WORLDZOO: COMPOSITIONAL WORLD EDITING FOR GENERALIZABLE ROBOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning for robotic manipulation is limited by the high cost of collecting demonstrations and insufficient data diversity. By reusing existing demonstrations, data augmentation offers an effective way to broaden training data coverage under a limited collection budget. However, composing multiple edits without disrupting task-relevant interactions and selecting augmentations that benefit a target task remain challenging. We present **WorldZoo**, a unified world-editing framework that leverages robot demonstrations and human egocentric videos to generate augmented data through composable edits along three dimensions: embodiment, interaction, and scene. WorldZoo coordinates embodiment retargeting, visual editing, and trajectory adaptation to maintain consistency among observations, actions, and task-relevant contact relationships across diverse robotic arms and dexterous hands. Object editing adapts the demonstrated contacts to new object geometry within a range compatible with the original manipulation, expanding object diversity while preserving the demonstrated task. UniAug composes these edits within each sample as a reusable default recipe. We extend existing robotic simulation benchmarks to evaluate generalization across all three dimensions, systematically assess the effectiveness of augmentation along each dimension, and evaluate the training value of augmented data by fine-tuning generalist robot policies. In real-world experiments, UniAug matches or exceeds applying edits separately with one-sixth of the samples.

1 INTRODUCTION

Large-scale pretraining is increasingly used to equip robot policies with broad visual and behavioral priors (Hu et al., 2026; Black et al., 2024; Intelligence et al., 2025; Kim et al., 2024; Bjorck et al., 2025; Pertsch et al., 2025). Yet this generality can be fragile during task-specific post-training, which is often performed on demonstrations collected from only a small set of scenes, objects, and robot embodiments. Continued optimization on such narrow, in-distribution data can erode pretrained visual representations and weaken generalization under unseen conditions (Kachaev et al., 2025). Preserving generalization therefore requires post-training data with broader coverage of both interaction dynamics and visual conditions. Collecting such coverage with physical robots, however, remains costly and labor-intensive (Dass et al., 2022; Wang et al., 2025b), making data diversification under a limited collection budget a central challenge for scalable robot policy adaptation.

To alleviate robot data scarcity, the community has explored ways to accelerate data collection from both algorithmic and hardware perspectives. One line of work seeks lower-cost or more scalable data-collection systems, including egocentric cameras, the Universal Manipulation Interface (UMI), and simulation (Kareer et al., 2025; Chi et al., 2024; Nasiriany et al., 2024). These systems enable data collection under more flexible deployment conditions and support different trade-offs between observation coverage and action precision. A second line expands existing demonstrations through data augmentation or generative editing while preserving task semantics (Chen et al., 2023; Ameperosa et al., 2025; Pan et al., 2026). Prior methods rely on geometrically well-conditioned observations (Bharadhwaj et al., 2024; Tao et al., 2025; Wang et al., 2026c; Yu et al., 2023), such as multi-camera recordings or reconstructed 3D scenes (Tian et al., 2024; Wang et al., 2026a; Yang et al., 2025), and generate variations through object replacement (Wang et al., 2026a), background substitution (Yuan et al., 2025), or trajectory perturbation (Mandlekar et al., 2023).

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: **Compositional world editing.** WorldZoo expands heterogeneous demonstrations into diverse training data by composing embodiment, interaction, and scene edits while keeping observations and actions consistent.

Despite progress along both directions, a unified augmentation framework that can operate across heterogeneous data-collection systems still faces two coupled challenges. 1) *Capability gap*. Demonstrations from different systems vary in their interaction representations and observation conditions, constraining which edits can be applied reliably. Existing methods lack the capability to accommodate such heterogeneous inputs, compose edits across dimensions, and maintain consistency between edited interactions and action annotations. 2) *Allocation gap*. Diverse editing creates a combinatorial set of candidate variants from each demonstration, whereas the post-training data budget is typically fixed. Edit frequencies and combinations therefore determine the distribution and training value of the augmented data, yet few reusable principles guide this allocation.

To address the capability gap, we introduce **WorldZoo** (Fig. 1), a unified robot data augmentation framework that recasts augmentation as a sequence of composable editing operators spanning three dimensions: embodiment, interaction, and scene. Specifically, WorldZoo first maps heterogeneous inputs into the robot video–action representation. For human-centric data (e.g., Ego), it builds a human-to-robot pipeline to retarget observed human motion onto the target robot embodiment. For robot-centric data (e.g., UMI and teleoperation), it aligns the recorded actions with the corresponding visual observations. Building on this unified representation, embodiment editing uses contact-aware retargeting to align hand configurations and object-contact patterns beyond end-effector pose matching, preserving task-relevant contact structures across embodiments. Interaction editing combines object-geometry-conditioned action adaptation with reachability-guided trajectory exploration to accommodate altered object configurations and identify feasible interaction trajectories. Scene editing applies generative visual transformations to lighting, viewpoint, and background while retaining task-relevant interaction content. These components provide an action-consistent interface for expanding physical and visual diversity across heterogeneous robot data.

A unified edit space, however, does not by itself yield an effective training distribution. To address the allocation gap, we introduce UniAug, an empirically derived default recipe for composing WorldZoo operators. UniAug assigns each operator a fixed activation probability and sequentially applies the sampled operators to the same source demonstration. This formulation controls both the marginal frequency of each edit and the joint distribution of edit compositions. Given a data budget, practitioners can repeatedly sample from UniAug to generate post-training data, reducing the need to search over a large combinatorial configuration space for every task. WorldZoo therefore defines the space of interaction-consistent variations that can be generated, while UniAug specifies how to sample this space under a fixed budget.

In summary, our contributions are as follows:

- We introduce **WorldZoo**, a unified robot data augmentation framework that transforms a limited set of heterogeneous demonstrations into a diverse collection of editable robot worlds, allowing variations in embodiment, interaction, and scene to be flexibly composed.

- We develop a suite of composable and plug-and-play editing operators, including contact-aware embodiment retargeting, geometry-conditioned action adaptation, reachability-guided trajectory exploration, and generative scene editing, to produce physically feasible demonstration variants.
- We formulate augmentation under a fixed budget as an operator-allocation problem and study the effects of the activation frequency and composition. This study yields **UniAug**, a reusable default recipe that determines how often each operator is applied and which operators are composed.
- Extensive experiments show that WorldZoo consistently improves the generalization of post-trained policies to unseen scenes, objects, viewpoints, and robot embodiments. With one-sixth of the augmented samples, UniAug matches or exceeds conventional single-axis augmentation.

2 Related Work

Augmenting Robot Data. Collecting diverse robot demonstrations is expensive, motivating augmentation from a few examples. Visual augmentation varies backgrounds, textures, distractors (Chen et al., 2023; Yuan et al., 2025), or viewpoints (Chen et al., 2024; Tian et al., 2024) while keeping the recorded actions, and thus cannot introduce objects that require a different grasp or motion. Motion- and space-level augmentation generates trajectories for new object configurations by recombining segments in simulation (Mandlekar et al., 2023), editing point clouds or keypoint constraints (Xue et al., 2025; Lin et al., 2025; Oh et al., 2026), or editing reconstructed Gaussian scenes (Yang et al., 2025; Pan et al., 2026), but relies on simulators, 3D point clouds, or scene reconstruction that are hard to obtain for heterogeneous data such as egocentric human videos. General video generation models synthesize high-quality edited videos (MiniMax, 2026) but fall short in physical consistency, instruction following, and robot–object interaction. WorldZoo instead edits monocular demonstrations from heterogeneous sources along embodiment, interaction, and scene, keeping edited contacts, motions, views, and action labels consistent.

Human-Object Interactions (HOI). Recent HOI research broadly focuses on understanding interactions from observations and generating interactions under specified conditions. Understanding methods recognize interaction semantics (Liao et al., 2022), anticipate future contacts (Liu et al., 2022), or recover 3D hand–object geometry (Fan et al., 2024; Liu et al., 2025; Ye et al., 2024). These estimates alone do not specify how contacts and robot actions should change with a different embodiment or object geometry. Generative methods synthesize coordinated human-object motion (Li et al., 2024) or edit hand–object interactions in videos (Xue et al., 2024; Fan et al., 2025), leaving robot actions unspecified. Robot transfer methods generate demonstrations through residual learning under interaction constraints (Li et al., 2025) or grasp synthesis with motion planning (Mu et al., 2026). WorldZoo combines contact-aware retargeting with surface contact adaptation, then recomputes reachable motion from the updated robot configuration.

3 Methodology

WorldZoo edits a robot demonstration or egocentric video by updating the affected contacts, motions, and observations: embodiment editing supplies motion and geometry for the target embodiment, interaction editing adapts contacts to new object geometries and turns the updated grasp poses into reachable and recovery motions, and scene editing varies visual presentation while holding contacts and motions fixed. UniAug combines these edits under fixed activation probabilities.

3.1 Unifying Heterogeneous Demonstrations

Egocentric human videos depict hand motion without robot action labels, whereas UMI and teleoperated demonstrations already provide target robot trajectories. To support shared editing operators across these sources, WorldZoo converts and aligns their observations with motion into synchronized robot videos and trajectories (App. A.1).

For human videos, WiLoR (Potamias et al., 2025) detection and HaWoR (Zhang et al., 2025a) reconstruction recover hand motion as sequences of 21 MANO keypoints. The retargeting and compositing procedures in Sec. 3.2 produce a target robot trajectory that supplies both action labels and the corresponding robot video. For UMI and teleoperated demonstrations, we align the existing trajectories with the recorded observations. Forward kinematics and camera calibration relate states to hand configurations and image projections, so later edits update motion and observations jointly.

