SkelWAM: A Skeleton-Guided World-Action Model for Zero-Shot Cross-Embodiment Manipulation

Pengjun Niu[1,*], Yujia Xie[1,*], Rui Peng[2], Hang Zhao[3,†], and Ke Liu[1,4,†]

[1]School of Advanced Manufacturing and Robotics, Peking University, Beijing, 100871, China

[2]Feagine

[3]IIIS, Tsinghua University

[4]Research Center for Robotics, Peking University, Beijing, 100871, China

[*]Equal contribution [†]Corresponding authors

hangzhao@tsinghua.edu.cn liuke@pku.edu.cn

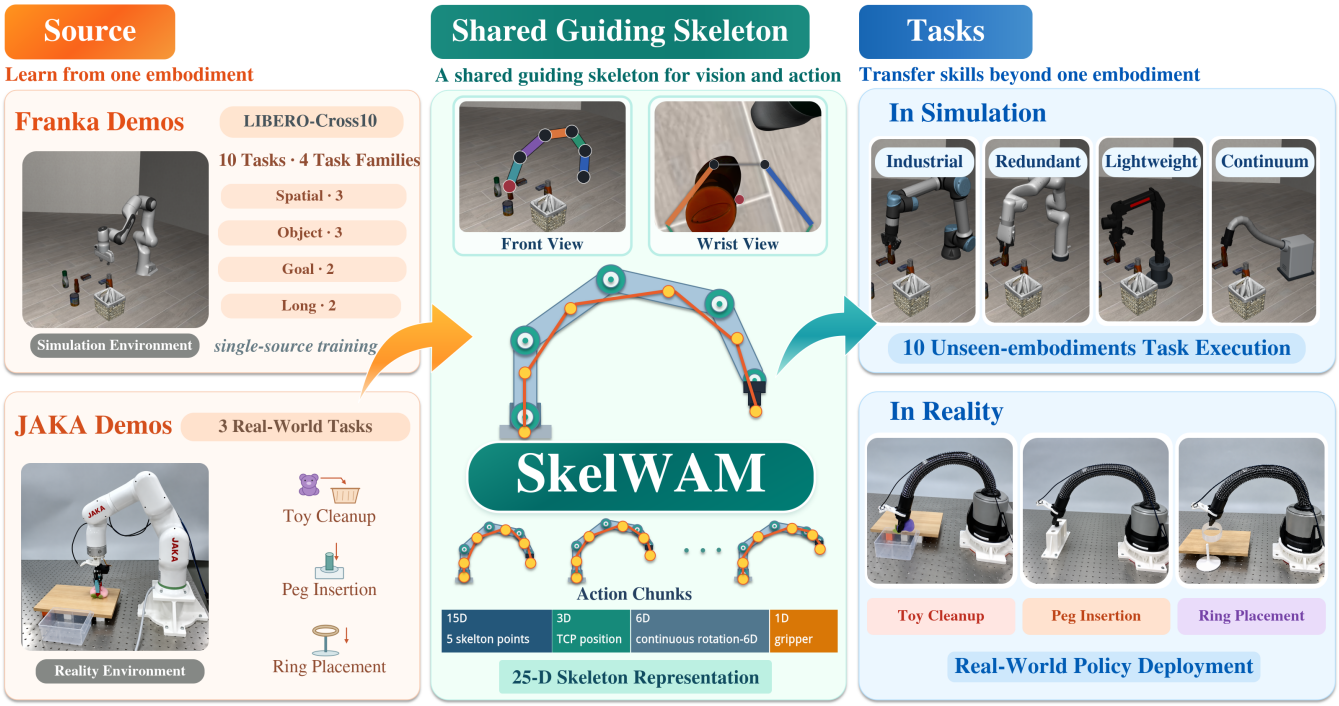

Fig. 1. **SkelWAM overview.** The same 25-D centerline/TCP/jaw geometry defines canonical visual observations and future action chunks. Embodiment-specific decoders translate the predicted whole-body motion intent into native controls on LIBERO-Cross10 targets. Franka demonstrations train the simulation policy. In physical experiments, a model trained on JAKA mini2 data is deployed on Feagine A03.

***Abstract*— Reusing manipulation experience across robot embodiments is important for scaling robot learning and reducing repeated task-specific data collection. However, changes in embodiment alter visual appearance, action dimensionality and semantics, and the whole-body configurations that can realize the same tool pose. We present *SkelWAM*, a skeleton-guided world–action model that couples perception and control through one explicit geometric representation for single-source cross-embodiment manipulation. Arm centerline geometry, tool-center-point (TCP) pose, and parallel-jaw commands form a shared 25-D state. The same definition underlies canonical third-person and wrist observations and future whole-body action targets. Trained with predictive visual supervision, a video–action mixture of transformers predicts canonical skeleton action chunks, which embodiment-specific constrained decoders convert into joint or continuum-robot controls. This formulation requires no one-to-one joint correspondence and uses no target-task demonstrations or target policy updates. We introduce LIBERO-Cross10, a source-only cross-embodiment transfer benchmark covering ten tasks and ten target embodiments across four morphological groups. On this benchmark, Franka-trained SkelWAM achieves 43.3% success over 1,000 episodes, exceeding the best-performing evaluated baseline by 36.2 percentage points. We further deploy a JAKA mini2-trained policy on the Feagine A03 continuum robot for three tabletop manipulation tasks, illustrating the approach's potential for real-world cross-embodiment manipulation. Project page: http://www.liukepku.com/skelwam/index.html**

