3 ↓18.3	76.9
$\pi_{0.5}$ (Black et al., 2025a)	96.9	53.0 ↓43.9	50.3 ↓46.6	65.7 ↓31.2	83.1 ↓13.8	77.3 ↓19.6	53.2 ↓43.7	72.7 ↓24.2	65.0
π_0 (Black et al., 2025b)	94.2	13.8 ↓80.4	6.0 ↓88.2	58.8 ↓35.4	85.0 ↓9.2	81.4 ↓12.8	79.0 ↓15.2	68.9 ↓25.3	53.6
MuxWAM (Ours)	98.8	**76.1** ↓22.7	49.5 ↓49.3	**97.4** ↓1.4	95.6 ↓3.2	90.1 ↓8.7	81.7 ↓17.1	77.3 ↓21.5	**81.1**

tiers, and LIBERO-Plus (Fei et al., 2026) tests the same LIBERO tasks under visual, language, layout, and robot-state perturbations. Physical evaluation uses Cobot Magic ALOHA (Zhao et al., 2023) for banana grasping, clothes folding, and phone assembly.

Implementation. We initialize the image encoder, vision–language backbone, and action expert from π_0 (Black et al., 2025b). Each active view uses $K = 2$ world tokens, with prediction length $H = 10$ in simulation and $H = 30$ on physical ALOHA. Meta-World uses one view; LIBERO, LIBERO-Plus, and physical ALOHA use two. The operation-task ID m selects router parameters; token content determines expert weights. Standard LIBERO supplies m alongside the instruction, independently of its wording. Detailed network dimensions, loss weights, and training settings are given in Appendix Table 6 and Appendix A.1. The supplied LIBERO-Plus summary omits variant-to-task mappings and run-level adaptation records (Appendix A.3).

Baselines. We compare VLAs (π_0, $\pi_{0.5}$, OpenVLA, OpenVLA-OFT) and WAMs (Cosmos Policy, LaWAM), with citations in the tables. Reference values include literature reports and author-supplied evaluations (Appendix A.7); literature results retain their source protocols. Table 5 contains our internal construction and routing comparisons.

5.2 Simulation Results

LIBERO. The four LIBERO suites test whether a fixed K-token world interface supports diverse manipulation tasks. With $K = 2$ tokens per view, MuxWAM reaches 98.8% mean success across four suite-specific evaluations and 98.4% on Long (Table 2). These scores support control with the same token budget across the evaluated tasks; the Long score measures completion of its finite sequences, not stability over arbitrary durations.

Meta-World. The single-view Meta-World evaluation tests the same per-view compact interface on a second task family. MuxWAM reaches 83.76% averaged equally over four difficulty groups, not a task-weighted mean over all 50 tasks, with 78.80% on Hard and 88.40% on Very Hard (Table 3). These group-level results extend the evidence for control with $K = 2$ to the Meta-World tasks.

LIBERO-Plus. The seven LIBERO-Plus perturbation axes probe how the fixed world interface behaves under shifts from LIBERO training conditions. The LIBERO-trained evaluations yield an 81.1% mean success rate across the seven perturbation categories (Table 1). Success under Light, Background, and Noise is 95.6%, 90.1%, and 81.7%, respectively, compared with 49.5% under

Table 2: Reported success rates (%) on **LIBERO**. The **best**, second-best, and third-best values are marked, with ties sharing rank. Parameter counts are in billions; ∼ marks backbone-scale estimates. Reference rows follow their stated variants and source protocols; MuxWAM reports suite-specific results. See Appendix A.7 for numerical sources and comparison scope.

Method	Params (B)	Spatial	Object	Goal	Long	Avg.
OpenVLA (Kim et al., 2025b)	7.0	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT (Kim et al., 2025a)	7.0	97.6	98.4	97.9	94.5	97.1
VLA-Adapter (Wang et al., 2026b)	0.5	97.8	99.2	97.2	95.0	97.3
ResVLA (Zhong et al., 2026)	∼2.0	96.0	**100.0**	97.4	92.8	96.6
$\pi_{0.5}$ (Black et al., 2025a)	3.5	98.8	98.2	98.0	92.4	96.9
π_0 (Black et al., 2025b)	3.5	96.8	98.8	95.8	85.2	94.2
WorldVLA (Cen et al., 2025b)	∼7.0	87.6	96.2	83.4	60.0	81.8
RynnVLA-002 (Cen et al., 2025a)	∼7.0	99.0	99.8	96.4	94.4	97.4
Fast-WAM (Yuan et al., 2026b)	6.0	98.2	**100.0**	97.0	95.2	97.6
Motus (Bi et al., 2026)	8.0	96.8	99.8	96.6	97.6	97.7
LingBot-VA (Li et al., 2026a)	5.3	98.5	99.6	97.2	**98.5**	98.5
LaWAM (Chen et al., 2026)	2.3	**99.4**	99.6	**98.4**	97.0	98.6
Cosmos Policy (Kim et al., 2026)	2.1	98.1	**100.0**	98.2	97.6	98.5
MuxWAM	∼3.5	98.8	**100.0**	97.8	98.4	**98.8**

Table 3: Reported success rates (%) on **Meta-World**. Avg. is the macro average of four difficulty groups, not a task-weighted MT50 mean. Training and evaluation budgets differ across sources (Appendix A.7); GR-1 and PAD cover 48 tasks.

Method	Params (B)	Easy	Medium	Hard	V. Hard	Avg.
TinyVLA-H (Wen et al., 2025)	∼1.3	77.6	21.5	11.4	15.8	31.6
SmolVLA (Shukor et al., 2025)	2.25	87.14	51.82	70.0	64.0	68.24
InternVL3-1B (Choi et al., 2026)	1.0	55.4	14.5	1.7	8.0	19.9
D2E (Choi et al., 2026)	1.0	53.6	18.2	8.3	20.0	25.0
MINT-4B (Huang et al., 2026)	4.0	82.1	72.4	58.3	56.0	67.2
Evo-1 (Lin et al., 2026)	0.77	89.2	**76.8**	77.2	79.2	80.6
$\pi_{0.5}$ (Black et al., 2025a)	3.5	71.1	20.9	24.0	10.0	31.5
π_0 (Black et al., 2025b)	3.5	71.8	48.2	41.7	30.0	47.9
GR-1 (Wu et al., 2024)	0.2	76.6	35.3	46.0	44.0	50.5
PAD (Guo et al., 2024)	0.66	81.8	65.1	56.7	87.2	72.7
MuxWAM	∼3.5	**92.21**	75.64	**78.80**	**88.40**	**83.76**

Robot perturbations, indicating uneven control performance through the fixed interface across the tested shifts. The unresolved task-ID mapping prevents interpreting the 97.4% Language score as evidence of stronger language understanding or unseen-task generalization (Appendix A.3). The separate LIBERO-Plus-trained evaluation yields an 88.6% suite-equal, seven-category mean; Appendix Table 9 gives both four-suite breakdowns.

5.3 Real-World Results

Clothes folding and phone assembly are long-horizon tasks on Cobot Magic ALOHA (Figure 2). Folding requires repeated grasping, lifting, and folding of a deformable garment into a neat stack. For phone assembly, we design a custom mechanical setup with a left-arm suction pump and a right-arm gripper. The right arm first grasps the data cable and places it in the box; the left arm then picks up the phone case with suction and places it in the box, followed by the lid to close it. These sequences demand deformable-object handling for folding and ordered bimanual placements with different end effectors for assembly.

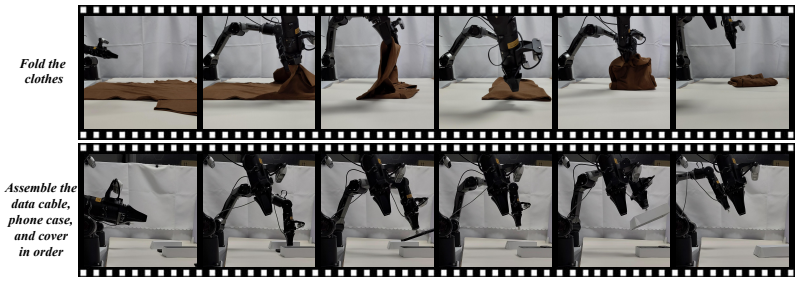


Figure 2: **Qualitative real-robot rollouts with MuxWAM.** Representative execution sequences for clothes folding and phone assembly. Frames progress from left to right.

Table 4: Success rates (%) on physical ALOHA tasks under Normal, Noise (lighting), and Background conditions. Cosmos denotes Cosmos Policy.