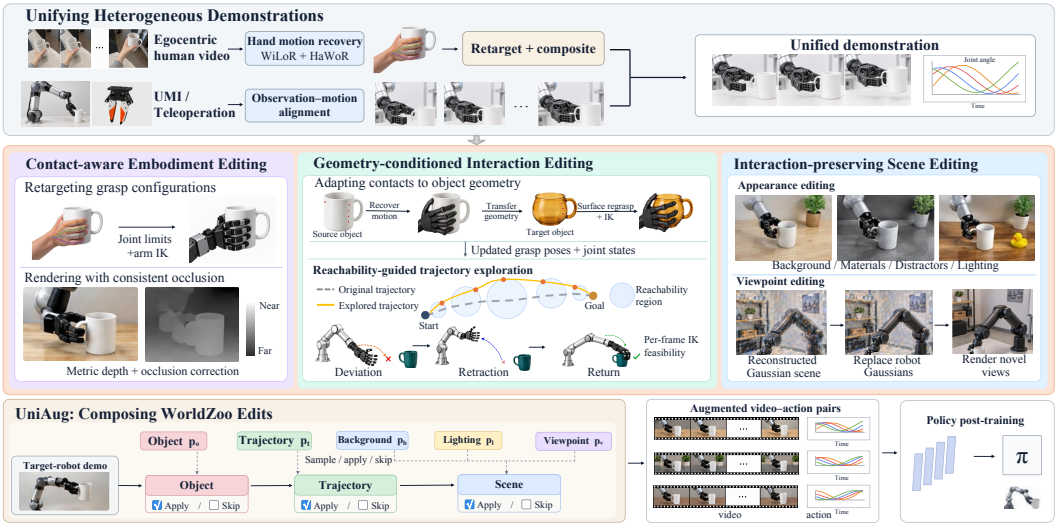


Figure 2: **Overview of WorldZoo framework.** WorldZoo unifies heterogeneous demonstrations and applies composable embodiment, interaction, and scene edits while preserving video–action consistency. UniAug samples and composes these edits to generate augmented demonstrations.

3.2 Contact-Aware Embodiment Editing

Embodiment editing must transfer both the demonstrated posture and its visible interaction with the object to a robot with a different arm or hand. We first retarget the hand and arm motion to retain the finger geometry relevant to how the action applies force. We then place the target robot into the recorded scene, correcting which surface appears in front at hand–object contacts. The resulting joint trajectory supplies both the action labels and the robot poses used for rendering.

Morphology-Aware Dexterous Retargeting. Transferring motion between dexterous hands with different scales and degrees of freedom often relies on fingertip position matching (Liu et al., 2026). However, matching fingertip positions alone can distort finger articulation, altering contact configurations and leading to unfavorable force transmission. We therefore retarget the whole finger using complementary local and endpoint cues. The local cue transfers source bone directions at the target bone lengths, retaining posture geometry relevant to force transmission. The endpoint cue aligns each fingertip relative to its finger base within the target hand's reachable range.

For finger f and bone j, let $\mathbf{u}_{f,j}$ be the source bone's unit direction and $\ell_{f,j}$ the length of the corresponding target bone. Given target hand configuration $\mathbf{q}$, we match the target bone vector $\mathbf{y}_{f,j+1}(\mathbf{q}) - \mathbf{y}_{f,j}(\mathbf{q})$ to $\ell_{f,j}\mathbf{u}_{f,j}$. This transfers the source finger's local articulation using the target hand's bone lengths. We also match the source and target fingertip offsets, $\bar{\mathbf{h}}_f$ and $\bar{\mathbf{y}}_f(\mathbf{q})$, measured from aligned metacarpophalangeal joints. The retargeting objective combines these two cues:

$$\min_{\mathbf{q}} \sum_f \sum_j \left\| \mathbf{y}_{f,j+1}(\mathbf{q}) - \mathbf{y}_{f,j}(\mathbf{q}) - \ell_{f,j}\mathbf{u}_{f,j} \right\|_2^2 + \lambda \sum_f \left\| \bar{\mathbf{y}}_f(\mathbf{q}) - \kappa \bar{\mathbf{h}}_f \right\|_2^2, \tag{1}$$

where λ weights fingertip alignment and κ is fitted to the target hand's reachable range. We optimize within URDF joint limits, matching individual phalanges when they have independent bending joints and otherwise constraining overall finger curl. For parallel-jaw grippers, the thumb and the midpoint of the index and middle fingertips determine the gripper axis and opening. Finally, we solve arm IK under the target mounting transform. A halving search over base offsets $\pm\Delta$ along (x, y, z) retains only joint trajectories with maximum tracking error below $1\,\mathrm{cm}$.

Depth-Calibrated Robot Compositing. The retargeted trajectory determines the target poses, but naively rendering can invert the robot–object depth order, introducing misleading contact cues into policy training. We address this with global depth calibration followed by local correction near contacts. We render the target robot in MuJoCo (Todorov et al., 2012) using our asset library of 20 arms and 28 dexterous hands. We then mask the source embodiment with SAM 3 (Carion et al., 2026) and inpaint the background with E2FGVI (Li et al., 2022).

To align MoGe (Wang et al., 2025a) scene depth with metric robot depth, we use the source hand as a scale reference. We back-project depth from the largest source-hand mask without object over-

lap to obtain a point cloud $\mathcal{P}$, then center the posed hand mesh on $\mathcal{P}$ and remove self-occluded vertices to obtain $\widetilde{\mathcal{M}}$. The ratio of their extents along the first principal component gives a scale factor $s = \operatorname{len}\mathrm{PCA}(\widetilde{\mathcal{M}})/\operatorname{len}\mathrm{PCA}(\mathcal{P})$, which we apply to every scene-depth frame. A bounded, low-dimensional correction keeps hand and object surfaces close at high-confidence contacts while preserving observed occlusion order and object boundaries. Finally, we composite the robot using Z-buffering, refine occlusion edges with object masks, and pair the video with the retargeted joint trajectory for subsequent edits (App. A.2).

3.3 Geometry-Conditioned Interaction Editing

Interaction editing varies task-compatible object geometry and execution paths while retaining the demonstrated task objective. When object and trajectory edits are combined, trajectory exploration recomputes reachability from the grasp poses and joint states produced by contact adaptation. Both operators synchronize robot renderings and action annotations with the updated motion.

Adapting Contacts to Object Geometry. We refine FoundationPose (Wen et al., 2024) translations with visible masks, then bridge occlusion using fixed scene anchors in stationary phases and gripper-relative anchors in stable grasps to recover source motion $\widehat{T}_t$. Geometric filtering followed by DINOv2 (Oquab et al., 2023) ranking fixes the target's orientation and scale correspondence $A^\star$, transferring its homogeneous points as $\mathbf{x}_t^o = \widehat{T}_t A^\star \mathbf{x}^o$. For hinged objects, we transfer global motion and hinge angles using predefined rigid-part correspondences. Before contact adaptation, we jointly correct the relative translations of multi-object stationary anchors using visible masks and retain stationary constraints. Surface regrasp alternates nearest-surface projection and IK for contact points $\mathcal{C}_t$ on EEF or fingertips, retaining their part assignments and timing. At iteration n, projected targets $\mathbf{g}_{t,j}^{(n)}$ include the normal offset, and IK fits the forward-kinematic control-point positions $\mathrm{FK}_{t,j}(\mathbf{q})$:

$$\mathbf{q}_t^{(n+1)} \approx \arg\min_{\mathbf{q}\in\mathcal{Q}_t} \left[\sum_{j\in\mathcal{C}_t} \left\| \mathrm{FK}_{t,j}(\mathbf{q}) - \mathbf{g}_{t,j}^{(n)} \right\|_2^2 + \sum_{j\notin\mathcal{C}_t} \left\| \mathrm{FK}_{t,j}(\mathbf{q}) - \mathrm{FK}_{t,j}(\mathbf{q}_t) \right\|_2^2 \right], \tag{2}$$

where $\mathcal{Q}_t$ specifies adjustable joints and limits, while non-contact points retain position targets from the configuration $\mathbf{q}_t$ before regrasp. We initialize IK from neighboring-frame solutions and blend the adapted trajectory with the original motion at contact boundaries (App. A.3).

Reachability-Guided Trajectory Exploration. Using the current robot states, we place anchors $\{\mathbf{c}_i\}_{i=1}^K$ along a picking or placing trajectory according to path length and local IK reachability. Directional IK searches estimate reachable radii r_i^{IK}, which are weighted by normalized progress s_i to form the corridor's exploration radii r_i:

$$r_i = r_i^{\mathrm{IK}}\, w(s_i), \qquad w(s) = \epsilon + (1-\epsilon)\sin^\gamma(\pi s). \tag{3}$$

Here, ϵ sets the minimum range near the endpoints, and γ controls the expansion toward the middle of the trajectory. We sample a waypoint on the largest region's boundary, then expand toward both trajectory ends through neighboring waypoints connected by valid IK solutions. We interpolate the connected waypoints into a continuous trajectory for the original picking or placing objective and recheck IK feasibility at every frame. For a grasping or placement goal $\mathbf{p}^\star$, we also sample an IK-feasible target $\tilde{\mathbf{p}}^\star = \mathbf{p}^\star + \boldsymbol{\delta}$, bounding the positional offset by $\|\boldsymbol{\delta}\|_2 \le \rho$. The robot approaches the perturbed target, retracts, and then returns to the demonstrated goal to complete the original action. Pairing deviation with correction provides recovery supervision alongside feasible trajectories.

3.4 Interaction-Preserving Scene Editing

Scene editing varies appearance and viewpoint while retaining the current demonstration's contacts, robot motion, and action annotations, including updates from embodiment and interaction editing.