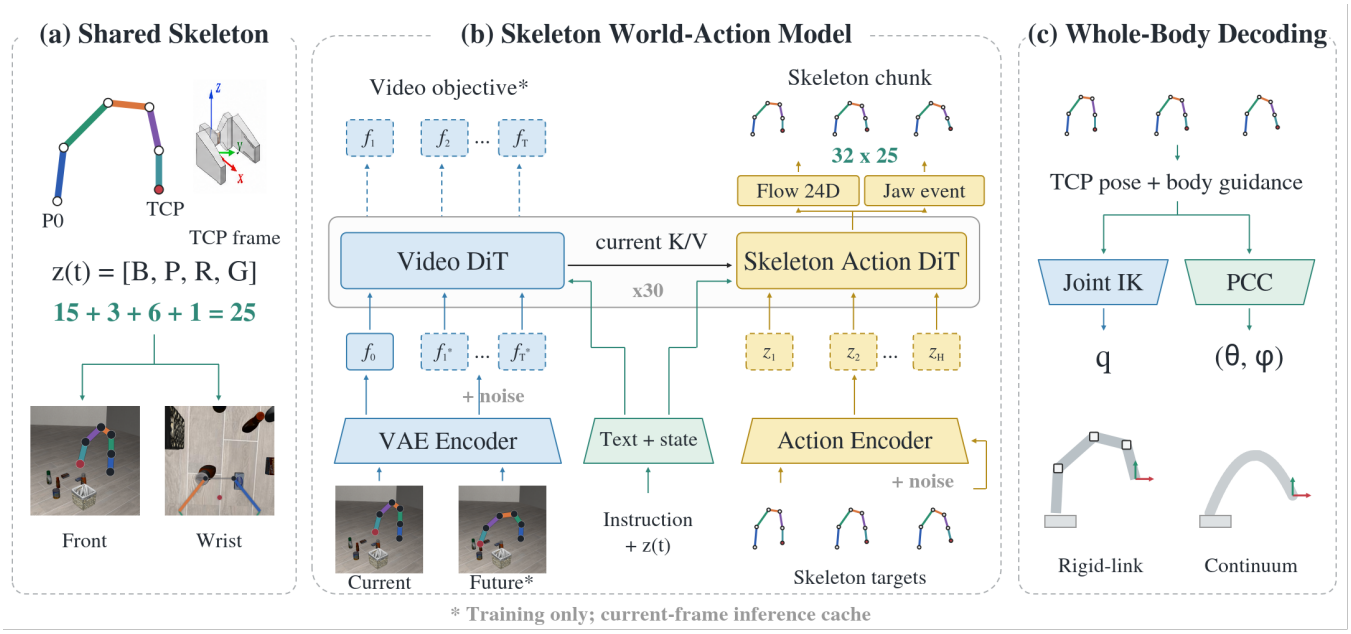

Fig. 2. **SkelWAM world–action model.** (a) The shared 25-D body/TCP/jaw state defines external and wrist proxies. (b) Thirty paired Video/Action DiT blocks predict a 32×25 skeleton chunk. Starred future-video paths provide supervision only during Stage A. Action attention accesses current-frame keys and values (K/V), which are cached at inference. The deployment heads combine 24-D continuous flow with a jaw event. (c) Native joint IK or PCC decoding prioritizes TCP pose and uses the centerline as posture guidance.

I. INTRODUCTION

Reusing manipulation experience across robot embodiments can scale robot learning while reducing repeated task-specific data collection. World–action models (WAMs) couple predictive visual representations with action generation, yet deploying a WAM trained on one robot onto another changes both its observation distribution and its action geometry. Multi-robot learning broadens data reuse [1], [2]. We study source-only task learning followed by deployment on target embodiments. Task demonstrations and policy optimization use one source robot, with no target-task demonstrations or target policy updates.

This setting introduces two coupled gaps. The visual-side embodiment gap arises from differences in appearance, body morphology, and camera-visible geometry. The ction-side embodiment gap includes different control coordinates, redundancy, and whole-body feasibility across native kinematic spaces. Visual adaptation can address appearance while leaving action realization to a controller. End-effector (EEF) or TCP retargeting supplies a common interaction goal, but the same TCP pose can correspond to different whole-body configurations. Cross-embodiment WAM transfer motivates a representation that bridges embodiment differences on both the visual and action sides.

We introduce SkelWAM to connect both sides through a shared skeleton representation (Fig. 1). Our key idea is to express robot morphology and motion intent in one canonical whole-body space and use the same geometry for the visual condition and action target of the WAM. Six centerline samples, TCP pose, and jaw command define a fixed 25-D canonical state. This state specifies external and wrist visual proxies and future skeleton action chunks. The shared policy predicts whole-body motion intent, which embodiment-specific decoders realize under each target's kinematic and execution constraints.

We introduce LIBERO-Cross10 to evaluate this source-only transfer setting. The benchmark uses Franka demonstrations from ten selected LIBERO tasks and ten target embodiments, comprising nine rigid-link arms and one continuum arm. SkelWAM achieves 43.3% mean success, compared with 7.1% for the best-performing evaluated baseline. Component interventions examine the shared representation and whole-body execution. A qualitative physical study complements the simulation results with JAKA mini2 source training and Feagine A03 deployment on three tabletop tasks.

Our contributions are:

- A shared whole-body skeleton representation that canonicalizes visual observations and action trajectories with the same geometry across embodiments.
- A skeleton-guided world–action model that combines predictive visual learning with skeleton action generation and embodiment-specific decoding for native execution.
- LIBERO-Cross10, a ten-task, ten-target source-only transfer benchmark spanning diverse rigid-link and continuum embodiments, with 43.3% mean success for SkelWAM.