Real Robot	Normal				Interference							
					Noise				Background			
	$\pi_{0.5}$	π_0	Cosmos	MuxWAM	$\pi_{0.5}$	π_0	Cosmos	MuxWAM	$\pi_{0.5}$	π_0	Cosmos	MuxWAM
Grasp banana	90.0	90.0	75.0	**100.0**	70.0	80.0	75.0	**70.0**	85.0	85.0	80.0	**100.0**
Fold clothes	25.0	20.0	5.0	**65.0**	10.0	20.0	5.0	**55.0**	25.0	20.0	5.0	**45.0**
Assemble phone	15.0	10.0	5.0	**70.0**	15.0	10.0	10.0	**55.0**	10.0	15.0	5.0	**50.0**
Average	43.3	40.0	28.3	**78.3**	31.7	36.7	30.0	**60.0**	40.0	40.0	30.0	**65.0**

Across the three physical tasks, MuxWAM averages 78.3% success under Normal conditions, including 65.0% for folding and 70.0% for assembly (Table 4). Lighting noise and background changes test robustness, yielding 60.0% and 65.0% mean success, respectively, or decreases of 18.3 and 13.3 percentage points from Normal. Appendix Figure 4 provides banana-grasping examples under normal observations and colored-light interference.

5.4 Ablation Studies

Task-conditioned routing. At $K = 2$ with the same expert-bank capacity, task-conditioned routing achieves 98.4% task-equal success on LIBERO-Long versus 92.8% for shared routing, a 5.6-percentage-point gain without enlarging the prediction interface (Table 5(c)). Both conditions retain four experts per query group and Top-2 weighting. Each condition uses one fixed checkpoint and 50 trajectories per task; seven tasks improve, two tie, and one declines (Appendix A.6).

MUX construction and capacity. Full MUX reaches a 98.8% four-suite mean, compared with 91.5% without Linear and 97.3% without MoE (Table 5(b)). In the Long sweep at $H = 10$, success is 96.0% at $K = 1$, 98.4% at $K = 2$, 98.2% at $K = 4$, and 95.4% at $K = 8$ (Appendix Table 10).

Perturbation behavior. Table 5(a) characterizes full-model token drift on 1,200 same-state clean–perturbed pairs across four suites and three perturbations, alongside clean success and perturbation-specific drops. These measurements describe representation sensitivity, not a comparison of drift across constructors (Appendix A.5).

5.5 Future-Token Prediction and Feedback Analysis

Within-chunk prediction. Under R1, ten future positions are predicted jointly, with real images refreshed after every ten-action chunk. Across 100 episodes per suite, the task-equal, two-view Step-10 cosine error is 0.356–0.369, while mean horizon drift is 0.0025–0.0079 (Figure 3(a,b)). On tasks containing both outcomes, successful episodes have lower mean paired end-step changes than failed episodes in Goal and Long (Figure 3(c)); this is an exploratory association.

Cross-chunk feedback. The R8 diagnostic refreshes real images every eight chunks and feeds back predicted tokens between refreshes while retaining observed robot state. Relative to refresh age

Table 5: **Construction, perturbation, and routing analyses.** (a) Full-model Drift ($\times 10^{-3}$), clean success (%), and Drop = clean minus perturbed success (pp; Table 9); (b) Branch success (%); Avg. equally weights four suites. (c) Long routing success (%) and Δ = Task-cond. − Shared (pp), with 50 trajectories per task. Protocols: Appendices A.5 and A.6.

(a) Full MuxWAM: perturbation and success.

	Drift ↓ ($\times 10^{-3}$)			Drop ↓ (pp)			Succ. ↑ (%)
Suite	Light	Background	Noise	Light	Background	Noise	Clean
Object	1.8	16.2	4.2	0.0	0.8	10.9	**100.0**
Spatial	2.7	11.9	3.8	0.5	5.0	7.3	**98.8**
Goal	1.6	3.7	5.4	1.0	11.7	22.1	**97.8**
Long	1.8	4.3	2.0	11.5	15.7	26.2	**98.4**

(b) Branch ablations (%).

Variant	Spa.	Obj.	Goal	Long	Avg.
w/o Linear	91.8	93.2	91.2	89.6	91.5
w/o MoE	97.0	98.6	97.2	96.4	97.3
Full	**98.8**	**100.0**	**97.8**	**98.4**	**98.8**

(c) Long routing success (%).

Task ID	0	1	2	3	4	5	6	7	8	9	Mean
Task-cond.	**98**	**100**	94	**100**	**98**	**100**	**98**	**100**	**100**	**96**	**98.4**
Shared	94	**100**	**96**	78	96	**100**	88	98	96	82	92.8
Δ (pp)	+4	0	−2	+22	+2	0	+10	+2	+4	+14	**+5.6**

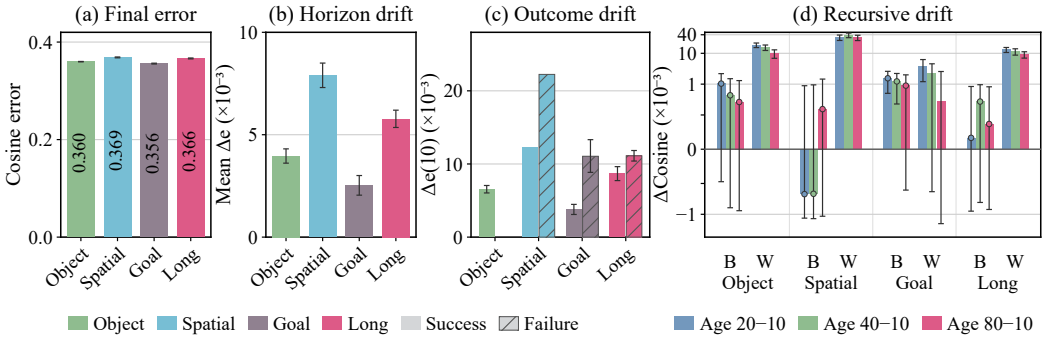


Figure 3: **Future-token prediction and latent feedback.** (a) Final-horizon cosine error $e(10)$. (b) Mean horizon drift, averaging $e(h) - e(1)$ over $h = 2$–10. (c) Outcome-grouped end-step drift for successful (solid) and failed (hatched) episodes; Object has no failures, and Spatial's two failures do not support interval bars. (d) Cross-chunk cosine drift for base (B) and wrist (W) views, comparing ages 20, 40, and 80 against age 10. Panel (d) uses a symmetric-log vertical scale (linear within $\pm 0.1 \times 10^{-3}$); dots mark Base means. Bars are task-equal means; whiskers denote 95% within-task episode-bootstrap intervals. Protocol: Appendix A.5.

10, base-view mean cosine-error increments at ages 20, 40, and 80 remain below 2.5×10^{-3} in magnitude; wrist-view mean increments are generally larger and vary by suite (Figure 3(d)). Refresh age counts executed control steps, rather than future positions within a chunk. R8 is separate from standard fresh-image replanning. Definitions and sampling procedures are in Appendix A.5.

6 Conclusion

MuxWAM uses its unified MUX construction to condition world-representation processing on the task while retaining a fixed K-token-per-view prediction interface. This representation supports joint world–action prediction and decoder-free inference. At $K = 2$, the LIBERO-Long routing comparison yields 92.8% success with shared routing and 98.4% with task-conditioned routing, while simulation and physical execution results support control through this compact interface. Together, these findings support treating task-conditioned construction and prediction-state size as separate design choices: a compact interface can accommodate task-conditioned computation.

AI USE STATEMENT

The authors used generative AI tools for language editing, manuscript comparison, and LaTeX migration. The authors reviewed the resulting text, equations, citations, and source files and remain responsible for every claim and artifact.

ETHICS STATEMENT

This work studies robot manipulation in simulation and on laboratory platforms. Physical trials use bounded workspaces, task-specific termination rules, and hardware safety stops. The study excludes human subjects and personal data.

REPRODUCIBILITY STATEMENT

Appendix A details the training, physical-robot, perturbation, efficiency, stability, and ablation protocols for MuxWAM.

REFERENCES

Arman Akbari, Ci Zhang, Arash Akbari, Lin Zhao, Yixiao Chen, Weiwei Chen, Xuan Zhang, Geng Yuan, and Yanzhi Wang. Flash-WAM: Modality-aware distillation for world action models. *arXiv preprint arXiv:2606.05254*, 2026. URL https://arxiv.org/abs/2606.05254.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://arxiv.org/abs/2404.08471.

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, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

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, pp. 17–40, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.09461.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent

Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://doi.org/10.15607/RSS.2023.XIX.025.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.014. URL https://www.roboticsproceedings.org/rss21/p014.html.

