DECOUPLED-WAM: DECOUPLING IMAGINATION AND ACTION FOR EFFICIENT AND FLEXIBLE WORLD ACTION MODEL

Anonymous authors
Paper under double-blind review

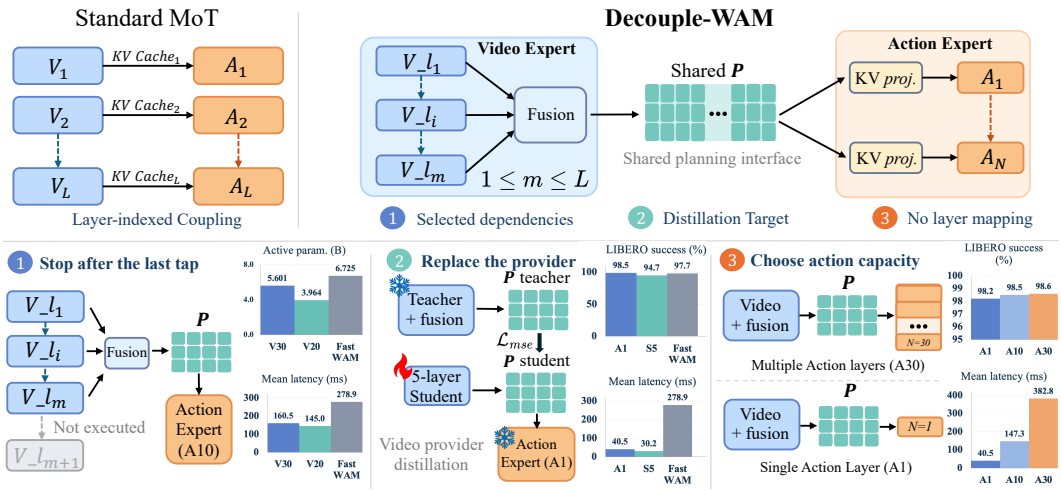

Figure 1: **Decoupled-WAM can adjust computation around a shared planning interface for efficient and flexible action generation.** *Top:* Standard Mixture-of-Transformers (MoT) exchanges layer-indexed KV caches between matched video and action blocks. Decoupled-WAM instead fuses selected video-layer outputs into one dense, fixed-shape planning feature $\mathbf{P}$; each action block constructs its own visual KV projections from $\mathbf{P}$, removing prescribed layer correspondence. *Bottom:* Decoupled-WAM exposes three compute controls: *(1)* Video Expert only need to execution m layer (V30: $m = 30$, V20: $m = 20$) at inference; *(2)* replace the Video Expert with a trainable five-layer student (S5) that matches the frozen teacher's inference-step interface; and *(3)* choose Action Expert depth through separately trained N-layer Action Expert (A1:$N = 1$, A10:$N = 10$, and A30:$N = 30$).

ABSTRACT

World Action Models (WAMs) transfer temporal and physical priors from video generative models to robot control. As future-video generation becomes less central at inference time, the representation transferred from the video model to the action policy becomes a key architectural question. Existing WAMs typically pass video-side attention states through layer-wise interfaces, binding the Action Expert to the internals of a particular video model. We introduce Decoupled-WAM, which instead places the video–action boundary at a shared dense planning feature. The Video Expert combines a design-time subset of intermediate layer outputs into a fixed-shape representation, while every Action Expert block constructs its own visual keys and values to read that representation through gated cross-attention. This interface supports Action Expert capacity variation, early termination after the deepest selected video tap, and replacement of the Video Expert while retaining a trained Action Expert. On LIBERO, ten-layer and one-layer Action Experts achieve 98.5% and 98.2% success, respectively; the one-layer model operates at 40.5 ms per action query and 24.7 queries/s. It further reaches 75.3% on LIBERO-Plus, while the default model obtains 91.9% on RoboTwin 2.0. With the one-layer Action Expert frozen, a five-layer student Video Expert retains 94.7% LIBERO success compared with 98.2% for its paired teacher. Retention

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.

is substantially weaker under distribution shifts, showing that interface regression does not preserve all aspects of the teacher's closed-loop behavior. These results characterize the capabilities and current limits of the shared interface.

1 INTRODUCTION

Video generative models offer a source of temporal and physical priors for robotic manipulation. World Action Models (WAMs) connect these models to action generation, allowing control policies to benefit from video modeling (Bi et al., 2026; Li et al., 2026b; Ye et al., 2026). Recent work suggests that much of this benefit can be retained without iterative future-video generation at inference time (Yuan et al., 2026). This shifts attention toward the representation transferred from the video model to the action policy, and toward the architectural commitments encoded by that transfer.

Existing WAMs commonly expose video-side attention states to the action stream through layer-wise Mixture-of-Transformer (MoT) interfaces (Bi et al., 2026; Li et al., 2026b; Ye et al., 2026). Recent docking architectures have already relaxed one-to-one depth correspondence; for example, DoT-Faster connects a shallow Action Expert to fused video key–value representations (Ma et al., 2026a). We build on this progress but revisit the module boundary itself: *what should a video model expose to an action policy—its internal attention states, or a shared representation that the action policy learns to read?* This distinction matters when changing either side of the system. Training policies with different Action Expert depths does not by itself establish that an existing Action Expert can use a replacement Video Expert.