II. RELATED WORK

A. Cross-Embodiment Manipulation Policies

Multi-robot datasets and generalist policies support learning from heterogeneous platforms [1], [2]. Visual adaptation methods reduce appearance shift: Mirage cross-paints a source robot into target observations [3], while RoVi-Aug synthesizes robot and viewpoint variation during training [4]. Shared action representations include physically interpretable

coordinates [5] and learned action tokens [6]. UniDex combines hand-masked observations with functionally aligned actuator coordinates [7]. OPFA learns geometry-aware action latents with a retargeting decoder [8]. Latent Action Diffusion uses contrastive encoders to align action embeddings and embodiment-specific decoders to recover native actions [9].

B. Structured Representations and World–Action Models

Motion Tracks uses image-space EEF trajectories [10]. X-Diffusion uses human-action supervision at high diffusion noise levels to train policies [11]. Robot–object relations support cross-hand grasping [12] and interaction-preserving retargeting [13]. XMoP uses link-wise geometry for cross-arm motion planning [14]. RT-Affordance conditions execution on key-stage robot poses [15], whereas A_0 predicts object-contact points and post-contact trajectories [16]. A_1 accelerates VLA inference with adaptive early exit [17]. World–action models incorporate future-predictive visual supervision into action learning [18], [19]. Fast-WAM studies video co-training with current-frame deployment [20], and LaWAM uses compact latent visual subgoals [21]. Video models also synthesize training trajectories [22]. Soft-arm kinematics can also be learned [23]. SkelWAM focuses on an explicit centerline/TCP geometry shared by visual conditions and action targets within a source-trained WAM.

III. METHOD

SkelWAM connects the visual and action sides of a world–action model through shared skeleton geometry. Figure 2 summarizes the canonical representation, source-only task learning, and embodiment-specific decoding that maps whole-body motion intent to native controls.

A. Problem Formulation

We separate source-only task learning from target-specific deployment. Let $\mathcal{D}_s = \{(I_t^f, I_t^w, x_t, a_t, \ell)\}$ contain synchronized external and wrist images, native state and action, and language from one source embodiment s. The target r provides calibrated geometry, cameras, and a kinematic controller. Policy optimization and model selection use only source demonstrations, with target embodiments excluded from the task-training set. Encoders E_r and decoders D_r connect native coordinates to a shared policy π_θ in canonical skeleton space $\mathcal{Z}$. Policy weights and source-fitted normalization remain frozen on targets.

B. Canonical Whole-Body Skeleton

We remove joint-space dependence from the shared policy interface by representing each embodiment with the same body/TCP/jaw state. Let $\gamma_r(x_r, u) \in \mathbb{R}^3$ be the ordered centerline from the mounted base to the tool center point (TCP), parameterized by normalized arc length $u \in [0, 1]$. We sample $p_i = \gamma_r(x_r, i/5)$ for $i = 0, \ldots, 5$, with p_5 at the TCP, and define $L_r = \sum_{i=0}^{4} \|p_{i+1} - p_i\|_2$. Given task-frame normalization $\nu_{\mathcal{F}}$, the continuous 6-D rotation representation $\rho(R)$ [24], and normalized jaw command g_t, the canonical state is

$$z_t = \left[\frac{p_0 - p_5}{L_r}, \ldots, \frac{p_4 - p_5}{L_r}, \nu_{\mathcal{F}}(p_5), \rho({}^{\mathcal{F}}R_5), g_t \right] \in \mathbb{R}^{25}. \tag{1}$$

The fields $[B, P, R, G]$ contain 15 body coordinates, 3 TCP position coordinates, a 6-D orientation, and a jaw command. The six samples follow centerline arc length rather than joint locations. Expressing body shape relative to the TCP and normalizing its length separates posture from global position and scale. The task-frame TCP retains the spatial interaction goal. Jaw commands $+1/-1$ denote opening/closing. Source-fitted per-coordinate normalization is applied before policy learning and held fixed at deployment. This statistical normalization follows the task-coordinate mapping $\nu_{\mathcal{F}}$.

C. Shared Visual–Action Representation

We use the same skeleton geometry to canonicalize what the policy sees. We construct canonical visual observations by masking the native robot, filling the background, and projecting the shared geometry into the image. The external view depicts the six-point centerline, while the wrist view depicts a perspective parallel-jaw skeleton in a fixed canonical TCP-relative camera frame. Let $V_r(\cdot; \mathcal{C}_r)$ denote this visual operator, where $\mathcal{C}_r$ contains the embodiment-specific calibration parameters. The policy observation is

$$o_t^c = \{V_r(I_t^f, I_t^w, \bar{z}_t; \mathcal{C}_r), \bar{z}_t, \ell\}, \tag{2}$$

where $\bar{z}_t = z_t$ for source demonstrations and follows the target feedback rule below during deployment. The same state definition supplies numerical policy input, both visual proxies, and future action supervision. One geometry therefore establishes correspondence between the posture visible to the policy and the motion coordinates predicted by the policy. In simulation, segmentation and robot-hidden renders provide the masks and backgrounds. The physical implementation uses calibrated masks and image filling (Sec. IV-E). Source images and future skeleton states are paired along the same demonstration timeline.

The external proxy retains scene context while exposing the arm arrangement within the workspace. The wrist proxy focuses on local object geometry and jaw state near the TCP. Their shared construction preserves correspondence between global posture, local interaction and the predicted action coordinates.