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

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025b. URL https://arxiv.org/abs/2506.21539.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026. URL https://arxiv.org/abs/2606.15768.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

Suhwan Choi, Jaeyoon Jung, Haebin Seong, Minchan Kim, Minyeong Kim, Yongjun Cho, Yoonshik Kim, Yubeen Park, Youngjae Yu, and Yunsung Lee. D2E: Scaling vision-action pretraining on desktop data for transfer to embodied AI. In *International Conference on Learning Representations*, pp. 46207–46236, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4cafca2c978ce63cafcea9f0484a93c0-Abstract-Conference.html.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 8469–8488, 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

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*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Jiarui Guan, Wenshuai Zhao, Yue Pei, Ziliang Chen, Arno Solin, and Juho Kannala. Point tracking improves world action models. *arXiv preprint arXiv:2605.23856*, 2026. URL https://arxiv.org/abs/2605.23856.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. In *Advances in Neural Information Processing Systems*, volume 37, pp. 112386–112410, 2024. doi: 10.52202/079017-3570. URL https://papers.nips.cc/paper_files/paper/2024/hash/cbe25fa0e7c7084049276888a09acc8d-Abstract-Conference.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24328–24346, 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Renming Huang, Chendong Zeng, Wenjing Tang, Jintian Cai, Cewu Lu, and Panpan Cai. Mimic intent, not just trajectories. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.206. URL https://www.roboticsproceedings.org/rss22/p206.html.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. WoVR: World models as reliable simulators for post-training VLA policies with RL. *arXiv preprint arXiv:2602.13977*, 2026. URL https://arxiv.org/abs/2602.13977.

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*, 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 2679–2713, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

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://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3744–3753, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

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 2026a. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-WAM: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026b. URL https://arxiv.org/abs/2606.08242.

Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang. M^3ViT: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28441–28457, 2022. doi: 10.52202/068431-2062. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b653f34d576d1790481e3797cb740214-Abstract-Conference.html.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13397–13406, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_Evo-1_Lightweight_Vision-Language-Action_Model_with_Preserved_Semantic_Alignment_CVPR_2026_paper.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

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. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Fulong Ma, Daojie Peng, Wenjun Yue, Jiahang Cao, Bintao Wang, Qiang Zhang, and Jun Ma. GeoSem-WAM: Geometry- and semantic-aware world action models. *arXiv preprint arXiv:2606.03188*, 2026. URL https://arxiv.org/abs/2606.03188.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 12–21, 2023. doi: 10.1109/WACV56688.2023.00010. URL https://openaccess.thecvf.com/content/WACV2023/html/Marin_Token_Pooling_in_Vision_Transformers_for_Image_Classification_WACV_2023_paper.html.

Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K. R. Zentner, Ryan Julian, Jordan Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Meta-World+: An improved, standardized, RL benchmark. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4960. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/da295a7037612bb3bfc1334a6f8fc8bd-Abstract-Datasets_and_Benchmarks_Track.html.

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, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint arXiv:2606.12217*, 2026. URL https://arxiv.org/abs/2606.12217.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.011. URL https://doi.org/10.15607/RSS.2025.XXI.011.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://arxiv.org/abs/2205.06175.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12786–12797, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1332–1344. PMLR, 2023. URL https://proceedings.mlr.press/v205/seo23a.html.

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.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. In *Computer Vision – ECCV 2026*, volume 17064 of *Lecture Notes in Computer Science*, pp. 478–497. Springer, 2026. doi: 10.1007/978-3-032-37196-6_26. URL https://link.springer.com/chapter/10.1007/978-3-032-37196-6_26.

Zuojin Tang, Shengchao Yuan, Xiaoxin Bai, Zhiyuan Jing, De Ma, Gang Pan, and Bin Liu. One token per frame: Reconsidering visual bandwidth in world models for VLA policy. *arXiv preprint arXiv:2605.07931*, 2026. URL https://arxiv.org/abs/2605.07931.

Junke Wang, Qihang Zhang, Shuai Yang, Yiming Luo, Yujun Shen, Zuxuan Wu, Yu-Gang Jiang, and Yinghao Xu. RepWAM: World action modeling with representation visual-action tokenizers. *arXiv preprint arXiv:2606.13674*, 2026a. URL https://arxiv.org/abs/2606.13674.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. VLA-Adapter: An effective paradigm for tiny-scale vision-language-action model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18638–18646, 2026b. doi: 10.1609/aaai.v40i22.38931. URL https://ojs.aaai.org/index.php/AAAI/article/view/38931.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. TinyVLA: Toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(4):3988–3995, 2025. doi: 10.1109/LRA.2025.3544909. URL https://doi.org/10.1109/LRA.2025.3544909.

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. In *International Conference on Learning Representations*, pp. 10641–10662, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2c37c5bcef24b9541550261dcd63261b-Abstract-Conference.html.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. MaskWAM: Unifying mask prompting and prediction for world-action models. *arXiv preprint arXiv:2606.13515*, 2026. URL https://arxiv.org/abs/2606.13515.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 1094–1100, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026a. URL https://arxiv.org/abs/2608.04996.

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*, 2026b. URL https://arxiv.org/abs/2603.16666.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems*, volume 38, pp. 24195–24228, 2025. doi: 10.52202/085713-0818. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/22d4f952efa13970f0b1ffb22170d416-Abstract-Conference.html.

Ruiteng Zhao, Zhengshen Zhang, Yue Su, Wenshuo Wang, Jiahui Li, Zhiyuan Yang, Francis E. H. Tay, Marcelo H. Ang, Jr., and Haiyue Zhu. SG-WAM: Self-guided world modeling in geometry-aware policy space. *arXiv preprint arXiv:2608.01397*, 2026. URL https://arxiv.org/abs/2608.01397.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://doi.org/10.15607/RSS.2023.XIX.016.

Yiming Zhong, Yaoyu He, Zemin Yang, Pengfei Tian, Yifan Huang, Qingqiu Huang, Xinge Zhu, and Yuexin Ma. From noise to intent: Anchoring generative VLA policies with residual bridges. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2604.21391.

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, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A SUPPLEMENTARY MATERIAL

This supplement details the MuxWAM training, physical-robot, perturbation, stability, and ablation protocols. Reported reference results are distinguished from within-configuration controls.

Our simulation benchmark success rates on LIBERO, LIBERO-Plus, and Meta-World are averaged over three evaluations. This averaging does not apply to the physical-robot experiments.

A.1 TRAINING HYPERPARAMETERS

Table 6 summarizes the shared model dimensions and benchmark-specific training settings. All settings use LoRA adaptation of the Gemma-2B backbone and the 300M-parameter action expert. The optimizer, learning-rate schedule, LoRA settings, and training seed are shared across benchmarks as listed in the table. LoRA adapts the query, key/value, and output projections in attention and the gate, up, and down projections in each feed-forward block of both transformers. LIBERO, LIBERO-Plus, and physical ALOHA use two active views, providing four current world tokens at $K = 2$; Meta-World uses one view and two tokens. The LIBERO-Plus column gives the 90K-update in-domain adaptation setting. LIBERO-Plus generalization uses the LIBERO-trained policies; the separate in-domain adapted results are reported in Table 9. Current and future images share the image encoder and MUX parameters, with gradients through both context and target construction.

MUX implementation details. MUX implements Definition 1 by constructing task-conditioned representations within a fixed K-token interface. Shared evidence aggregation forms view-specific Linear summaries and learned-query summaries from the same patches. Task-conditioned MoE refinement acts on the latter, and normalized global fusion completes the construction. Table 7 summarizes the implementation; the equations below specify these computations.

Shared evidence aggregation. For patches $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N]^\top$, MUX forms Linear and learned-query summaries from the same visual evidence. Linear uses K affine scoring maps for each active view c, with positive scores normalized over patches:

$$a^c_{nk} = \frac{\sigma(\boldsymbol{p}_n^\top \boldsymbol{w}^c_k + \beta^c_k)}{\sum_{i=1}^N \sigma(\boldsymbol{p}_i^\top \boldsymbol{w}^c_k + \beta^c_k) + \epsilon}, \qquad (\boldsymbol{U}_L)_k = \sum_{n=1}^N a^c_{nk} \boldsymbol{p}_n, \tag{14}$$

where σ is the sigmoid function and $\epsilon = 10^{-6}$. Linear produces K weighted summaries using view-specific scoring parameters shared across tasks.

