EMERGENT DEPTH-WISE SPECIALIZATION IN A UNIFIED EMBODIED MODEL: FROM SEMANTIC UNDERSTANDING THROUGH VISUAL FORESIGHT TO ACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies emphasize semantic understanding, whereas world-action models (WAMs) learn predictive representations of environment dynamics. Systems that expose a policy to both sources often still concentrate action computation on a single expert. We present EWAM, an action-centric unified embodied model whose asymmetric joint attention lets action tokens read semantic, current-visual, predicted-future, and action information at every layer while the perceptual experts retain their distinct roles. Without layer-wise supervision, EWAM develops an *emergent depth-wise specialization*: action queries attend mainly to vision-language features in shallow layers, to predicted future frames in intermediate layers, and to action tokens themselves in deep layers. This handoff replicates across tasks and is stable across denoising steps. Checkpoint tracking and causal interventions show that it is learned and that action generation depends on it. EWAM is pretrained in two separate regimes, one on cross-embodiment robot trajectories and one on human egocentric video. In simulation and real-robot experiments, it surpasses existing VLA, WAM, and hybrid baselines. Human egocentric data improve both cross-embodiment transfer and real-robot robustness, and subtask-phase supervision improves long-horizon completion. Together, these results suggest that unified embodied learning can induce an ordered internal progression from semantic understanding, through visual foresight, to action formation. Code will be made publicly available.

1 INTRODUCTION

Kenneth Craik's classic account of internal models proposed that an intelligent organism carries a "small-scale model" of external reality, allowing it to "react to future situations before they arise" (Craik, 1943). This view casts purposeful action as an ordered computation: understanding the present situation, anticipating how it may evolve, and then determining what to do. Generalist robot policies face an analogous challenge. They must connect at least three forms of computation: understanding what an instruction and the current scene imply, anticipating how the physical world may evolve, and producing precise actions.

Two dominant paradigms emphasize different parts of this pipeline. Vision-language-action (VLA) models inherit rich semantic representations from vision-language models, but their static pretraining provides limited explicit supervision for physical dynamics (Brohan et al., 2023; Driess et al., 2023; Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2025b;a). World-action models (WAMs), in contrast, learn from temporal visual data and can acquire useful motion priors (Ye et al., 2026b; Li et al., 2026b; Yuan et al., 2026b), yet predicting plausible futures does not by itself guarantee semantic reasoning or fine-grained control (Zhang et al., 2026e).

Prior attempts to obtain both capabilities generally follow one of two paths. The first remains centered on a single paradigm, adding auxiliary outputs while leaving action generation predominantly dependent on either vision-language or video features (Wu et al., 2026b; Zhang et al., 2026b; Sun et al., 2026). The second attaches an independently operating world model or understanding module to a policy (Physical Intelligence et al., 2026; Long et al., 2026). Such composition adds capabilities, but does not necessarily induce structured cooperation inside the action policy. Even a multi-expert

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

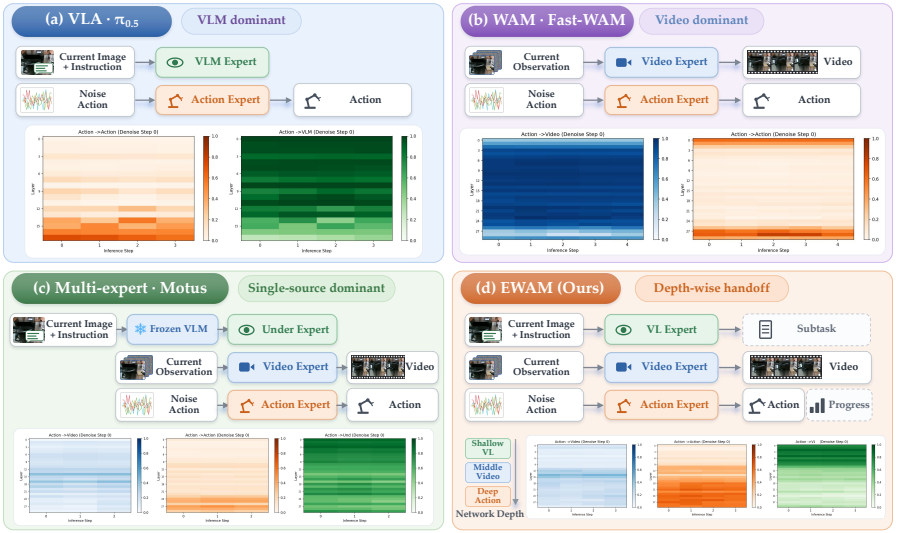


Figure 1: **Architectures and action-attention routing across embodied-model paradigms.** Each panel pairs a simplified architecture with action-query attention maps. Representative VLA, WAM, and multi-expert policies concentrate action attention on one primary source. In EWAM, the dominant source changes from vision-language features in shallow layers, through predicted-future features in intermediate layers, to action self-attention in deep layers. Colour encodes attention mass on an absolute zero-to-one scale, the horizontal axis is the policy inference step at denoising step 0, and the vertical axis is the layer index. Appendix B.1 gives further details.

model (Bi et al., 2026) can leave action computation concentrated on one expert. This raises a more fundamental question than how to add another module: *when semantic understanding, visual prediction, and action generation are truly unified, do they continue to compete throughout the network, or does the model develop an ordered progression reminiscent of purposeful human action—first understanding the task, then imagining what should happen next, and finally committing to an action?* Figure 1 makes this distinction concrete by visualizing where action queries retrieve information across representative paradigms. In $\pi_{0.5}$ (Black et al., 2025a), action attention is consistently concentrated on VLM tokens, reflecting a VLA-style computation in which semantic representations are the primary source for control (Figure 1(a)). Fast-WAM (Yuan et al., 2026b) exhibits the complementary pattern: action queries attend predominantly to video tokens, consistent with a WAM whose control computation is centered on learned visual dynamics (Figure 1(b)). Motus (Bi et al., 2026) integrates multiple expert modules through an externally attached frozen VLM, yet its action attention remains dominated by the understanding expert (Figure 1(c)). This observation suggests that merely stacking experts may still yield a single-expert bias resembling the VLA paradigm, rather than inducing an internal division of labor.

To determine whether a unified policy can move beyond this single-expert bias, we develop EWAM, a unified embodied model containing a vision-language (VL) expert, a video expert, and an action expert. At every layer, action queries can attend to semantic tokens, predicted visual tokens, and action tokens. An asymmetric mask prevents video and vision-language queries from reading action tokens or each other, making the action stream the integration site while preserving the distinct predictive and semantic roles of the two perceptual experts. Video and action outputs are trained jointly with flow-matching objectives.

Despite receiving no explicit layer-wise supervision, EWAM develops a consistent internal progression (Figure 1(d)). In shallow layers, action tokens attend primarily to vision-language tokens, including both language and current-image content. In intermediate layers, attention to current-image tokens declines while attention to predicted future-frame tokens rises. In deep layers, action self-attention becomes dominant as the policy consolidates the information into a control sequence. We refer to this depth-dependent handoff as *emergent depth-wise specialization*: an ordered progression from semantic understanding, through visual foresight, to action formation.

We validate this claim along four dimensions: observational evidence, developmental dynamics, causal interventions, and policy performance. Our observational analysis confirms that the hierar-

chical handoff is stable across tasks, while tracking training checkpoints shows that this internal organization emerges during training. We then examine depth-specific functional dependencies on vision-language, observed-frame, and predicted-future features through layer-targeted training and fixed-checkpoint interventions. Same-task counterfactual future-feature injection across five tasks further tests whether actions depend on the specific content of predicted futures. Finally, we evaluate the resulting policy in simulation and real-world robot settings. We study two separate pretraining regimes, one on cross-embodiment robot trajectories and the other on human egocentric video. Initialized from cross-embodiment robot pretraining, EWAM achieves 72.1% on RoboTwin clean-to-random (C2R) evaluation and a 92.9% average in the in-domain setting. It also obtains 98.8% on LIBERO and demonstrates consistent performance across multiple real-world robot embodiments. Human pretraining raises success on a held-out robot embodiment from 40.1% to 66.9%, and egocentric co-training raises real-robot success from 10% to 60% under scene variations. Subtask-phase supervision during post-training further improves long-horizon completion in both simulated and real-world tasks.

Our contributions are:

- We develop EWAM, an action-centric unified embodied model that exposes semantic, predictive, and action representations to a common action stream through asymmetric joint attention while preserving the provenance of the perceptual experts.
- We identify and characterize emergent depth-wise specialization: a semantic-to-foresight-to-action handoff that replicates across diverse tasks and remains stable across denoising steps. Checkpoint tracking, layer-targeted interventions, and counterfactual future-feature injection show that it is learned and functionally consequential.
- We show that EWAM is competitive across complementary evaluation protocols, reaching 72.1% on RoboTwin C2R, a 92.9% in-domain average, and 98.8% on LIBERO, with consistent performance across three real-robot embodiments. Human egocentric data improve cross-embodiment transfer via pretraining and robustness under scene variations via co-training. Subtask-phase supervision further improves long-horizon completion.

2 Related Work

Vision-language-action models. VLA policies adapt vision-language representations to robot control, providing a direct route for transferring semantic and web-scale knowledge into action prediction. RT-1 (Brohan et al., 2023) demonstrated large-scale tokenized action prediction, while PaLM-E (Driess et al., 2023) connected embodied observations to a multimodal language model. RT-2 (Zitkovich et al., 2023) casts actions as language tokens and co-trains robot trajectories with vision-language data. Octo (Ghosh et al., 2024) learns a cross-embodiment generalist policy from heterogeneous robot datasets, while OpenVLA (Kim et al., 2024) scales an open VLA built from pretrained vision and language components. The π family (Black et al., 2025b) couples a pretrained VLM to a flow-matching action expert, extending continuous generative policy learning beyond the earlier Diffusion Policy (Chi et al., 2023), and $\pi_{0.5}$ (Black et al., 2025a) further combines multi-robot data, semantic subtask prediction, and web data for open-world operation. These systems establish the strength of semantic pretraining for instruction following and generalization. However, when predictive visual dynamics are introduced alongside VLM features, it remains unclear how the action stream coordinates these complementary sources across network depth.

World-action models. WAMs instead build robot policies on predictive video representations and jointly model future observations and actions. DreamZero (Ye et al., 2026b) adapts a pretrained video diffusion backbone into a large joint video-action policy, demonstrating zero-shot transfer across embodiments. LingBot-VA (Li et al., 2026b) learns causal frame prediction and policy execution in a shared latent space with closed-loop rollout. Fast-WAM (Yuan et al., 2026b) separates video co-training from test-time imagination, showing that predictive supervision can benefit action learning even when future frames are not generated at inference. A recent controlled robustness study (Zhang et al., 2026e) reports complementary strengths between VLA and WAM policies, with their relative performance shaped by architecture and training-data diversity. Our work does not seek to determine which paradigm is universally superior; instead, it examines how their semantic and predictive representations are internally coordinated when exposed to one action stream.

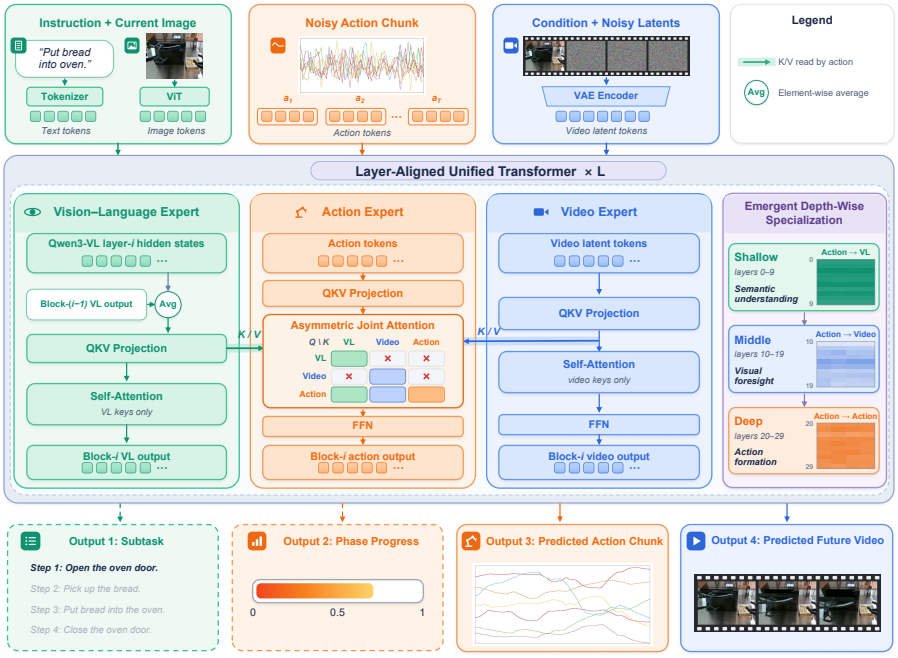


Figure 2: **Overview of our unified embodied model, EWAM.** Language and image observations, noisy action sequences, and noisy video latents are tokenized into layer-aligned VL, action, and video streams. Action queries attend to VL, video, and action keys, while VL and video queries remain within their own streams. The model predicts an action sequence and future video, and optionally predicts the active subtask and its corresponding phase progress.

Hybrid VLA–WAM systems. Recent systems begin to combine the semantic strengths of VLAs with the visual foresight of world models, but most retain a modular pipeline. At inference, $\pi_{0.7}$ (Physical Intelligence et al., 2026) uses a high-level semantic policy to produce subtask instructions and a separately trained world model based on BAGEL (Deng et al., 2025) to generate subgoal images; these external outputs then condition its VLA action policy. Cortex 2.0 (Aida et al., 2026) similarly augments an existing VLA with a distinct world-model planning module: the world model rolls out candidate futures, a reward module selects one, and the selected trajectory is passed to the downstream controller. Motus (Bi et al., 2026) brings video generation and action prediction into a Mixture-of-Transformers, but introduces semantic understanding through an externally attached frozen VLM. These approaches establish the value of combining semantics, foresight, and control, while preserving explicit module boundaries or one-way intermediate interfaces. In contrast, our model couples semantic, future-visual, and action representations within one layer-aligned architecture: the action stream can interact with all three sources throughout depth, allowing the understanding–foresight–action division of labor to emerge internally rather than being prescribed by a serial pipeline.

3 A Unified Embodied Model

3.1 Architecture Overview

As illustrated in Figure 2, given an image observation o_t and a language instruction ℓ, our model jointly predicts a future visual trajectory $v_{t+1:t+H}$ and an action chunk $a_{t:t+K-1}$. It contains three layer-aligned experts. A vision-language expert encodes the instruction and current image into semantic tokens h^L; a video expert processes the condition-frame latent together with noisy future-video latents h^V; and an action expert embeds the noisy action chunk into h^A. The video expert is initialized from Wan2.2-TI2V-5B (Wan Team et al., 2025). The vision-language expert is initialized from Qwen3-VL-2B (Bai et al., 2025). Unlike a pipeline in which a separately generated future is passed to a downstream VLA, the three streams are executed in one forward process and interact at every network depth through the action stream.

Training proceeds in two stages. During large-scale pretraining, we optimize future prediction and action generation over cross-embodiment trajectories. During task-specific post-training, we retain these objectives and introduce subtask-phase supervision. Long-horizon demonstrations are partitioned into semantic phases; the same partition supplies a textual label of the active phase and a scalar phase index. The post-trained model therefore produces

$$(\hat{v}_{t+1:t+H}, \hat{a}_{t:t+K-1}, \hat{p}_t, \hat{y}_t) = f_\theta(o_t, \ell), \tag{1}$$

where $\hat{y}_t$ is the currently active subtask and $\hat{p}_t$ is the normalized index of that phase in the subtask sequence.

3.2 Action-Centric Asymmetric Attention

Unrestricted trimodal attention can blur the pretrained roles of semantic and predictive representations. We instead make action tokens the sole integration interface. For video (V), action (A), and vision-language (L) tokens, the attention mask permits

$$A \leftarrow \{V, A, L\}, \qquad V \leftarrow V, \qquad L \leftarrow L. \tag{2}$$

Thus, action queries can retrieve semantic and predictive evidence, while the video and vision-language streams retain modality-specific computation and cannot read one another or the action stream. In implementation, the three query-side computations in Figure 2 are evaluated through a single concatenated masked-attention operation, with each stream retaining its own normalization and projection parameters.

The instruction and current image are first processed by Qwen3-VL (Bai et al., 2025), yielding 28 hidden states. To expose these semantic features across the 30 unified blocks, we align them to the unified depth using the fixed mapping detailed in Appendix A. Let $q_l^L \in \mathbb{R}^{N_L \times 2{,}048}$ denote the representation aligned to block l, and let z_{l-1}^L denote the VL state accumulated through the preceding unified block. At unified block l, we form

$$x_l^L = \begin{cases} q_l^L, & l = 0, \\ \frac{1}{2}\left(q_l^L + z_{l-1}^L\right), & l > 0. \end{cases} \tag{3}$$

Thus, x_l^L combines the layer-matched Qwen3-VL feature with the VL updates accumulated through unified depth; its VL-only attention output is projected back to 2,048 dimensions and added residually to give z_l^L. The video stream requires no such alignment, since the 30 unified blocks follow the native depth of the Wan2.2 video expert. Both streams expose their keys and values to action queries, making the action stream the sole cross-modal reader; the action objective shapes semantic and predictive representations only through these exposed keys and values. This separation also makes action-query attention a well-defined measure of which source is retrieved.