We introduce Decoupled-WAM, which places the video–action boundary at a single shared planning feature. Rather than exporting video-native attention caches, the Video Expert combines a design-time subset of its intermediate layer outputs into one dense representation of fixed shape. Every Action Expert block reads that same representation, constructing its own visual keys and values through dedicated projections before attending to it with gated cross-attention. The interface contract is therefore the delivered representation itself. For a fixed observation configuration, its shape is independent of both the number of tapped video layers and the number of Action Expert blocks; the cost of reading it remains controlled by the chosen Action Expert.

This boundary supports two distinct operations on the Video Expert side. First, once the deepest tapped layer has been computed, the shared representation is complete, so later video layers can be omitted at inference time without changing the interface consumed by the Action Expert. Second, a shallower student Video Expert can be trained to approximate the representation delivered by the teacher while the trained Action Expert remains frozen. Early termination is a structural consequence of the selected feature source, whereas Video Expert replacement depends on how accurately the student reproduces the interface and is therefore an empirical question.

Our experiments examine Action Expert capacity variation, interface-only video distillation, feature-source selection, and deployment efficiency. The default model achieves 98.5% success on LIBERO and 91.9% on RoboTwin 2.0. Reducing the Action Expert to one layer retains 98.2% LIBERO success at 40.5 ms and 24.7 queries/s, while reaching 75.3% under LIBERO-Plus perturbations. Starting from this one-layer policy, a five-layer student Video Expert retains 94.7% LIBERO success compared with 98.2% for its paired teacher, without updating the Action Expert. Its weaker retention under distribution shifts further shows that matching the shared feature does not preserve every aspect of the teacher's closed-loop behavior.

Our contributions are:

- We introduce a shared dense planning-feature interface that separates responsibilities across the video–action boundary: the Video Expert delivers reusable features, while each Action Expert block constructs the visual attention states needed to read them.
- We instantiate early termination after the deepest required video tap and interface-only Video Expert distillation with a frozen Action Expert, demonstrating provider replacement while identifying its robustness gap.
- We evaluate the resulting architecture on LIBERO, LIBERO-Plus, and RoboTwin 2.0, and characterize its action-capacity, efficiency, feature-source, and replacement trade-offs.

2 Related Work

Efficient Vision-Language-Action Policies. Vision-language-action (VLA) policies (Brohan et al., 2023; Kim et al., 2024) inherit broad visual and semantic priors from pretrained vision-language models, but their scale makes both adaptation and deployment expensive. SmolVLA (Shukor et al., 2025) reduces this burden with a compact VLA and an asynchronous inference stack for responsive action-chunk execution. VLA-Adapter (Wang et al., 2025) instead studies the bridge from vision-language representations to action and introduces a lightweight policy module with Bridge Attention, reducing dependence on large backbones and robot-data pretraining. Together, these works suggest that efficiency depends not only on compressing the perception backbone, but also on designing an effective interface to the action generator. We investigate the analogous interface between video-based planning and action generation in WAMs.

World Action Models for Robot Control. World Action Models augment control learning with video prediction, providing a route to learn temporal and physical structure from video supervision. Motus introduces a Mixture-of-Transformers (Liang et al., 2024) architecture that unifies understanding, video generation, and Action Experts (Bi et al., 2026; Ma et al., 2026b). LingBot-VA (Li et al., 2026b) jointly learns frame prediction and policy execution in a shared latent space with layer-wise MoT coupling, while DreamZero (Ye et al., 2026) scales a pretrained video diffusion model into a closed-loop policy and demonstrates transfer from video-only human and cross-embodiment data. These systems establish the Video Expert/Action Expert decomposition as a strong WAM design pattern, but typically couple the two streams through repeated layer-wise state exchange. Our work asks whether that depth-aligned communication is necessary.

Efficient World Action Models Architecture. Recent work attacks WAM cost at several locations. Light-WAM (Li et al., 2026c) uses a compact Video Expert, downsampled latent supervision, and a state-fusion action decoder; Efficient-WAM (Li et al., 2026a) transfers a smaller Video Expert, sparsifies video latents, and allocates fewer denoising steps to video than to actions. Fast-WAM (Yuan et al., 2026) removes test-time future imagination while retaining video-model pretraining benefits. Two concurrent designs further reduce cross-expert cost: DoT-Faster (Ma et al., 2026a) docks a shallow Action Expert to representations aggregated from all video layers, whereas Faster-WAM (Zhao et al., 2026) uses SparseMoT and Interval KV-Fusion to compute future context once and reuse selected interactions during action denoising. In contrast, Decoupled-WAM exposes selected video-layer outputs as a single fixed-width planning feature that is shared by independently scalable action blocks. This makes the cross-expert state explicit and reusable, enabling the Video Expert and Action Expert to be varied without reinstating layer-wise MoT coupling.