MoE aggregates evidence with two learned-query groups, indexed by $r \in \{1, 2\}$, that differ in attention and refinement capacity. Each group has K learned queries $\boldsymbol{Q}_r \in \mathbb{R}^{K \times d_z}$ and reads the same patch set:

$$\begin{aligned} \boldsymbol{V}_{M,1} &= \boldsymbol{Q}_1 + \boldsymbol{W}_1^O \mathcal{A}_1(\boldsymbol{W}_1^Q \boldsymbol{Q}_1, \boldsymbol{W}_1^K \boldsymbol{P}, \boldsymbol{W}_1^V \boldsymbol{P}), \\ \boldsymbol{V}_{M,2} &= \boldsymbol{W}_2^Q \boldsymbol{Q}_2 + \mathcal{A}_2(\boldsymbol{W}_2^Q \boldsymbol{Q}_2, \boldsymbol{W}_2^K \boldsymbol{P}, \boldsymbol{W}_2^V \boldsymbol{P}). \end{aligned} \tag{15}$$

Each $\boldsymbol{W}$ is an affine projection applied row-wise, and $\mathcal{A}_r$ includes splitting and concatenating eight attention heads. The first group RMS-normalizes projected queries and keys, scales attention by $\sqrt{d_h}$ with $d_h = 256$, and uses an output projection. The second group uses $\sqrt{d_z}$ scaling and a projected-query residual. The groups have independent projection parameters; each group's parameters are shared across camera views and tasks.

Task-conditioned refinement. Linear summaries pass through refinement unchanged. For a token $\boldsymbol{v} = (\boldsymbol{V}_{M,r})_k$ and $\boldsymbol{u} = \mathrm{LN}(\boldsymbol{v})$, task identity m selects router $R_{m,r}$ from the query group's task-indexed router bank. The selected router maps normalized token content to expert weights:

$$\begin{aligned} \boldsymbol{\rho} &= R_{m,r}(\boldsymbol{u})/\tau_r, \qquad T = \mathrm{Top}_q(\boldsymbol{\rho}), \\ g_e &= \frac{\mathbf{1}\{e \in T\} \exp(\rho_e)}{\sum_{j \in T} \exp(\rho_j)}, \qquad (\boldsymbol{U}_{M,r})_k = \boldsymbol{v} + \sum_{e=1}^{E_{\mathrm{moe}}} g_e f_{r,e}(\boldsymbol{u}). \end{aligned} \tag{16}$$

Each group uses four experts, Top-2 routing, and temperature $\tau_r = 1$. The first group's experts have hidden width 2048 and GELU activations; the second group's experts have hidden width 512 and

Table 6: Training hyperparameters and model dimensions across the three simulation settings and physical ALOHA. The LIBERO-Plus column describes in-domain adaptation; generalization uses LIBERO-trained policies. World-loss weights are listed once per active camera view.

Hyperparameter	**LIBERO**	**LIBERO-Plus**	**Meta-World**	**Physical ALOHA**
Input resolution	224×224	224×224	224×224	224×224
Patch tokens per view	256	256	256	256
Active camera views	2	2	1	2
World tokens per view (K)	2	2	2	2
Total current world tokens	4	4	2	4
World-token width (d_z)	2048	2048	2048	2048
MoE attention heads	8	8	8	8
Experts per query group / selected	$4/2$	$4/2$	$4/2$	$4/2$
Action/state representation width	32	32	32	32
Action-expert parameters	300M	300M	300M	300M
Prediction length (H)	10	10	10	30
Parameter adaptation	LoRA	LoRA	LoRA	LoRA
LoRA rank / alpha (backbone)	$16/16$	$16/16$	$16/16$	$16/16$
LoRA rank / alpha (action expert)	$32/32$	$32/32$	$32/32$	$32/32$
LoRA target layers	Attn. + FFN	Attn. + FFN	Attn. + FFN	Attn. + FFN
Default training updates	30K	90K	30K	40K
Global batch size	32	32	32	32
Optimizer	AdamW	AdamW	AdamW	AdamW
AdamW β_1/β_2	$0.9/0.95$	$0.9/0.95$	$0.9/0.95$	$0.9/0.95$
AdamW ϵ / weight decay	$10^{-8}/10^{-10}$	$10^{-8}/10^{-10}$	$10^{-8}/10^{-10}$	$10^{-8}/10^{-10}$
Peak learning rate	2.5×10^{-5}	2.5×10^{-5}	2.5×10^{-5}	2.5×10^{-5}
Final learning rate	2.5×10^{-6}	2.5×10^{-6}	2.5×10^{-6}	2.5×10^{-6}
Learning-rate schedule	Cosine	Cosine	Cosine	Cosine
LR schedule endpoint (updates)	30K	30K	30K	30K
Warmup updates	1K	1K	1K	1K
Gradient-norm clip	1.0	1.0	1.0	1.0
Training seed	42	42	42	42
Loss weights (action/world)	$1.0/0.1/0.1$	$1.0/0.2/0.2$	$1.0/0.2$	$1.0/0.2/0.2$
Router-balance weight	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Temporal-router weight (λ_t)	0	0	0	0
Benchmark evaluation repeats	3	3	3	1

Table 7: MUX implementation details.

Property	**Linear summary**	**MoE refinement**
Patch aggregation	Learned scoring weights	Learned-query attention
Task conditioning	Shared across tasks	Task-specific routers
Expert refinement	None	Four experts; Top-2
Internal query groups	None	Two complementary groups
Representation size	$K \times d_z$	$K \times d_z$

Swish activations. Expert banks are separate between groups and shared across tasks, views, frames, and token slots within each group.

Normalized global fusion. Token-wise RMS normalization aligns the scales of the Linear and refined query-group summaries. Let a_L and $\{a_{M,r}\}_{r=1}^{2}$ be global learned coefficients, and define $S_M = \sum_{r=1}^{2} \exp(a_{M,r})$. Fusion combines the normalized MoE query-group summaries and fuses

the result with the normalized Linear summaries to form the final MUX output:

$$\begin{aligned} \eta_r &= \frac{\exp(a_{M,r})}{S_M}, & \overline{\boldsymbol{U}}_L &= \mathrm{RMS}(\boldsymbol{U}_L), \\ \pi_L &= \frac{\exp(a_L)}{\exp(a_L) + S_M}, & \overline{\boldsymbol{U}}_M &= \sum_{r=1}^{2} \eta_r \,\mathrm{RMS}(\boldsymbol{U}_{M,r}), \\ \pi_M &= 1 - \pi_L, & \boldsymbol{Z} &= \sum_{b\in\{L,M\}} \pi_b \overline{\boldsymbol{U}}_b. \end{aligned} \tag{17}$$

RMS normalization uses $\epsilon = 10^{-6}$. The coefficients are initialized as $a_L = -1$, $a_{M,1} = 1$, and $a_{M,2} = 0.5$, and are shared across tasks, samples, views, frames, and slots. The final representation remains $\boldsymbol{Z} \in \mathbb{R}^{K\times d_z}$.

Task identity and routing. Each instruction is associated with an operation-task identity m. For LIBERO-10, the ten task identities select ten router parameter sets within each MoE query group. Token content determines the Top-2 mixture weights, while the expert bank is shared across tasks. The operation-task identity accompanies the language instruction during evaluation. The four expert responses are evaluated before applying the Top-2 weights.

Prediction length and objective. Each training sequence contains a current observation and H future positions, with $H = 10$ for simulation and $H = 30$ for physical ALOHA. The action-loss weight is 1.0 in all settings. Each active view's world loss has weight 0.1 for LIBERO and 0.2 for LIBERO-Plus adaptation, Meta-World, and physical ALOHA. The action and per-view token losses average absolute velocity errors over Hd_a and HKd_z coordinates, respectively. The router-balance weight is 10^{-3} and the temporal-router weight is $\lambda_t = 0$ in all settings. Flow time follows $\tau = 0.001 + 0.999\zeta$ with $\zeta \sim \mathrm{Beta}(1.5, 1)$.

Router regularization. For learned-query group $r \in \{1, 2\}$ in MUX and active view $c \in \mathcal{C}$, let $\mathcal{I}^r_{m,c}$ index minibatch tokens associated with task m. The mean routing probability and Top-q selection load are

$$\bar{p}^r_{m,c,e} = \frac{1}{|\mathcal{I}^r_{m,c}|} \sum_{i\in\mathcal{I}^r_{m,c}} \mathrm{softmax}(\boldsymbol{\rho}_i^{c,r})_e, \tag{18}$$

$$\bar{l}^r_{m,c,e} = \frac{1}{|\mathcal{I}^r_{m,c}|} \sum_{i\in\mathcal{I}^r_{m,c}} \mathbf{1}\{e \in \mathrm{Top}_q(\boldsymbol{\rho}_i^{c,r})\}. \tag{19}$$

With $\mathrm{CV}^2(\boldsymbol{v}) = \mathrm{Var}_e(v_e)/(\mathrm{Mean}_e(v_e)^2 + \epsilon_{\mathrm{cv}})$ and $\epsilon_{\mathrm{cv}} = 10^{-10}$, the balance penalty is

$$\mathcal{R}_{\mathrm{bal}} = \sum_{r=1}^{2} \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \frac{1}{|\mathcal{M}_{\mathcal{B}}|} \sum_{m\in\mathcal{M}_{\mathcal{B}}} \left[\mathrm{CV}^2(\bar{\boldsymbol{p}}^r_{m,c}) + \mathrm{CV}^2(\bar{\boldsymbol{l}}^r_{m,c})\right], \tag{20}$$

where $\mathcal{M}_{\mathcal{B}}$ contains the task identities present in the minibatch. We compute each view's coefficient of variation, average over tasks and views, and sum over the two query groups within MoE. The logits $\boldsymbol{\rho}$ include the router temperature in equation 8. Together with the action and world-token losses, this balance penalty gives equation 13 in all training settings.