3.3 Joint Video–Action Flow Matching

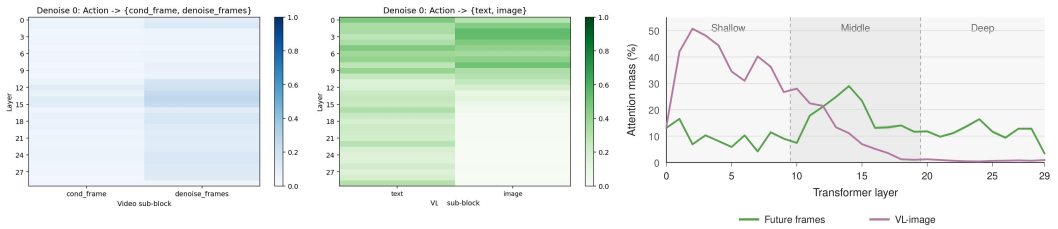
Let x_v and x_a denote clean future-video and action targets. We independently sample noise levels for the two modalities, construct their noisy interpolants, and regress the corresponding flow velocities. The observed condition-frame latent is kept fixed, while future-video and action latents are updated jointly. The base objective is

$$\mathcal{L}_{\text{base}} = \lambda_v \|\hat{u}_v - u_v^\star\|_2^2 + \lambda_a \|\hat{u}_a - u_a^\star\|_2^2, \tag{4}$$

where $\hat{u}_v$ and $\hat{u}_a$ are the predicted video and action velocities, $u_v^\star$ and $u_a^\star$ are the corresponding target flow velocities, and λ_v and λ_a are loss weights. At inference, we jointly integrate the predicted video and action velocity fields. The two trajectories are therefore denoised concurrently within the same forward process, allowing evolving future-video features to participate directly in action computation rather than serving as a separately generated visual prompt.

3.4 Subtask-Aware Post-Training

During task-specific post-training, we partition each long-horizon demonstration into semantic subtask phases using annotated transition checkpoints. The same partition supplies two lightweight prediction heads on the unified representation: a textual description of the active phase and a scalar phase index $p_t = k_t / \max(N-1, 1)$. The scalar is the normalized position of the current phase in the subtask sequence. Both losses augment the base video–action objective only during post-training, and no external high-level policy is required. Appendix A.4 gives the complete formulation.



(a) Fine-grained action-attention allocation across depth

(b) Cross-task image-to-future visual handoff

Figure 3: **Fine-grained visual handoff across depth.** (a) Action attention to observed and predicted video features and to VL text and image features on the ADJUST BOTTLE task. (b) Cross-task VL-image and future-frame attention, macro-averaged over 50 tasks and five denoising steps.

4 EMERGENT DEPTH-WISE SPECIALIZATION

4.1 OBSERVATIONAL EVIDENCE OF DEPTH-WISE SPECIALIZATION

We analyze how the information retrieved by action queries changes with model depth. Let $\mathcal{Q}_A$ denote the set of action queries, $\mathcal{K}$ the set of all keys visible to them, and $\mathcal{K}_m \subseteq \mathcal{K}$ the keys from source m. With $\alpha_{qk}^{(l)}$ the post-softmax attention weight from query q to key k in block l, the allocation to source m is

$$R_l^m = \frac{\sum_{q\in\mathcal{Q}_A}\sum_{k\in\mathcal{K}_m}\alpha_{qk}^{(l)}}{\sum_{q\in\mathcal{Q}_A}\sum_{k\in\mathcal{K}}\alpha_{qk}^{(l)}}. \tag{5}$$

We distinguish five sources: VL-text, VL-image, condition-frame, future-frame, and action tokens, macro-averaged over 50 RoboTwin 2.0 tasks and five denoising steps, using the robot-pretrained in-domain checkpoint. This statistic shows where action tokens retrieve information; Section 4.3 tests functional necessity.

Shallow understanding. In layers 0–9, VL tokens receive 75.4% of action attention, compared with 14.1% for video and 10.6% for action tokens. Action computation therefore begins by retrieving the instruction and current-scene information needed to identify and ground the task.

Intermediate visual foresight. In layers 10–19, Figure 3(a) and (b) show that attention to VL-image tokens drops sharply while attention to predicted future-frame tokens increases, with a median crossover at layer 13; attention to the observed condition frame remains comparatively weak. The model thus retains linguistic guidance from the VL expert while shifting its visual dependence from the current observation toward predictive representations of future scene evolution.

Deep action formation. In layers 20–29, action self-attention rises to 59.5%, while VL and video attention fall to 25.2% and 15.3%. Deep layers therefore primarily refine motor trajectories while preserving concise semantic and predictive context. Together, these regimes form a gradual understanding–foresight–action handoff rather than three isolated modules.

Cross-task stability. The image-to-future handoff appears in all 50 tasks and remains present in 49 after token-count normalization. VL attention is largest in the shallow stage for all tasks, and action attention is largest in the deep stage for all tasks. The profiles are also stable across denoising steps (mean pairwise correlation 0.998); Appendices B.2 and B.3 report the complete statistics.

4.2 LEARNING THE VISUAL HANDOFF DURING TRAINING

We trace action-attention routing across twelve checkpoints spanning 50 to 40,000 optimizer steps of one training run without robot pretraining, to determine whether the depth-wise division of labor is present from the start or emerges through training. At 50 steps, action queries attend to VL-image tokens at every depth, and VL-text receives only 2.9% of shallow attention; no layer-wise division is yet visible. A coarse organization appears within the first few hundred steps. By 150 steps, shallow text attention has risen to 22.6%, image attention concentrates in shallow layers, and raw future-frame attention mass first exceeds VL-image mass at layer 8. Continued training

refines this organization until, at 40,000 steps, the three regimes of Section 4.1 are established: shallow layers read the instruction and current image, and raw future-frame attention mass exceeds VL-image mass at every layer from 13 onward (Figure 4). The mask determines which sources action queries can access, but not the depth at which each source is used. The layer-specific routing therefore emerges during the training of our model. Appendix B.4 and Table 11 give full values, including token-normalized measurements.

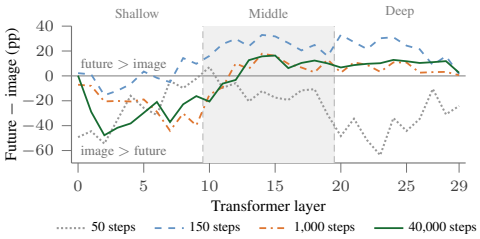


Figure 4: **Emergence of the visual handoff.**

4.3 Causal Validation through Layer-Targeted Interventions

Table 1 combines trained structural ablations with inference-time key masking at a fixed robot-pretrained checkpoint, testing the utility of the access pattern and the model's learned dependence on it. Removing shallow VL or intermediate video access during training reduces success from 83% to 78%, whereas action-only deep layers preserve the baseline. At the fixed checkpoint, shallow VL masking costs 16.27 points, while shallow video masking has little effect. Masking intermediate future features alone nearly collapses performance, reducing success from 92.80% to 3.55%. These results corroborate the mechanism of early semantic grounding, intermediate visual integration, and late action refinement.

Table 1: **Layer-targeted training and fixed-checkpoint interventions on RoboTwin Randomized.**

Training results (without pretraining)			**Inference results (robot-pretrained model)**		
Configuration	**Score**	**Δ**	**Configuration**	**Score**	**Δ**
Full unified model	83	–	Full unified model	92.80	–
Symmetric full attention	81	−2	Mask all video, layers 0–9	91.98	−0.82
No VL access, layers 0–9	78	−5	Mask VL, layers 0–9	76.53	−16.27
No video access, layers 10–19	78	−5	Mask condition, layers 10–19	81.40	−11.40
Action only, layers 24–29	83	0	Mask future, layers 10–19	3.55	−89.25
			Mask all video, layers 10–19	0.25	−92.55
			Mask all video, layers 20–29	65.82	−26.98

Counterfactual future-feature injection. To test whether actions depend on the content of future features rather than merely on their presence, we conduct same-task counterfactual interventions across five tasks. We inject future-frame keys and values from successful donor seed 100 into target seeds 43–47 at layers 10–19 during the first three policy calls. All 25 target rollouts succeed without injection but only 4/25 after injection (Table 2). On Adjust Bottle, injection transfers the donor's spatial target and arm choice, providing content-sensitive causal evidence that future representations influence action generation. Appendix B.5 provides further details.

Table 2: **Counterfactual future-feature injection across five tasks.**

Task	**No injection**	**Injection setting**	**Observed behavior or outcome**	**Success**
Adjust Bottle	5/5	Left-facing (43, 46)	Donor-directed first grasp misses; both recover after cutoff	2/2
		Right-facing (44, 45, 47)	Donor-consistent right arm instead of required left arm; no recovery	0/3
Place Empty Cup	5/5	All targets (43–47)	All injected rollouts fail	0/5
Stack Blocks Two	5/5	All targets (43–47)	All injected rollouts fail	0/5
Click Bell	5/5	All targets (43–47)	Only seed 46 remains successful	1/5
Grab Roller	5/5	All targets (43–47)	Only seed 45 remains successful	1/5
Overall	**25/25**	All injected targets	Success decreases from 100% to 16%	**4/25**

5 Experiments

We evaluate EWAM in three parts. Section 5.1 assesses the robot-pretrained policy on RoboTwin 2.0 (Chen et al., 2026b) under clean-to-random and in-domain protocols, on LIBERO (Liu et al., 2023), and on seven real-world tasks across Franka, Dobot, and Unitree G1-D, comparing against VLA, WAM, and hybrid VLA–WAM baselines. Section 5.2 examines human

egocentric pretraining through cross-embodiment transfer and real-robot co-training. Section 5.3 tests subtask-phase supervision on instruction-conditioned block tasks and a real-world sorting sequence. Appendices E and C.5 detail pretraining data and inference latency.

5.1 Robot-Pretrained Policy Evaluation

RoboTwin Clean-to-Random Evaluation. Following the official RoboTwin 2.0 protocol, we adapt one cross-embodiment robot-pretrained policy on clean demonstrations of all 50 tasks and evaluate it under clean-to-clean (C2C) and clean-to-random (C2R) conditions; no target-embodiment randomized data are used for adaptation (Table 3). EWAM reaches 82.2% C2C and 72.1% C2R, exceeding GigaBrain-0.7 by 4.2 C2R points and $\pi_{0.5}$ by 26.1. Overall, EWAM achieves the highest C2R and average scores, generalizing well from clean to randomized scenes.

RoboTwin In-Domain Results. In the in-domain setting, all methods train and evaluate on both clean and randomized demonstrations. EWAM achieves the highest average, 92.9%, with 93.0% on clean scenes and 92.8% under randomization (Table 4); its 0.2-point gap indicates consistent performance across conditions. Without pretraining, the same architecture reaches 83.0, showing that it remains effective from scratch while benefiting substantially from cross-embodiment pretraining.

Table 3: **RoboTwin 2.0 clean-to-random.**

Paradigm	Model	C2C (%)	C2R (%)	Average (%)
VLA	starVLA (StarVLA Community, 2026)	46.52	3.16	24.84
	GalaxeaVLA (Jiang et al., 2025)	62.70	12.72	37.71
	Xiaomi Robotics-0 (Cai et al., 2026b)	62.90	18.20	40.55
	X-VLA (Zheng et al., 2026)	68.00	20.90	44.45
	ABot-M0 (Yang et al., 2026b)	57.40	30.36	43.88
	$\pi_{0.5}$ (Black et al., 2025a)	70.70	46.00	58.35
WAM	Fast-WAM (Yuan et al., 2026b)	77.80	1.90	39.85
	ImageWAM (Zhang et al., 2026d)	84.4	18.3	51.35
	X-WAM (Guo et al., 2026)	70.00	25.80	47.90
	4D-WAM (Yang et al., 2026a)	81.5	41.8	61.6
VLA+WAM	GigaBrain-0.7 (GigaBrain Team et al., 2026)	66.8	67.9	67.3
	EWAM (Ours)	**82.2**	**72.1**	**77.2**

Table 4: **RoboTwin 2.0 in-domain results.**

Paradigm	Model	Clean (%)	Randomized (%)	Average (%)
VLA	$\pi_{0.5}$ (Black et al., 2025a)	82.7	76.8	79.8
	ABot-M0 (Yang et al., 2026b)	86.1	85.1	85.6
	LingBot-VLA (Wu et al., 2026a)	88.6	86.7	87.7
	JoyAI-RA (Zhang et al., 2026c)	90.5	89.3	89.9
	HyVLA-0.5 (Zhang et al., 2026a)	90.9	90.1	90.5
	ACE-Ego-0 (Li et al., 2026a)	91.1	90.6	90.9
WAM	GigaWorld-Policy (Ye et al., 2026a)	86.4	85.0	85.7
	LaWAM (Chen et al., 2026a)	92.6	89.8	91.2
	LingBot-VA (Li et al., 2026b)	92.9	91.5	92.2
	Fast-WAM (Yuan et al., 2026b)	91.9	91.8	91.9
VLA+WAM	Motus (Bi et al., 2026)	88.7	87.0	87.9
	InternVLA-A1 (Cai et al., 2026a)	89.4	89.6	89.5
	WLA-0 (Yang et al., 2026c)	92.9	90.0	91.5
	EWAM (Ours)	**93.0**	**92.8**	**92.9**

LIBERO Simulation Results. We evaluate EWAM on LIBERO's spatial, object-centric, goal-conditioned, and long-horizon suites (Liu et al., 2023). EWAM achieves the highest overall average, 98.8% (Table 5); per-suite results are reported in Appendix Table 15.

Table 5: **LIBERO average success rates (%).**

Paradigm	Model	Average
VLA	$\pi_{0.5}$ (Black et al., 2025a)	96.9
	ABot-M0 (Yang et al., 2026b)	98.6
	Qwen-VLA-Instruct (Wang et al., 2026c)	97.9
WAM	Fast-WAM (Yuan et al., 2026b)	97.6
	LingBot-VA (Li et al., 2026b)	98.5
	LaWAM (Chen et al., 2026a)	98.6
VLA+WAM	Motus (Bi et al., 2026)	97.7
	WLA-0 (Yang et al., 2026c)	98.6
	EWAM (Ours)	**98.8**

Real-World Evaluation. We evaluate seven real-world tasks across a Franka single arm, a Dobot dual-arm platform, and a Unitree G1-D dual-arm humanoid. As shown in Tables 6 and 7, EWAM attains the highest success rate on every task, exceeding $\pi_{0.5}$ on six of the seven tasks and tying it on the Dobot towel task. These results demonstrate consistent real-world performance across diverse embodiments and manipulation tasks.

Table 6: **Real-world success rates on Franka (%).**

Model	Stack Bowls	Object into Box
$\pi_{0.5}$ (Black et al., 2025a)	70	80
Motus (Bi et al., 2026)	60	20
LingBot-VA (Li et al., 2026b)	70	60
Xiaomi Robotics-0 (Cai et al., 2026b)	70	70
Fast-WAM (Yuan et al., 2026b)	60	30
EWAM (Ours)	**80**	**90**

Table 7: **Real-world success rates on Dobot and Unitree G1-D (%).**

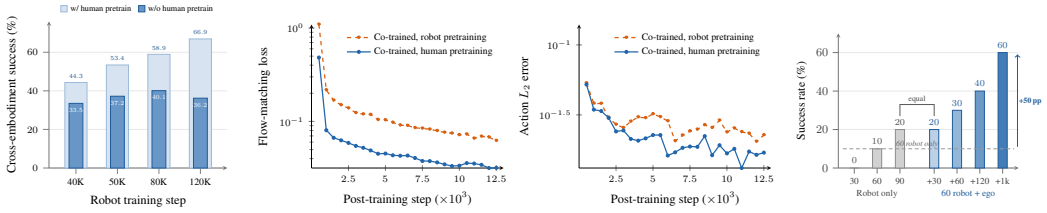
	Dobot			Unitree G1-D	
Model	Pour Water	Tidy Desk	Towel	Kettle Pouring	Pour Beans
$\pi_{0.5}$ (Black et al., 2025a)	30	20	**40**	60	60
Motus (Bi et al., 2026)	30	10	30	30	40
Fast-WAM (Yuan et al., 2026b)	20	70	0	20	30
EWAM (Ours)	**70**	**80**	**40**	**70**	**80**

5.2 Human Pretraining and Co-Training

Cross-Embodiment Transfer from Human Pretraining. To evaluate the cross-embodiment transfer of human pretraining, we train otherwise identical policies on four RoboTwin 2.0 embodiments and evaluate on the unseen Aloha-Agilex-2 embodiment. The policies differ only in whether they use the human-pretrained initialization of Appendix D.2. Human pretraining improves cross-embodiment success at every evaluated checkpoint (Figure 5(a)). At matched 120K steps, success rises from 36.2% to 66.9%, a 30.7-point gain; comparing each policy's best checkpoint gives 40.1% versus 66.9%, a 26.8-point gain.