3 Background: World-Action Models

A language-conditioned manipulation policy observes the current scene, reads a task instruction, and emits a short chunk of future actions. World Action Models (WAMs) keep this control objective but train the policy together with an explicit future-visual prediction task, so that temporal evolution and physical interaction are also learned from video supervision (Pai et al., 2025; Kim et al., 2026; Yuan et al., 2026) through the diffusion transformer architecture (Peebles & Xie, 2023): a *Video Expert* models how the scene will evolve, and an *Action Expert* emits the action chunk while reading a visual representation produced by the video pathway.

Formally, at control step t, the policy model receives a visual observation o_t, a language instruction ℓ, and a proprioceptive state s_t, and predicts an action chunk $A_t = a_{t+1:t+H}$ of horizon H, whereas a conventional visuomotor policy directly models $p(A_t \mid o_t, \ell, s_t)$. Let E_v be a visual encoder, $z_t = E_v(o_t)$ the latent representation of the current observation, and $Z_t = z_{t+1:t+K}$ a sequence of future video latents. A generative WAM contains a Video Expert f_{vid} for modeling Z_t and an Action Expert f_{act} for modeling A_t. Both experts are conditioned on the current context $c_t = (z_t, \ell, s_t)$, while an internal visual representation R_t^v produced by the video pathway additionally conditions action generation:

$$p_\theta(A_t \mid o_t, \ell, s_t) = p_\theta(A_t \mid c_t, R_t^v). \tag{1}$$

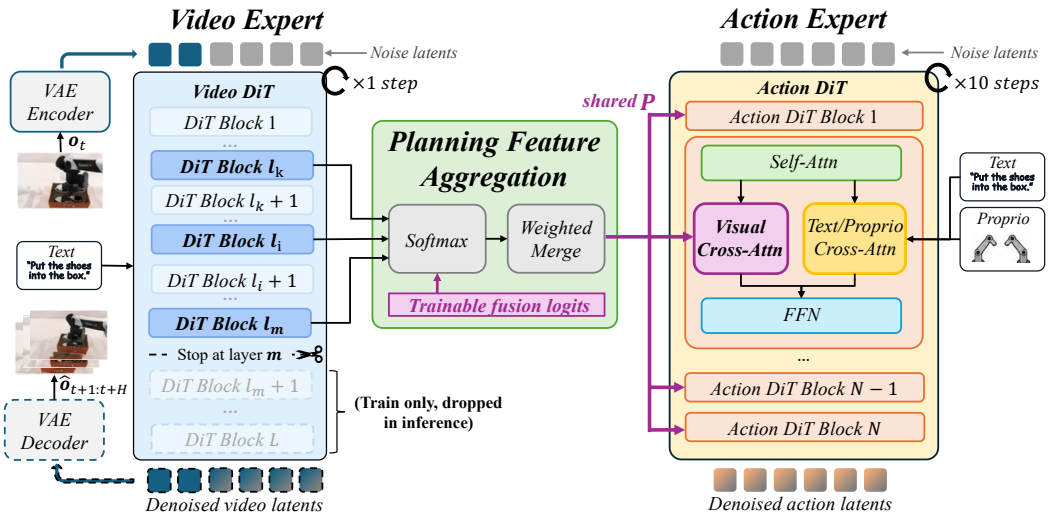

Figure 2: **Overview of Decoupled-WAM.** The Video Expert runs a single Video DiT pass over the encoded observation o_t, and a design-time tap set $\mathcal{S} \subseteq \{1, \ldots, L\}$ exposes same-shape features that Planning Feature Aggregation merges, under softmax-normalized trainable logits, into one fixed-shape Planning Feature $\mathbf{P}$. Blocks after the deepest tap $\max \mathcal{S}$ and the VAE decoder for $\hat{o}_{t+1:t+H}$ are train-only and dropped at inference. Every ActionDiT block reads the same $\mathbf{P}$ through its own visual cross-attention, in parallel with text and proprioceptive conditioning, and $\mathbf{P}$ is computed once per replan yet reused at every flow-integration step. This one-way interface replaces depth-aligned Video–Action coupling, so the two experts can be scaled or replaced independently.

The representation R_t^v may correspond to generated future latents, a current-frame world representation, or intermediate features of the Video Expert. In widely used MoT-based WAMs, video and action tokens exchange internal states repeatedly across aligned Transformer layers.

4 Method

4.1 Overview

Decoupled-WAM implements the video-side representation R_t^v introduced in Section 3 as a shared *Planning Feature* $\mathbf{P}$ that connects the Video Expert to the Action Expert (Figure 2). Conditioned on the current observation and language instruction, the Video Expert produces intermediate features at selected layers. Planning Feature Aggregation combines these features through a learned weighted sum into a single dense tensor, preserving the video-token grid. Each Action Expert block reads the same $\mathbf{P}$ through gated visual cross-attention, using its own key and value projections alongside language and proprioceptive conditioning. For a fixed observation configuration, the shape of $\mathbf{P}$ is independent of both the number of selected video layers and the number of action blocks, removing the need for layer-wise correspondence between the two experts.