A.2 Physical-Robot Protocol

Table 4 in the main text reports the per-task physical success rates and their condition-wise averages for π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), Cosmos Policy (Kim et al., 2026), and MuxWAM. The protocol below specifies the MuxWAM evaluation.

Hardware and sensing. Figure 5 shows the dual-arm AgileX Piper and Cobot Magic ALOHA platforms documented for physical evaluation. The reported control protocol uses a stationary bimanual setup (Zhao et al., 2023). For phone assembly, our custom mechanical setup combines a left-arm suction pump with a right-arm gripper. A fixed Intel RealSense D435 RGB camera provides the base view, while two additional D435 cameras provide the left-wrist and right-wrist views. The



Figure 4: **Additional real-robot rollouts with MuxWAM.** Representative banana-grasping sequences under normal observations (top) and colored-light interference (bottom). Frames progress from left to right.

Table 8: Physical-robot data volume and task success criteria. The main-text comparison uses banana grasping, clothes folding, and phone assembly.

Physical task	Train/val. demos	Time limit	Success predicate
Grasp banana	50/20	60 s	The robot places the banana on the plate.
Stack cube	80/20	90 s	The upper cube rests only on the lower cube.
Open drawer	100/20	90 s	Open the drawer and successfully insert the cube.
Fold clothes	200/20	180 s	Keep the stacked clothes neat and tidy.
Assemble phone	200/20	180 s	The right-arm gripper places the data cable in the box; the left-arm suction pump then places the phone case in the box and closes it with the lid.

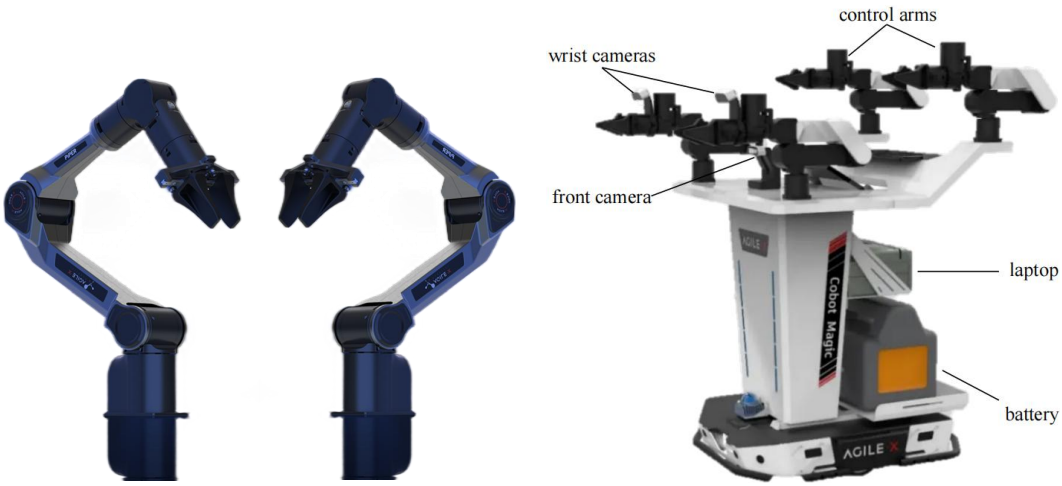
policy and compressor receive the base and left-wrist streams; the right-wrist stream is retained for visualization only.

The cameras capture 640×480 RGB observations at 30 Hz, and we disable depth. We synchronize all three streams using host timestamps, center-crop each frame to a square, and resize it to 224×224 before SigLIP encoding.

Control and chunk execution. The robot servo loop runs at 50 Hz, while observations and policy targets run at 10 Hz. Each action is a 14-dimensional absolute target containing six joint positions and one gripper command for each arm; we zero-pad the vector to the model dimension of 32. A prediction horizon of $H = 30$ therefore spans 3 s. At each policy query, we execute the first five targets over 0.5 s, linearly interpolate them at 50 Hz, acquire a new synchronized observation, and replan the remaining motion. Each replan discards the remaining 25 unexecuted targets, with temporal ensembling disabled. Before interpolation, we clip joint increments to 0.12 rad per policy step, gripper increments to 8% of full travel, and all targets to the manufacturer's soft limits.

Data collection and evaluation. We keep the camera mounts and viewpoints fixed and place objects within the pose range used for training demonstrations. We collect ALOHA joint-space teleoperation at 50 Hz and subsample it to 10 Hz after anti-aliasing filtering. We define the validation split using complete trajectories, with all validation frames excluded from training. For physical evaluation, we use one fixed MuxWAM checkpoint from a single training run. We select the checkpoint before testing based on validation performance and do not change it according to physical test outcomes. At evaluation time, we independently sample object positions within a ± 4 cm planar range and yaw angles within $\pm 15^\circ$ of the nominal reset. For the three main-text tasks, we conduct 20 independent trials per task under each evaluation condition, resulting in 60 trials per condition and 180 trials per MuxWAM. Reference-baseline rows follow their source-specific protocols. We record success only when the task-specific predicate in Table 8 remains satisfied continuously for 2 s; operator judgment cannot override the predicate.

Failure and safety termination. A trial fails when it reaches its time limit, an object leaves the reachable tabletop, an irreversible wrong assembly occurs, or recovery requires a manual reset. The safety supervisor stops the robot when a human enters the marked workspace or an unexpected self-



(a) Agilex Piper (b) ALOHA

Figure 5: Hardware platforms for physical evaluation. (a) Dual-arm AgileX Piper. (b) Cobot Magic ALOHA with two follower arms, two leader arms, a front camera, and wrist cameras.

or table-collision occurs. It also stops on entry into the 5° joint-limit buffer, joint speed above 1.5 rad/s, or motor current above 80% of its rated continuous value for 0.25 s. Each safety-stopped trial remains in the failure denominator without retry or replacement.

A.3 LIBERO-Plus Results and Aggregation

For the LIBERO-trained results in Table 9, within-suite averages and category summaries across suites use trial-count weighting. The summary Avg. in that table and the MuxWAM Avg. in Table 1 instead equally average the seven category scores, yielding 81.1% for generalization. The Original entry for MuxWAM uses its standard LIBERO result in Table 2; comparison models retain their source-reported standard LIBERO results.

The in-domain adaptation results cover all four LIBERO-Plus suites, with seven-category arithmetic means of 91.8%, 92.5%, 81.2%, and 88.9% for Spatial, Object, Goal, and Long, respectively. For the in-domain MuxWAM row in Table 1, each category is averaged equally across the four suites; the mean of these seven category averages is 88.6%. Aggregation uses the supplied precision before rounding for display. These adapted results remain separate from the generalization results and their ranking. The Fast-WAM and SG-WAM values in Table 1, including their source-reported Avg. entries, follow Zhao et al. (2026). The VLA-JEPA values, including its reported Avg., follow Sun et al. (2026). DreamWAM uses the joint video–action inference results and Original score of Yuan et al. (2026a). The remaining comparison values, apart from $\pi_{0.5}$, follow (Fei et al., 2026; Zhong et al., 2026), retaining their variants and evaluation scopes. The $\pi_{0.5}$ rows use separate reported evaluations with Original success of 96.9% in both training settings. They are not reruns under our aggregation protocol.

Task identity and evaluation scope. The source report separates LIBERO-trained results from Plus-adapted results. During training, each operation task has a fixed identity m. Standard LIBERO evaluation supplies this task identity alongside the language instruction to select the corresponding router. The task identity is independent of the instruction wording. The Plus-specific variant-to-original-task mapping and run-level adaptation records are not available in the supplied evaluation summary. Consequently, the Language column is not evidence of stronger language understanding or unseen-task generalization; interpreting it as robustness with original-task routing requires that mapping to be verified. The seven single-axis evaluations also do not establish unseen-embodiment or combined-shift generalization.