Effect of Human Pretraining on Co-Training. To test whether human pretraining helps the policy exploit human data during post-training, we co-train Unitree G1-D on a long-horizon pouring task with 60 robot and 30 retargeted human episodes, initializing from either the robot-pretrained or the human-pretrained checkpoint. On the same mixture, human pretraining reduces both final training loss (Figure 5(b)) and held-out action error (Figure 5(c)), while raising physical-robot success from 40% to 60% (Appendix Table 20).

Effect of Egocentric Co-Training Data. Without pretraining and with 60 robot episodes fixed, adding 30 to 1,000 egocentric episodes raises real-robot success under tablecloth and cup changes from 10% to 60% without saturating (Figure 5(d); Appendix D.7). Adding just 30 egocentric episodes reaches 20%, matching robot-only training with 90 episodes. These results show that low-cost egocentric data can effectively improve robustness.



(a) Cross-embodiment transfer (b) Training loss (c) Held-out action error (d) Egocentric data scaling

Figure 5: **Human pretraining and egocentric co-training.** (a) Cross-embodiment transfer to Aloha-Agilex-2. (b) Training loss and (c) held-out action error when co-training from robot or human pretraining. (d) Real-robot success without (gray) and with (blue) egocentric data.

5.3 Subtask-Phase Supervision

We next test the subtask-phase prediction heads of Section 3.4 as a current-goal signal for long-horizon control. The same architecture is post-trained with and without this supervision, then evaluated on three instruction-conditioned block tasks and on a Unitree G1-D dual-basket sorting sequence (Appendix C.4). Table 8 shows higher later-stage completion and higher full-episode success on all three simulation tasks: instructed ranking rises from 83.5% to 91.0%, instructed stacking from 44.5% to 58.5%, and instructed ranking & stacking from 1.5% to 16.0%. The same supervision raises later-stage completion on dual-basket sorting, from a mean of $1.0/4$ objects placed without the heads to $2.2/4$ with them. The gains are consistent with training the policy to recognize and complete the currently active subtask, rather than with adding an independently executed planner.

Table 8: **Effect of subtask-phase supervision on long-horizon block tasks.** Grasp, Place, and Stack are stage-wise rates; Full is complete-episode success. Dashes mark inapplicable stages.

Task	Setting	Grasp	Place	Stack	Full
Instructed Ranking	w/o subtask	94.0	89.5	–	83.5
	w/ subtask	93.5	97.5	–	**91.0**
Instructed Stacking	w/o subtask	91.5	–	44.5	44.5
	w/ subtask	98.5	–	58.5	**58.5**
Instructed Ranking & Stacking	w/o subtask	60.0	37.0	24.5	1.5
	w/ subtask	63.0	51.5	39.5	**16.0**

6 Conclusion

We present EWAM, a unified embodied model in which action queries read from vision-language and video experts through asymmetric attention. Its computation organizes into a depth-wise handoff from semantic grounding, through predicted visual futures, to action refinement. This organization emerges during training, and layer-targeted masking and counterfactual injection show that action generation depends on it. Across simulation benchmarks and real-robot embodiments, EWAM exceeds strong VLA, WAM, and hybrid baselines. Low-cost human egocentric data further improve cross-embodiment transfer and robustness, and subtask-phase supervision improves long-horizon completion. These results suggest that how unified policies coordinate semantic and predictive computation offers a principled path toward more capable embodied models.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing of the manuscript and to assist with LaTeX formatting, layout, and figure plotting. All AI-assisted text and figures were reviewed and edited by the authors, and all plotted values were checked against the original experimental logs.

REFERENCES

Adriana Aida, Walid Amer, Katarina Bankovic, Dhruv Behl, Fabian Busch, Annie Bhalla, Minh Duong, Florian Gienger, Rohan Godse, Denis Grachev, Ralf Gulde, Elisa Hagensieker, Junpeng Hu, Shivam Joshi, Tobias Knobloch, Likith Kumar, Damien LaRocque, Keerthana Lokesh, Omar Moured, Khiem Nguyen, Christian Preyss, Ranjith Sriganesan, Vikram Singh, Carsten Sponner, Anh Tong, Dominik Tuscher, Marc Tuscher, and Pavan Upputuri. Cortex 2.0: Grounding world models in real-world industrial deployment. *arXiv preprint arXiv:2604.20246*, 2026. URL https://arxiv.org/abs/2604.20246.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 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 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

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

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, pp. 33383–33407, 2025c.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026a.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026b.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, pp. 138846–138923, 2026.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, et al. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning (ICML)*, 2026b. URL https://arxiv.org/abs/2506.18088.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Kenneth J. W. Craik. *The Nature of Explanation*. Cambridge University Press, 1943.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, et al. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 753–771, 2018.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, 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.

GigaBrain Team et al. Gigabrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture. *arXiv preprint arXiv:2608.15875*, 2026.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5843–5851, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Ryan Hoque, Peide Huang, David Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, pp. 4218–4237, 2026.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, et al. RoboMIND 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025. URL https://arxiv.org/abs/2512.24653.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and G0 dual-system VLA model. *arXiv preprint arXiv:2509.00576*, 2025. URL https://arxiv.org/abs/2509.00576.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233, 2025.

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 *Conference on Robot Learning*, 2024.

Hao Li, Ganlong Zhao, Yufei Liu, Haotian Hou, Guoquan Ye, et al. ACE-Ego-0: Unifying egocentric human and robotic data for vla pretraining. *arXiv preprint arXiv:2606.17200*, 2026a.

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.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

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.

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*, pp. 44776–44791, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, pp. 29982–30009, 2025.

Qian Long, Yueze Wang, Jiaxi Song, Junbo Zhang, Peiyan Li, Wenxuan Wang, Yuqi Wang, Haoyang Li, Shaoxuan Xie, Guocai Yao, Hanbo Zhang, Xinlong Wang, Zhongyuan Wang, Xuguang Lan, Huaping Liu, and Xinghang Li. Scaling world model for hierarchical manipulation policies. *arXiv preprint arXiv:2602.10983*, 2026.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674–679, 1981.

Yunchao Ma, Ze Chen, Erjin Zhou, Ziming Liu, Haowei Zhang, Kai Liu, and Haoqiang Fan. Table30 V2: Evaluating generalized models by real robots at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4461–4467, 2026.

Macrodata Labs. Turning egocentric video into robot actions. https://macrodata.co/blog/turning-egocentric-video-into-3d-hand-actions, 2026. Technical blog, August 6, 2026. Accessed September 10, 2026.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 655–677, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos. *arXiv preprint arXiv:2606.19333*, 2026.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. URL https://arxiv.org/abs/2604.15483.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12254, 2025.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017.

Ropedia. Xperience-10M: A large-scale egocentric multimodal dataset with structured 3d/4d annotations. `https://huggingface.co/datasets/ropedia-ai/xperience-10m`, 2026. Dataset.

Abraham Savitzky and Marcel J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac60214a047.

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

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 *European Conference on Computer Vision (ECCV)*, pp. 478–497, 2026. doi: 10.1007/978-3-032-37196-6_26.

Wan Team et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Gaotian Wang, Kejia Ren, Andrew Morgan, Yiting Chen, Howard H. Qian, Podshara Chanrungmaneekul, and Kaiyu Hang. EgoInfinity: A web-scale 4d hand-object interaction data engine for any-view robot retargeting and video-to-action robot learning. *arXiv preprint arXiv:2606.17385*, 2026a.

Hongwu Wang, Chenhao Yu, Youhao Hu, Jiachen Zhang, Yuanyuan Li, and Shaqi Luo. BifrostUMI: Bridging robot-free demonstrations and humanoid whole-body manipulation. *arXiv preprint arXiv:2605.03452*, 2026b.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-VLA: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026c. URL `https://arxiv.org/abs/2605.30280`.

Ziyu Wang, Chenyuan Liu, Yushun Xiang, Runhao Zhang, Qingbo Hao, Hongliang Lu, Houyu Chen, Zhizhong Feng, Kaiyue Zheng, Dehao Ye, et al. The Great March 100: 100 detail-oriented tasks for evaluating embodied ai agents. *arXiv preprint arXiv:2601.11421*, 2026d. URL `https://arxiv.org/abs/2601.11421`.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, et al. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 10641–10662, 2024.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS)*, 2025a. URL `https://www.roboticsproceedings.org/rss21/p152.pdf`.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025b. URL `https://arxiv.org/abs/2511.17441`.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026a.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving VLA models in practice. *arXiv preprint arXiv:2607.06403*, 2026b.

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-Robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories. *arXiv preprint arXiv:2607.15330*, 2026.

Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, Erjin Zhou, Fan Jia, Haitao Yang, Haoqiang Fan, Haowei Zhang, Hongyang Peng, et al. RoboChallenge: Large-scale real-robot evaluation of embodied policies. *arXiv preprint arXiv:2510.17950*, 2025. URL https://arxiv.org/abs/2510.17950.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026a.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, et al. ABot-M0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026b.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, et al. World-language-action model for unified world modeling, language reasoning, and action synthesis. *arXiv preprint arXiv:2606.05979*, 2026c.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world-action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, et al. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, pp. 28213–28239, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b. URL https://arxiv.org/abs/2602.15922.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-RobotManip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a.

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.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, et al. Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026a.

Jianke Zhang, Yucheng Hu, Yanjiang Guo, Xiaoyu Chen, Yichen Liu, Wenna Chen, Chaochao Lu, and Jianyu Chen. UniJEPA: Enhancing robot policy via unified continuous and discrete representation learning. In *International Conference on Machine Learning*, 2026b.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1805–1815, 2025.

Tianle Zhang, Zhihao Yuan, Dafeng Chi, Peidong Liu, Dongwei Li, et al. JoyAI-RA 0.1: A foundation model for robotic autonomy. *arXiv preprint arXiv:2604.20100*, 2026c.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–21, 2022.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, et al. ImageWAM: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026d.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, Feng Wen, Xinyu Wang, Xingyue Quan, and Yingxue Zhang. Do world action models generalize better than VLAs? a robustness study. *arXiv preprint arXiv:2603.22078*, 2026e. URL https://arxiv.org/abs/2603.22078.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, pp. 60580–60606, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183, 2023.

A Model and Training Details

A.1 Problem Formulation

Let o_t denote the current image observation, ℓ a language instruction, $v_{t+1:t+H}$ future visual observations, and $a_{t:t+K-1}$ an action chunk. The policy models future-video and action trajectories conditioned on (o_t, ℓ). Both output spaces are trained with flow matching (Lipman et al., 2023). For a clean target x and Gaussian noise ϵ, we construct

$$x_\tau = (1-\tau)x + \tau\epsilon, \qquad u^\star(x_\tau, \tau) = \epsilon - x. \tag{6}$$

We regress a velocity field from the noisy sample to $u^\star$. Video and action use independently sampled noise levels.

A.2 Architecture

EWAM contains three depth-aligned transformer experts for semantic understanding, future-visual prediction, and action generation. The vision-language expert receives the language instruction and current image. The video expert receives a latent representation of the observed condition frame and noisy future-video latents. The action expert receives a noisy action chunk. Corresponding blocks from the three experts are executed at the same network depth, allowing the action representation to aggregate information progressively rather than waiting for a separately generated visual subgoal.

The attention mask is asymmetric. Vision-language queries attend only within the vision-language stream, and video queries attend only within the video stream. Action queries may attend to vision-language, video, and action keys. This design prevents the auxiliary generative stream from overwriting semantic representations while making both sources directly available to action computation. It also gives a well-defined quantity for measuring how action queries allocate attention among modalities at each depth.

VL layer alignment and recurrent fusion. The 28 Qwen3-VL (Bai et al., 2025) hidden states are extracted in a single Qwen3-VL forward pass before the 30-block unified computation. Writing the original Qwen3-VL outputs as $(\bar{q}_0^L, \ldots, \bar{q}_{27}^L)$, the aligned sequence is

$$(q_0^L, \ldots, q_{29}^L) = (\bar{q}_0^L, \ldots, \bar{q}_{27}^L, \bar{q}_{26}^L, \bar{q}_{27}^L). \tag{7}$$

At each unified block, Eq. 3 combines the aligned Qwen3-VL state with the accumulated VL state. The normalized fused tokens are mapped by block-specific query, key, and value projections from 2,048 dimensions to 24 heads of width 128. Video, action, and VL tensors are then concatenated and processed by one masked-attention call. For the VL query rows, the mask retains only VL keys; the same VL keys and values remain visible to action queries.

The VL-side attention result is mapped back to 2,048 dimensions by a block-specific output projection and applied through a residual update,

$$z_l^L = x_l^L + W_{O,l}^L \, \mathrm{Attn}_l^L(x_l^L). \tag{8}$$

Here Attn_l^L denotes the VL-query rows of the shared masked attention operation. The resulting z_l^L is carried to the next fusion step. No additional VL feed-forward layer follows this update because each q_l^L already includes the native Qwen3-VL layer's feed-forward computation.

The implementation used in our analysis contains 30 aligned transformer blocks, indexed from 0 to 29. Based on the observed attention transitions, we refer to blocks 0–9 as the shallow stage, blocks 10–19 as the intermediate stage, and blocks 20–29 as the deep stage. These equal-width ranges are descriptive partitions rather than architecturally hard-coded modules.

A.3 Training Objectives

Training follows two stages. During cross-embodiment pretraining, EWAM jointly optimizes future-video and action flow-matching losses. Independent noise levels are sampled for video and action targets so that the model cannot solve one modality by relying on an identical denoising state in the

other. Padded action dimensions introduced by cross-embodiment alignment are masked from the loss. The base objective is

$$\mathcal{L}_{\text{base}} = \lambda_v \mathcal{L}_{\text{video}} + \lambda_a \mathcal{L}_{\text{action}}. \tag{9}$$

Task-specific post-training retains both base losses and adds subtask-phase supervision. Annotated phases of each demonstration supply a textual label of the active subtask and a normalized phase index in $[0, 1]$ derived from the same partition:

$$\mathcal{L}_{\text{post}} = \mathcal{L}_{\text{base}} + \lambda_p \mathcal{L}_{\text{prog}} + \lambda_s \mathcal{L}_{\text{sub}}. \tag{10}$$

The scalar is regressed from the action-side representation; the textual label identifies the currently active semantic phase. Both terms are applied only during task-specific post-training.

A.4 Subtask-Aware Post-Training Details

Subtask-phase partition. Long-horizon demonstrations are divided into N semantically meaningful phases using annotated transition checkpoints. If the current observation lies in phase $k_t \in \{0, \ldots, N-1\}$, we associate a short textual description $y_t = (y_{t,1}, \ldots, y_{t,M})$ of the active subtask and define a normalized phase index

$$p_t = \frac{k_t}{\max(N-1, 1)}. \tag{11}$$

The scalar p_t is the position of the current phase in this partition, not elapsed trajectory time. It is therefore derived from the same subtask annotation rather than an independently defined progress target.

Subtask prediction. The vision-language stream receives only the high-level instruction. After the joint forward pass, a lightweight autoregressive decoder predicts y_t from a concatenated cross-attention memory, obtained by linearly projecting the final action representation h_t^A and the aligned VL tokens h_t^L with W_A and W_L:

$$\text{Mem}_t = \left[W_A\, h_t^A;\ W_L\, h_t^L\right]. \tag{12}$$

It reuses the Qwen3-VL token embedding and language-model output head:

$$p_\theta(y_t \mid \text{Mem}_t) = \prod_{j=1}^{M} p_\theta(y_{t,j} \mid y_{t,<j}, \text{Mem}_t), \tag{13}$$

where p_θ is the decoder distribution with parameters θ and $y_{t,<j}$ denotes the preceding tokens, with token-level objective

$$\mathcal{L}_{\text{sub}} = -\sum_{j=1}^{M} m_{t,j} \log p_\theta(y_{t,j} \mid y_{t,<j}, \text{Mem}_t), \tag{14}$$

where $m_{t,j}$ masks padding.

Phase-index head. We mean-pool the final action representation and regress p_t as

$$\bar{h}_t^A = \text{MeanPool}(h_t^A), \qquad \hat{p}_t = W_2\ \text{SiLU}(W_1 \bar{h}_t^A), \tag{15}$$

where W_1 and W_2 are the weights of a two-layer MLP head and $\hat{p}_t$ is the predicted phase index, with $\mathcal{L}_{\text{prog}} = \|\hat{p}_t - p_t\|_2^2$.

B Attention Analysis Details

B.1 Cross-Paradigm Attention Measurement

Figure 1 compares action-query attention routing across four policies. The measurement details are as follows.