D. Skeleton-Guided World–Action Model

With both modalities expressed in canonical skeleton space, a WAM learns their correspondence. We adopt paired video–action experts with cached current-frame conditioning and train the action interface in canonical skeleton space. The model pairs a 30-layer Video Expert with a 30-layer Action Expert, both using diffusion-transformer blocks [25]. Transformer weights are initialized from the Wan2.2 video backbone in the Wan model family [26], [27]. The 25-D state encoder, action input/output projections, and gripper-event

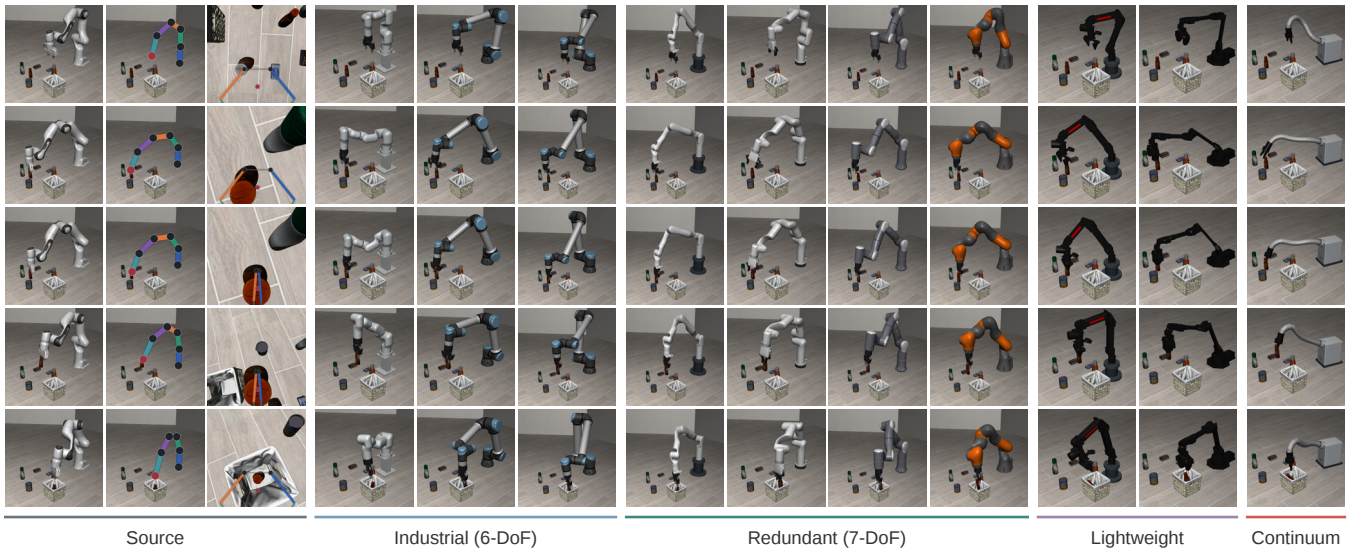

Fig. 3. BBQ-sauce-to-basket on Cross-Object (task 3). The first three columns show one Franka source demonstration and its front/wrist skeleton views. Target columns follow the robot order in Table I and show SkelWAM control replays from recorded states. Rows align initialization, approach, grasp, transport, and the LIBERO goal state across independently selected trials.

heads are newly initialized and trained on source demonstrations. In each paired block, action queries attend to current-frame video keys and values (K/V) and other action tokens. Future-video tokens are excluded from action attention. The future-video branch supplies an auxiliary predictive objective during training, while the action attention mask preserves current-observation conditioning for deployment.

Training has two source-only stages. Stage A jointly adapts the trainable Video Expert blocks and Action Expert through video prediction and skeleton action learning:

$$\mathcal{L}_A = \lambda_v \mathcal{L}_{\text{video}} + \mathcal{L}^{25D}_{\text{action}}. \tag{3}$$

The video objective supervises future scene evolution, and the action objective supervises the associated geometric trajectory. Stage B freezes the Video Expert and optimizes action generation conditioned on current-frame features. Flow matching [28] predicts 24 geometric coordinates, while a factorized event head predicts whether the jaw state switches and the time of its first switch:

$$\mathcal{L}_B = \mathcal{L}^{24D}_{\text{flow}} + \lambda_g(\mathcal{L}_{\text{presence}} + \mathcal{L}_{\text{timing}}). \tag{4}$$

Here, $\mathcal{L}_{\text{presence}}$ and $\mathcal{L}_{\text{timing}}$ supervise the occurrence and timing of the first jaw-state change. Separating jaw events from continuous geometry preserves the discrete opening/closing semantics. The predictions form a 32×25 skeleton action chunk of future absolute canonical states $\hat{Z}_{t+1:t+32}$, with at most one jaw-state switch per chunk. Subsequent switches are handled by replanning. At inference, the Video Expert encodes the current observation once. Its cached current-frame K/V condition the Action Expert throughout sampling. Future-video sampling is unnecessary at deployment.