Table 9: Detailed MuxWAM robustness and generalization on LIBERO-Plus. Entries are success rates (%). For LIBERO-trained results, suite averages and category summaries use trial-count weighting; the summary Avg. equally averages the seven category scores. In-domain Avg. entries are seven-category macro averages (†); the in-domain summary averages each category equally across the four suites.

Task	Camera	Robot	Language	Light	Background	Noise	Layout	Avg.
LIBERO-Spatial	93.9	53.4	98.2	98.3	93.8	91.5	86.8	87.7
LIBERO-Object	81.8	49.5	99.4	100.0	99.2	89.1	75.7	83.3
LIBERO-Goal	75.2	44.0	96.8	96.8	86.1	75.7	67.3	76.0
LIBERO-Long	55.6	51.9	95.3	86.9	82.7	72.2	81.4	73.7
Average	76.1	49.5	97.4	95.6	90.1	81.7	77.3	81.1
Trained on LIBERO-Plus (In-domain Evaluation)								
LIBERO-Plus-Spatial	98.40	59.71	97.44	99.66	98.84	97.15	91.43	91.80†
LIBERO-Plus-Object	99.49	63.31	99.44	99.66	99.60	87.10	99.05	92.52†
LIBERO-Plus-Goal	90.4	45.2	88.5	93.2	93.6	90.2	67.3	81.2†
LIBERO-Plus-Long	96.9	58.5	97.1	95.3	95.5	94.7	84.3	88.9†
Average (suite-equal)	96.3	56.7	95.6	97.0	96.9	92.3	85.5	88.6†

A.4 Perturbation Protocol

Simulation perturbations. Simulation perturbation evaluations use the official LIBERO-Plus variants (Fei et al., 2026). Tables 1 and 9 report the seven benchmark categories. Appendix A.5 specifies the same-state image pairing and official variant selection used for token-drift measurements.

Physical lighting. The clean physical workspace measures 550 ± 50 lux at the table center with a 5000 K white point. Two dimmable LED panels vary illuminance between 250 and 850 lux and correlated color temperature between 2700 and 6500 K; each episode samples both endpoints and linearly traverses between them over 8 s.

Physical background changes. In the physical setup, we uniformly select one of six printed rear panels and one of three table skirts, producing 18 combinations. The panels remain outside the robot workspace and preserve contact geometry, object poses, and illumination.

A.5 Latent Stability and Ablation Protocol

Cosine error and drift. Success measures control, while future-token cosine error and its paired change measure prediction against encoded observations. For each action chunk (chunk index omitted), let $\widehat{\boldsymbol{z}}^c_{i,h,k}$ denote predicted token k for episode i, view c, and future step h. We obtain target $\boldsymbol{z}^c_{i,h,k}$ from the frame actually reached after executing the predicted actions, using the checkpoint's SigLIP encoder, view-specific constructor, task ID m_i, and identical preprocessing in evaluation mode without random augmentation. We compute

$$e^c_{i,h} = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{\langle \widehat{\boldsymbol{z}}^c_{i,h,k}, \boldsymbol{z}^c_{i,h,k}\rangle}{\|\widehat{\boldsymbol{z}}^c_{i,h,k}\|_2 \|\boldsymbol{z}^c_{i,h,k}\|_2 + 10^{-8}}\right), \tag{21}$$

$$\Delta e^c_{i,h} = e^c_{i,h} - e^c_{i,1}. \tag{22}$$

Fixed slot indices through the joint flow rollout remove the need for Hungarian matching. We first compute paired differences within the same chunk, then average complete chunks within each episode, episodes within each task, and tasks equally. Within-chunk summaries average base and wrist errors equally after averaging slots. Mean absolute error (MAE) averages $|\widehat{z} - z|$ over token slots and latent coordinates; recursive diagnostics report the two views separately.

Prediction and recursive evaluation. The diagnostics use Object, Spatial, Goal, and Long (LIBERO-10), with ten tasks per suite and the fixed first ten episodes per task and condition. Each

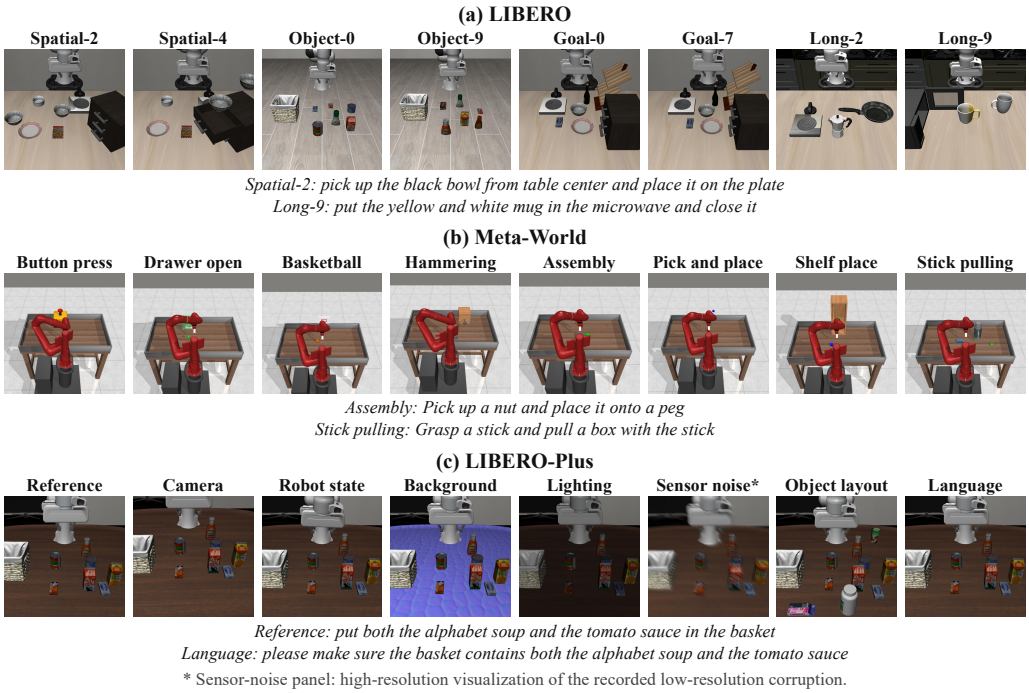


Figure 6: Simulation task coverage. Representative observations from (a) the LIBERO Spatial, Object, Goal, and Long suites; (b) Meta-World tasks; and (c) a LIBERO-Plus reference scene with seven perturbation categories. Example instructions appear beneath each group. The starred sensor-noise tile visualizes the recorded low-resolution corruption at higher resolution.

suite uses one checkpoint: 30,000 updates for Object, Spatial, and Goal, and 38,000 for Long. We fix $K = 2$, $H = 10$, and ten Euler evaluations per policy query. Each chunk executes ten actions before replanning. R1 refreshes real images every chunk; R8 refreshes them every eight chunks. Between R8 refreshes, the final predicted latent from the preceding chunk supplies the next visual prefix, while the actual robot state remains available. Target images are encoded only for diagnostics between refreshes.

R1 statistics retain 6,891 complete ten-step chunks and exclude 307 partial chunks from these diagnostics only; episode outcomes remain unchanged. For R8, refresh age counts executed actions since the last real-image refresh. Each episode first averages its available cycles at each age, then forms paired error differences between ages 20, 40, or 80 and age 10. All 100 episodes per suite contribute paired differences. Available cycle sets may differ across ages; these are differences of age-specific means, not comparisons restricted to a common cycle set or cumulative sums.

Episode bootstrap and outcome groups. We use 10,000 within-task episode-bootstrap replicates (seed 20260923 for R1 and 20260922 for R8), preserving paired steps, views, and ages within each sampled episode. Tasks retain equal weight and are not resampled; checkpoints are not resampled. Intervals are the 2.5th and 97.5th percentiles and do not include training-seed uncertainty. Outcome comparisons use tasks containing both outcomes, with equal task weights in each group: Spatial has two such tasks (18 successful/two failed episodes), Goal four (23/17), and Long eight (47/33). Object shows its 100 successful episodes descriptively, with the unavailable failure bar left empty. Spatial's outcome bars have no intervals because only two failures are available. Outcome differences are exploratory associations, not causal effects of prediction accuracy on success.

Prediction-length and token-budget ablations. Table 10 organizes the ablations by the Spatial, Object, Goal, and Long suites. The prediction-length sweep uses $H \in \{5, 10, 15, 20\}$ at fixed $K = 2$; the token-budget sweep uses $K \in \{1, 2, 4, 8\}$ at fixed $H = 10$. The $H = 10$, $K = 2$ reference point appears in both column groups. At this reference setting, all four suites reuse the MuxWAM results in Table 2. The default LIBERO training schedule uses 30K updates and batch size 32.