Table 9: **Cross-task visual handoff before and after token-count normalization.** Each entry is the macro mean over 50 RoboTwin 2.0 tasks with its 95% task-bootstrap confidence interval in brackets. We report individual layers around the transition and averages outside that interval. Enrichment is measured relative to a task-specific uniform-token reference.

	VL-image		Future frame	
Layers	Mass (%)	Enrichment	Mass (%)	Enrichment
0–9	36.75 [36.45, 37.05]	1.74 [1.73, 1.76]	9.58 [9.49, 9.66]	0.23 [0.23, 0.23]
10	27.99 [27.72, 28.26]	1.33 [1.32, 1.34]	7.42 [7.34, 7.51]	0.18 [0.17, 0.18]
11	22.42 [22.22, 22.60]	1.06 [1.06, 1.07]	17.79 [17.64, 17.94]	0.42 [0.42, 0.43]
12	21.48 [21.30, 21.66]	1.02 [1.01, 1.03]	21.15 [20.89, 21.42]	0.50 [0.50, 0.51]
13	13.37 [13.15, 13.61]	0.63 [0.62, 0.65]	24.85 [24.65, 25.05]	0.59 [0.58, 0.59]
14	11.09 [10.94, 11.24]	0.53 [0.52, 0.53]	28.96 [28.47, 29.47]	0.69 [0.67, 0.70]
15	6.96 [6.80, 7.12]	0.33 [0.32, 0.34]	23.37 [23.06, 23.72]	0.55 [0.55, 0.56]
16–29	1.31 [1.28, 1.34]	0.06 [0.06, 0.06]	11.79 [11.42, 12.19]	0.28 [0.27, 0.29]

Checkpoints. All baseline weights are publicly released. Figure 1(b) uses the released Fast-WAM (Yuan et al., 2026b) weights and Figure 1(c) the released Motus (Bi et al., 2026) weights. Figure 1(a) uses the $\pi_{0.5}$ baseline checkpoint distributed with Motus, that is, a $\pi_{0.5}$ policy trained on RoboTwin 2.0 by the Motus authors. Figure 1(d) uses the in-domain EWAM checkpoint of Table 4, which is also the checkpoint used for the inference-time interventions of Table 1.

Evaluation condition. Every panel is measured on the same single RoboTwin 2.0 ADJUST BOTTLE episode under the randomized protocol, at denoising step 0. Attention allocation is stable across tasks and denoising steps, so a single episode at one denoising step is representative (Appendix B.3). The horizontal axis is the policy inference step, that is, the successive action-chunk queries within the episode; the number of steps differs across panels because the four policies complete the episode in different numbers of chunks.

Key grouping. For each model we assign every attended key to one of the sources that model possesses and apply Eq. 5 over those groups. The available sources differ by architecture: $\pi_{0.5}$ has no video stream, and Fast-WAM has no separate vision-language stream.

B.2 TOKEN-COUNT-NORMALIZED ATTENTION

The allocation R_l^m in Eq. 5 measures the total attention mass received by source m. Because the five sources contain different numbers of keys, we additionally measure attention enrichment relative to a uniform-token reference:

$$E_l^m = \frac{R_l^m}{|\mathcal{K}_m| / \sum_j |\mathcal{K}_j|}, \tag{16}$$

where $E_l^m > 1$ indicates that a source receives more attention than expected if attention were distributed uniformly over all available keys. We compute this statistic on one representative sequence from each of the 50 RoboTwin 2.0 tasks, using the robot-pretrained in-domain checkpoint of Table 4. Every sequence contains 120 condition-frame, 240 future-frame, 21 action, and 120 VL-image tokens, while VL-text length varies from 59 to 85 tokens. We therefore compute E_l^m using each task's own token counts, average over its five measured denoising steps, and then macro-average with equal task weight.

Table 9 reports both total attention mass and token-normalized enrichment for the two visual sources involved in the observed handoff. VL-image tokens are strongly enriched in shallow layers but decline rapidly with depth, while future-frame attention rises. In the cross-task macro average, future-frame raw mass first exceeds VL-image mass at layer 13, and token-normalized future enrichment first exceeds image enrichment at layer 14. The handoff therefore persists after controlling for the two-to-one difference in future-frame and VL-image token counts.

B.3 CROSS-TASK MECHANISM REPLICATION

Using the same 50-task macro-averaging protocol and the same robot-pretrained in-domain checkpoint, we next quantify how often the depth-wise mechanism occurs. For stage-level allocations,

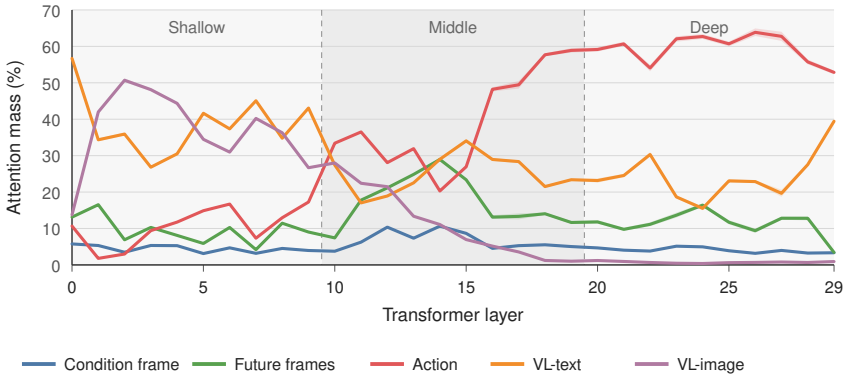


Figure 6: **Complete raw cross-task action-attention allocation.** Attention mass to all five sources is macro-averaged over 50 RoboTwin 2.0 tasks after averaging five denoising steps within each task; shaded regions show 95% task-bootstrap confidence intervals. Dashed vertical lines mark the shared shallow (0–9), intermediate (10–19), and deep (20–29) stages.

Table 10: **Cross-task replication of depth-wise specialization.** Stage allocations are macro means with 95% task-bootstrap confidence intervals. Mechanism prevalence is measured across 50 tasks; its uncertainty uses 95% Wilson intervals. Each task contributes one representative sequence.

Stage-level action-attention mass (%)			
Stage	VL	Video	Action
Shallow (0–9)	75.38 [75.24, 75.51]	14.05 [13.93, 14.16]	10.58 [10.54, 10.61]
Middle (10–19)	36.55 [36.28, 36.79]	24.31 [23.90, 24.76]	39.14 [38.86, 39.41]
Deep (20–29)	25.22 [25.02, 25.40]	15.32 [14.83, 15.84]	59.46 [59.00, 59.90]
Mechanism prevalence and transition location			
Criterion	Coverage	Estimate	Uncertainty
Shallow VL largest	50/50	100%	95% CI [92.9, 100.0]
Raw image-to-future handoff	50/50	100%	95% CI [92.9, 100.0]
Raw stable crossover	50/50	median layer 13	IQR 12–13
Normalized image-to-future handoff	49/50	98%	95% CI [89.5, 99.6]
Normalized stable crossover	50/50	median layer 14	IQR 14–14
Deep action largest	50/50	100%	95% CI [92.9, 100.0]
Denoising-step stability	50 tasks	mean $r = 0.998$	minimum $r = 0.995$

uncertainty is estimated with 95% task-bootstrap confidence intervals (CIs), while mechanism prevalence uses 95% Wilson CIs. We define the stable crossover as the first of three consecutive layers for which future-frame attention exceeds VL-image attention, and summarize its location by the median and interquartile range (IQR). Figure 6 reports the complete raw five-source routing profile. Table 10 summarizes the stage-level attention values, raw and token-normalized mechanism prevalence, crossover locations, and denoising-step stability.

The upper block of Table 10 quantifies the three-stage reallocation. VL tokens receive 75.38% of shallow attention, compared with 14.05% for video and 10.58% for action. The middle stage is substantially more distributed (36.55% VL, 24.31% video, and 39.14% action), consistent with a gradual handoff rather than a hard module boundary. In the deep stage, action attention rises to 59.46%, while VL and video attention fall to 25.22% and 15.32%, respectively.

The lower block measures how consistently this organization appears across tasks. All 50 tasks exhibit shallow VL dominance, the intermediate raw image-to-future handoff, and deep action dominance. The stable raw crossover is tightly concentrated around layers 12–13. After controlling for each task's token counts, 49 of 50 tasks still exhibit the handoff over layers 13–15, and all tasks eventually reach a stable normalized crossover, with a median at layer 14. The mean correlation of 0.998 across denoising-step profiles further shows that these depth trends are largely invariant to denoising time.

Table 11: **Checkpoint evolution of the visual handoff under the asymmetric mask.** Token-normalized enrichment E of action-query attention to VL-image and future-frame tokens by stage, with the shallow-stage raw mass of text and image attention. The crossover column gives the first layer at which token-normalized future enrichment exceeds image enrichment for three consecutive layers, followed in parentheses by the percentage of subsequent layers that maintain the ordering; "—" marks checkpoints at which no such layer exists; this token-normalized criterion can lag the raw-mass crossover of Figure 4. Measured on one representative sequence averaged over five denoising steps, using the token counts recorded for each checkpoint.

	VL-image E			Future frame E			Shallow mass (%)		
Step	0–9	10–19	20–29	0–9	10–19	20–29	Text	Image	Crossover
50	2.07	1.74	2.33	0.40	0.56	0.28	2.9	44.2	—
100	1.11	0.95	0.75	0.56	0.86	0.77	14.3	23.5	23 (71)
150	1.10	0.61	0.61	0.53	0.89	0.81	22.6	23.3	11 (84)
200	1.02	0.27	0.25	0.27	0.55	0.46	37.1	21.9	11 (95)
250	1.04	0.21	0.31	0.15	0.46	0.45	45.8	22.0	12 (89)
300	1.12	0.28	0.35	0.10	0.39	0.38	49.8	23.9	12 (78)
350	1.11	0.27	0.26	0.09	0.40	0.34	52.2	23.4	12 (78)
400	1.04	0.31	0.21	0.09	0.39	0.30	54.8	22.1	12 (89)
600	1.21	0.50	0.20	0.08	0.35	0.24	58.1	25.6	14 (62)
1,000	1.30	0.43	0.21	0.09	0.35	0.24	59.7	27.6	14 (56)
5,000	1.79	0.64	0.49	0.07	0.38	0.20	53.4	38.2	—
40,000	1.75	0.56	0.04	0.21	0.41	0.24	38.7	37.4	14 (100)

The only task without a late-middle normalized handoff is STAMP SEAL, for which future-frame enrichment is slightly below VL-image enrichment when averaged over layers 13–15 (margin -0.020). Its normalized stable crossover occurs later, at layer 17, and it still exhibits the complete raw understanding–foresight–action progression.

B.4 CHECKPOINT EVOLUTION OF THE VISUAL HANDOFF

Table 11 reports the checkpoint trajectory of the asymmetric model over twelve checkpoints, and Figure 7 plots it against training step in the same layout as the bidirectional-mask trajectory of Figure 10. The table resolves the concurrent redistribution of attention from image to text tokens and separates the two halves of the handoff.

Three phases are visible. At 50 steps, action queries attend to VL-image tokens at every depth: image enrichment is 2.07, 1.74, and 2.33 across the three stages, and text accounts for only 2.9% of shallow attention. Between 100 and 400 steps the shallow stage rapidly acquires its linguistic character. Shallow text mass rises from 14.3% to 54.8%, while deep image enrichment falls from 0.75 to 0.21. A future-over-image crossover is already detectable during this phase, but its onset shifts from layer 23 at 100 steps to layers 11–12 thereafter, and its persistence varies from 71% to 95%.

The subsequent refinement is not monotonic. At 600 and 1,000 steps, stable crossovers begin at layer 14 but persist over only 62% and 56% of the remaining layers. At 5,000 steps, shallow image enrichment rebounds to 1.79 and no stable crossover is detected. By 40,000 steps, shallow image enrichment remains high at 1.75, deep image enrichment falls to 0.04, and the layer-14 crossover persists through every subsequent layer. Thus, coarse semantic-to-action depth organization appears within the first few hundred steps, whereas the specific shallow-image-to-intermediate-future handoff fluctuates before stabilizing at convergence. All checkpoints come from the same thirty-two-GPU run, so this trajectory is not confounded by a change in global batch size.

B.5 INTERVENTION AND COUNTERFACTUAL INJECTION DETAILS

Layer-targeted interventions. The training-time variants in Table 1 are optimized independently without pretraining and can adapt to their masks; they therefore test architectural utility rather than dependence of a fixed policy. The inference-time variants instead mask selected keys in one fixed

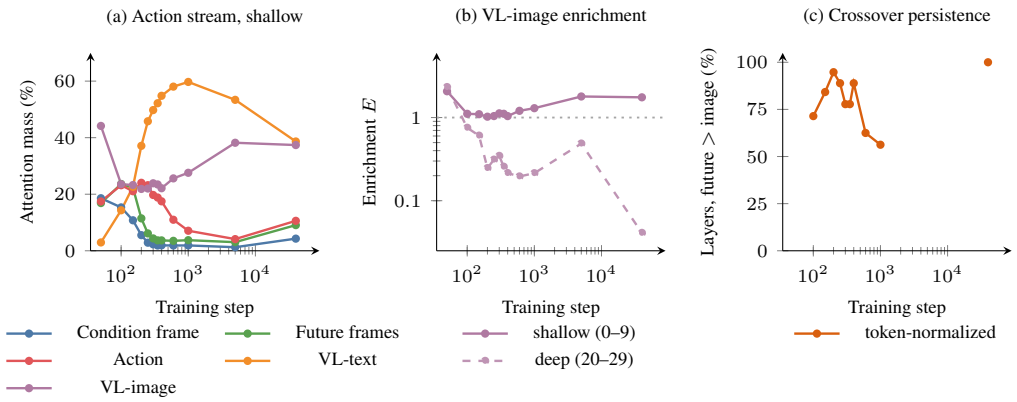


Figure 7: **Checkpoint evolution of depth-wise attention routing under the asymmetric mask.** Stage-level quantities across the trajectory of Table 11, laid out to match Figure 10 so that the two attention designs can be compared panel by panel. (a) Shallow-stage action attention over the five key sources. (b) Token-normalized VL-image enrichment at the two ends of the network, with the dotted line marking the uniform-token reference $E = 1$. (c) Fraction of layers at or beyond the stable crossover for which token-normalized future-frame enrichment exceeds VL-image enrichment, with the curve broken at checkpoints that have no stable crossover. All circles belong to the same thirty-two-GPU training trajectory.

cross-embodiment robot-pretrained checkpoint on all 50 randomized tasks, using 100 episodes per task with matched seeds.

Counterfactual injection protocol and behavior. For each of five tasks, future-frame keys and values from successful donor seed 100 are cached at layers 10–19 during the first three policy calls and injected into target seeds 43–47 at the corresponding calls. Each target's condition-frame, VL, and action representations remain unchanged. All 25 targets succeed without injection.

ADJUST BOTTLE provides an interpretable view of the intervention. A bottle whose opening faces left should be adjusted with the right arm, whereas a right-facing opening requires the left arm. Injection causes all five targets to initially follow the donor's right-arm behavior despite changes in bottle identity, orientation, and position. For left-facing seeds 43 and 46, the donor-directed first grasp misses; after injection stops, both policies re-localize the bottle and succeed on a second grasp. For right-facing seeds 44, 45, and 47, the transferred right-arm choice conflicts with the required left-arm strategy and all three episodes fail. These outcomes show that action generation depends on the content of the intermediate future-frame representations rather than merely on their presence: replacing only these features transfers the donor's arm choice and spatial target. The recovery of seeds 43 and 46 after injection stops further indicates that this influence is exerted online, while the representations are present.