We jointly train the two experts with video and action flow-matching objectives, allowing the action loss to update both the feature aggregation weights and the selected video representations. At inference, the Video Expert computes $\mathbf{P}$ once per replan and stops after the deepest selected layer. The Action Expert then reuses this feature throughout action flow integration, without executing the remaining video layers or decoding future frames. Section 4.2 details the feature construction, action-side reader, and joint training objective. Section 4.3 describes action-capacity variation and interface-only distillation for replacing the Video Expert while keeping a trained Action Expert frozen.

4.2 Constructing and Consuming the Planning Feature

Planning Feature Aggregation. Let z_t denote the latent encoding of the current observations and ϵ_f Gaussian future latent slots. At each replan, the Video Expert receives $X_v = [z_t; \epsilon_f]$ and exposes

the output of any design-time subset of layers $\mathcal{S} \subseteq \{1, \dots, L\}$:

$$F_l = V_\theta^{(1:l)}(X_v, t_v^{(1)}, \ell) \in \mathbb{R}^{B \times N_v \times d_v}, \qquad l \in \mathcal{S}. \tag{2}$$

Here $V_\theta^{(1:l)}$ denotes the first l blocks of the Video Expert, and $t_v^{(1)}$ the single video flow time queried at deployment. The future slots follow the deployed policy input and contain noise rather than ground-truth future observations. Because all tapped features have the same shape, G_ω can combine a variable number of sources without changing the interface delivered to the Action Expert. With trainable logits $\{w_l\}_{l \in \mathcal{S}}$, we use

$$\alpha_l = \mathrm{softmax}(\{w_k\}_{k \in \mathcal{S}})_l, \qquad \mathbf{P} = G_\omega(\{F_l\}_{l \in \mathcal{S}}) = \sum_{l \in \mathcal{S}} \alpha_l F_l. \tag{3}$$

The method defines $\mathcal{S}$ as an arbitrary subset; the tap locations used by our main instance are an empirical configuration reported in Section 5. The Video Expert communicates with the Action Expert only through $\mathbf{P} \in \mathbb{R}^{B \times N_v \times d_v}$, which preserves the dense video-token grid and, for a fixed observation configuration, has a shape independent of both $|\mathcal{S}|$ and the number of Action Expert blocks. Thus *fixed shape* denotes depth decoupling rather than token or channel compression.

Interface-conditioned Action Expert. The Action Expert is an ActionDiT with N blocks. Every block reads the same $\mathbf{P}$ through its own gated visual cross-attention module, in parallel with text and proprioceptive conditioning. Notating $\bar{x}_j$ for the post-self-attention state of block j,

$$\Delta x_j^{\mathrm{vis}} = g_j \odot \mathcal{C}_j(\bar{x}_j, \mathbf{P}), \qquad x_{j+1} = \mathrm{FFN}_j\Big(\bar{x}_j + \Delta x_j^{\mathrm{txt/prop}} + \Delta x_j^{\mathrm{vis}}\Big), \tag{4}$$

where g_j is a learned per-block residual gate and FFN_j the feed-forward sublayer of block j. Crucially, $\mathbf{P}$ crosses the Video–Action boundary as a dense feature tensor rather than as precomputed attention caches. The key and value projections used to read it are constructed inside each action block and are independently parameterized across blocks. Hence the blocks share *what* the Video Expert provides, while retaining control over *how* that representation is read. This removes the layer-indexed dependency between the two experts and lets Action Experts of different depth use the same interface form. The Video Expert forms $\mathbf{P}$ once per replan, after which the action flow reuses it throughout iterative refinement.

Joint training. Both experts are trained with conditional flow matching (Lipman et al., 2023). For a clean target x, either the future-video latents Z_t or the action chunk A_t, and noise $\epsilon \sim \mathcal{N}(0, I)$, we form the interpolated state $x^\tau = (1-\tau)x + \tau\epsilon$ with $\tau \in [0, 1]$, where $\tau = 0$ is data and $\tau = 1$ is noise, and regress a conditional velocity network v_θ onto the target velocity $\epsilon - x$; the video and action flow times are sampled independently. Writing $\mathcal{L}_{\mathrm{FM}}(x; c) = \mathbb{E}_{x,\epsilon,\tau}\big[\|v_\theta(x^\tau, \tau; c) - (\epsilon - x)\|_2^2\big]$, training minimizes the following combined losses:

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{FM}}(Z_t; c_t) + \mathcal{L}_{\mathrm{FM}}(A_t; c_t, \mathbf{P}), \tag{5}$$

which is the standard WAM training objective; the only change is the path taken by the action gradient, which passes through Eq. 3 to the selected video representations and the mixer.

4.3 What the Shared Interface Enables

The proposed Decoupled-WAM enables three operations that depth-aligned coupling does not support, each a consequence of the interface being a named object rather than a per-layer exchange: the Action Expert may be resized without changing the Video Expert, the Video Expert may be replaced while the Action Expert stays frozen, and video execution may stop once $\mathbf{P}$ is complete. This design enables more efficient and effective robot control. Below, we give the details of each function.