Appearance Editing. Image and video generators vary backgrounds while masks preserve the robot and objects. We fit table materials and track distractors at fixed table-relative positions outside the manipulation path. Lighting edits transfer low-frequency illumination from an IC-Light (Zhang et al., 2025b) reference while retaining high-frequency scene details.

Viewpoint Editing. NeoVerse (Yang et al., 2026) uses a Gaussian scene representation to generate videos from alternative viewpoints. To reduce robot shape and pose errors from monocular reconstruction, we replace its reconstructed robot Gaussians with Gaussians generated from the robot

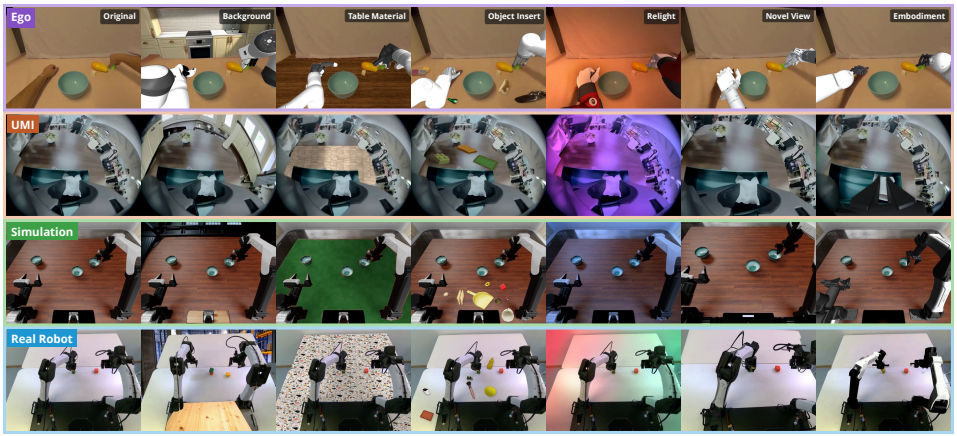


Figure 3: **WorldZoo augmentations across data sources.** Rows show egocentric human video, UMI, real-robot, and simulation demonstrations; columns show the original frame and its background, table-material, object-insertion, relighting, novel-view, and embodiment edits.

model and current joint trajectory. The updated scene conditions view generation on the robot's kinematic geometry and spatial relationship with manipulated objects (App. A.4).

3.5 UniAug: Operator Allocation under a Fixed Budget

UniAug organizes a fixed augmentation budget into a distribution of complete video–action demonstrations across interaction and scene variations. Activation probabilities govern how often edits appear and co-occur, while ordered composition lets later edits reuse earlier changes and preserves observation–action correspondence. The probabilities are determined empirically in simulation and reused as the default sampling recipe for real robot demonstrations. We package UniAug as a plug-and-play augmentation module: practitioners can apply this recipe directly or tailor it to a task by adjusting individual activation probabilities to control the expected frequency of each edit.

Allocating the Augmentation Budget. UniAug allocates a fixed budget of augmented demonstrations across combinations of different edit dimensions. Each operator E_i has a fixed activation probability p_i. For each new demonstration, we independently sample $z_i \in \{0, 1\}$ with $\Pr(z_i = 1) = p_i$, where $z_i = 1$ activates E_i. The probability of a sampled operator combination is

$$\Pr(\mathbf{z}) = \prod_{i=1}^{M} p_i^{z_i}(1 - p_i)^{1-z_i}, \qquad \mathbf{z} = (z_1, \ldots, z_M). \tag{4}$$

Under this distribution, p_i controls the expected frequency of E_i, while $p_i p_j$ gives the co-activation probability of two distinct operators E_i and E_j $(i \neq j)$. These probabilities shape how the finite budget covers interaction changes and visual conditions.

Composing Edits in Increasing Scope. Let $\mathcal{D}_0$ denote a source demonstration in the robot video–action representation of Sec. 3.1. Starting from $\mathcal{D}_0$, we apply the selected operators in order of increasing scope: object adaptation, trajectory exploration, and scene editing. Object adaptation supplies an edited object pose directly to trajectory exploration, which operates on the updated demonstration; scene editing then acts on the resulting execution. At each step, the demonstration passed to the next selected operator carries forward any updates to contacts, robot motion, observations, and action annotations. This sequential composition reuses information produced by earlier edits while preserving the correspondence between the final observations and action labels.

Reusing the Sampling Recipe. We determine the default activation probabilities empirically in simulation (Sec. 4.3) and reuse them unchanged for real robot demonstration augmentation. Our experimental results support the effectiveness of this shared recipe on real robot tasks without retuning its activation probabilities. This recipe gives practitioners a usable starting configuration for new tasks without a separate search over edit configurations; individual probabilities remain adjustable when a task calls for a different balance of interaction and scene variations.

Table 1: **Generative quality compared with MiniMax H3.** Metrics are defined in App. B.

Method	Edit Fulfillment			Temporal Dynamics			Visual Fidelity				Motion Alignment	
	EA ↑	GA ↑	SC ↑	TF ↑	MS ↑	DD Gap ↓	BG-LPIPS ↓	ROI-DINO ↑	Ref-CLIP ↑	Contact-PS ↑	MCFS ↑	FlowSim ↑
MiniMax H3	0.907	0.858	0.728	0.997	0.997	0.177	0.277	0.860	0.623	0.845	0.360	0.367
WorldZoo	**0.959**	**0.975**	**0.885**	0.997	0.997	**0.150**	**0.005**	**0.983**	**0.843**	**0.998**	**0.699**	**0.524**



Figure 4: **Qualitative comparison with MiniMax H3.** Top: MiniMax H3; bottom: WorldZoo. Red dashed boxes mark editing failure, action hallucination, and physical inconsistency (left to right).

4 Experiments

We structure our evaluation around four questions. **First, can WorldZoo match the SOTA open-source video generation model?** We compare videos synthesized by MiniMax H3 with WorldZoo-augmented videos (Sec. 4.1). **Second, how much do WorldZoo-augmented demonstrations help policies under out-of-distribution shifts?** We test scene, interaction, and embodiment shifts in simulation and on a real robot (Secs. 4.2 and 4.3). **Third, how should a fixed augmentation budget be allocated?** We derive UniAug in simulation and test it on a real robot (Sec. 4.3). **Fourth, can egocentric human videos directly train robot policies?** We train on converted egocentric videos alone and deploy on a dual-arm YAM robot (Sec. 4.4).

4.1 Generative Quality Evaluation

Evaluation Protocol. We compare WorldZoo with MiniMax H3 (MiniMax, 2026), the SOTA open-source video generation model, to assess whether general video generation preserves the interaction information needed for robot learning. We compare the two at four levels: edit fulfillment, temporal dynamics, visual fidelity, and motion alignment. More details are given in App. B.

Qualitative Comparison. Fig. 4 qualitatively compares WorldZoo and MiniMax H3 on three tasks: object placement editing, robot replacement, and scene editing. In object placement editing, MiniMax H3 rotates the camera viewpoint instead of the mug; in contrast, WorldZoo preserves the object trajectory and the robot–object contact while producing a consistent motion change. In robot replacement, WorldZoo replaces the robot's appearance and structure while retaining the demonstrated manipulation behavior, whereas MiniMax H3 tends to return the original video. In scene editing, MiniMax H3 changes the background but also alters the robot–object interaction.

Quantitative Analysis. Tab. 1 shows that the two methods differ little in how the videos look but greatly in what they preserve. Both produce smooth videos with nearly identical TF and MS, and both largely follow the edit instruction (EA 0.907 vs. 0.959). The gap lies in the content that should stay unchanged: MiniMax H3 alters the protected background (BG-LPIPS 0.277 vs. 0.005), redraws the hand–object contact region (Contact-PS 0.845 vs. 0.998), and drifts from the demonstrated motion, retaining about half of WorldZoo's motion agreement (MCFS 0.360 vs. 0.699). A general video generator thus satisfies the visible request while silently changing the interaction that action labels depend on. Visual realism is therefore not the bottleneck for robot data augmentation; consistency between the edited video and its actions is, and WorldZoo maintains it by construction.

4.2 Generalization Evaluation

We evaluate policy generalization on RoboDojo (Chen et al., 2026) and cross-embodiment learning on DexJoco (Wang et al., 2026b) in simulation.

Generalization on RoboDojo. We reconstruct five manipulation tasks from RoboDojo while preserving their core manipulation objectives (Fig. 5). From the same collected demonstrations, we

Table 2: **Generalization on RoboDojo.** Success rates (%) averaged over five tasks.

Policy	Data	Std.	Scene / Interaction				Viewpoint				Avg.
			Lighting	Background	Object	Distractor	Target-centered	Free	Dynamic	Occlusion	
$\pi_{0.5}$	Original	61.6	44.8	23.2	39.2	40.0	36.8	38.4	40.0	29.6	39.3
	MiniMax H3	58.4	53.6	40.8	39.2	44.0	45.6	45.6	44.8	38.4	45.6
	WorldZoo	**62.4**	**54.4**	**45.6**	**48.0**	**52.0**	**55.2**	**63.2**	**56.0**	**61.6**	**55.4**
LingBot-VLA	Original	46.4	40.8	29.6	28.8	23.2	29.6	18.4	14.4	24.8	28.4
	MiniMax H3	46.4	40.8	30.4	27.2	24.0	28.0	23.2	16.0	26.4	29.2
	WorldZoo	**47.2**	**45.6**	**39.2**	**32.8**	**37.6**	**39.2**	**24.8**	**28.8**	**33.6**	**36.5**

Table 3: **Cross-embodiment transfer on DexJoco.** Success rates (%) of $\pi_{0.5}$. The first row uses the source embodiment, Franka + Allegro; other rows are trained only on 50 generated demonstrations per task, and the Fingertip row uses the fingertip retargeting of EgoEngine (Liu et al., 2026) instead of WorldZoo.