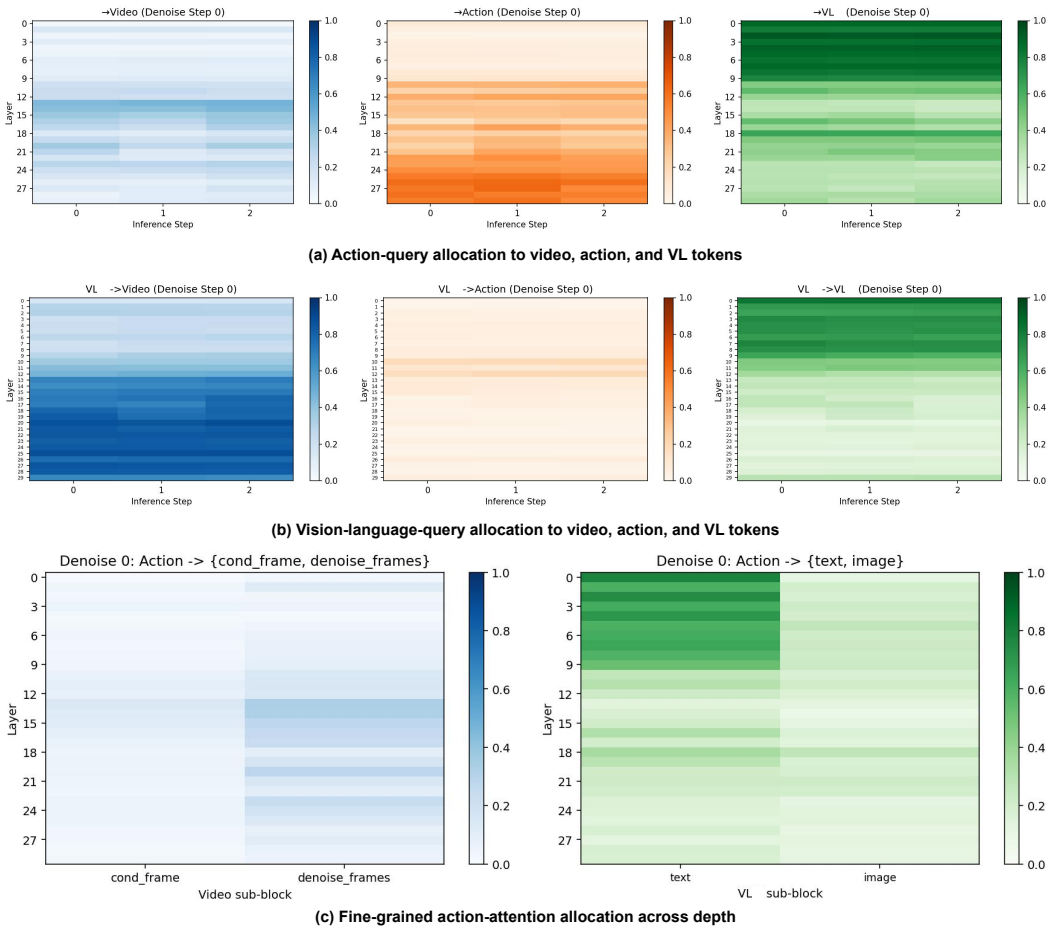


Figure 8: **Attention allocation under a fully bidirectional mask at 40,000 steps.** Rows are the thirty transformer layers. All panels use an absolute colour scale from zero to one, so intensities are comparable across panels within this figure. (a) Action-query allocation over the three key sources, the bidirectional counterpart of Figure 1(d). (b) Vision-language-query allocation, which has no counterpart in the main text because the asymmetric mask fixes this stream to self-attention at every depth. (c) Action-query allocation resolved into the two video sub-blocks and the two vision-language sub-blocks, the bidirectional counterpart of Figure 3(a). The vision-language column of (a) is dark at every depth and the image column of (c) is uniformly pale, in contrast to the shallow-to-deep gradient that the asymmetric mask produces.

B.6 Attention Routing under a Fully Bidirectional Mask

The asymmetric mask in Eq. 2 restricts the two perceptual streams to their own tokens. To determine what this restriction contributes, we train a variant in which every stream may attend to every other stream, so that $A \leftarrow \{V, A, L\}$, $V \leftarrow \{V, A, L\}$, and $L \leftarrow \{V, A, L\}$, and apply the same depth-wise attention analysis. As for the asymmetric trajectory of Appendix B.4, the variant is trained without robot pretraining; all other components, objectives, and data are unchanged. We report two independently trained runs: a trajectory of checkpoints between 50 and 5,000 optimizer steps trained on four GPUs, and a converged 40,000-step run trained on thirty-two GPUs. Because the two runs use different global batch sizes, we treat the converged run as an independent replication rather than as a continuation of the trajectory. Both are analysed on a single representative sequence with ten denoising steps.

The coarse depth organization does not depend on the mask. At convergence, the bidirectional variant reproduces the three-stage action-attention profile reported in Section 4, which is visible in the aggregate maps of Figure 8(a). VL tokens receive 85.6% of shallow action attention, the intermediate stage is distributed across VL, video, and action tokens (40.6%, 29.0%, and 30.5%),

Table 12: **Depth-differentiated use of VL-image tokens under the two masks.** Token-normalized enrichment of action-query attention to VL-image tokens, by stage. Values above one indicate that a source receives more attention than a uniform allocation over all keys would give. Both rows are converged 40,000-step runs without robot pretraining; asymmetric values are the 40,000-step row of Table 11.

Mask	Shallow (0–9)	Middle (10–19)	Deep (20–29)	Shallow/Deep
Asymmetric (ours)	1.75	0.56	0.04	43.8
Fully bidirectional	0.81	0.70	0.67	1.2

and action self-attention rises to 49.9% in the deep stage. The concentration of predicted-future usage in intermediate layers also recurs: future-frame enrichment rises from 0.14 in the shallow stage to 0.50 in the middle stage before falling to 0.31, peaking at layer 14, against a corresponding 0.21–0.41–0.24 profile for the asymmetric model at the same step. At the level of the three-source aggregate, and on the future-frame side of the visual transition, the depth-wise organization studied in this paper is therefore not produced by the asymmetric mask and is not an artifact of one particular choice of cross-stream connectivity. We regard this as a partial replication that strengthens the generality of the phenomenon. As in Section 4, however, the stage-level aggregate conceals the sub-modality structure, and it is there that the two configurations diverge.

The role of the semantic stream does depend on the mask. The two configurations differ in what the vision-language stream itself computes. Under the asymmetric mask the vision-language stream attends only to its own tokens at every depth, by construction. Under the bidirectional mask it does not, as Figure 9 and Figure 8(b) show: its shallow layers retain self-attention, but from roughly layer 11 onward video tokens become the dominant key source, and in the deep stage the vision-language stream directs 82.0% of its attention to video tokens while retaining only 15.3% on its own. In other words, the semantic stream ceases to act primarily as a source of semantic keys and becomes a consumer of the generative stream.

The consequence on the action side is that the visual handoff itself does not survive, even though its future-frame half does. A handoff requires two things: an early reliance on the observed image, and its subsequent release in favour of predicted future frames. Table 12 shows that the bidirectional variant has neither. Under the asymmetric mask, action queries use VL-image tokens in a sharply depth-specific way: image enrichment is 1.75 in the shallow stage, well above the uniform-token reference, and falls to 0.04 in the deep stage, a shallow-to-deep ratio of roughly 44. The model reads the current image where it grounds the instruction and gives it up once actions are being refined. Under the bidirectional mask this differentiation is absent. Figure 8(c) shows the pattern directly: the image column is pale at every depth, whereas the text column is dark throughout the shallow stage. Image enrichment is flat across depth (0.81, 0.70, and 0.67 for the three stages, a ratio of 1.2), never reaches the uniform-token reference at the stage level, and attains its per-layer maximum at layer 18 rather than in the shallow stage. Its shallow stage is instead almost purely linguistic, with text enrichment of 6.29 against image enrichment of 0.81, so the shallow visual grounding that the asymmetric model performs has no counterpart here.

The bidirectional variant does exhibit layers at which future-frame enrichment numerically exceeds image enrichment, at layers 13–15. That inequality, however, holds between two signals that both remain below the uniform-token reference throughout the network, and it is produced entirely by the rise of future-frame attention rather than by any release of the observed image: image enrichment recovers to 0.67 in the deep stage rather than collapsing. We therefore do not regard it as the same phenomenon. Consistent with this, the inequality is not sustained. Measuring the fraction of layers at or beyond the stable crossover for which token-normalized future enrichment exceeds image enrichment, the bidirectional variant reaches only 24%, whereas the asymmetric model maintains the ordering at every layer from the crossover to layer 29.

Training dynamics. Figure 10 traces these quantities across training. Three observations follow. First, the vision-language stream moves in opposite directions at the two ends of the network. Its shallow layers reclaim their own tokens early, with self-attention rising from 27.1% at 50 steps to 43.0% at 150 and 66.4% at 250. Its deep layers do not: deep self-attention recovers only to 39.6% near 350 steps and then declines steadily, reaching 27.0% at 5,000 steps in the four-GPU run and

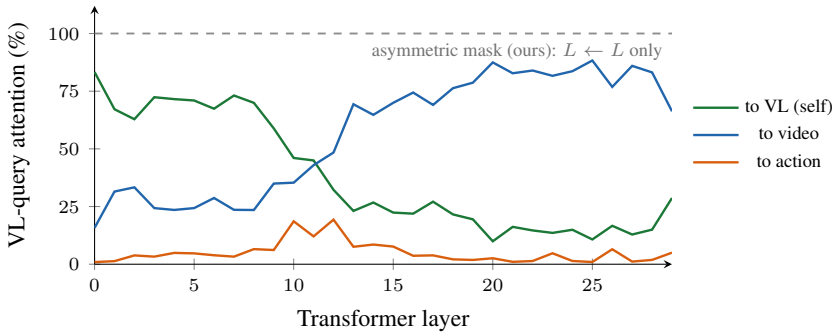


Figure 9: **Vision-language-query attention allocation across depth under a fully bidirectional mask.** The vision-language stream retains self-attention only in shallow layers; from roughly layer 11 onward video tokens become its dominant key source, reaching 82.0% in the deep stage. The dashed line marks the asymmetric mask of Eq. 2, under which this curve is fixed at 100% self-attention by construction. Curves are averaged over ten denoising steps at 40,000 steps.

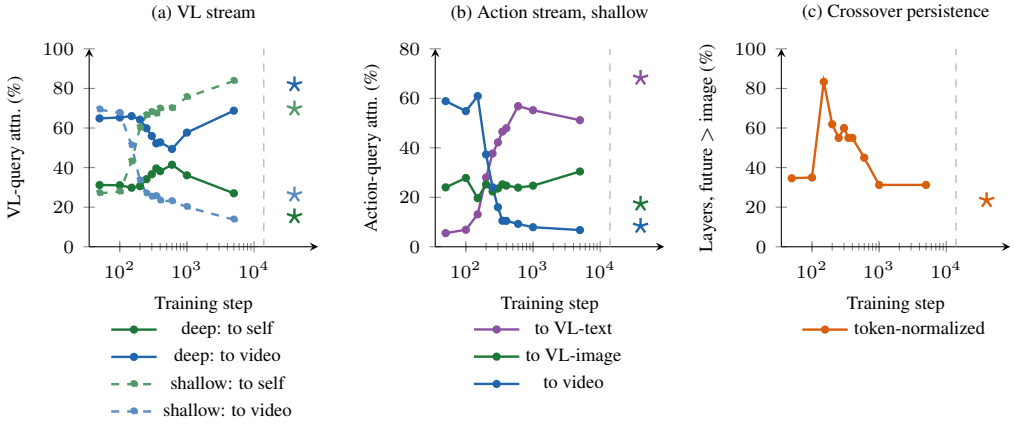


Figure 10: **Attention routing under a fully bidirectional mask across training.** (a) Vision-language-query allocation, split into self-attention and attention to video, for the deep stage (layers 20–29, solid) and the shallow stage (layers 0–9, dashed). (b) Shallow-stage action-query allocation. (c) Fraction of layers at or beyond the stable crossover for which token-normalized future-frame enrichment exceeds VL-image enrichment. Circles joined by lines are the four-GPU checkpoint trajectory; stars to the right of the dashed rule are the independently trained thirty-two-GPU run at 40,000 steps, which uses a different global batch size and is therefore shown as a separate measurement rather than as a continuation of the trajectory.

15.3% in the independently trained converged run. Attention to video moves oppositely at the two ends over the same span: the shallow share falls from 69.3% to 13.7% between 50 and 5,000 steps and stands at 26.4% in the converged run, whereas the deep share rises from 64.9% to 68.8% over the same interval and reaches 82.0% at convergence. The deep-layer capture is therefore progressive rather than a property of initialization. Second, the action stream reorganizes over the same short interval: between 150 and 250 steps, shallow action attention to VL-text rises from 13.1% to 37.7% while shallow action attention to video falls from 60.9% to 24.0%. Because the shallow vision-language recovery occupies exactly this window, the two streams reorganize together rather than in a resolvable order. Third, crossover persistence declines monotonically over training, from 83% at 150 steps to 55% at 400 steps and 31% at 5,000 steps, with the converged run at 24%. Under a bidirectional mask the future-over-image ordering therefore becomes less rather than more sustained as training proceeds, and does not consolidate into a handoff.

Scope. These measurements characterize where each stream retrieves information, not what its representations encode. Attention to video keys does not by itself establish that semantic content has been displaced from the vision-language representation, since residual connections can carry shallow-layer semantics through deeper blocks irrespective of where those blocks attend. Estab-

lishing whether the routing change is accompanied by a representational change requires decoding semantic targets directly from the deep vision-language activations of both configurations, which we leave to future work. We therefore restrict the claim to routing: the asymmetric mask is not what produces the coarse depth-wise organization, but it is what makes the observed image a source that shallow layers rely on and deeper layers release, and what keeps the vision-language stream a pure semantic source. It is therefore also what makes token identity an interpretable indicator of information source in the analysis of Section 4.

C Experimental Details

C.1 RoboTwin 2.0

RoboTwin 2.0 contains 50 bimanual manipulation tasks and applies structured randomization to clutter, illumination, background, tabletop height, and language (Chen et al., 2026b). For the standard multi-task experiment, we evaluate one policy across all tasks in both Clean and Randomized scenes. Each reported aggregate is the unweighted mean of per-task success rates, with 100 evaluation episodes per task and setting.

For clean-to-random evaluation, no target-embodiment randomized data are used during downstream adaptation; the adapted policy is evaluated directly in the fully randomized Hard setting. In contrast, the in-domain results in Table 4 use both clean and randomized data during downstream adaptation. Table 13 summarizes the RoboTwin evaluation settings detailed above. Table 14 reports the complete task-level Clean and Randomized results for the in-domain multi-task evaluation.

Table 13: **RoboTwin evaluation protocols used in this work.**

Protocol	Tasks	Evaluation/task	Test domain
Standard Clean	50	100	Clean
Standard Randomized	50	100	Randomized
Clean-to-Random	50	100	Randomized Hard

C.2 LIBERO

LIBERO evaluates knowledge transfer across spatial, object-centric, goal-conditioned, and long-horizon manipulation suites (Liu et al., 2023). We train and evaluate a single EWAM policy under the same observation and action interface used by the comparison methods. The primary metric is average task success, reported separately for LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long in addition to the overall mean.

C.3 Real-World Evaluation Details

We evaluate the same model family on a Franka single-arm platform, a Dobot dual-arm platform, and a Unitree G1-D humanoid. The selected tasks cover single-arm object rearrangement, bimanual coordination, and contact-rich liquid transfer. Franka evaluation includes stacking bowls and placing an object into a box. Dobot evaluation includes pouring water, tidying a desk, and manipulating a towel. Unitree G1-D evaluation includes kettle pouring and pouring beans. Together, these tasks test grasp stability, object rearrangement, deformable-object manipulation, bimanual coordination, and controlled pouring. Figure 11 illustrates the real-world robot evaluation setup and corresponding tasks. Table 16 summarizes the embodiments and tasks included in this evaluation.

Every method is evaluated with 10 trials per task on each platform, and all real-world success rates reported in Tables 6, 7, and 20 are computed over those 10 trials. An episode is counted as successful only when the task-level terminal condition is satisfied, rather than when an intermediate grasp or motion succeeds. Initial object poses are varied across trials within the reachable workspace. We use the same task instruction for EWAM and the corresponding baseline and do not manually intervene after execution begins.

Table 14: **RoboTwin 2.0 per-task success rates (%).** Each baseline is the strongest Randomized-setting representative of its paradigm. EWAM is evaluated on 100 episodes per task and setting.