E. Embodiment-Specific Decoding and Feedback

The policy is shared across embodiments, while native execution remains embodiment-specific. For desired state $\hat{z}$, decoder D_r prioritizes TCP pose and jaw commands, subject to native coordinate limits and feasibility checks. In addition to its EEF objective, it uses the posture and continuity regularizer

$$\sum_{i=0}^{4} w_i \|f_{r,i}(x_r) - \hat{p}_i\|_2^2 + \lambda_q \|x_r - x_{r,t-1}\|_2^2, \tag{5}$$

where $f_{r,i}(x_r)$ are target-kinematics centerline samples and $\hat{p}_i$ are reference points decoded at the target scale. Simulation decoding rescales body offsets with the target's sampled centerline length at decoder initialization and adds the task-frame TCP position. The rigid-link decoder uses constrained joint IK, whereas the continuum decoder operates on piecewise-constant-curvature (PCC) parameters [32] and available tool coordinates. The TCP term anchors task interaction, and the centerline term provides soft whole-body posture guidance subject to target feasibility. Exact centerline agreement is not required. Continuity discourages abrupt changes between successive native commands. Redundant arms can redistribute joint motion, while the continuum arm realizes the reference through segment curvature.

The policy state combines canonical intent with interaction feedback. The simulation bridge uses measured geometry when no prior intent exists. For rigid-link targets, the previous body intent is retained while measured TCP position, orientation, and jaw command update the next policy state. For the continuum target, body and interaction-orientation intent are retained, with measured TCP position and jaw feedback. The decoder uses the measured native configuration for its kinematic and feasibility computations. This separates the policy's posture reference from the target's measured body shape. Execution is receding-horizon: after the configured prefix is applied, fresh observations update the state and initiate a new chunk.

IV. EXPERIMENTS

We evaluate source-only cross-embodiment transfer on LIBERO-Cross10, compare against representative policies,

TABLE I

Zero-shot success on the ten target embodiments of LIBERO-Cross10 (%). Columns follow the four target reporting groups, and rows are grouped by method category. Each target has 100 episodes (ten tasks, ten trials per task).

Family	Method	Industrial (6-DoF)			Redundant (7-DoF)				Lightweight		Continuum	Mean
		JAKA mini2	UR5e	UR10e	Kinova Gen3	xArm7	Rizon4	IIWA	ARX-L5	ViperX300	Feagine A03	
IL	Diffusion Policy [29]	0.0	2.0	3.0	0.0	3.0	0.0	5.0	0.0	0.0	0.0	1.3
VLA	$\pi_{0.5}$ [30]	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.1
	OpenVLA-OFT [31]	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.2
WAM	FastWAM [20]	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.4
	LaWAM [21]	0.0	0.0	1.0	4.0	2.0	0.0	0.0	0.0	2.0	0.0	0.9
X-Emb.	Mirage [3]	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.4
	RoVi-Aug [4]	3.0	**15.0**	4.0	7.0	13.0	8.0	9.0	10.0	2.0	0.0	7.1
Ours	**SkelWAM**	**59.0**	13.0	**62.0**	**49.0**	**19.0**	**62.0**	**22.0**	**56.0**	**32.0**	**59.0**	**43.3**

Bold and underlined values indicate the highest and second-highest distinct rates in each column, respectively, including ties. Mean is the equally weighted average over ten targets. Non-SkelWAM methods use their native policy interface with target-specific EEF-IK. Mirage and RoVi-Aug use FastWAM as the underlying policy. Ablations are reported separately.

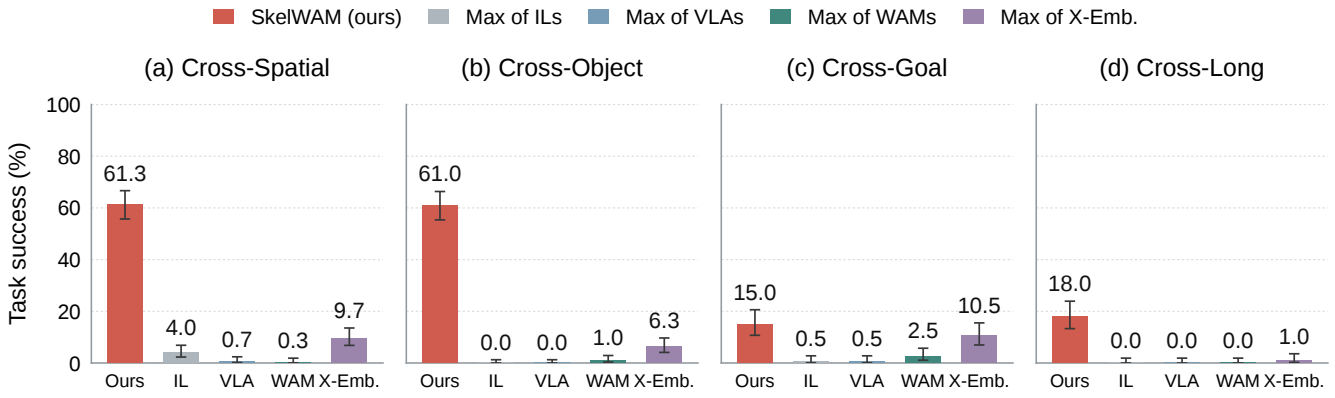

Fig. 4. Success on the four selected task subsets, averaged over ten target embodiments. Each baseline bar is the maximum complete-method average within its family. Error bars show descriptive, episode-level Wilson 95% confidence intervals from binary outcomes (300, 300, 200, and 200 episodes per displayed method in the four panels), without adjustment for maximum selection. Task subsets are defined in the benchmark section.

and examine the roles of the shared representation and whole-body decoding. We also deploy a JAKA mini2-trained policy on a physical Feagine A03 continuum arm.