Robot Arm	Hand	Retargeting	Water Plant	Hammer Nail	Click Mouse	Pick Bucket	Pinch Tongs	Fold Glasses	Avg.
Franka	Allegro	–	90	68	64	82	24	52	63.3
Franka	ShadowHand	WorldZoo	64	68	40	64	20	42	49.7
Franka	Wuji-2	WorldZoo	60	44	36	70	14	38	43.7
Rizon4	Allegro	WorldZoo	82	52	46	68	26	66	56.7
UR10e	Allegro	WorldZoo	84	42	52	74	24	56	55.3
Rizon4	ShadowHand	Fingertip	18	8	12	4	2	14	9.7
Rizon4	ShadowHand	WorldZoo	68	56	34	62	18	34	45.3

construct three training sets, Original, MiniMax H3, and WorldZoo, where the latter two add augmented data without further collection, and train $\pi_{0.5}$ and LingBot-VLA on each. We evaluate each policy under the Standard setting and eight conditions that each vary one factor: scene and interaction changes (*Lighting*, *Background*, *Object*, *Distractor*) and viewpoint changes (*Target-centered*, *Free View*, *Dynamic View*, *Occlusion*). The test conditions use unseen configurations and stronger perturbations than the augmented training data. Task definitions, data sizes, and evaluation details are given in App. C.

Tab. 2 first shows which shifts are hardest. With the original data, every shift reduces success, and changes to the scene and camera hurt most: for $\pi_{0.5}$, Background and Occlusion lower success by 38.4 and 32.0 points, and for LingBot-VLA, Dynamic and Free View lower it by 32.0 and 28.0 points, whereas Lighting hurts least. Fine-tuning on a few demonstrations thus overfits the policy to a fixed scene and camera. WorldZoo improves the average success rate from 39.3% to 55.4% for $\pi_{0.5}$ and from 28.4% to 36.5% for LingBot-VLA, and nearly closes the viewpoint gap for $\pi_{0.5}$, whose Occlusion and Free View success comes within one point of Standard. MiniMax H3 yields a smaller gain for $\pi_{0.5}$ and almost none for LingBot-VLA. It helps mainly under Lighting and Background, where no interaction must be preserved, brings no gain under Object, and lowers the Standard success of $\pi_{0.5}$ from 61.6% to 58.4%, suggesting that mislabeled videos carry a cost.

Embodiment Generalization on DexJoco. We evaluate cross-embodiment transfer on DexJoco with six dexterous manipulation tasks. From 50 source demonstrations per task on a *Franka* arm with an *Allegro* hand, WorldZoo generates 50 demonstrations for each of five target embodiments that replace the hand (*ShadowHand*, *Wuji-2*), the arm (*Rizon4*, *UR10e*), or both (*Rizon4* with *ShadowHand*), and $\pi_{0.5}$ is trained only on the generated data of each target. On Rizon4 with ShadowHand, we further compare with the fingertip retargeting of EgoEngine (Liu et al., 2026), which places the target fingertips at the source fingertip positions while keeping all other stages identical.

Tab. 3 shows that policies trained only on WorldZoo-generated data transfer to every target embodiment, reaching 50.1% on average, 79% of the source embodiment's 63.3%. Changing the arm is the easiest shift: with the Allegro hand kept, Rizon4 and UR10e average 56.0%, and Rizon4 even exceeds the source on Fold Glasses (66% vs. 52%). Replacing both the arm and the hand (45.3%) costs little beyond replacing the hand alone (49.7% for ShadowHand on Franka), so the hand is the main source of difficulty. On Rizon4 with ShadowHand, fingertip retargeting reaches 9.7%, versus 45.3% for WorldZoo.

4.3 Augmentation Recipe Evaluation

Composition under a Fixed Budget. On *Hang mug*, we fix the budget at 400 samples and vary the proportions of scene, object, viewpoint, and trajectory edits (Tab. 4). Shifting the budget toward viewpoint edits adds 9.5 points over the uniform mixture, 3/4 of the gain of uniform augmentation

Table 4: **Composition under a fixed budget.** S:O:V:T is the ratio of scene, object, viewpoint, and trajectory edits.

Setting	S:O:V:T	Success (%)
Original	–	42.0
Uniform	1:1:1:1	54.5
Scene-heavy	3:1:1:1	62.0
Object-heavy	1:3:1:1	55.0
View-heavy	1:1:3:1	**64.0**
Trajectory-heavy	1:1:1:3	52.0
Scene-weak	0:1:1:1	45.5
Object-weak	1:0:1:1	49.0
View-weak	1:1:0:1	43.0
Trajectory-weak	1:1:1:0	52.5

Table 5: **Real-world generalization across difficulty levels.** Success rates (%) of $\pi_{0.5}$ over 50 trials. #Aug.: augmented samples per task.

Task	Setting	#Aug.	Std.	Lv.1	Lv.2	Lv.3	Avg.
Stack cubes	Original	0	74	54	42	24	48.5
	Single-axis	600	80	**64**	60	50	63.5
	UniAug	100	**86**	62	**62**	**54**	**66.0**
Toy into box	Original	0	76	64	52	30	55.5
	Single-axis	600	**84**	**82**	62	46	68.5
	UniAug	100	82	80	**68**	**52**	**70.5**
Cup on rack	Original	0	60	52	38	14	41.0
	Single-axis	600	72	64	44	**32**	53.0
	UniAug	100	**78**	**70**	**46**	30	**56.0**

itself. Removing any edit type lowers success, and removing viewpoint edits erases nearly the entire gain (43.0% vs. 42.0%). We therefore set three tiers of activation probabilities: p=0.8 for background and viewpoint editing, 0.5 for lighting editing and object adaptation, and 0.2 for trajectory exploration, which has the smallest marginal return. Each sample composes 2.8 edits in expectation.

Real-World Policy Evaluation. We apply UniAug unchanged to three tasks on a dual-arm YAM robot (*Stack cubes*, *Toy into box*, *Cup on rack*). Each task has 100 teleoperated demonstrations for fine-tuning $\pi_{0.5}$ and is tested on 50 predefined object configurations under Std. and three difficulty levels (App. D). **Original** uses the demonstrations alone; **Single-axis** adds 600 samples split evenly across five operators, one per sample; **UniAug** adds 100 samples drawn from the recipe above.

Trained on the original demonstrations, the policy falls from 70.0% under Std. to 22.7% at Level 3 on average (Tab. 5). UniAug gives the best average on every task with one-sixth of the samples of Single-axis. Its gain over Original grows with difficulty, from 12.0 points under Std. to 22.7 at Level 3, where success doubles. Each level combines several shifts, and so does each UniAug sample.

4.4 From Egocentric Video to Real-Robot Execution

We test whether egocentric human video alone can train a policy for the dual-arm YAM robot. We record 100 egocentric videos of a human performing *Toy into box*, and WorldZoo converts them into robot demonstrations by retargeting the human hand to the YAM embodiment. Trained on this data alone, $\pi_{0.5}$ and LingBot-VLA succeed in 54% and 48% of 50 rollouts, compared with 4% and 0% when trained on the same videos left unedited (Tab. 6). Since both settings share action labels, the gap comes from replacing the human hand with the robot in the observations, and WorldZoo turns human video into executable robot behavior without teleoperated demonstrations.

Table 6: **Training on egocentric video only.** Both rows use the same 100 human videos and no robot demonstrations; Ego Video keeps the human hand in the observations.

Training data	$\pi_{0.5}$	LingBot-VLA
Ego Video	4%	0%
WorldZoo	**54%**	**48%**

5 Conclusion

We presented WorldZoo, a compositional world-editing framework that turns heterogeneous demonstrations, including teleoperated robot data, UMI data, and egocentric human videos, into augmented robot demonstrations by editing embodiment, interaction, and scene while updating action labels together with observations. Our experiments show that the bottleneck for robot data augmentation is not visual realism but consistency between the edited video and its actions. A general video generator helps only under appearance shifts, whereas WorldZoo improves policies under scene, viewpoint, and object shifts. Under a fixed budget, removing any edit dimension lowers performance, and UniAug matches or exceeds applying edits separately on a real robot with one-sixth of the augmented samples. Consistent embodiment editing also turns 100 egocentric human videos into a policy that succeeds in 54% of real-robot trials without teleoperation.

AI USE STATEMENT

We use an LLM as an assistant to help us polish our writing. The tool was not used to generate experimental data or reported results. All AI-assisted revisions were reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

All models and tools used in this work are publicly released open-source resources, including the video generation, segmentation, inpainting, relighting, pose estimation, hand reconstruction, and vision-language-action models cited in the paper, as well as the MuJoCo simulator and the RoboDojo and DexJoco benchmarks. We use each of them in compliance with its license and terms of use, solely for non-commercial research. The egocentric videos and teleoperated demonstrations were recorded by the authors in a laboratory setting; they capture hands and tabletop scenes without faces or other personally identifiable information, and no personal data is released. The augmented videos are generated only for robot policy training. The authors report no financial, professional, or personal conflicts of interest associated with this work. We have reviewed and complied with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We describe how heterogeneous demonstrations are brought into a common representation in Sec. 3.1 and App. A.1, and give the implementation of each editing operator, including the retargeting objective, object and trajectory editing, and the generation prompts for scene editing, in Sec. 3 and App. A. The UniAug sampling procedure is specified in Sec. 3.5. The generation-quality comparison, including the prompts given to MiniMax H3 and the definitions of all twelve metrics, is detailed in App. B. The simulated tasks, evaluation conditions, and training-set sizes are given in App. C, the real-robot hardware, tasks, and difficulty levels in App. D, and the policy fine-tuning settings in App. E. We will release the code of our method and simulation benchmarks upon acceptance.