	ACE-Ego-0 (Li et al., 2026a)		Fast-WAM (Yuan et al., 2026b)		WLA-0 (Yang et al., 2026c)		EWAM	
Task	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	100	100	100	100	100	100	100	100
Beat Block Hammer	98	92	99	97	95	87	94	95
Blocks Ranking RGB	98	97	100	100	98	98	99	97
Blocks Ranking Size	89	91	94	98	93	85	76	85
Click Alarmclock	52	38	100	100	99	100	100	100
Click Bell	66	71	100	100	100	100	100	100
Dump Bin Bigbin	100	97	97	96	90	94	97	97
Grab Roller	100	100	100	100	100	100	100	100
Handover Block	96	85	95	81	96	87	94	92
Handover Mic	91	94	99	100	92	93	98	98
Hanging Mug	29	31	58	62	69	47	72	69
Lift Pot	100	100	100	100	100	100	100	99
Move Can Pot	100	98	90	88	98	99	96	97
Move Pillbottle Pad	100	100	100	99	100	97	96	99
Move Playingcard Away	100	98	100	100	99	100	100	100
Move Stapler Pad	90	89	77	64	92	75	68	66
Open Laptop	100	98	98	100	99	100	97	99
Open Microwave	91	85	62	45	97	92	100	100
Pick Diverse Bottles	84	86	80	85	95	79	85	91
Pick Dual Bottles	89	88	100	96	100	83	95	89
Place A2B Left	95	96	95	93	77	76	95	86
Place A2B Right	90	94	93	99	75	75	96	91
Place Bread Basket	92	93	91	93	91	91	95	99
Place Bread Skillet	94	89	90	93	94	85	92	96
Place Burger Fries	98	100	96	99	95	98	99	97
Place Can Basket	78	82	71	69	87	78	82	89
Place Cans Plasticbox	100	98	99	96	100	98	100	100
Place Container Plate	98	100	96	100	99	99	99	98
Place Dual Shoes	95	96	94	88	94	92	95	95
Place Empty Cup	100	100	100	100	99	100	100	99
Place Fan	94	93	96	96	94	94	96	95
Place Mouse Pad	96	95	83	89	89	88	86	74
Place Object Basket	93	89	89	88	82	84	89	83
Place Object Scale	95	92	90	97	99	96	93	95
Place Object Stand	95	94	90	94	99	92	97	97
Place Phone Stand	91	98	97	99	95	98	94	82
Place Shoe	100	100	96	99	100	99	98	99
Press Stapler	98	98	90	97	99	97	94	100
Put Bottles Dustbin	94	93	95	90	89	85	84	81
Put Object Cabinet	82	79	94	89	82	84	61	71
Rotate QRcode	94	95	93	89	91	91	96	89
Scan Object	95	97	89	92	96	95	76	90
Shake Bottle	100	100	100	100	99	100	100	100
Shake Bottle Horizontally	100	100	100	100	99	100	100	100
Stack Blocks Three	87	82	95	97	95	91	100	99
Stack Blocks Two	100	100	100	100	100	100	100	100
Stack Bowls Three	80	85	80	81	86	84	91	88
Stack Bowls Two	96	98	92	98	96	99	97	97
Stamp Seal	94	100	90	94	95	84	99	99
Turn Switch	59	57	61	59	39	32	78	78
Average	91.12	90.62	91.88	91.78	92.94	90.02	**92.98**	**92.80**

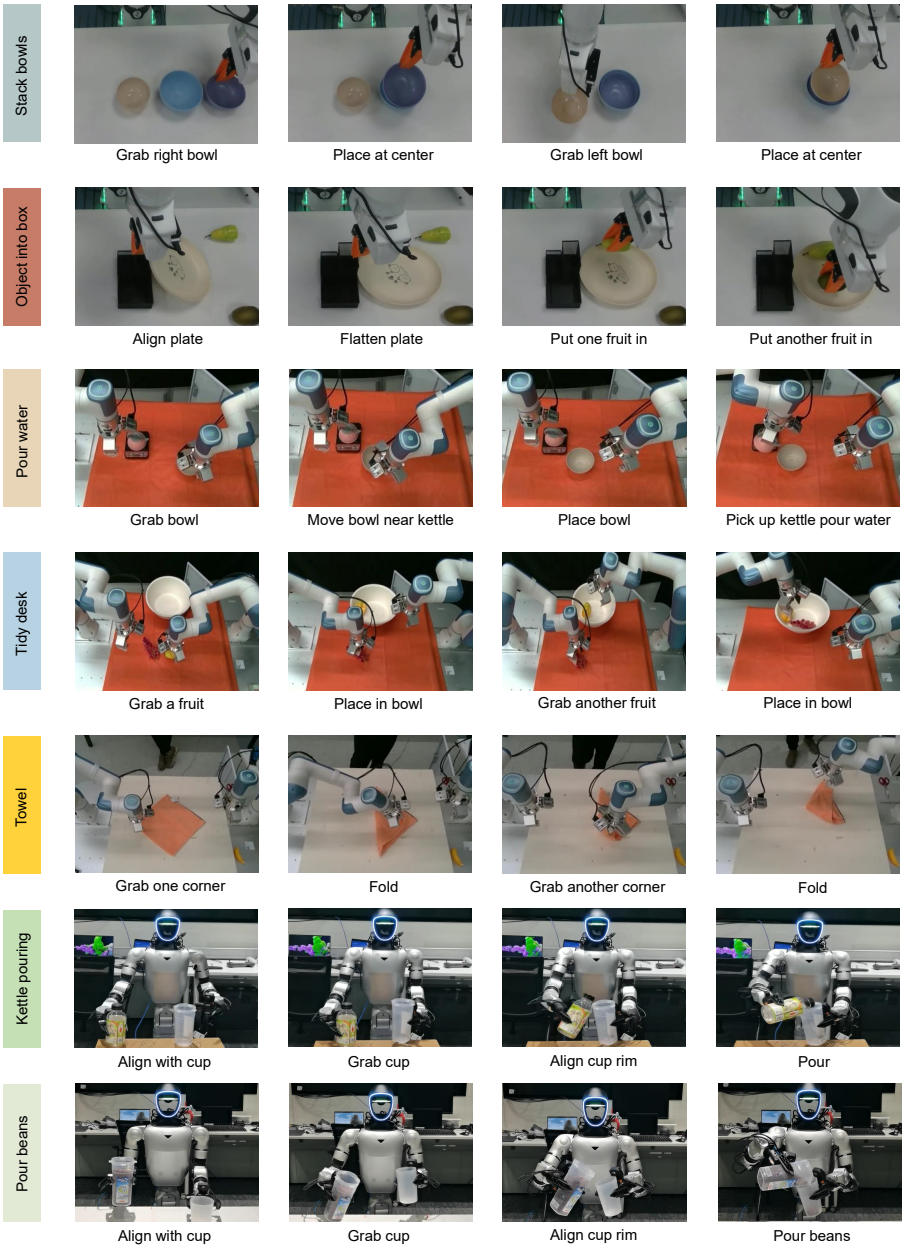


Figure 11: **Seven real-world manipulation tasks evaluated on Franka, Dobot, and Unitree G1-D platforms.**

Table 15: **Full LIBERO results by suite.** Success rates (%); dashes indicate unreported suite-level scores.

Paradigm	Model	Spatial	Object	Goal	Long	Average
VLA	$\pi_{0.5}$ (Black et al., 2025a)	98.8	98.2	98.0	92.4	96.9
	ABot-M0 (Yang et al., 2026b)	98.8	99.8	**99.0**	96.6	98.6
	Qwen-VLA-Instruct (Wang et al., 2026c)	–	–	–	–	97.9
WAM	Fast-WAM (Yuan et al., 2026b)	98.2	**100.0**	97.0	95.2	97.6
	LingBot-VA (Li et al., 2026b)	98.5	99.6	97.2	**98.5**	98.5
	LaWAM (Chen et al., 2026a)	**99.4**	99.6	98.4	97.0	98.6
VLA+WAM	Motus (Bi et al., 2026)	96.8	99.8	96.6	97.6	97.7
	WLA-0 (Yang et al., 2026c)	99.0	**100.0**	97.8	97.6	98.6
	EWAM (Ours)	98.6	99.8	98.6	98.2	**98.8**

Table 16: **Overview of the seven real-world evaluation tasks.** The evaluation spans single-arm, dual-arm, and humanoid dual-arm embodiments.

Platform	Embodiment	Tasks
Franka	Single arm	Stack bowls; object into box
Dobot	Dual arm	Pour water; tidy desk; towel manipulation
Unitree G1-D	Humanoid dual arm	Kettle pouring; pour beans

C.4 Long-Horizon Subtask Evaluation

This appendix details the two long-horizon evaluations of Section 5.3: three instruction-conditioned block tasks in simulation, and Dual-Basket Sorting on a Unitree G1-D.

C.4.1 Instruction-Conditioned Block Tasks

The three tasks used in Section 5.3 are adapted from the Blocks Ranking and Blocks Stacking layouts of RoboTwin and are distinct from the 50-task suite evaluated in Sections 5.1 and 5.1. In the original designs, three colored blocks (red, green, and blue) are initialized at random table positions and must be arranged in a fixed red–green–blue order, from left to right for ranking and from bottom to top for stacking. Our variants preserve the object set, workspace, and embodiment, and modify only the language interface. Each instruction specifies a sequence of local objectives that changes within the episode, so later stages remain conditioned on completing the currently active subtask.

Instructed Ranking. The policy grasps the three blocks in one instructed order and places them along a left-to-right line in a second instructed order. The two orders need not coincide. For example, an instruction may require grasping blue–red–green and placing red–green–blue from left to right.

Instructed Stacking. The policy grasps the three blocks in an instructed order and stacks them bottom to top in that same order. With three blocks this admits only $3! = 6$ instruction variants. Because the stacking order is not specified independently, an episode that completes the stack necessarily completes the full task, so Stack and Full coincide in Table 8. For example, an instruction may require grasping blue–red–green and stacking blue–red–green from bottom to top.

Instructed Ranking & Stacking. This task composes the two skills within a single episode. The policy first completes an instructed ranking, grasping in one specified order and placing from left to right in a second, and then restacks the same blocks bottom to top in a third instructed order. For example, an instruction may require grasping blue–red–green, placing red–green–blue from left to right, and then stacking green–red–blue from bottom to top.

Evaluation protocol. Each task is evaluated for 200 episodes. Both variants share the same unified backbone, pretraining, and post-training data, and differ only in whether the subtask-phase heads are trained. Grasp, Place, and Stack are the percentages of episodes that complete the named stage, with Place and Stack counted only when the currently instructed target is met; they are computed

independently over the same 200 episodes and are therefore not nested, so a later-stage rate may exceed an earlier-stage rate. Full is complete-episode success, counted only when the terminal spatial configuration satisfies every specified order. Dashes in Table 8 mark stages a task does not define: Stack is absent from instructed ranking, and Place is absent from instructed stacking. Because the commanded orders differ from the fixed convention of the original tasks, success requires finishing each active subtask and then the remaining sequence, rather than reproducing a memorized layout.

C.4.2 Real-World Dual-Basket Sorting

We evaluate the same with- versus without-supervision pair on a Unitree G1-D Dual-Basket Sorting sequence. Four objects are placed on the table: a doll, a marker, a crumpled tissue, and a used water bottle. The policy must place useful objects (the doll and the marker) in the right basket and waste (the tissue and the bottle) in the left basket, thereby clearing the table. We score each trial by the number of objects placed in the designated basket (out of four); a trial is fully successful only when all four objects are cleared.

With subtask-phase supervision, ten trials complete $4, 3, 0, 1, 3, 3, 2, 3, 1, 2$ objects (mean $2.2/4$); one trial clears the table. After repeated failures on the marker, the policy typically switches to another remaining object and later returns, rather than persisting on the same failed grasp. Without this supervision, ten trials complete $0, 1, 1, 1, 2, 1, 2, 1, 1, 0$ objects (mean $1.0/4$), with no full clearance. After a failed grasp the policy typically retries the same object and does not reallocate to an easier remaining target.

Together with the block tasks, these results indicate that subtask-phase labels supply a current-goal signal. Training the policy to recognize and complete the active phase increases later-stage completion and full-sequence success on both the instructed block tasks and the physical multi-object sequence.

Figure 12 illustrates the behavioral contrast. With subtask-phase supervision (top), the policy maintains a clear local objective and clears the table: it successfully grasps each of the four objects and places them in the designated baskets. Without this supervision (bottom), rollouts frequently trigger grasp height errors and grasp pose-and-position errors. The policy then retries the same failed grasp rather than switching to another object, so progress stalls and the table is not cleared.

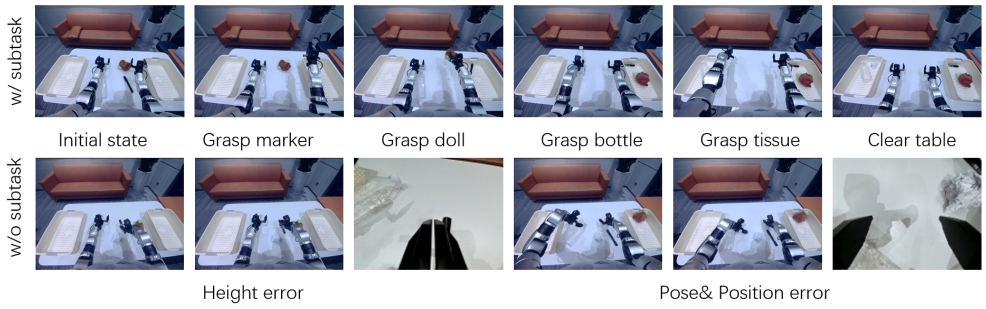


Figure 12: **Dual-Basket Sorting on Unitree G1-D.** Top: a successful rollout with subtask-phase supervision that clears the table (initial state, grasp marker, grasp doll, grasp bottle, grasp tissue, and clear table). Bottom: two failure modes that occur frequently without this supervision. The first three frames show grasp height errors; the last three show grasp pose-and-position errors. These errors lead to repeated attempts on the same object and stalled task progress.

C.5 Inference Latency and Real-Time Deployment

Inference configuration and timing. EWAM generates a chunk of 16 actions, each with 16 dimensions, in 258 ms on a single RTX 5090 using five denoising (flow-integration) steps, including chunk-overlap correction (Table 17). Timings are measured at batch size one in bfloat16 after warm-up, with GPU synchronization around each inference call. We report the minimum of three repetitions, which differ by at most 1 ms.

Reducing inference cost. We omit VAE decoding of predicted future frames because robot control requires only the action output. The video expert and the latent future features used by action

Table 17: **Inference latency per action chunk on a single RTX 5090.** Each chunk contains 16 actions; step counts refer to denoising. The baseline includes future-frame VAE decoding and does not use full-graph compilation. Subsequent rows apply the listed changes cumulatively.

Configuration	10 steps (ms)	5 steps (ms)
Baseline inference	1668	–
Without future-frame VAE decoding	1128	578
+ graph-compatible RoPE and full-graph compilation	461	254
+ chunk-overlap correction	464	**258**

Table 18: **Action-output inference latency under a common benchmark configuration.** All models use a single RTX 5090, batch size one, bfloat16, and chunks of 16 actions with 16 dimensions each. Baseline weights are randomly initialized. The final column reports the marginal cost of one denoising step.

Model	5 steps (ms)	10 steps (ms)	Per step (ms)
VLA			
$\pi_{0.5}$ (Black et al., 2025a)	50	60	1.9
Xiaomi Robotics-1 (Xiaomi Robotics Team et al., 2026)	186	321	27.1
LingBot-VLA-4B (Wu et al., 2026a)	293	504	42.3
WAM			
Fast-WAM (Yuan et al., 2026b)	216	401	37.1
VLA+WAM			
Motus (Bi et al., 2026)	901	1901	200.0
EWAM (Ours)	258	464	41.2

tokens remain active. Rewriting rotary position embeddings to avoid graph breaks enables full-graph compilation of each denoising step. At ten denoising steps, these changes reduce chunk latency from 1,668 to 461 ms; overlap correction adds 3 ms, giving 464 ms in total and a $3.6\times$ speedup over the original implementation. With five steps, the corresponding timings are 254 ms before correction and 258 ms after correction. The reduction in denoising steps changes the sampling procedure and is separate from the implementation optimizations. Neither float16 nor the tested FP8 implementations improved on compiled bfloat16.

Action execution and replanning. With a temporal stride of four, a 16-action prediction has a nominal horizon of $16 \times 4/30 = 2.13$ s at a 30 Hz command rate. The measured 258 ms inference time is approximately 12% of this horizon, providing a latency margin relative to the nominal prediction horizon. Real-time chunking (Black et al., 2025c) generates a new chunk while the previous one executes and constrains generation using the remaining actions to improve continuity. The command execution rate and the observation-conditioned replanning rate are distinct: the latter depends on inference latency and the execution schedule.

Comparison with other policies. Table 18 compares action-output inference on the same GPU, with batch size one, bfloat16, and the same action-chunk shape. Baseline models use randomly initialized weights, so the results characterize their execution cost under this benchmark configuration. At five denoising steps, EWAM takes 258 ms versus 901 ms for Motus, while remaining slower than $\pi_{0.5}$, Xiaomi Robotics-1, and Fast-WAM.

D Human Egocentric Data: Retargeting, Pretraining, and Co-Training

D.1 Related Work on Human Egocentric Data

Learning manipulation from human egocentric video. Robot demonstrations are expensive and embodiment-specific, which has motivated a long line of work on transferring manipulation knowledge from human video. Large egocentric corpora such as Ego4D (Grauman et al., 2022), EPIC-Kitchens (Damen et al., 2018), and Something-Something v2 (Goyal et al., 2017) supply broad coverage of everyday manipulation, and methods such as R3M (Nair et al., 2022) and VC-

1 (Majumdar et al., 2023) use them to pretrain visual representations that are subsequently frozen or fine-tuned inside a policy. Because these methods supply no action supervision, the resulting transfer is confined to the perceptual encoder. A second family extracts a surrogate action signal, either through future-frame prediction in GR-1 (Wu et al., 2024) or by quantizing inter-frame change into latent action tokens in LAPA (Ye et al., 2025); the surrogate is learned jointly with control but does not correspond to a physical motion of the hand. A third family makes the human hand itself the end-effector. VITRA (Li et al., 2025) converts in-the-wild egocentric video into hand-centric vision-language-action episodes and pretrains a policy on them, EgoDex (Hoque et al., 2026) provides large-scale egocentric video with on-device hand tracking, and EgoMimic (Kareer et al., 2025) co-trains robot demonstrations with human egocentric data captured through matched hardware. These approaches establish that human hand motion is a usable action signal, but they generally either keep the human stage in its own action space and adapt afterwards, or rely on hardware that makes the human and robot viewpoints identical by construction. Our human stage differs in scope rather than in ambition: we place three heterogeneous egocentric corpora and the downstream robot data in a single camera-relative end-effector action space, so the same joint video–action objective and the same layer-aligned experts are trained throughout, and the human stage transfers a shared perceptual and predictive representation rather than a separately parameterized module.