Action-capacity variation. Because all action blocks consume the same $\mathbf{P}$, the Action Expert depth need not match either the Video Expert depth or the number of selected taps. We instantiate one-layer and ten-layer Action Experts as separately jointly trained Video Expert–Action Expert systems. This tests whether the interface supports different action capacities; it does not assume that one frozen Video Expert has already been reused by those Action Experts.

Video Expert replacement with a frozen Action Expert. Because the interface is an explicit object, the composition $G_\omega \circ V_\theta$ becomes a direct regression target. A teacher Video Expert produces $\mathbf{P}^T$ under the deployed input construction, and a shallower student Video Expert produces $\hat{\mathbf{P}}$. We optimize the student using interface regression in the normalized coordinates read by the frozen Action Expert:

$$\mathcal{L}_P = \frac{1}{BN_v d_v} \sum_{b,n,d} \Big(\mathrm{LN}_0(\hat{\mathbf{P}})_{bnd} - \mathrm{LN}_0(\mathbf{P}^T)_{bnd} \Big)^2, \tag{6}$$

where LN_0 denotes LayerNorm over the channel dimension without affine parameters, i.e. the coordinates in which the frozen Action Expert reads the interface, so that the student is matched there rather than in raw feature scale. Only the student Video Expert is optimized in this stage; the trained Action Expert remains frozen, and no action-generation loss is used. Closed-loop evaluation then determines whether matching the designated interface is sufficient to preserve behavior. Detailed implementation settings are provided in Appendix A.

Early termination and reuse at inference. Since Eq. 3 is complete once the deepest selected tap $\max \mathcal{S}$ has been computed, the Video Expert runs once through $\max \mathcal{S}$ to form $\mathbf{P}$, after which later video blocks and future-frame decoding are skipped without changing the interface consumed by the Action Expert. The same $\mathbf{P}$ is then shared across ActionDiT blocks and reused at every action flow-integration step. Exact dimensions, tap locations, integration horizon, and deployment-aligned input conventions are listed in Appendix A and Section 5.

5 Experiments

We evaluate Decoupled-WAM along five questions: whether the shared planning interface preserves control performance on standard manipulation benchmarks, remains competitive under controlled distribution shifts, scales to bimanual multi-task control, supports modular changes on the Video and Action sides, and yields favorable deployment trade-offs.

5.1 Experimental Setup

Benchmarks. We evaluate on LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and RoboTwin 2.0 (Chen et al., 2025). LIBERO contains four ten-task suites—Spatial, Object, Goal, and Long—and reports suite-level and average task success. LIBERO-Plus evaluates the LIBERO-trained policy without adaptation under seven controlled shifts: camera, robot, language, lighting, background, sensor noise, and layout. RoboTwin 2.0 evaluates multi-task bimanual manipulation in clean and randomized domains. Success rate is the primary metric; for LIBERO, we additionally report matched latency and throughput.

Implementation details. Unless otherwise specified, the default Decoupled-WAM configuration pairs a 30-layer Wan2.2 TI2V-5B Video Expert with a 10-layer, width-1024 ActionDiT. The planning interface fuses dense features from Video Expert layers $\{10, 15, 20\}$ and conditions the generation of a 32-step action chunk over ten flow-integration steps, using a single Video Expert scheduler call. During inference, video computation terminates after layer 20, the deepest selected tap. Appendix A and Appendix B provide the complete optimization and evaluation protocols.

5.2 Main Benchmark Results

Across the three benchmarks, we compare with VLA baselines π_0 (Black et al., 2024), $\pi_{0.5}$ (Black et al., 2025), π_0-FAST (Pertsch et al., 2025), UniVLA (Bu et al., 2025), OpenVLA-OFT (Kim et al., 2024), X-VLA (Zheng et al., 2026), WorldVLA (Cen et al., 2025), together with WAM baselines Motus (Bi et al., 2026), LingBot-VA (Li et al., 2026b), Flash-WAM (Akbari et al., 2026), Light-WAM (Li et al., 2026c), Efficient-WAM (Li et al., 2026a), Fast-WAM (Yuan et al., 2026), Image-WAM (Zhang et al., 2026), DoT-Faster (Ma et al., 2026a), and Faster-WAM (Zhao et al., 2026). Below, we give the detailed results of each benchmark.

Table 1: **LIBERO success (%) with matched model-compute latency (ms) and throughput (query/s).** `PT.` denotes embodied pretraining. Protocol details are provided in Appendix B.3.

Method	PT.	Spat.	Obj.	Goal	Long	Avg.	Latency ↓	Throughput↑
Motus	✓	96.8	99.8	96.6	97.6	97.7	1690.3	0.59
LingBot-VA	✓	98.5	99.6	97.2	98.5	98.5	2850.5	0.35
Flash-WAM	✓	97.0	92.8	96.4	98.0	95.7	397.2	2.25
Light-WAM	×	98.2	99.6	97.8	93.0	97.2	57.3	17.5
Fast-WAM	×	98.2	**100.0**	97.0	95.2	97.6	278.9	3.6
Image-WAM	×	97.2	99.2	98.8	98.4	98.4	147.8	6.8
Faster-WAM	×	**99.6**	99.8	98.2	**98.2**	**99.0**	263.4	3.8
Decoupled-WAM	×	98.0	99.8	**98.4**	97.8	98.5	147.3	6.8
Decoupled-WAM (V20A1)	×	97.8	99.8	98.0	97.2	98.2	**40.5**	**24.7**

Table 2: LIBERO-Plus success (%) under seven OOD shifts. `PT.` denotes embodied pretraining and column abbreviations follow Sec. 5.1.