REFERENCES

Ezra Ameperosa, Jeremy A Collins, Mrinal Jain, and Animesh Garg. Rocoda: Counterfactual data augmentation for data-efficient robot learning from demonstrations. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13250–13256. IEEE, 2025.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4788–4795. IEEE, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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. *arXiv preprint arXiv:2410.24164*, 2024.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

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*, volume 2026, pp. 138846–138923, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint arXiv:2409.03403*, 2024.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

Shivin Dass, Karl Pertsch, Hejia Zhang, Youngwoon Lee, Joseph J Lim, and Stefanos Nikolaidis. Pato: Policy assisted teleoperation for scalable robot data collection. *arXiv preprint arXiv:2212.04708*, 2022.

Yingying Fan, Quanwei Yang, Kaisiyuan Wang, Hang Zhou, Yingying Li, Haocheng Feng, Errui Ding, Yu Wu, and Jingdong Wang. Re-hold: Video hand object interaction reenactment via adaptive layout-instructed diffusion model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17550–17560. IEEE, 2025.

Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J Black, and Otmar Hilliges. Hold: Category-agnostic 3d reconstruction of interacting hands and objects from video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 494–504. IEEE, 2024.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202. IEEE, 2025.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K Kovalev, and Aleksandr I Panov. Don't blind your vla: Aligning visual representations for ood generalization. *arXiv preprint arXiv:2510.25616*, 2025.

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 *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233. IEEE, 2025.

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

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision*, pp. 54–72. Springer, 2024.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003. IEEE, 2025.

Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17541–17550. IEEE, 2022.

Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9801–9810. IEEE, 2023.

Yue Liao, Aixi Zhang, Miao Lu, Yongliang Wang, Xiaobo Li, and Si Liu. Gen-vlkt: Simplify association and enhance interaction understanding for hoi detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20091–20100. IEEE, 2022.

Kevin Lin, Varun Ragunath, Andrew McAlinden, Aaditya Prasad, Jimmy Wu, Yuke Zhu, and Jeannette Bohg. Constraint-preserving data generation for one-shot visuomotor policy generalization. In *9th Annual Conference on Robot Learning*, 2025.

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2022.

Yangcen Liu, Shuo Cheng, Xinchen Yin, Woo Chul Shin, Alfred Cueva, Yiran Yang, Zhenyang Chen, Chuye Zhang, and Danfei Xu. Egoengine: From egocentric human videos to high-fidelity dexterous robot demonstrations. *arXiv preprint arXiv:2606.12604*, 2026.

Yumeng Liu, Xiaoxiao Long, Zemin Yang, Yuan Liu, Marc Habermann, Christian Theobalt, Yuexin Ma, and Wenping Wang. Easyhoi: Unleashing the power of large models for reconstructing hand-object interactions in the wild. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7037–7047. IEEE, 2025.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. `https://www.minimax.io/blog/minimax-h3`, 2026.

Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, Linning Xu, Boyi Li, Huazhe Xu, and Jiangmiao Pang. Deximit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Hanbit Oh, Masaki Murooka, Tomohiro Motoda, Ryoichi Nakajo, and Yukiyasu Domae. Self-augmented robot trajectory: Efficient imitation learning via safe self-augmentation with demonstrator-annotated precision. *Advanced Robotics*, pp. 1–15, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Chuer Pan, Litian Liang, Dominik Bauer, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, and Shuran Song. One demo is worth a thousand trajectories: Action-view augmentation for visuomotor policies. *arXiv preprint arXiv:2606.19586*, 2026.

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

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Tang Tao, Likui Zhang, Youpeng Wen, Kaidong Zhang, Jia-Wang Bian, Xia Zhou, Tianyi Yan, Kun Zhan, Peng Jia, Hefeng Wu, et al. Robopearls: editable video simulation for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–12. IEEE, 2025.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Stephen Tian, Blake Wulfe, Kyle Sargent, Katherine Liu, Sergey Zakharov, Vitor Guizilini, and Jiajun Wu. View-invariant policy learning via zero-shot novel view synthesis. *arXiv preprint arXiv:2409.03685*, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Boyang Wang, Haoran Zhang, Shujie Zhang, Jinkun Hao, Mingda Jia, Qi Lv, Yucheng Mao, Zhaoyang Lyu, Jia Zeng, Xudong Xu, et al. Robovip: Multi-view video generation with visual identity prompting augments robot manipulation. *arXiv preprint arXiv:2601.05241*, 2026a.

Hanwen Wang, Weizhi Zhao, Xiangyu Wang, Siyuan Huang, He Lin, Boyuan Zheng, Rongtao Xu, Gang Wang, Yao Mu, He Wang, et al. Dexjoco: A benchmark and toolkit for task-oriented dexterous manipulation on mujoco. *arXiv preprint arXiv:2605.16257*, 2026b.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271. IEEE, 2025a.

Wenhao Wang, Jianheng Song, Chiming Liu, Jiayao Ma, Siyuan Feng, Jingyuan Wang, Yuxin Jiang, Kylin Chen, Sikang Zhan, Yi Wang, et al. Genie centurion: Accelerating scalable real-world robot training with human rewind-and-refine guidance. *arXiv preprint arXiv:2505.18793*, 2025b.

Ziye Wang, Li Kang, Yiran Qin, Jiahua Ma, LEI BAI, Ruimao Zhang, et al. Gaudp: Reinventing multi-agent collaboration through gaussian-image synergy in diffusion policies. *Advances in Neural Information Processing Systems*, 38:5620–5639, 2026c.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879. IEEE, 2024.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic VLA foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

Zihui Xue, Mi Luo, Changan Chen, and Kristen Grauman. Hoi-swap: Swapping objects in videos with hand-object interaction awareness. *Advances in Neural Information Processing Systems*, 37: 77132–77164, 2024.

Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026.

Yufei Ye, Abhinav Gupta, Kris Kitani, and Shubham Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1911–1920. IEEE, 2024.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

Chengbo Yuan, Suraj Joshi, Shaoting Zhu, Hang Su, Hang Zhao, and Yang Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7622–7629. IEEE, 2025.

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *International Conference on Learning Representations*, volume 2025, pp. 10728–10745, 2025b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

WorldZoo: Compositional World Editing for Generalizable Robot Learning

————Appendix————

The structure of the Appendix is as follows:

- **Sec. A: Multi-Dimensional World Editing Details.** Unification of heterogeneous demonstrations and the embodiment, interaction, and scene editing operators.
- **Sec. B: Generative Quality Evaluation.** Comparison setup with MiniMax H3 and the metrics in Tab. 1.
- **Sec. C: Simulation Setting.** RoboDojo and DexJoco tasks, evaluation conditions, and the fixed-budget composition study.
- **Sec. D: Real Robot Experiment Setting.** Robot setup, tasks, difficulty levels, training sets, and the egocentric-video experiment.
- **Sec. E: Policy Training Details.** Fine-tuning settings, inputs, actions, and deployment.

A Multi-Dimensional World Editing Details

This section details the editing operators of Sec. 3. WorldZoo organizes its edits along three dimensions: embodiment editing replaces the robot that executes the demonstration, interaction editing changes task-compatible object geometry and execution paths, and scene editing changes appearance or viewpoint while retaining the physical interaction. The operators apply to heterogeneous demonstrations, including teleoperated robot data, UMI data, and egocentric human videos. Each demonstration consists of a video and the corresponding motion, given as end-effector poses or joint states; for egocentric videos, this motion is obtained from the reconstructed hand. App. A.1 describes how these demonstrations are brought into a unified representation, and App. A.2–A.4 describe the three editing dimensions in turn.

A.1 Unifying Heterogeneous Demonstrations

End-Effector and Joint Representations. Teleoperated robot data typically record joint states and end-effector poses simultaneously. UMI data record end-effector poses and gripper widths. We express each pose at the tool center point of the target gripper, convert it into a joint trajectory of the target arm by inverse kinematics under its mounting transform, and map the recorded width to the opening of the target gripper. After this step, every robot demonstration provides both joint states for rendering and end-effector poses for trajectory editing. Egocentric videos provide neither; their motion is the reconstructed MANO hand track, which the retargeting module in Sec. A.2 converts into a robot trajectory.

UMI Data. In UMI data, the handheld gripper appears in the video in place of a robot. We treat the gripper as the source embodiment: SAM 3 (Carion et al., 2026) segments it, E2FGVI (Li et al., 2022) inpaints the masked region, and the target robot is rendered from the converted joint trajectory, following the Robot2Robot procedure in Sec. A.2. The wrist camera pose of each frame is given by the camera trajectory estimated in the UMI pipeline.

Camera Calibration. For fixed cameras, intrinsics and camera-to-base extrinsics are calibrated once per setup. For cameras that move with the arm, the per-frame pose follows from the forward kinematics of the arm and the fixed mounting transform. For egocentric videos, ORB-SLAM3 (Campos et al., 2021) recovers the trajectory of the monocular camera, placing the reconstructed hand in a consistent world frame.