A. *LIBERO-Cross10 Benchmark*

We introduce LIBERO-Cross10 to evaluate source-trained manipulation policies under changes in robot morphology. Built on LIBERO [33], it retains objects, language instructions, object reset states, and success predicates while registering target-specific arms, grippers, installations, and cameras. We use a fixed set of ten tasks: task IDs 0, 2, and 3 from both Spatial and Object, and task IDs 5 and 8 from both Goal and LIBERO-10. We refer to these selected subsets as Cross-Spatial, Cross-Object, Cross-Goal, and Cross-Long, respectively. Franka Panda is the source and is excluded from all ten-target averages. We report results for four fixed, non-overlapping groups: *Industrial* (6-DoF: JAKA mini2, UR5e, and UR10e), *Redundant* (7-DoF: Kinova Gen3, xArm7, Rizon4, and IIWA), *Lightweight* (ARX-L5 and ViperX300), and *Continuum* (Feagine A03). The first three groups contain nine rigid-link arms. Feagine A03 supplies a continuum target with PCC kinematics. All targets use parallel-jaw grippers with their native or shared contact geometries. The benchmark uses MuJoCo [34] and robosuite [35]. Figure 3 illustrates the common task across the source and target roster.

TABLE II

Component interventions on the same 1,000 episodes per variant. Δ is ablation minus full, in percentage points.

Variant	V	A	W	D	SR (%)	Δ (pp)
Full	✓	✓	✓	✓	**43.3**	–
w/o V^R	×	✓	✓	✓	0.0	-43.3
w/o A^R	✓	×	✓	×	0.1	-43.2
w/o W^R	✓	✓	×	✓	43.0	-0.3
w/o D^I	✓	✓	✓	×	40.9	-2.4

V, A, W, and D denote skeleton vision, body-action supervision, wrist input, and the decoder body objective, respectively. R denotes source retraining, and I denotes an inference-only intervention using the full checkpoint. Removing A also disables D, coupling changes in learning and execution.

B. *Evaluation Protocol*

Source training. The source set contains 467 successful Franka demonstrations from the ten selected tasks, split by episode into 447 training and 20 validation episodes. Each

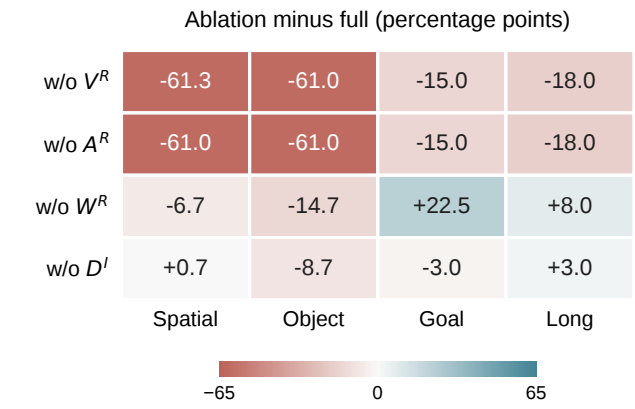

Fig. 5. Task-dependent effects of the interventions in Table II. Cells show ablation minus full in percentage points on the four Cross-* subsets. Red indicates a decrease and blue an increase in success.

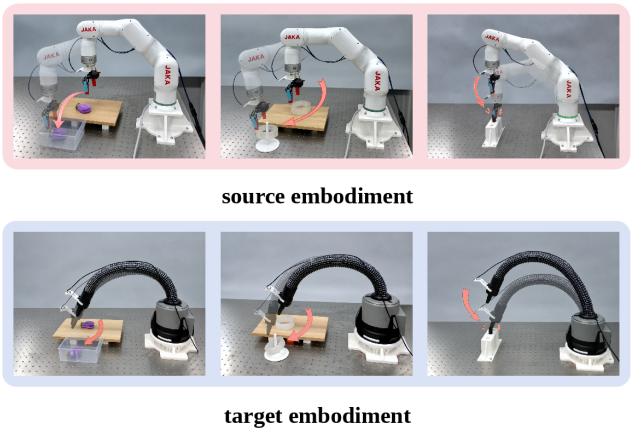

Fig. 6. **Real-robot task setup.** Columns show Toy Cleanup, Ring Placement, and Peg Insertion, while rows show source JAKA mini2 and target Feagine A03. Arrows and translucent poses illustrate the task motions.

window contains two 224×224 canonical views, language, the current 25-D state, and 32 future absolute states. Video targets are sampled at offsets $0, 4, \ldots, 32$. Stage A trains the final four video blocks, video output head, and Action Expert for 6k updates. Stage B then freezes the Video Expert and trains the action pathway for 24k updates. Model selection uses held-out Franka episodes and selects Stage-B step 23k after the 6k Stage-A updates. Inference uses 20 flow-matching steps and executes ten predicted states before replanning.

Baselines and adaptation. We evaluate four method families: imitation learning (Diffusion Policy [29]), vision–language–action policies ($\pi_{0.5}$ [30] and OpenVLA-OFT [31]), world–action models (Fast-WAM [20] and LaWAM [21]), and cross-embodiment visual adaptation (Mirage [3] and RoVi-Aug [4]). The last two are adaptations of these approaches to a common Fast-WAM backbone.

Diffusion Policy is source-trained, whereas the VLA and WAM baselines use released checkpoints. Mirage reuses Fast-WAM weights with visual cross-painting, whereas RoVi-Aug uses source-data visual augmentation and a 4k adapted checkpoint. Pretraining and source-adaptation budgets differ across these methods. Each baseline retains its method-specific image processing and native action contract. Cartesian outputs pass through target-specific EEF inverse kinematics in joint or PCC coordinates (EEF-IK). No baseline receives SkelWAM skeleton images, centerline state, or a body-tracking objective. Results compare deployed systems, including their adapters.