Method	PT.	Backg.	Cam.	Lang.	Light.	Layout	Rob.	Sens.	Avg.
UniVLA	✓	81.0	1.8	69.6	69.0	31.9	46.2	21.2	42.9
OpenVLA-OFT	✓	93.3	56.4	79.5	88.7	74.2	31.9	75.8	69.6
π_0	✓	81.4	13.8	58.8	85.0	68.9	6.0	79.0	53.6
π_0-FAST	✓	73.2	65.1	61.0	73.2	68.8	21.6	74.4	61.6
X-VLA	✓	96.0	23.4	75.7	88.2	71.8	89.7	62.7	71.4
WorldVLA	✓	17.1	0.1	41.6	43.7	38.0	27.9	10.9	25.0
Fast-WAM	×	45.7	18.8	70.1	83.2	62.7	45.7	29.8	49.1
DoT-Faster	×	57.0	**67.9**	92.1	94.3	**82.7**	49.0	**82.3**	75.0
Faster-WAM	×	61.3	53.8	**94.7**	**96.3**	79.1	71.6	63.6	73.6
Decoupled-WAM	×	**62.0**	51.6	93.9	94.2	80.5	69.9	74.3	74.9
Decoupled-WAM (V20A1)	×	56.7	55.0	93.6	94.4	81.8	**71.7**	74.3	**75.3**

Table 3: RoboTwin 2.0 success (%) on both clean and random task suite. `PT.` denotes embodied pretraining.

Method	PT.	Clean	Rand.	Avg.
π_0	✓	65.9	58.4	62.2
$\pi_{0.5}$	✓	82.7	76.8	79.8
Motus	✓	88.7	87.0	87.8
LingBot-VA	✓	**92.9**	91.5	92.2
Flash-WAM	✓	88.4	82.7	85.5
Light-WAM	×	76.4	76.3	76.4
Efficient-WAM	×	86.7	85.7	86.2
Fast-WAM	×	91.9	**91.8**	91.8
DoT-Faster	×	89.7	88.6	89.2
Decoupled-WAM	×	92.2	91.7	91.9

LIBERO. Table 1 evaluates both control quality and inference efficiency on standard single-arm manipulation. The default Decoupled-WAM achieves 98.5% average success, matching LingBot-VA (Li et al., 2026b), outperforming Fast-WAM (Yuan et al., 2026), Motus (Bi et al., 2026), and Light-WAM (Li et al., 2026c) by 0.9, 0.8, and 1.3 points, respectively, and remaining 0.5 points below Faster-WAM (Zhao et al., 2026). Under the matched efficiency protocol, it reduces Fast-WAM's latency from 278.9 to 147.3 ms while increasing throughput from 3.6 to 6.8 queries/s. The one-layer Action Expert further retains 98.2% success at only 40.5 ms and 24.7 queries/s, reducing latency by 72.5% relative to the default model with only a 0.3-point decrease in success.

RoboTwin 2.0. Table 3 evaluates whether the shared interface remains effective for multi-task bimanual manipulation. Decoupled-WAM achieves 92.2% success in the clean domain and 91.7% under randomization, yielding a 91.9% average. This result exceeds Fast-WAM (Yuan et al., 2026), DoT-Faster (Ma et al., 2026a), and Efficient-WAM (Li et al., 2026a) by 0.1, 2.7, and 5.7 points, respectively, while remaining 0.7 points below Faster-WAM (Zhao et al., 2026) and 0.3 points below LingBot-VA (Li et al., 2026b). The clean-to-randomized decrease is limited to 0.5 points, showing that the competitive average is maintained under domain randomization and extends beyond the standard single-arm LIBERO setting.

LIBERO-Plus. Table 2 evaluates the LIBERO-trained policies without adaptation under seven controlled distribution shifts. The default Decoupled-WAM obtains 74.9% average success, only 0.1 points below DoT-Faster (Ma et al., 2026a), while exceeding Faster-WAM (Zhao et al., 2026) and Fast-WAM (Yuan et al., 2026) by 1.3 and 25.8 points, respectively. The one-layer Action Expert further reaches 75.3%, the highest result among methods in the table without embodied-data pretraining. Both configurations remain strong under language and lighting shifts, and the one-layer model additionally improves the camera, layout, and robot-state results. These results indicate that

action-side compression preserves, rather than systematically degrades, the aggregate OOD robustness enabled by the shared interface.