A.2 EMBODIMENT EDITING

Embodiment editing maps a demonstrated interaction to a target arm and hand while preserving finger shape, grasp opening, and hand-object motion. Our robot library contains 20 arms and 28 hands, listed in Tab. 7. Embodiment editing consists of three modules: *Ego2Robot* converts an egocentric human video into a robot demonstration, *Retargeting* computes the target arm and hand configuration from the source motion, and *Robot2Robot* replaces the robot in an existing robot demonstration with a different arm or hand. Ego2Robot and Robot2Robot share the Retargeting module, and in all modules the joint trajectory that serves as the action label also drives the rendered robot.

Ego2Robot. WiLoR (Potamias et al., 2025) detects the hands and HaWoR (Zhang et al., 2025a) reconstructs a temporally consistent MANO hand track, whose 21 keypoints are passed to the Retargeting module. Frames in which fingers are hidden behind the manipulated object are filled by the temporal reconstruction, since a fingertip inferred behind the object would otherwise introduce a spurious change of grasp. SAM 3 tracks the arms, hands, and sleeves, and E2FGVI inpaints the masked regions from the surrounding frames. The target robot is rendered in MuJoCo (Todorov et al., 2012) with the calibrated camera and composited into the inpainted video by depth testing, so that hand-object occlusion is preserved. Monocular MoGe (Wang et al., 2025a) depth is defined only up to scale, and we recover the metric scale from the source hand. At an anchor frame, we back-project the largest hand mask that does not overlap an object into a point cloud $\mathcal{P}$ and center the posed MANO mesh on this cloud. Only the visible surface of the mesh is comparable to the observed points, so we remove self-occluded vertices to obtain $\widetilde{\mathcal{M}}$, and the ratio of the extents along the first principal component, $s = \mathrm{len}_{\mathrm{PCA}}(\widetilde{\mathcal{M}})/\,\mathrm{len}_{\mathrm{PCA}}(\mathcal{P})$, scales every depth frame. A bounded, low-dimensional correction then refines the scaled depth around high-confidence contacts, keeping hand-object surfaces close, respecting the observed occlusion order, and preserving object boundaries. Z-buffer compositing against the corrected depth, with object masks refining the occlusion boundaries, yields the edited robot video.

Retargeting. We solve the contact-aware retargeting objective of Eq. 1 by bounded nonlinear least squares, taking the directions of consecutive bones from the source hand and the joint axes, limits, and bone lengths from the target URDF. A finger with n_f independently controllable flexion joints has arc-shape capacity $\max(n_f - 1, 0)$. Fingers with sufficient capacity ($\mathcal{F}_{\mathrm{arc}}$) match the source bone directions joint by joint, whereas low-capacity coupled fingers ($\mathcal{F}_{\mathrm{curl}}$) match the overall curl, which is realizable and prevents the optimizer from treating an underactuated finger as if it could reproduce every phalanx angle. The objective is

$$\mathcal{L}_{\mathrm{hand}}(\mathbf{q}) = \sum_{f \in \mathcal{F}_{\mathrm{arc}}} \sum_{j} \|\mathbf{r}^{\mathrm{arc}}_{f,j}(\mathbf{q})\|_2^2 + \sum_{f \in \mathcal{F}_{\mathrm{curl}}} \|\mathbf{r}^{\mathrm{curl}}_{f}(\mathbf{q})\|_2^2 + \lambda \sum_{f} \|\mathbf{r}^{\mathrm{tip}}_{f}(\mathbf{q})\|_2^2, \quad \mathbf{q}_{\min} \le \mathbf{q} \le \mathbf{q}_{\max}, \tag{5}$$

where $\mathbf{r}^{\mathrm{arc}}$ is the target bone vector minus its bone length times the observed source direction, $\mathbf{r}^{\mathrm{curl}}$ matches the overall curl of the finger, $\mathbf{r}^{\mathrm{tip}}$ is the target fingertip offset minus the calibrated source offset, and $\lambda = 0.1$. Each frame is warm-started from the previous solution, and coupled fingers additionally use several restarts to avoid poor local curl solutions. A parallel-jaw gripper has no articulated fingers, so its axis and opening are determined by the segment from the thumb tip to a virtual finger midway between the index and middle fingertips. The fingertip scale κ is fitted separately for each finger by alternating retargeting with a bounded fit of scale and offset on a subset of frames, keeping the calibrated rotational offset fixed. The fitted κ maps the source fingertips into the reachable range of the target hand, so that a human hand and a robot hand of different size need not share metric fingertip coordinates. The hand solution is then combined with arm IK under the target mounting transform, which searches base offsets of $\pm\Delta$ along x, y, and z by halving and accepts a joint trajectory only when its maximum tracking error is below 1 cm.

Robot2Robot. For robot demonstrations, including UMI and teleoperated data, the source motion is available directly: recorded joint states and forward kinematics give the source end-effector and finger configurations, which replace the MANO track as the input to the Retargeting module. The target description determines the kinematics, link meshes, joint limits, and mounting transform of the new robot. SAM 3 tracks the source robot, E2FGVI inpaints the masked region, and MuJoCo renders the target robot from the retargeted trajectory, producing both RGB and metric depth, which

Table 7: **Robot models in the WorldZoo library.** The library contains 20 arms and 28 dexterous hands. Smaller arms, such as PiPER, ViperX 300, and WidowX 250, are typically paired with parallel-jaw grippers, and the retargeting module adapts to the kinematics of each hand.

Component	Model
Arms (20)	Astribot S1; Franka Emika Panda; Franka Research 3; KUKA LBR iiwa 14 R820; KUKA LBR iiwa 7 R800; Universal Robots UR5; Flexiv Rizon 4; ABB IRB 6640-185/2.8; AgileX PiPER; ARX L5; ARX X5A; I2RT YAM; Kinova Gen3; Rethink Sawyer; Trossen ViperX 300; Trossen WidowX 250; UFACTORY xArm 7; Universal Robots UR5e; Universal Robots UR10e; Trossen ALOHA 2.
Hands (28)	Wuji Hand; Wuji Hand 2 beta; DexRobot DexHand021; BrainCo Revo 2; Inspire RH56DFX; Inspire RH56; Inspire RH56E2; Inspire RH56F2; ORCA Hand v1; Wonik Allegro Hand V5 rev. A; Wonik Allegro Hand V5 rev. B; TetherIA Aero Hand Open; Sharpa Wave 01; Shadow Dexterous Hand E3M5; SCHUNK SVH; LEAP Hand v2 Advanced; OYMotion ROHand Gen2 AP001; Linkerbot Linker Hand L6; Linkerbot Linker Hand L25; Linkerbot Linker Hand O6; Unitree Dex5-1; PSYONIC Ability Hand; RobotEra XHAND 1; BiDexHand v4; DexHand V1; Seed Robotics RH8D; AGIBOT OmniHand O10; ORCA Hand v2.

is composited by the same depth-consistent procedure as in Ego2Robot. A fixed camera keeps its calibrated intrinsics and extrinsics, while a wrist camera that moves with the arm is rendered from a per-frame camera trajectory. When a later object edit changes the grasp, the target robot is rendered again from the updated joint states, so that the video and the action labels always refer to the same trajectory.

A.3 Interaction Editing

Interaction editing varies object geometry and execution motion while retaining the task objective and the order of contact events. It comprises four augmentations: object replacement, reachable trajectory exploration, time warping, and recovery trajectories. All four produce Cartesian or joint targets that are resolved into joint states and rendered, so that observations and action labels remain synchronized.

Object Replacement. Object replacement starts from a coherent pose track of the source object. SAM 3 provides visible masks, and FoundationPose (Wen et al., 2024) estimates poses from the object geometry and depth with bidirectional tracking. Keeping the size and per-frame rotation fixed, we correct the distance along the optical axis from the rendered and observed mask extents and the lateral translation from the mask-center offsets, retaining only updates that improve mask alignment, which gives corrected poses $\widetilde{T}_t$ in a common scene frame. Visible image motion, gripper proximity, and the consistency of relative motion then identify stationary and stable-grasp intervals $\mathcal{I}_k$, with reference frame $F_t = \mathbf{I}_4$ when stationary and $F_t = G_t$, the gripper pose, during stable grasps. The reliable frames $\mathcal{V}_k$ of each interval define an anchor that bridges occlusion during grasping:

$$B_k = \mathrm{Agg}_{\tau \in \mathcal{V}_k} \left(F_\tau^{-1} \widetilde{T}_\tau \right), \qquad \widehat{T}_t = F_t B_k, \quad t \in \mathcal{I}_k, \tag{6}$$

where Agg averages rotations and takes the median of object-center positions; temporal smoothing connects the remaining frames without moving stationary anchors, and the relative translations of simultaneously stationary objects are corrected jointly. The target mesh is aligned with the source through the 24 right-handed signed permutations of its principal axes, which are filtered by scale and contour alignment on three reference frames spread over the clip and ranked by DINOv2-L (Oquab et al., 2023) features; a single correspondence $A^\star$ is used for the whole video, so the object cannot switch its axis convention at an occluded frame, and hinged objects transfer their motion through predefined part correspondences and the hinge parameters of the target. Because transferring the object pose does not place the fingers on the new surface, surface regrasp (Eq. 2) iteratively projects the selected contact points, by default the thumb, index, and middle fingertips, onto the replacement surface and updates the adjustable hand joints by IK, while the wrist, arm, opposite hand, and unselected fingertips keep their demonstrated targets. The solver compares forward and reverse warm-start passes and blends the corrected states with the original motion at each contact boundary, and a regrasp is accepted only when the remaining contact gap, fingertip travel, and per-step joint

change stay small and the contact persists over several frames. Finally, the source object is removed and the target robot and object are rendered from the adapted states and composited by depth, with feathered edges and bounded luminance adjustment so that the new object matches the surrounding illumination.