Table 10: Prediction-horizon and token-capacity ablations on the four LIBERO suites, measured by success rate (%). Horizon ablations fix $K = 2$; token ablations fix $H = 10$. The $H = 10$ and $K = 2$ columns show the same reference setting, using the MuxWAM results for all four suites in Table 2. The sampler uses H Euler evaluations, so varying H changes both prediction length and solver compute.

Suite	**Prediction horizon H**				**World tokens per view K**			
	5	10	15	20	1	2	4	8
LIBERO-Spatial	98.6	98.8	98.2	98.8	97.0	98.8	98.2	96.8
LIBERO-Object	99.6	100.0	99.0	100.0	98.6	100.0	99.0	98.0
LIBERO-Goal	96.0	97.8	98.2	97.6	97.8	97.8	97.0	97.2
LIBERO-Long	95.8	98.4	98.2	97.6	96.0	98.4	98.2	95.4

Parameter accounting. MUX uses $K = 2$ by default. With two active views, increasing K by one adds $2(d_z + 1)$ parameters to the Linear branch's view-specific scoring heads and $2d_z$ parameters to the two learned-query groups within MoE. At $d_z = 2048$, this totals $4d_z + 2 = 8194$ parameters. Expert banks and task routers do not scale with K. The count assumes view-specific Linear heads and shared MoE queries. Changing H introduces no learned prediction-position parameters: flow time uses a fixed sinusoidal embedding, while the transformer uses rotary position embeddings.

Full-model perturbation and branch ablations. Table 5(a) reports full-model perturbation drift, success drops, and clean success across four suites. Panel (b) reports the branch-ablation results for w/o Linear and w/o MoE. The full-model clean suite results reproduce Table 2. The full-model Long entry also provides the task-conditioned mean in the routing comparison below.

In panel (a), Succ. reports clean-task success (%). For each suite, the full-model Drop is the reported clean success minus the corresponding Light, Background, or Noise success in Table 9, in percentage points. The three categories are reported separately; negative differences are retained. These success rates reuse the existing control evaluations; the 1,200-pair image diagnostic measures token drift. Background denotes the official LIBERO-Plus background-texture category.

For suite s and perturbation p (illumination, background texture, or sensor noise), let $\boldsymbol{z}_{i,k}^{c,G} = G(E_\phi(\boldsymbol{x}_i^c); m_i, c)_k$ and $\widetilde{\boldsymbol{z}}_{i,k}^{c,G} = G(E_\phi(\widetilde{\boldsymbol{x}}_{i,p}^c); m_i, c)_k$ be tokens from paired clean and perturbed observations under fixed encoder, constructor, preprocessing, task ID, view, and slot correspondence. Define

$$D_{\text{pert}}^{s,p}(G) = \frac{1}{|\mathcal{I}_{s,p}|} \sum_{(i,c,k)\in\mathcal{I}_{s,p}} \left(1 - \frac{\langle \boldsymbol{z}_{i,k}^{c,G}, \widetilde{\boldsymbol{z}}_{i,k}^{c,G} \rangle}{\|\boldsymbol{z}_{i,k}^{c,G}\|_2 \|\widetilde{\boldsymbol{z}}_{i,k}^{c,G}\|_2 + 10^{-8}}\right), \tag{23}$$

where $\mathcal{I}_{s,p}$ contains valid matched observation–view–slot triples. We average slots, active views, ten fixed states per task, and ten tasks with equal weights. This clean–perturbed representation metric differs from future-token prediction error.

The full-model diagnostic uses the first ten archived initial states of each task in all four suites, giving 1,200 clean–perturbed pairs across three LIBERO-Plus categories. Object, Spatial, and Goal use 30K-update checkpoints; Long uses the specified 38K-update checkpoint. Official lighting and background-texture variants are selected before encoding, using a fixed seed and round-robin sampling across available difficulty levels. When fewer than ten variants exist, the fixed order repeats across distinct initial states. Both renders receive exactly the same simulator state; body poses, physical parameters, camera parameters, and proprioception are checked for equality without executing an action or physics step. The encoder and constructor process both inputs in evaluation mode with identical parameters and preprocessing.

The official sensor-noise category comprises motion, Gaussian, zoom, and glass blur and fog, with the selected official severity. It affects only the base camera; the wrist image is unchanged. We apply this official postprocessing to the paired raw base render before the original image flip and resize. Table 5(a) averages both views; the archive also reports each view separately. Repeated clean encodings must be bitwise equal. The raw images, latent arrays, physical state identifiers, selected

perturbations, and evaluation-version records are retained. Drift for the other two constructors is left unmeasured.

A.6 Task-Router Comparison

Conditions and evaluation. Table 5(c) compares task-conditioned and shared routing on LIBERO-Long. The task-conditioned router selects its parameters using the operation task ID. Each shared router instead uses one content-based parameter set across tasks, with separate expert banks for the two query groups within MoE. The fixed $K = 2$ interface, four experts per group, Top-2 weighting, and Linear–MoE constructor are retained. Each condition uses a fixed checkpoint across all ten tasks and evaluates 50 trajectories per task, for 500 trajectories per condition. Task IDs 0–9 follow the default LIBERO ordering.

Aggregation and results. We compute each task's success rate from its 50 trajectories and take the equal-weight mean over all ten tasks. Task-conditioned routing succeeds on 492 of 500 trajectories (98.4%); shared routing succeeds on 464 of 500 (92.8%), giving a 5.6-percentage-point difference. The table includes all ten tasks. Its task-conditioned mean is also the LIBERO-Long entry in the main benchmark table. These control results are separate from the same-state perturbation measurements in panel (a).

A.7 Reference Systems and Comparison Scope

Tables 2 and 3 collate reported reference results, not uniformly retrained systems. Each row cites the method; the numerical sources below identify evaluations reported by other works where applicable. Training data, inputs, model size, adaptation, and trial budgets differ. Column ranks are descriptive and do not establish matched causal gains. The MuxWAM rows retain the author's reported values. The separately evaluated routing comparison is described in Appendix A.6.

LIBERO numerical sources and variants. OpenVLA, π_0, and OpenVLA-OFT follow Table I of Kim et al. (2025a). The $\pi_{0.5}$ row follows Table 1 of Chen et al. (2026). WorldVLA uses 512×512 inputs (Cen et al., 2025b); the RynnVLA-002 row uses the continuous-action head reported in its official model zoo (Cen et al., 2025a). The Fast-WAM, Motus, and LaWAM rows follow Table 1 of Chen et al. (2026). LingBot-VA's success rates and total parameter count follow its published results and architecture description (Li et al., 2026a). VLA-Adapter denotes the original published model (Wang et al., 2026b). ResVLA jointly trains on the four suites from scratch for 30K steps (Zhong et al., 2026). Cosmos Policy reports 500 trials per suite over three seeds (Kim et al., 2026). The parameter column uses reported model sizes where available. The approximate entries for WorldVLA and RynnVLA-002 identify their 7B backbones; ResVLA uses a 2B backbone; MuxWAM's parameter entry is a backbone-scale estimate.

Meta-World metric and numerical sources. The four difficulty tiers contain 28, 11, 6, and 5 tasks. Table 3 uses the unweighted mean of tier success rates, not the task-weighted mean of all 50 tasks. Task coverage follows each source, including the 48-task GR-1 and PAD evaluations described below. TinyVLA-H uses Table I of Wen et al. (2025). The robot-pretrained π_0 row is the evaluation in Table 2 of Shukor et al. (2025), which also supplies SmolVLA-2.25B. InternVL3-1B is the D2E study's action-policy baseline, and D2E is its no-pseudo-label pretraining variant; both follow Table 8 of Choi et al. (2026). MINT-4B and Evo-1 follow their own main result tables (Huang et al., 2026; Lin et al., 2026). The $\pi_{0.5}$ four-tier values are the author's supplied evaluation. GR-1 and PAD use the per-task results in Table 5 of Guo et al. (2024), including that study's evaluation of GR-1. We average these results within the difficulty groups defined in Appendix F of Seo et al. (2023), then average the four group means equally. The source excludes the two handle-pull tasks, so these rows cover 26/11/6/5 tasks across the four groups, or 48 tasks in total. This calculation reproduces the displayed 50.5% and 72.7% macro averages. The parameter counts round the 195M total parameters reported for GR-1 (Wu et al., 2024, Appendix A.1) and the 661M model size reported for PAD (Guo et al., 2024, Table 4). Parameter entries use the reported model size or, where marked, the backbone-scale estimate.