5.3 Modularity through the Shared Planning Interface

Action Expert Depth Flexibility						
Configuration	D_A	Spatial	Object	Goal	Long	Avg.
Decoupled-WAM (V20A1)	1	97.8	99.8	98.0	97.2	98.2
Decoupled-WAM (V20A10)	10	98.0	99.8	98.4	97.8	<u>98.5</u>
Decoupled-WAM (V20A30)	30	98.4	100.0	98.6	97.4	**98.6**
Video Expert Replacement with Frozen Action Expert						
Configuration	D_A	Spatial	Object	Goal	Long	Avg.
Decoupled-WAM (V20A1)	1	97.8	99.8	98.0	97.2	98.2
5-layer Video Student	1	97.2	99.8	94.6	87.2	94.7
Fast-WAM teacher	30	98.2	100.0	97.0	95.2	97.6
5-layer KV student	30	26.0	17.8	14.2	5.0	15.8

Table 4: **Interface modularity on LIBERO.** Action-depth variation and frozen-consumer provider replacement.

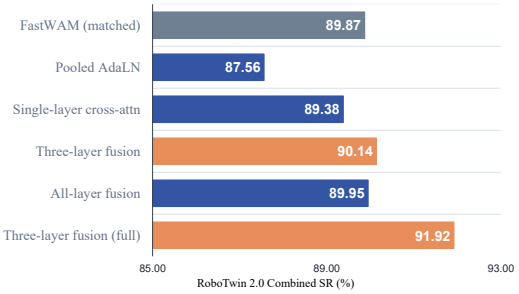

Figure 3: **Planning-feature construction on RoboTwin 2.0.** Combined success across interface variants. Matched indicated the same constrained data recipe while full for standerd recipe

Action-capacity variation. The fixed-shape planning interface removes the need for a depth-wise correspondence between the Video and Action Experts. Table 4 evaluates action-side capacity flexibility by comparing separately trained provider–consumer pairs with 1-, 10-, and 30-layer Action Experts under the same planning-feature construction. The resulting policies achieve 98.2%, 98.5%, and 98.6% success on standard LIBERO, respectively. Success varies by only 0.4 percentage points across this 30-fold depth range, with the single-layer consumer remaining competitive with substantially deeper alternatives. We adopt the 10-layer model as the default configuration and use the shallower and deeper variants to characterize the available accuracy–compute range.

Provider replacement with a frozen consumer. Table 4 also evaluates video-side replacement while retaining an already trained action consumer. Starting from the one-layer teacher policy, we train a five-layer video student to reproduce its deployment-time planning feature while freezing the complete Action Expert. The student achieves 94.7% success on standard LIBERO, compared with 98.5% for its paired teacher, preserving most in-distribution control performance after replacing the provider. Under LIBERO-Plus perturbations, success decreases from 74.9% to 38.1%, identifying distributional robustness as the primary remaining gap. Under their respective native replacement procedures, the shared-feature student also substantially outperforms the evaluated Fast-WAM layer-wise-KV student (94.7% versus 15.75%). These results support the shared planning feature as a practical replacement target for a frozen action consumer.

5.4 Planning-Feature Construction and Sensitivity

Figure 3 compares how the representation crossing the interface is constructed across four completed RoboTwin systems. Replacing a pooled single-layer feature consumed through AdaLN with dense visual cross-attention raises combined success from 87.56% to 89.38%. Fusing the sparse tap set $\{10, 15, 20\}$ through the same fixed-shape interface further reaches 90.14%, whereas extending fusion to all 30 Video Expert layers yields 89.95%. The results favor dense feature access and sparse source fusion: the selected taps provide the highest combined success among the evaluated variants, surpassing the matched trained FastWAM performance, while additional sources do not yield a monotonic improvement. Equal-weight and one-hot interventions produce further task-level variation, confirming that the feature mixture materially affects control performance; the complete intervention results and contribution diagnostics are reported in the Appendix.

5.5 Efficiency and Deployment Trade-offs

Figure 4 broadens the efficiency comparison from the principal WAM baselines to a wider set of WAM and VLA policies. Across both families, Decoupled-WAM variants occupy a favorable quality–latency frontier. The default A-U10 model reaches 98.5% LIBERO success at 147.271 ms

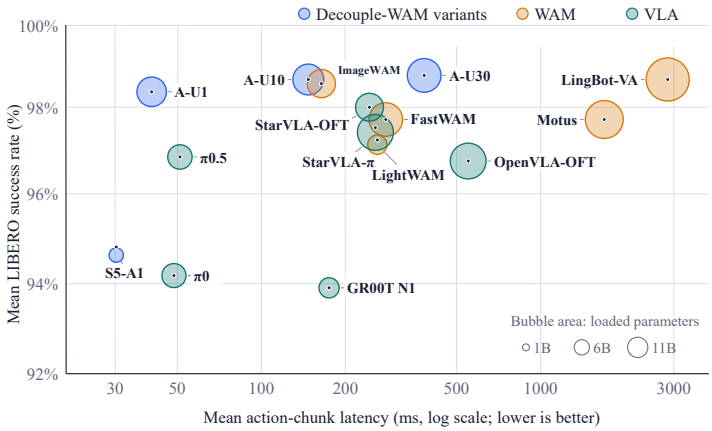

Figure 4: **LIBERO quality–efficiency landscape.** Mean success versus action-chunk latency; bubble area denotes loaded parameters. The plot extends Table 1 with representative WAMs and VLAs.

per action chunk, corresponding to a $1.89\times$ latency speedup over Fast-WAM at comparable success. The V20A1, V20A30, and 5-layer Video Student points further delineate a broad range from compact low-latency deployment to higher-capacity action generation.