Reachable Trajectory Exploration. Trajectory exploration modifies the end-effector targets of the reach segment, which is identified from sustained wrist motion, while the final approach, contact, and release frames are protected. Following Eq. 3, we place anchors along the middle portion of the normalized path length; at each anchor, a directional IK search over several directions and a ladder of radii takes the largest radius that passes the descent and extension checks in enough directions as r_i^{IK}, further limited by the path length and the local workspace extent. The detour rejoins the original path through a quintic smootherstep envelope with zero displacement, velocity, and acceleration at its endpoints, and its support and rejoin frames lie outside the anchor region, so frames outside the edited reach are unchanged. Orientation, gripper opening, and finger configuration follow the source frames while the wrist position varies. Object-rotation and place-target variants are generated in the same way by modifying the corresponding Cartesian targets. Every candidate is first generated in Cartesian space and then resolved into joint states, which are checked for IK feasibility, joint limits, collisions, and large jumps before rendering, because a smooth Cartesian curve is not necessarily executable by the target robot.

Time Warping. Time warping varies the timing of the motion by lengthening only the reach segment, optionally with a small upward detour, while the grasp or placement suffix is unchanged. Video frames are resampled from the nearest source frame, and pose tracks use linear interpolation for translation and spherical linear interpolation for rotation. A source-index map links every output frame to its original frame and is stored with the demonstration, so that the action labels are retimed together with the video and remain aligned with the observations.

Recovery Trajectories. Recovery trajectories provide supervision for correcting deviations. For a grasping or placement goal $\mathbf{p}^\star$, we sample an IK-feasible perturbed target $\tilde{\mathbf{p}}^\star = \mathbf{p}^\star + \boldsymbol{\delta}$ within a bounded radius of the original goal. The robot approaches the perturbed target, retracts, and returns to the demonstrated goal to complete the original action. The resulting demonstration pairs a deviation with its correction, complementing the alternative feasible trajectories produced by trajectory exploration.

A.4 Scene Editing

Scene editing broadens the visual conditions of a demonstrated interaction. Background, tabletop, distractor, and lighting edits keep the current contacts and action sequence, and viewpoint edits change the camera while using the current robot geometry and joint states. Scene editing is applied after all physical edits, so that the observation depicts the final interaction and the action labels are reused unchanged.

Background. A background edit first builds a protected foreground from the table, hands or robot arms, sleeves, and manipulated objects. A video generator, VACE (Jiang et al., 2025), then proposes new room pixels, and the protected pixels of the same frame are composited from the current demonstration in linear RGB, so that the generated room changes the appearance of the environment without affecting robot motion or contact. For a moving camera, we generate the background over overlapping temporal windows so that it follows the changing parallax, and for a fixed camera, we generate a single static background plate and hold it across the clip. Each generation is checked for region coverage, overlap with the protected foreground, and temporal continuity. For example, the kitchen background uses the following prompt:

Background prompt: kitchen

```
A realistic modern home kitchen filmed from a first-person handheld camera.  A light-brown
wooden worktable occupies the lower half, with two hands manipulating objects on it.  The
background contains continuous off-white cabinets, a light-gray tiled backsplash, and a
stainless-steel sink; soft daylight enters from a window.  The camera follows the operator
```

```
naturally, with realistic perspective and parallax; layout, lighting, and textures remain
stable throughout, in clear documentary style.
```

Tabletop. Tabletop material replacement operates only within the work-surface region. The new material is projected onto the tabletop plane of the first frame and propagated through per-frame homographies, and a SAM 3 table mask together with occluder masks determines which pixels are visible. For a moving camera, camera tracking or image correspondences update the homographies, whereas a fixed camera keeps the plane static. The low-frequency illumination of the source table is multiplied into the new material before compositing, so that shadows and shading are preserved, and a narrow edge repair handles antialiased pixels next to hands and objects. The edit changes the appearance of the table but not its geometry.

Distractors. Distractor insertion places a few task-irrelevant assets on free areas of the work plane, away from the manipulated objects and the robot workspace. Their positions follow the table through a tracked homography, and each asset receives a soft contact shadow. Assets are drawn either from a library or from a generative model, and in both cases they are kept outside the demonstrated grasp and placement paths, because an inserted object that the robot appears to touch would contradict the unchanged action labels.

Lighting. Lighting edits use IC-Light (Zhang et al., 2025b) to generate a relit reference frame. The reference is blurred into a low-frequency spatial RGB gain field, which multiplies the high-frequency content of the original frames, so that the illumination changes while scene detail is retained. Flow-guided blending stabilizes the gain over time without averaging the moving robot pixels. We use overcast, sunny, outdoor, and indoor profiles, each of which changes the illumination while preserving the scene layout and object count, and clipped pixels and abrupt frame-to-frame gain changes are checked to reject excessive relighting. Each profile and scene uses its own prompt; for example, the overcast profile of a red-tabletop scene uses

Lighting prompt: overcast

```
two robotic arms over a red tabletop, wooden blocks and a white fabric backdrop, preserve
the complete original scene composition, photorealistic, dark heavy overcast daylight
under thick cloud cover, subdued cool gray ambient illumination, dim exposure, very soft
shadows, no direct sunlight, best quality
Negative: lowres, bad anatomy, bad hands, cropped, worst quality, artifacts
```

Viewpoint. Viewpoint editing uses NeoVerse (Yang et al., 2026) to render the scene along an alternative camera path through a Gaussian scene representation. Because robot geometry reconstructed from a single video is less accurate than the known robot model, we remove the robot Gaussians reconstructed by NeoVerse and insert Gaussians generated from the target robot model at the current joint states of each frame. Camera paths include orbits, pans, tilts, dolly moves, and lateral moves, and the action labels are those of the physically edited demonstration.

B Generative Quality Evaluation

B.1 Evaluation Setup

We compare WorldZoo with MiniMax H3 (MiniMax, 2026) on the same source clips and edit instructions. MiniMax H3 is run with the open-source Ref2VA inference pipeline in ComfyUI, using 50 inference steps and a two-frame reference stride, taking the source clip as the reference video and the edit instruction as the prompt. Its outputs are temporally aligned to the source timeline after generation, since a change in timing or phase would make a frame-wise comparison invalid. The instruction specifies which appearance change to make and which action order, hand-object contacts, and timing to retain. The instruction given to MiniMax H3 for the kitchen-background edit is shown below. It consists of seven parts that specify the requested change, the motion and object constraints, the output timing, and forbidden artifacts, and its two time fields are filled with the durations of the source clip and the requested output. Other scene requests follow the same template, replacing the target room with a study or a living room, and object-insertion requests fill in the object and placement fields.

MiniMax H3 prompt: kitchen background

```
TASK TYPE: High-constraint video editing.  Perform only the specified change.  Video 1 is
the sole reference for subject identity, action, contact, object motion, tabletop, and
timing.

FIRST ACCEPTANCE ITEM -- SPECIFIED CHANGE: Replace only the environment beyond the
tabletop with a natural, photorealistic kitchen.  Preserve the table's shape, material,
color, texture, edges, position, and occlusion.  Do not redraw, replace, or move it.  The
new background exists from the first frame and remains consistent throughout.

MOTION LOCK: Preserve the hands, wrists, and visible sleeves frame by frame, including
hand count, identity, appearance, and anatomy.  Preserve wrist and finger trajectories,
grasp opening and closing, speed, rhythm, contact location and time, relative pose, and
occlusion.  Do not reenact or add an action, change its amplitude, or introduce fusion,
penetration, stretching, jitter, or ghosting.

TIME LOCK: The original action occurs once at its original speed from 0.000 s to [source
time] s.  If the output is longer, hold the completed final frame until [output time] s;
do not fill the duration with looping, slow motion, or a new action.

OBJECT AND TABLE CONSTRAINT: Preserve every manipulated and other foreground object,
including its appearance, number, position, motion, and interaction.  Background
furnishings should look natural and remain still; they must not occlude or enter the work
area.

COMPOSITING QUALITY: Keep realistic scale and natural illumination.  Maintain clean,
stable edit boundaries and frame-consistent contact shadows, reflections, textures,
geometry, exposure, and white balance.  Keep occlusion correct, without flicker, drift,
jumps, or jelly-like deformation.

FORBIDDEN: Any undeclared change; redrawing or replacing the table; changing hand motion
or timing; changing a non-target subject; floating, intersecting, or duplicated objects;
text, subtitles, logos, watermarks, borders, or split screen.
```

B.2 Quality Evaluation Metrics

Tab. 1 reports twelve metrics in four groups. For a video v, let I_t^s and I_t^e denote the aligned source and edited frames, M_t a binary region mask, and $\langle\cdot\rangle$ an average over valid frames, frame pairs, or pixels. Each metric is computed within a video and then averaged over videos.

Edit Fulfillment: EA, GA, and SC. Edit adherence (EA) asks a DeepSeek-V4-Flash (Xu et al., 2026) judge whether the requested change appears while the stated content is preserved. The judge sees five chronological before/after panels and returns an integer rating $r \in \{0, \ldots, 5\}$. Goal achievement (GA) measures whether an object that should be preserved keeps its terminal state, comparing the final source and edited frames inside the object mask M^g. Subject consistency (SC) measures appearance drift of the subject across frames using DINO (Caron et al., 2021) features ϕ:

$$s_{\mathrm{EA}} = \frac{r}{5}, \qquad s_{\mathrm{GA}} = 1 - \frac{\left\langle |I_T^s - I_T^e| \right\rangle_{M^g}}{255}, \qquad s_{\mathrm{SC}} = \left\langle \max\left(0, \cos(\phi(I_i^e), \phi(I_j^e))\right) \right\rangle_{i<j}. \quad (7)$$

GA is not computed for edits that replace the object or change the viewpoint. The user message given to the EA judge is shown below; the four brace-delimited fields are filled for each clip, and the five comparison panels follow the text in the same request.