Acquiring action labels from human demonstrations. Wearable acquisition systems such as BifrostUMI combine human motion sensing with instrumented manipulation interfaces to collect demonstrations without the target robot (Wang et al., 2026b). Video-based processing pipelines reconstruct hand motion and camera trajectories to obtain metric action supervision (Macrodata Labs, 2026). EgoInfinity and Do as I Do further recover hand–object interactions and map reconstructed motion to robot embodiments (Wang et al., 2026a; Paliwal et al., 2026). Our acquisition pipeline combines measured wrist motion with visual hand articulation and interaction-state estimation to produce the observation and action streams required by our bimanual training representation.

D.2 Pretraining Setup

Robot demonstrations are costly and embodiment-specific, whereas egocentric human video records a far wider range of manipulation behavior from a viewpoint that already resembles a head-mounted robot camera. The obstacle to using it is not perception but action: a human hand and a robot arm share no action space, so human trajectories cannot simply be written into the padded representation that aligns heterogeneous robot embodiments.

A camera-relative end-effector action space. Our starting point is that the egocentric camera is the one reference frame that a human recording and a robot platform both provide. We therefore express manipulation as wrist motion relative to that camera. Let (R_{we}, t_{we}) denote the wrist pose in world coordinates at an anchor frame, $(R_{we^\star}, t_{we^\star})$ its pose at a later frame, and R_{wc} the orientation of the egocentric camera at the anchor. We first form the motion in the wrist's own frame,

$$R_{ee^\star} = R_{we}^\top R_{we^\star}, \qquad t_{ee^\star} = R_{we}^\top \big(t_{we^\star} - t_{we}\big), \tag{17}$$

where $R_{ee^\star}$ and $t_{ee^\star}$ are the rotation and translation of the wrist from the anchor to the later frame, expressed in the anchor wrist frame; we then map it into the camera frame through $R_{ce} = R_{wc}^\top R_{we}$:

$$\Delta R_c = R_{ce} R_{ee^\star} R_{ce}^\top, \qquad \Delta t_c = R_{ce}\, t_{ee^\star}, \qquad \Delta\omega_c = \log_{SO(3)}\big(\Delta R_c\big), \tag{18}$$

where ΔR_c and Δt_c are the same wrist motion expressed in the anchor camera frame and $\log_{SO(3)}$ maps ΔR_c to its axis–angle vector $\Delta\omega_c$. Each hand contributes a translation and a rotation vector together with two reserved slots, giving a sixteen-dimensional action

$$a = \big[\, \Delta t_c^L,\, \Delta\omega_c^L,\, 0,\, 0,\, \Delta t_c^R,\, \Delta\omega_c^R,\, 0,\, 0 \,\big], \tag{19}$$

where superscripts L and R denote the left and right hands, and the two reserved dimensions for each hand are excluded from the loss using the same validity mask as for cross-embodiment padding. All actions in a chunk are referenced to the same anchor pose and the same anchor camera, so the i-th action is the displacement from the anchor to frame $i{+}1$ rather than a step-to-step increment; the state is the current wrist pose expressed in that camera frame. This construction removes dependence on the world-frame origin and axis convention while preserving metric translation magnitudes; it does not normalize embodiment scale or remove kinematic differences. It gives human wrists and robot end-effectors a common task-space layout. On the robot side, the reserved slots carry gripper commands.



Figure 13: **Egocentric demonstration collection system.** A Lumos Ego headset records fisheye RGB, time-of-flight depth, and camera motion, while two wrist-mounted Vive Tracker 3.0 devices measure metric six-degree-of-freedom wrist motion.

Reducing heterogeneous sources to a common frame. The three corpora we pretrain on expose camera geometry in different forms, and each requires its own route to the anchor camera pose. VITRA-1M (Li et al., 2025) publishes world-to-camera extrinsics, which we invert; EgoDex (Hoque et al., 2026) stores a camera-to-world transform directly; and Xperience (Ropedia, 2026) provides a SLAM body pose T_{wb} together with a stereo-rectified calibration T_{cb}, from which the camera pose follows as $T_{wc} = T_{wb}T_{cb}^{-1}$. Once a source is reduced to world-frame wrist poses and an anchor camera pose, all three enter the identical computation in Eq. 18, so a given dimension carries the same physical meaning across the mixture. Per-hand tracking validity, derived from each source's own confidence or visibility signal, masks that hand's six dimensions whenever the estimate is unreliable, which keeps invalid reconstructions from contributing gradients. No action normalization is applied at this stage. The coordinate conversion preserves each source's metric scale, so consistent units and calibration remain necessary.

D.3 Robot-Compatible Demonstrations from Egocentric Capture

Egocentric human demonstrations acquired through wearable capture (Wang et al., 2026b) and visual reconstruction (Macrodata Labs, 2026) provide manipulation experience, but their motions cannot be used directly as robot actions: head motion changes the observed hand trajectory, and human wrists and robot tools have different interaction frames. We construct robot-compatible demonstrations by separating task-space motion and grasp timing from the joint configurations that realize them. Metric wrist tracking and visual hand geometry define the human-side reference; interaction-aware tool alignment and sequence optimization convert it into bimanual robot trajectories. Paired with egocentric images, these references support task-specific human–robot co-training on G1-D, complementing the wrist-motion pretraining in Section D.2.

Metric hand-motion reconstruction. Recovering manipulation motion requires separating wrist movement from camera movement while retaining finger articulation. We combine a Lumos Ego headset, which records fisheye RGB, time-of-flight depth, and camera poses from simultaneous localization and mapping (SLAM), with two wrist-mounted Vive Tracker 3.0 devices that measure metric six-degree-of-freedom motion (Figure 13).

HaWoR (Zhang et al., 2025) estimates wrist poses and hand parameters from the egocentric video; MANO (Romero et al., 2017) decodes the hand parameters into articulated geometry. Camera poses

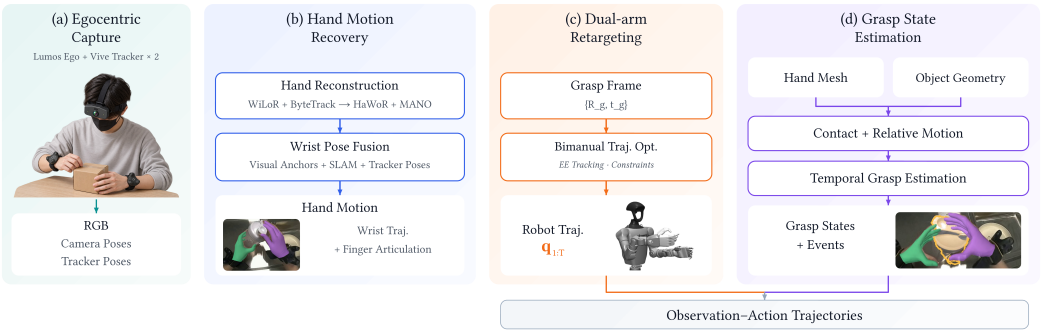


Figure 14: **Egocentric capture and observation–action trajectory construction.** (a) A Lumos Ego headset and two wrist-mounted Vive Trackers record egocentric RGB, camera poses, and metric wrist motion. (b) Visual hand reconstruction and wrist-pose fusion recover hand motion from RGB, SLAM, and Tracker observations. (c) A grasp-frame construction followed by bimanual trajectory optimization produces robot-compatible arm trajectories. (d) Projected hand geometry and two-dimensional object masks provide evidence for temporal grasp-state estimation. The inferred commands supply grasp/release timing for target construction and robot output. FK-derived robot reference trajectories and grasp states are paired with the captured human images to form observation–action trajectories.

map the visual estimates into the SLAM world frame, where the registered Tracker measurements provide metric motion constraints. This combines measured wrist motion with the finger geometry needed for tool alignment and grasp inference. Appendix D.4 gives the visual tracking and temporal processing details.

Interaction-aware action alignment. Robot action targets must specify tool motion and when to grasp or release. We infer grasp timing from the relation between the hand and the manipulated object. SAM 3 masks (Carion et al., 2026), propagated by Lucas–Kanade optical flow (Lucas & Kanade, 1981), provide image-space object geometry. Projected finger-to-mask distances, object occupancy within the pinch region, and hand-aperture changes supply grasp evidence. Distinct temporal conditions for grasp and release convert this evidence into stable binary gripper commands.

Human-to-robot retargeting (Wang et al., 2026a; Paliwal et al., 2026) requires geometric human-to-gripper alignment (Yuan et al., 2026a), which must also account for the difference between the anatomical wrist and the tool center point (TCP). We estimate a wrist-local interaction point from the median thumb–index midpoint in valid closed-hand frames. Fixing this offset separates the reference motion from instantaneous finger articulation: the point follows wrist translation and rotation while the fingers open or close.

A shared rigid transform places both hands in the robot workspace using a real-robot reference trajectory, retaining metric scale and the relative geometry of the two reference points. The median thumb–index direction and the orthogonalized wrist-to-interaction direction define the tool's closing and approach axes, with signs fixed across the episode. These targets refer to the robot's nominal pinch-center TCP. Together with the grasp schedule, they specify the task-space reference that the retargeter must track. Figure 14 summarizes this construction; Appendix D.4 gives the coordinate definitions.

Kinematically constrained retargeting. The same tool motion can be realized by different robot joint configurations, and independent framewise inverse kinematics can introduce abrupt joint changes. We therefore optimize the 14 arm joints over the full sequence, balancing tool tracking, reference-posture regularization, and temporal smoothness. The tool targets and grasp schedule remain fixed, so optimization changes how the robot realizes the reference without redefining that reference.

Grasp timing also determines orientation tolerance. Tighter tolerances during closure prioritize the closing and approach directions; looser tolerances when open allow posture adjustment. A real-robot reference regularizes redundant shoulder–elbow and wrist configurations, while temporal penalties control velocity, acceleration, and jerk on the original 30 Hz timeline. Joint and derivative bounds

constrain the solve, and workspace and collision checks assess the resulting trajectory. Appendix D.4 details the objective and optimization procedure.

Observation–action supervision. Each episode pairs the captured human images with robot end-effector trajectories and inferred gripper commands on a common timeline. Forward kinematics (FK) recomputes the end-effector trajectories from the optimized joint reference using the robot data's tool convention. This pairing supplies the correspondence between visual evolution and motion needed for EWAM's joint video–action learning. In the co-training setting of Section 5.2, human and robot episodes use the same sixteen-dimensional end-effector action interface. All reconstruction and retargeting are performed offline. Figure 15 illustrates demonstration replay, while Sections D.5 and 5.2 evaluate retargeting fidelity and learned-policy performance, respectively.

D.4 Egocentric Reconstruction and Retargeting Details

This section gives the coordinate definitions and implementation details for the demonstration construction in Section D.3.

Hand representation and interaction reference. Let $s \in \{\mathrm{L}, \mathrm{R}\}$ index the hands, W the reconstruction world frame, H_s the wrist frame, and G the robot base frame. The recovered wrist pose is $\mathbf{T}^W_{H_s}(t) = [\mathbf{R}^W_{H_s}(t), \mathbf{p}^W_{H_s}(t)]$; the articulated joints $\mathbf{J}^{H_s}(t)$ are expressed locally. A local joint is placed in the world as $\mathbf{j}^W = \mathbf{R}^W_{H_s}\mathbf{j}^{H_s} + \mathbf{p}^W_{H_s}$. We estimate a fixed local interaction offset $\bar{\mathbf{c}}^H_s$ as the median thumb–index midpoint in valid closed-hand frames, giving

$$\mathbf{c}^W_s(t) = \mathbf{R}^W_{H_s}(t)\bar{\mathbf{c}}^H_s + \mathbf{p}^W_{H_s}(t). \tag{20}$$

where t is the frame index and $\mathbf{c}^W_s(t)$ is the interaction reference point of hand s in the world frame. The point follows wrist translation and rotation while remaining independent of instantaneous finger articulation. Its use assumes that one wrist-local interaction reference is appropriate for the episode.

Shared robot-frame mapping. One episode-level rigid transform $(\mathbf{C}_{G\leftarrow W}, \mathbf{b}_{G\leftarrow W})$ places both hands in the robot workspace. The tool targets are

$$\begin{aligned} \mathbf{p}^\star_{t,s} &= \mathbf{C}_{G\leftarrow W}\mathbf{c}^W_s(t) + \mathbf{b}_{G\leftarrow W}, \\ \mathbf{R}^\star_{t,s} &= \mathbf{C}_{G\leftarrow W}\mathbf{R}^W_{H_s}(t)\mathbf{B}_s. \end{aligned} \tag{21}$$

where $\mathbf{p}^\star_{t,s}$ and $\mathbf{R}^\star_{t,s}$ are the target TCP position and orientation of hand s at frame t in the robot base frame. Here $\mathbf{C}_{G\leftarrow W}$ aligns the axes and $\mathbf{b}_{G\leftarrow W}$ sets the placement using a real-robot reference trajectory. The transform is shared across hands and has unit scale. The fixed basis $\mathbf{B}_s$ maps the wrist orientation to the tool orientation using an episode-consistent closing axis and an orthogonalized approach axis. The targets refer to the nominal pinch-center TCP and are held fixed during joint optimization.

Sequence optimization. Let $Q = \{\mathbf{q}_t\}_{t=0}^{T-1}$, with $\mathbf{q}_t \in \mathbb{R}^{14}$ containing seven joints per arm. The retargeter seeks to minimize

$$\underset{Q\in\mathcal{F}_{\mathrm{solve}}}{\text{minimize}} \sum_{t=0}^{T-1} [\lambda_{\mathrm{task}}E_{\mathrm{task}}(\mathbf{q}_t) + \lambda_{\mathrm{post}}E_{\mathrm{post}}(\mathbf{q}_t)] + \lambda_{\mathrm{temp}}E_{\mathrm{temp}}(Q). \tag{22}$$

where T is the number of frames and λ_{task}, λ_{post}, and λ_{temp} are the weights of the corresponding terms. The task term E_{task} groups TCP position error and closing- and approach-axis errors outside phase-dependent tolerances. The posture term E_{post} regularizes redundant shoulder–elbow and wrist configurations around a real-robot reference posture. The temporal term E_{temp} penalizes velocity, acceleration, and jerk through finite differences on the original 30 Hz timeline. $\mathcal{F}_{\mathrm{solve}}$ denotes the constraints enforced during the solve, including joint, velocity, and acceleration bounds. Workspace and collision conditions are also checked on the returned trajectory. This is a grouped description of the objective; the returned reference need not be a global minimizer. Output admissibility is reported separately in Section D.5.

Visual tracking and temporal processing. The visual branch applies WiLoR hand detection (Potamias et al., 2025) and ByteTrack association (Zhang et al., 2022) to fisheye-rectified images. We retain detections for each anatomical hand, reject tracks dominated by image boundaries, and interpolate short detection gaps only when the endpoint detections have sufficient overlap. The resulting contiguous clips are supplied to HaWoR. Short, bounded gaps in wrist-local articulation are interpolated, and valid articulation segments are smoothed with a Savitzky–Golay filter (Savitzky & Golay, 1964). World-frame wrist poses, wrist-local geometry, and reconstruction-source and validity indicators are retained.

Staged optimization and initialization. We solve sparse quadratic subproblems within a trust region after local linearization, using multiple initial inverse-kinematics candidates and staged warm starts. The selected refinement first regularizes the full trajectory and then increases position-tracking priority. A final prefix refinement frees the initial joint pose and uses a stable later segment to initialize the first 1.2 s, keeping the remainder fixed. This avoids locking the sequence to an unfavorable first-frame solution. Later frames are available because the procedure operates offline. Throughout refinement, the task-space targets, gripper commands, and original timeline are held fixed.

Articulated-hand adaptation. For articulated robot hands, an additional branch aligns palm frames and hand scales, then solves fingertip-position inverse kinematics with joint limits and temporal regularization. The G1-D two-finger interface instead uses the inferred binary gripper commands alongside the arm trajectory.

Output audit. Before export, we check finite values, joint and derivative limits, workspace conditions, collision clearance at recorded and interpolated configurations, FK consistency, and preservation of target, gripper, and time signals. The exported trajectories are reference actions, distinct from controller commands and hardware measurements. Grasp states are inferred from image-space evidence and do not measure contact forces. Mechanical admissibility is evaluated under the tested robot model and sampled configurations, separately from physical task performance.