Trials and aggregation. Every method uses the same ten targets, ten tasks, and ten paired trial indices per embodiment–task cell. Let $\mathcal{R}$ and $\mathcal{T}$ denote the target-embodiment and task sets, respectively, and let J be the number of trials per cell. Let $y_{r,k,j} = 1$ when trial j on embodiment r and task k satisfies the original LIBERO success predicate, and 0 otherwise.

$$\mathrm{SR} = \frac{1}{|\mathcal{R}|\,|\mathcal{T}|\,J} \sum_{r\in\mathcal{R}} \sum_{k\in\mathcal{T}} \sum_{j=1}^{J} y_{r,k,j}. \tag{6}$$

We report SR as a percentage. Here, $|\mathcal{R}| = |\mathcal{T}| = J = 10$, giving 1,000 episodes per method. All embodiment–task cells are equally weighted, with the four task families contributing 300, 300, 200, and 200 episodes. The paired trial indices use the same initializations across methods. Error bars use 95% Wilson intervals for pooled episode proportions. Zero-shot evaluation fixes the policy after source training and permits task-independent target calibration.

C. Zero-Shot Transfer Results

Comparison with prior methods. Table I compares zero-shot success across the complete target matrix. SkelWAM achieves 43.3% success over 1,000 episodes (95% Wilson CI: 40.3–46.4%). The best-performing evaluated baseline, the Fast-WAM-based RoVi-Aug implementation with EEF-IK, reaches 7.1%, giving SkelWAM an absolute improvement of 36.2 percentage points. Fast-WAM and its Mirage cross-painted variant both obtain 0.4%.

SkelWAM achieves the highest success rate on nine targets. On UR5e, RoVi-Aug obtains 15.0% compared with 13.0% for SkelWAM. Figure 4 reports the highest average success within each baseline family for each task subset, with every method averaged over all ten target embodiments.

Task and morphology analysis. Figure 4 shows 61.3% and 61.0% success on Cross-Spatial and Cross-Object, versus 15.0% and 18.0% on Cross-Goal and Cross-Long. Transfer performance varies substantially with the task subset despite the shared interface.

The target matrix also spans different embodiment groups: the mean success rates are 44.7% for Industrial, 38.0% for Redundant, 44.0% for Lightweight, and 59.0% for the single Continuum target. Variation within groups includes the 19.0%–62.0% range among the seven-DoF arms.

D. Component Analysis

Ablation design. Table II examines four components: skeleton vision (V), body-action supervision (A), wrist input (W), and the decoder's body objective (D). The V, A, and W variants are separately trained on Franka data. The D intervention alone reuses the selected full-model weights. All

Fig. 7. **Physical visual–action interface.** JAKA mini2 is shown on the left and Feagine A03 on the right. For Toy Cleanup, the five rows show external RGB, mask overlays, skeleton composites, wrist RGB, and wrist illustrations. Lower bands: Peg Insertion and Ring Placement. Each side shows four selected times.

scores use the same 100 target-embodiment–task cells and ten paired trial indices.

Representation and execution. The w/o V^R and w/o A^R variants obtain 0.0% and 0.1%, respectively. The action variant retains the 25-D architecture, removes supervision on its 15 body coordinates, and uses an EEF-only decoder objective. This intervention changes both learning and execution, whereas removing D changes execution alone. The visual variant is trained with native source RGB.

Task-dependent effects. Removing the wrist changes overall success by only -0.3 points, yet its effects are $-6.7/-14.7/+22.5/+8.0$ points across the four task subsets. Figure 5 reveals these task-dependent reversals. Removing only the decoder body objective changes the mean by -2.4 points. Its effects are also task dependent: -8.7 on Cross-Object but $+3.0$ on Cross-Long. Thus, the wrist and whole-body objective contribute differently across task families.

E. Real-Robot Deployment

Setup and source training. JAKA mini2 and Feagine A03 each use external and wrist RGB cameras for Toy Cleanup, Ring Placement, and Peg Insertion (Fig. 6). The 277 quality-controlled JAKA mini2 teleoperation demonstrations are split by episode into 249 training and 28 validation trajectories, then compiled into 10 Hz skeleton windows. We train the Action Expert and final two video blocks on JAKA mini2 data for 6k updates with the action objective. Model selection uses source validation only.

Physical interface. Calibrated JAKA mini2 forward kinematics and Feagine A03 PCC models encode native feedback into the shared 25-D representation. Task-frame, tool, and camera calibration registers the numerical skeleton with both camera views. The TCP is defined at the gripper contact center, accounting for the different tool lengths. Embodiment-specific constrained inverse solvers map predicted skeleton trajectories to native commands. Feagine A03 uses its PCC decoder with interaction feedback. For vision, classical computer vision combines static-ROI feature matching with RANSAC, forward–backward Lucas–Kanade flow, and color-assisted TCP relocking for masking and skeleton-point tracking.

Three-task deployment. We deploy the JAKA mini2-trained policy on Feagine A03 with fixed weights and a

target-specific PCC decoder for Toy Cleanup, Ring Placement, and Peg Insertion. No Feagine A03 task demonstrations are used for fine-tuning. Figure 7 shows the qualitative deployments and the associated external and wrist-view processing.

Transferring the JAKA mini2-trained policy to Feagine A03 illustrates SkelWAM's potential for real-world cross-embodiment manipulation.

V. DISCUSSION AND LIMITATIONS

The results support shared visual–action geometry as a backbone for transferring source-trained WAMs across embodiments. Cross-Goal and Cross-Long remain lower than Cross-Spatial and Cross-Object, and the ablations show that wrist context and posture guidance interact with the manipulation objective. The comparison uses a common target protocol, although released pretraining and source adaptation differ across methods. The coupled action intervention also limits component-level attribution.