The frontier reflects three independent deployment levers exposed by the fixed interface. First, Action Expert depth can be adjusted without changing the Video Expert interface. Second, provider replacement yields the 30.225 ms 5-layer Video Student point, compared with 286.023 ms for the evaluated 5-layer KV student. This contrast is consistent with the shared interface enabling provider compression within a compact consumer pipeline. Third, stopping Video Expert execution at the deepest tap L20 reduces latency by 9.64% while preserving the produced action chunks. Physically pruning the unused later layers further reduces loaded parameters from 5.601B to 3.964B. Together, Figure 4 and the controlled variants establish a family of deployment configurations spanning action capacity, provider size, and executed video depth.

6 Limitations and Future Work

Our evaluation is currently limited to simulation benchmarks because of constrained experimental resources; we do not yet validate Decoupled-WAM on physical robots. Its reliability under real-world sensing noise, dynamics, and deployment constraints therefore remains to be established. Future work will complement the current study with real-robot experiments and investigate edge deployment enabled by the decoupled interface. Because the Video and Action Experts communicate through a fixed-bandwidth planning feature and the Action Expert can be independently downsized, a compact Action Expert could potentially run on a resource-constrained edge device while receiving planning features from a separate Video Expert. We view this as a promising direction rather than a capability demonstrated by the present experiments.

7 Conclusion

We introduced Decoupled-WAM, which replaces layer-indexed video–action states with a dense, fixed-shape planning feature $\mathbf{P}$. The Video Expert merges selected intermediate features, while each Action Expert block constructs the attention states needed to read them, allowing action capacity to vary and video execution to terminate after the deepest tap. The default model achieves 98.5% success on LIBERO and 91.9% on RoboTwin 2.0. A one-layer Action Expert retains 98.2% LIBERO success at 40.5 ms and 24.7 queries/s, and reaches 75.3% on LIBERO-Plus. With this Action Expert frozen, a five-layer video student retains 94.7% success compared with 98.2% for its paired teacher. Weaker retention under perturbations shows that matching the interface does not preserve all closed-loop behavior. Together, these results establish the shared planning feature as a practical route toward more configurable WAMs while identifying robust Video Expert replacement as the main remaining challenge.

ETHICS STATEMENT

This work studies robot learning in simulation and controlled manipulation settings and does not involve human subjects or personally identifiable data. Although our experiments do not deploy policies on physical robots, learned manipulation systems may cause physical damage or be repurposed for unsafe applications if transferred without appropriate validation. Any real-world deployment should therefore include task-specific safety constraints, human oversight, and testing under the sensing and dynamics conditions of the target platform.

AI USE STATEMENT

Generative AI tools were used to assist with language revision, bibliography organization, LaTeX editing, and the prototyping of scientific figures. They were not used to generate or alter experimental measurements. The authors reviewed and verified the resulting text, citations, figures, code changes, and reported results, and remain responsible for the content of the paper.

REPRODUCIBILITY STATEMENT

The main paper and Appendix A–E document the model architecture, planning-interface construction, joint-training and provider-distillation procedures, benchmark protocols, efficiency-measurement boundaries, and additional ablations. Reported comparisons distinguish paper-reported baseline results from measurements obtained under our matched local protocol, and identify cases in which success and efficiency use different checkpoints. The appendix also records evaluation denominators, checkpoint-selection rules, and the frozen or trainable status of the relevant Video and Action Expert components. Upon publication, we will release the training and evaluation code, experiment configurations, and trained model weights required to reproduce the reported Decoupled-WAM results.

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.

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 (CVPR)*, pp. 35101–35113, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

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

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

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: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. URL https://arxiv.org/abs/2510.13626.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, Meng Guo, and Shanghang Zhang. Efficient-wam: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026a. URL https://arxiv.org/abs/2606.10040.

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

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*, 2026c. URL https://arxiv.org/abs/2606.08242.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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*, 2023. URL https://arxiv.org/abs/2306.03310.

Liheng Ma, Rui Heng Yang, Zhanguang Zhang, Mateo Clemente, Ziwen Hu, Tongtong Cao, and Yingxue Zhang. Faster-wam: Do world action models need deep action modules? *arXiv preprint arXiv:2608.02365*, 2026a. URL https://arxiv.org/abs/2608.02365.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026b.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Mustafa Shukor et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. URL https://arxiv.org/abs/2506.01844.

Yihao Wang et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025. URL https://arxiv.org/abs/2509.09372.

Seonghyeon Ye et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

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

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

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026. URL https://arxiv.org/abs/2608.04404.

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