D.5 Retargeting Quality and Robot Replay

We assess the offline retargeting pipeline on five recorded human demonstration sequences, examining compliance with the specified kinematic checks and positional agreement with the human-derived tool targets. TCP position error is computed between the positions obtained by forward kinematics from the retargeted joint references and the corresponding tool targets. We report the 95th percentile (P95) of this distance separately for the left and right tools. These measurements characterize the offline references rather than measured tracking error during physical execution. As shown in Table 19, all five references pass the constraint and consistency checks defined in Appendix D.4. Nevertheless, the largest per-sequence P95 errors are 77.01 mm for the left tool and 19.69 mm for the right tool, indicating residual target mismatch despite compliance with the specified checks.

Table 19: **Kinematic checks and TCP position errors on five retargeted demonstrations.** TCP P95 measures the distance between FK-derived reference positions and human-derived tool targets. "Pass" indicates satisfaction of the specified constraint and consistency checks.

Session	TCP P95 (mm)		Checks
	Left	Right	
013312	56.79	2.92	Pass
013326	77.01	14.65	Pass
013335	42.53	1.66	Pass
013352	18.63	1.30	Pass
222650	4.07	19.69	Pass

As a qualitative illustration, Figure 15 shows a recorded bimanual pouring demonstration adapted for replay in simulation and on the physical Unitree G1-D robot. The comparison aligns seven visible action phases, from grasping through pouring to placement and release, providing a view of the demonstrated action sequence across human and robot embodiments. The robot sequences are

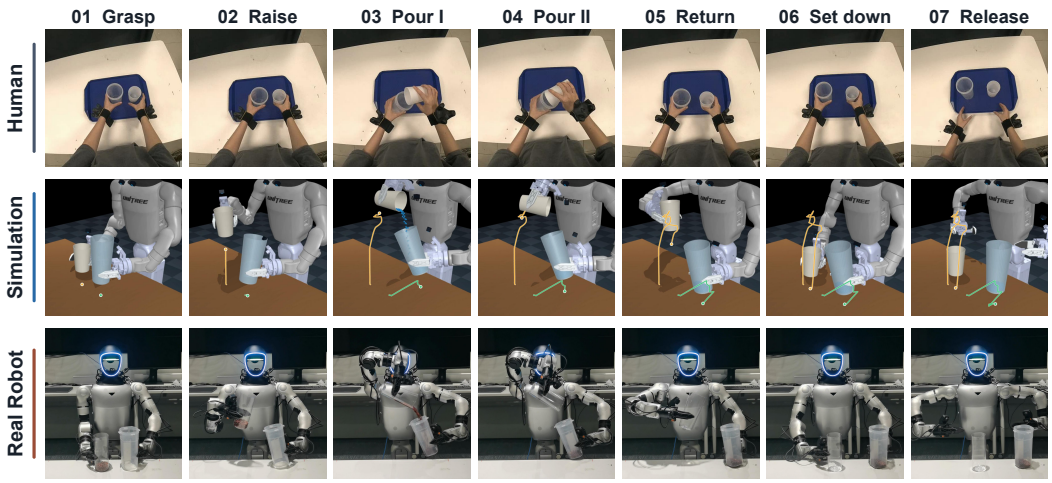


Figure 15: **Human demonstration and robot replay.** Rows show the human egocentric recording, simulation replay, and physical Unitree G1-D replay. Columns correspond to seven visually matched action phases: grasp, raise, two pouring stages, return, placement, and release. Yellow and green curves show projected trajectories of the source and receiver cup bases in simulation, respectively. The simulation represents the transferred material using rigid particles.

replays of demonstration-derived trajectories. The effect of human demonstrations on learned-policy performance is evaluated separately in Section 5.2.

D.6 Additional Human Pretraining Results

Table 20 reports the full G1-D pouring comparison summarized in Section 5.2. Every policy is post-trained on 90 episodes, evaluated for held-out action error on the same 30 robot episodes, and tested on the physical robot over 10 trials per condition. Losses are means over the final five evaluations; training loss is measured on each row's post-training data and is directly comparable only between the two co-trained rows. On the same 60 robot + 30 human mixture, initializing from the human-pretrained rather than the robot-pretrained checkpoint lowers the final training loss from 0.068 to 0.034 and the held-out L_2 error from 0.0231 to 0.0164, and raises success from 40% to 60%. The robot-only policy trained on 90 robot episodes remains the strongest, with 0.0098 held-out L_2 error and 70% success.

Table 20: **Human pretraining and co-training on G1-D pouring.**

Post-training data	Pretraining	Train loss	Held-out L_2	Success (%)
90 robot episodes	Robot	0.055	**0.0098**	**70**
60 robot + 30 human	Robot	0.068	0.0231	40
60 robot + 30 human	Human	0.034	0.0164	60

D.7 Generalization under Tabletop and Cup Changes

Evaluation setting. This section details the real-robot pouring evaluation summarized in *Effect of Egocentric Co-Training Data* (Section 5.2), which changes table colours or tablecloths and replaces the pouring cup. All seven configurations omit EWAM robot and human pretraining and use end-effector (EEF) actions. We vary robot-only demonstration counts or add egocentric data to 60 robot demonstrations. Each configuration is evaluated over 10 physical trials, randomly switching among two to three tablecloth and cup variants. Rates pool these trials. This evaluation is separate from Table 20.

Observed results. Robot-only training remains at 0–20% success (Figure 5(d)). With robot demonstrations fixed at 60, adding 1,000 egocentric demonstrations raises success from 10% to 60%, a gain of 50 percentage points. Two findings follow from these results. First, a small amount of egocentric data can substitute for an equal amount of robot data: 60 robot plus 30 egocentric demonstrations reach 20%, matching 90 robot-only demonstrations, while egocentric data are far

cheaper to collect. Second, performance continues to scale with egocentric data: as egocentric demonstrations increase from 30 to 1,000, success rises from 20% to 30%, 40%, and 60%, with no sign of saturation in this range. Egocentric data thus offer a cost-effective way to improve robustness.

E PRETRAINING DATA

This section describes the data used in EWAM's two separate pretraining regimes, with further details provided in Sections E.1 and E.2. The human pretraining regime uses approximately 2,084 hours of egocentric video, in which wrist motion supplies the action target. Independently, the robot pretraining regime uses 300K cross-embodiment trajectories, comprising approximately 1,800 hours of curated robotic manipulation data collected across multiple robot embodiments, and provides synchronized observations, language instructions, robot states, actions, and future visual observations. In each regime, heterogeneous state and action spaces are mapped into a shared padded representation, while validity masks ensure that unavailable dimensions do not contribute to the action loss.

E.1 HUMAN EGOCENTRIC DATA

The human stage of Section D.2 draws on three egocentric corpora summarized in Table 21. Combined, they provide approximately 2,084 hours of first-person manipulation video paired with per-frame hand pose and camera geometry. The datasets offer complementary coverage: **VITRA-1M** (Li et al., 2025), compiled from unconstrained internet footage, provides the broadest diversity of objects, environments, and activities; **EgoDex** (Hoque et al., 2026) contains scripted tabletop manipulation with high-quality on-device hand tracking; **Xperience** (Ropedia, 2026) supplies SLAM-registered recordings captured via a head-mounted stereo rig.

Table 21: **Human egocentric corpora used for human pretraining.** The last column names the quantity from which each source's anchor camera pose is obtained; all three then enter the identical computation of Eq. 18. VITRA-1M is used through four of its five subsets; its EgoExo4D subset is excluded.

Corpus	Source material	Hours	Anchor camera pose from
VITRA-1M (Li et al., 2025)	Ego4D, EPIC-Kitchens, SSv2	283	Inverted extrinsics
EgoDex (Hoque et al., 2026)	Apple Vision Pro	829	Camera-to-world transform
Xperience (Ropedia, 2026)	Head-mounted stereo rig	972	SLAM pose and calibration
Total		2,084	

VITRA-1M provides MANO-based hand reconstructions together with corrected camera intrinsics and extrinsics for episodes derived from Ego4D, EPIC-Kitchens, and Something-Something v2 (Grauman et al., 2022; Damen et al., 2018; Goyal et al., 2017). We use its `ego4d_cooking_and_cleaning`, `ego4d_other`, `epic`, and `ssv2` subsets, retaining 1,155,865 episodes; the EgoExo4D subset is excluded. Frames flagged as untracked by the released per-hand masks are treated as invalid. **EgoDex** contributes 338,000 demonstrations across 194 tabletop tasks recorded at 30 Hz, with per-frame hand tracking confidences; we discard hand observations whose confidence falls below 0.3. **Xperience** is a partially released corpus recorded with a head-mounted stereo rig, for which hand poses are stored in the per-frame rectified stereo-left camera frame and must first be lifted to world coordinates using the SLAM trajectory before the shared computation applies. Episodes lacking a SLAM pose, lacking stereo calibration, or shorter than one training window are dropped at indexing time.

Every retained episode is reduced to world-frame wrist poses plus an anchor camera pose and then converted by Eq. 18, so the sixteen action dimensions carry identical physical meaning across the mixture. Sampling is weighted in proportion to the number of indexed windows in each corpus. Because the targets are camera-relative displacements whose scale is already comparable across sources, no action normalization is applied during this stage.

E.2 ROBOTIC MANIPULATION DATA

During pretraining, EWAM leverages multiple open-source robotic manipulation datasets containing approximately 3,250 hours of raw recordings, including **GM-100** (Wang et al., 2026d), **Robo-**

COIN (Wu et al., 2025b), **RoboMIND1.0** (Wu et al., 2025a), **RoboChallenge1.0** (Yakefu et al., 2025), **RoboChallenge2.0** (Ma et al., 2026), **RDT** (Liu et al., 2025), and **Simulation data** (Chen et al., 2026b; Hou et al., 2025; Wu et al., 2025a). To focus on tabletop manipulation, we apply task-relevance filtering, data cleaning, and quality assessment. After removing invalid, low-quality, and task-irrelevant samples, we obtain approximately 1,800 hours of high-quality pretraining data. Figure 16 shows the retained duration and corresponding proportion of each source dataset.

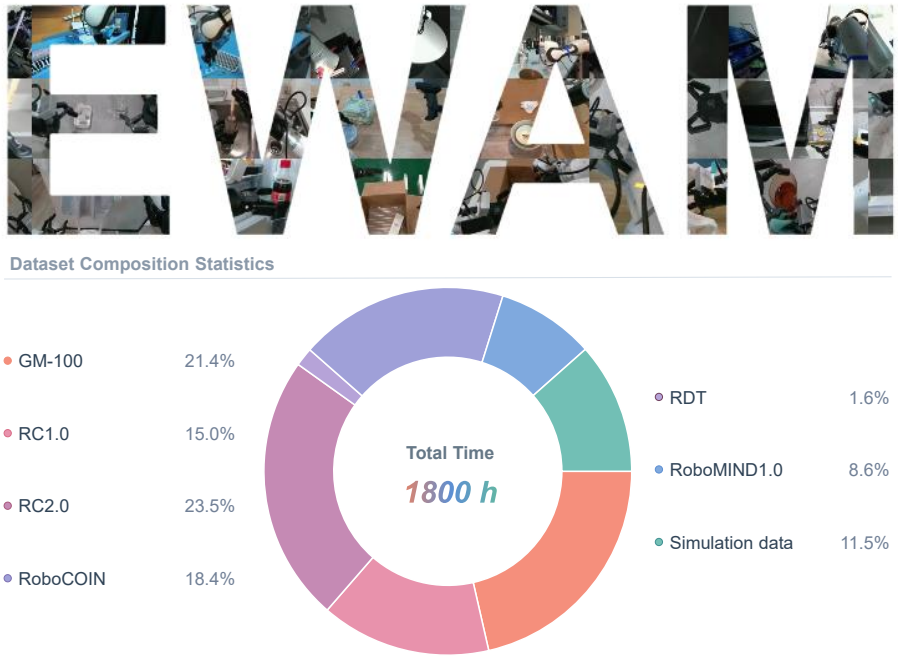


Figure 16: **Source composition and retained duration distribution of the high-quality robotic manipulation data used for EWAM pretraining.**

The pretraining data of EWAM cover a wide range of robot platforms and manipulation tasks. Substantial disparities exist among robots in terms of embodiment, degrees of freedom, and manipulation approaches. The data encompass representative manipulation tasks, including grasping, placing, moving, organizing, switch operation, and cleaning, together with a wide variety of interactive objects. As shown in Figure 17, the composition of robot platforms and the keyword distributions of manipulation tasks and interactive objects are presented.

To systematically characterize the composition of the pretraining data, we summarize the kinematic joint dimensions across various robot platforms in Table 22. The reported statistics correspond to joint-state dimensionality. As several platforms support both single-arm and bimanual setups, joint dimensions differ accordingly.

Table 22: **Joint-state dimensions by robot platform.**

Mode	Robot Platform	Joint Dim.
Single Arm	Franka	8
	UR5	7
	ARX-5	7
Dual Arm	Franka	16
	UR5	14
	Agilex	14
	AgiBot G1	16
	Galaxea R1	16
	ALOHA	14
	ARX-5	14
	DOS-W1	14



Figure 17: **Robot platforms, manipulation tasks, and interacted objects covered by the EWAM pretraining data.**

During data processing, we find substantial differences among the open-source datasets in terms of acquisition devices, data formats, annotation protocols, and data quality. Common defects include corrupted videos, black frames, severe occlusion, prolonged inactivity, and missing frames. Certain episodes also exhibit state-action misalignment, abrupt signal changes, and numerical outliers. These issues may disrupt the temporal continuity of action sequences and the logical consistency between states and actions, thereby impairing model pretraining.

According to the overall quality inspection results, most datasets exhibit varying degrees of state-action misalignment, abrupt numerical changes, and anomalous values. Numerical outliers have the most detrimental effect on model optimization. After the model has fitted the normal data, anomalous samples substantially increase the difficulty of optimization, causing the loss to rise sharply and potentially preventing training from converging. Several datasets contain a substantial number of anomalies, including extreme outliers with magnitudes as high as tens of billions. Regarding state-action alignment, some samples exhibit state signals that precede action signals or motion trends that are inconsistent between the two. Further analysis indicates that these problems typically arise from inconsistent sampling frequencies or temporal synchronization errors among different signals. Moreover, some episodes contain prolonged static segments, predominantly near their beginning and end, with occasional occurrences during task execution. Mid-trajectory inactivity often indicates hesitation or pauses during data collection, potentially encouraging the model to learn uninformative or suboptimal action patterns.

To address these issues, we design a two-stage data cleaning and quality inspection pipeline. The first stage detects file corruption, black frames, occlusions, static frames, and state-action misalignment, and removes unqualified data at the trajectory level. The second stage detects abrupt numerical changes and anomalous values, and removes abnormal data at the frame level. This pipeline adopts a coarse-to-fine filtering strategy that preserves the temporal continuity and state-action consistency of the cleaned data while maintaining a stable data retention rate.

Overall, systematic curation, format standardization, and two-stage quality assurance yield approximately 1,800 hours of high-quality robotic manipulation data for pretraining EWAM on tabletop manipulation. Spanning diverse robot platforms, objects, and task categories, the resulting corpus combines reliable state-action synchronization with broad behavioral coverage. It therefore provides a solid foundation for learning generalizable visuomotor representations and supports transfer to downstream tasks.

F DISCUSSION

From a fixed handoff to adaptive computation. The emergent depth-wise handoff suggests that semantic grounding, visual foresight, and action refinement need not follow a fixed global schedule. Different tasks and execution phases may require different amounts of each form of computation. Ambiguous instructions may demand prolonged semantic grounding, uncertain interactions may require more predictive computation, and well-specified motions may permit an earlier transition to action refinement. Future models could therefore learn task- and state-conditioned routing that dynamically shifts the transition between experts.

From visual foresight to counterfactual planning. The counterfactual future-feature intervention provides a further direction. Changing the content of intermediate future representations alters spatial targeting and arm selection, indicating that these representations carry control-relevant information rather than serving only as an auxiliary prediction target. The current model produces a future trajectory jointly with its action prediction, but does not explicitly compare alternative outcomes. A natural extension is to generate multiple action-conditioned futures and evaluate their consequences before committing to an action. This would move unified world–action modeling from a single predictive trajectory toward counterfactual planning, while retaining semantic grounding from the vision-language expert.

Limitations and future directions. Our analysis characterizes information routing rather than the complete contents of the learned representations. Attention allocation identifies where action queries retrieve information, and the interventions establish functional dependence on selected sources, but neither provides a complete causal decomposition of the computation. Representation probing, activation patching, and cross-layer causal mediation could test whether semantic and predictive information itself changes together with the observed routing. Finally, although the deployed system achieves real-time operation through chunked execution, maintaining three large experts remains computationally expensive. The layer-wise division of labor observed in this work suggests a route to structured optimization. Rather than compressing all three experts uniformly, future work could optimize computation over the expert–depth grid: model capacity could be allocated according to the measured contribution of each expert at different depths, while structured pruning or parameter sharing could target regions with consistently low functional contribution. These improvements could reduce the model's computational overhead while preserving its functional organization.