Transfer depends on reachability, gripper contact geometry, camera calibration, and the quality of background filling under object occlusion. These factors remain in the target interface.

VI. CONCLUSION

Cross-embodiment world–action model (WAM) transfer faces visual-side and action-side embodiment gaps. SkelWAM connects both through a shared 25-D whole-body skeleton: the same geometry defines visual conditions, canonical state, and future skeleton action chunks. A source-trained WAM predicts motion intent, and embodiment-specific decoders realize it under native constraints. On LIBERO-Cross10, Franka-trained SkelWAM achieves 43.3% mean success across ten targets, including nine rigid-link and one continuum embodiment, 36.2 percentage points above the best-performing evaluated baseline. The qualitative JAKA mini2-to-Feagine A03 study demonstrates physical deployment of the shared interface.

References

[1] A. O'Neill, *et al.*, "Open X-Embodiment: Robotic learning datasets and RT-X models : Open X-Embodiment Collaboration0," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2024, pp. 6892–6903.

[2] D. Ghosh, *et al.*, "Octo: An open-source generalist robot policy," in *Proc. Robot. Sci. Syst. (RSS)*, 2024.

[3] L. Y. Chen, K. Hari, K. Dharmarajan, C. Xu, Q. Vuong, and K. Goldberg, "MIRAGE: Cross-embodiment zero-shot policy transfer with cross-painting," in *Proc. Robot. Sci. Syst. (RSS)*, 2024.

[4] L. Y. Chen, *et al.*, "RoVi-Aug: Robot and viewpoint augmentation for cross-embodiment robot learning," in *Proc. Conf. Robot Learn. (CoRL)*, ser. PMLR, vol. 270, 2025, pp. 209–233.

[5] S. Liu, *et al.*, "RDT-1B: a diffusion foundation model for bimanual manipulation," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2025, pp. 29 982–30 009.

[6] Y. Liu, *et al.*, "G0.5: One autoregressive stream for robot reasoning and action," 2026, arXiv:2608.11739.

[7] G. Zhang, *et al.*, "UniDex: A robot foundation suite for universal dexterous hand control from egocentric human videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2026, pp. 1841–1852.

[8] J. Mu, *et al.*, "One-Policy-Fits-All: Geometry-aware action latents for cross-embodiment manipulation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2026, in press. Preprint: arXiv:2603.14522.

[9] E. Bauer, E. Nava, and R. K. Katzschmann, "Latent action diffusion for cross-embodiment manipulation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2026, in press. Preprint: arXiv:2506.14608.

[10] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg, "Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 8802–8810.

[11] M. A. Pace, *et al.*, "X-Diffusion: Training diffusion policies on cross-embodiment human demonstrations," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2026, in press. Preprint: arXiv:2511.04671.

[12] Z. Wei, *et al.*, "$\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 4982–4988.

[13] J. Wu, *et al.*, "TopoRetarget: Interaction-preserving retargeting for dexterous manipulation," 2026, arXiv:2606.16272.

[14] P. K. Rath and N. Gopalan, "XMoP: Whole-body control policy for zero-shot cross-embodiment neural motion planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 8299–8306.

[15] S. Nasiriany, *et al.*, "RT-Affordance: Affordances are versatile intermediate representations for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 8249–8257.

[16] R. Xu, *et al.*, "A0: An affordance-aware hierarchical model for general robotic manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025, pp. 13 491–13 501.

[17] K. Zhang, *et al.*, "A1: Adaptive truncated vision-language-action model from affordance to action," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Findings*, 2026, pp. 1503–1514.

[18] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta, "Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets," in *Proc. Robot. Sci. Syst. (RSS)*, 2025.

[19] Y. Shen, *et al.*, "VideoVLA: Video generators can be generalizable robot manipulators," in *Adv. Neural Inf. Process. Syst.*, vol. 38, 2025, pp. 95 597–95 621.

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

[21] J. Chen, *et al.*, "LaWAM: Latent world action models for efficient dynamics-aware robot policies," 2026, arXiv:2606.15768.

[22] J. Jang, *et al.*, "DreamGen: Unlocking generalization in robot learning through video world models," in *Proc. Conf. Robot Learn. (CoRL)*, ser. PMLR, vol. 305, 2025, pp. 5170–5194.

[23] X. Chen, Y. Xiong, and L. Wen, "Consistency-driven dual LSTM models for kinematic control of a wearable soft robotic arm," 2026, arXiv:2603.17672.

[24] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 5745–5753.

[25] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 4195–4205.

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

[27] Team Wan, "Wan2.2," GitHub repository, 2025. [Online]. Available: https://github.com/Wan-Video/Wan2.2

[28] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[29] C. Chi, *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, vol. 44, no. 10–11, pp. 1684–1704, Sept. 2025, doi: 10.1177/02783649241273668.

[30] K. Black, *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," in *Proc. Conf. Robot Learn. (CoRL)*, ser. PMLR, vol. 305, 2025, pp. 17–40.

[31] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," in *Proc. Robot. Sci. Syst. (RSS)*, 2025.

[32] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010, doi: 10.1177/0278364910368147.

[33] B. Liu, *et al.*, "LIBERO: Benchmarking knowledge transfer for lifelong robot learning," in *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 44 776–44 791.

[34] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 5026–5033.

[35] Y. Zhu, *et al.*, "robosuite: A modular simulation framework and benchmark for robot learning," 2020, arXiv:2009.12293.