These sources share the four-tier metric but not every evaluation detail: TinyVLA uses three seeds with multiple training-stage evaluations; SmolVLA uses 10 trials per task; D2E reports 50K training

steps with LeRobot v0.4.1; and Evo-1 uses 50 demonstrations per task and 10 trials per task over five runs. We therefore do not pool uncertainty or infer a matched-budget advantage. MT10 scores, point-cloud-only comparisons, and task-weighted averages are not substituted for the four-tier macro average. ResVLA does not report Meta-World results in its cited paper and is not assigned an estimated score.

A.8 Theoretical Results and Proofs

This appendix proves the low-rank result in Section 3.1, then states and proves the codec-cycle error bound.

A.8.1 Proof of Low-Rank Task Sufficiency

Proof of Proposition 1. Fix an observed task m. Because $r_m \le K$, extend the r_m rows of $\boldsymbol{U}_m^\top$ to a K-row matrix $\boldsymbol{M}_m$. Equivalently, choose $\boldsymbol{M}_m$ whose row space contains $\mathrm{row}(\boldsymbol{U}_m^\top)$. There is therefore a matrix $\boldsymbol{B}_m \in \mathbb{R}^{r_m \times K}$ satisfying $\boldsymbol{U}_m^\top = \boldsymbol{B}_m \boldsymbol{M}_m$. For $\boldsymbol{Z}_m = \boldsymbol{M}_m \boldsymbol{P}$,

$$\boldsymbol{B}_m \boldsymbol{Z}_m = \boldsymbol{B}_m \boldsymbol{M}_m \boldsymbol{P} = \boldsymbol{U}_m^\top \boldsymbol{P} = T_m(\boldsymbol{P}).$$

Hence $T_m(\boldsymbol{P})$ is a deterministic function of $\boldsymbol{Z}_m$. The assumed conditional independence gives

$$p(\boldsymbol{R}_t \mid \boldsymbol{P}, \ell, m, \boldsymbol{s}_t) = p(\boldsymbol{R}_t \mid T_m(\boldsymbol{P}), \ell, m, \boldsymbol{s}_t)$$

and the tower property gives

$$\begin{aligned} p(\boldsymbol{R}_t \mid \boldsymbol{Z}_m, \ell, m, \boldsymbol{s}_t) &= \mathbb{E}[p(\boldsymbol{R}_t \mid \boldsymbol{P}, \ell, m, \boldsymbol{s}_t) \mid \boldsymbol{Z}_m, \ell, m, \boldsymbol{s}_t] \\ &= p(\boldsymbol{R}_t \mid T_m(\boldsymbol{P}), \ell, m, \boldsymbol{s}_t), \end{aligned}$$

because $T_m(\boldsymbol{P}) = \boldsymbol{B}_m \boldsymbol{Z}_m$. Thus $\boldsymbol{Z}_m$ is task-sufficient. A predictor that observes $\boldsymbol{Z}_m$ can recover the sufficient statistic and attain the same conditional Bayes decision as a predictor that observes $\boldsymbol{P}$; conversely, $\boldsymbol{Z}_m$ is a function of $\boldsymbol{P}$, so it cannot provide a lower Bayes risk than $\boldsymbol{P}$. The two optimal risks are equal conditional on $(\ell, m, \boldsymbol{s}_t)$. Applying the construction to every observed task proves the proposition. Finally,

$$\|T_m(\boldsymbol{P}) - \boldsymbol{B}_m \boldsymbol{Z}_m\|_F = \|(\boldsymbol{U}_m^\top - \boldsymbol{B}_m \boldsymbol{M}_m)\boldsymbol{P}\|_F \le \|\boldsymbol{U}_m^\top - \boldsymbol{B}_m \boldsymbol{M}_m\|_2 \|\boldsymbol{P}\|_F,$$

which proves equation 6. □

A.8.2 The Codec-Cycle Error Channel

Consider a pixel-feedback WAM with encoder E, transition F, and decoder D. Its recurrent decode–re-encode loop is

$$\begin{aligned} \boldsymbol{z}_h &= E(\boldsymbol{x}_h), \qquad \widehat{\boldsymbol{z}}_{h+1} = F(\widetilde{\boldsymbol{z}}_h, \widehat{\boldsymbol{a}}_h), \\ \widehat{\boldsymbol{x}}_{h+1} &= D(\widehat{\boldsymbol{z}}_{h+1}), \quad \widetilde{\boldsymbol{z}}_{h+1} = E(\widehat{\boldsymbol{x}}_{h+1}) = (E \circ D)(\widehat{\boldsymbol{z}}_{h+1}). \end{aligned} \tag{24}$$

For codec cycle $\mathcal{T} = E \circ D$, define latent and action errors e_h, e_h^a, dynamics error η_h, and codec-cycle residual κ_h:

$$\begin{aligned} e_h &= \|\widetilde{\boldsymbol{z}}_h - \boldsymbol{z}_h\|, & e_h^a &= \|\widehat{\boldsymbol{a}}_h - \boldsymbol{a}_h\|, \\ \eta_h &= \|F(\boldsymbol{z}_h, \boldsymbol{a}_h) - \boldsymbol{z}_{h+1}\|, & \kappa_h &= \|\mathcal{T}(F(\boldsymbol{z}_h, \boldsymbol{a}_h)) - F(\boldsymbol{z}_h, \boldsymbol{a}_h)\|. \end{aligned} \tag{25}$$

Assumption 2 (Lipschitz feedback dynamics)**.** *The codec cycle $\mathcal{T}$ is $L_{\mathcal{T}}$-Lipschitz, and F is L_z-Lipschitz in its latent input and L_a-Lipschitz in its action input.*

Proposition 2 (Pixel-cycle error bound)**.** *Under Assumption 2,*

$$e_{h+1} \le a_\star e_h + L_{\mathcal{T}} L_a e_h^a + \eta_h + \kappa_h, \qquad a_\star = L_{\mathcal{T}} L_z. \tag{26}$$

After H rollout steps,

$$e_H \le a_\star^H e_0 + \sum_{j=0}^{H-1} a_\star^{H-1-j} \left(L_{\mathcal{T}} L_a e_j^a + \eta_j + \kappa_j \right). \tag{27}$$

If $\delta_j = L_{\mathcal{T}} L_a e_j^a + \eta_j + \kappa_j \le \bar{\delta}$, then

$$e_H \le \begin{cases} a_\star^H e_0 + \bar{\delta} \dfrac{1 - a_\star^H}{1 - a_\star}, & a_\star \ne 1, \\ e_0 + H\bar{\delta}, & a_\star = 1. \end{cases} \tag{28}$$

Each codec residual enters subsequent updates through powers of $a_\star$. Under the uniform bound, the upper bound is linear in H at $a_\star = 1$, geometric at $a_\star > 1$. The proof follows.

Proof of Proposition 2. Using the definitions in equation 25, add and subtract $\mathcal{T}(F(\boldsymbol{z}_h, \boldsymbol{a}_h))$ and $F(\boldsymbol{z}_h, \boldsymbol{a}_h)$:

$$
\begin{aligned}
e_{h+1} &= \|\mathcal{T}(F(\widetilde{\boldsymbol{z}}_h, \widehat{\boldsymbol{a}}_h)) - \boldsymbol{z}_{h+1}\| && (29)\\
&\le \|\mathcal{T}(F(\widetilde{\boldsymbol{z}}_h, \widehat{\boldsymbol{a}}_h)) - \mathcal{T}(F(\boldsymbol{z}_h, \boldsymbol{a}_h))\| && (30)\\
&\quad + \|\mathcal{T}(F(\boldsymbol{z}_h, \boldsymbol{a}_h)) - F(\boldsymbol{z}_h, \boldsymbol{a}_h)\| && (31)\\
&\quad + \|F(\boldsymbol{z}_h, \boldsymbol{a}_h) - \boldsymbol{z}_{h+1}\| && (32)\\
&\le L_{\mathcal{T}} \left(L_z e_h + L_a e_h^a\right) + \kappa_h + \eta_h. && (33)
\end{aligned}
$$

This is equation 26. Repeated substitution yields

$$
e_H \le a_\star^H e_0 + \sum_{j=0}^{H-1} a_\star^{H-1-j} \left(L_{\mathcal{T}} L_a e_j^a + \eta_j + \kappa_j\right), \qquad a_\star = L_{\mathcal{T}} L_z,
$$

which is equation 27. If $\delta_j = L_{\mathcal{T}} L_a e_j^a + \eta_j + \kappa_j \le \bar{\delta}$, substituting $\bar{\delta}$ and evaluating the geometric sum gives equation 28. □