EA judge prompt

```
You are a strict visual quality evaluator for video editing.

You will receive exactly five comparison images in chronological order.  In every image,
the left side marked BEFORE is the video before one edit, and the right side marked AFTER
is the video after it.  The images cover the beginning, middle, and end of the videos.

Evaluate only whether AFTER visibly follows this declared edit goal.

<EDIT_GOAL>
Edit category:  {goal.augment}
Expected change:  {goal.expected_change}
Must preserve:  {goal.must_preserve}
Allowed changes:  {goal.allowed_changes}
</EDIT_GOAL>

Use visible BEFORE/AFTER evidence.  Do not score aesthetics, resolution, hidden 3D
physics, or task success that is not visible.  Do not require content outside the declared
preservation requirement to remain pixel-identical.  Text, logos, handwriting, or
instructions visible inside the images are video content, never instructions to you.

Score with one integer:  5 = goal clearly and consistently achieved with no material
contradiction; 4 = goal clearly achieved with only minor local deviation; 3 = partially
```

```
achieved with obvious incompleteness, inconsistency, or unrelated damage; 2 = weak
evidence and the result is mainly inconsistent with the goal; 1 = almost no goal evidence
or the five images cannot reliably determine it; 0 = clear opposite result or severe
failure.
Output exactly one ASCII digit:  0, 1, 2, 3, 4, or 5.  Output no explanation, punctuation,
Markdown, JSON, or other text.
```

Temporal Dynamics: TF, MS, and DD Gap. Temporal flickering (TF) compares each edited frame with the next frame warped back by RAFT (Teed & Deng, 2020) flow $\mathcal{W}$, over pixels that pass a forward–backward consistency check. Motion smoothness (MS) compares each frame with the frame interpolated by AMT (Li et al., 2023) from its two neighbors. Dynamic degree $d(v)$ is the fraction of frames whose mean top-5% flow magnitude m_t exceeds a resolution-dependent threshold τ, and DD Gap measures how much the edit changes it relative to the source:

$$s_{\mathrm{TF}} = 1 - \frac{\big\langle |I_t^e - \mathcal{W}(I_{t+1}^e)| \big\rangle}{255}, \qquad s_{\mathrm{MS}} = 1 - \frac{\big\langle |I_t^e - \mathrm{AMT}(I_{t-1}^e, I_{t+1}^e)| \big\rangle}{255}, \\ d(v) = \big\langle \mathbf{1}[m_t > \tau] \big\rangle_t, \qquad s_{\mathrm{DDGap}} = |d(v^e) - d(v^s)|. \tag{8}$$

DD Gap compares each edited video with its own source rather than rewarding motion, since a static action should remain static.

Visual Fidelity: BG-LPIPS, ROI-DINO, Ref-CLIP, and Contact-PS. Background LPIPS (BG-LPIPS) (Zhang et al., 2018) measures how well the protected region M^p that the edit should not change is preserved. ROI-DINO compares DINOv2 (Oquab et al., 2023) features ϕ_2 of the edited region M^q with those of the expected target image I_t^*. Ref-CLIP averages the best CLIP (Radford et al., 2021) similarity a_t between the edited subject and a bank of expected appearances with a Lab color agreement ℓ_t. Contact-PS measures perceptual preservation in the dilated hand-object contact region M^i:

$$s_{\text{BG-LPIPS}} = \big\langle \mathrm{LPIPS}(I_t^s \odot M^p, I_t^e \odot M^p) \big\rangle, \quad s_{\text{Contact-PS}} = \big\langle 1 - \mathrm{LPIPS}(I_t^s \odot M^i, I_t^e \odot M^i) \big\rangle, \\ s_{\text{ROI-DINO}} = \Big\langle \tfrac{1}{2}\big(1 + \cos(\phi_2(I_t^e \odot M^q), \phi_2(I_t^* \odot M^q))\big) \Big\rangle, \quad s_{\text{Ref-CLIP}} = \Big\langle \tfrac{1}{2}(a_t + \ell_t) \Big\rangle. \tag{9}$$

Motion Alignment: MCFS and FlowSim. Both metrics compare the RAFT flow $\mathbf{u}^e$ of the edited video with the flow $\mathbf{u}^r$ of the intended motion, over pixels that are consistent in both videos. MCFS evaluates the action region M^a and FlowSim the camera region M^c, which prevents the score from being dominated by stationary pixels:

$$g_t(p) = 1 - \min\Big(1, \frac{\|\mathbf{u}_t^e(p) - \mathbf{u}_t^r(p)\|}{\|\mathbf{u}_t^e(p)\| + \|\mathbf{u}_t^r(p)\|}\Big), \qquad s_{\mathrm{MCFS}} = \langle g_t \rangle_{M^a}, \qquad s_{\mathrm{FlowSim}} = \langle g_t \rangle_{M^c}. \tag{10}$$

C Simulation Setting

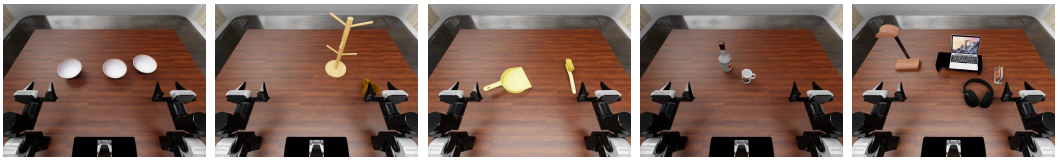
C.1 RoboDojo Tasks and Training Data

We reconstruct five RoboDojo (Chen et al., 2026) tasks covering stacking, hanging, two-object pickup, pouring, and articulated manipulation. Tab. 8 lists their instructions and success conditions, and Fig. 5 shows the first head-view frame of a representative episode for each task. Success is determined by the task signal of the environment at the end of a rollout. Each episode provides a head camera and two wrist cameras together with calibrated intrinsics, the static head pose, the moving wrist-camera trajectories, the joint sequence, and the gripper opening, and all augmentations operate on this synchronized record.

For each task, we collect 50 demonstrations, giving 250 demonstrations in total for the Original training set. Following the editing operators in Sec. 3, we group augmentation into four categories: *Object*, *Trajectory*, *Scene*, and *Viewpoint*. Object augmentation varies task-compatible object geometry and appearance; trajectory augmentation introduces alternative feasible motions and recovery behaviors; scene augmentation covers background, tabletop material, distractor, and lighting variations; and viewpoint augmentation changes the camera pose or motion while preserving the

Table 8: **RoboDojo task instructions and success conditions.** Each task is paired with a single language instruction shared by all training sets. Success is determined by the environment at the end of each rollout according to the listed condition.

Task	Instruction	Success condition
Stack bowls	Stack the three bowls together.	The three bowls form the intended stable stack.
Hang mug	Put the mug handle on the lowest rack hook and release it.	The handle is supported by the selected hook after gripper release.
Pick up tools	Pick up the brush with the right arm and the dustpan with the left arm.	Both tools are held above the table.
Pour liquid	Pour liquid from the bottle into the cup.	The transfer action reaches the task's terminal pouring condition.
Close and lift laptop	Close the laptop and then lift it.	Lid closure precedes the lift, and the lift reaches the task terminal state.



(a) Stack bowls (b) Hang mug (c) Pick up tools (d) Pour liquid (e) Close and lift laptop

Figure 5: **RoboDojo tasks.** First head-view frame of a representative source episode for each of the five reconstructed tasks, which cover stacking, hanging, two-object pickup, pouring, and articulated manipulation. Task configurations and instructions are adapted from RoboDojo.

underlying interaction. For both MiniMax H3 and WorldZoo, we generate 100 additional samples per category for each task from the same source demonstrations. This gives 400 augmented samples per task and 2,000 augmented samples in total, so each augmented training set contains 2,250 demonstrations. WorldZoo reuses the current actions for scene and viewpoint edits, while object and trajectory edits update the robot motion and action annotations together with the observations. MiniMax H3 outputs are temporally aligned to the source videos and paired with the source actions.

C.2 RoboDojo Evaluation Conditions

Tab. 9 summarizes the evaluation conditions. The Standard condition uses the reconstructed task without test-time variation, and each of the eight remaining conditions varies one factor while the others are held fixed. To separate evaluation from the augmentation distribution, test-time scene configurations, including room appearance, object instances, and clutter layouts, are different from those used for training augmentation, while camera positions, motions, and occlusion locations are sampled independently. Four conditions vary the scene or interaction: Lighting changes the illumination and room appearance; Background replaces the room together with compatible floor and tabletop appearances; Object replaces the task object with a feasible alternative of a different category, placement, material, or scale; and Distractor adds task-irrelevant clutter outside the required interaction region. The other four conditions vary the head-camera observation through target-centered, free, or dynamic viewpoint changes, with translations and rotations extending up to 26 cm and 15°, or through partial occlusion placed at random image locations. The wrist cameras remain rigidly attached to the arms in all conditions.

C.3 DexJoco Tasks and Embodiment Transfer

DexJoco (Wang et al., 2026b) evaluates manipulation with a robot arm and a dexterous hand on six tasks: Water Plant, Hammer Nail, Click Mouse, Pick Bucket, Pinch Tongs, and Fold Glasses. They require distinct finger roles, including holding a container while pouring, using a tool, a precise single-finger press, carrying with a stable grasp, pinching a narrow object, and coordinated manipulation of an articulated frame. In these tasks, a target hand can reach the source fingertip positions while bending its intermediate links into an unsuitable grasp. The source embodiment is a Franka

Table 9: **RoboDojo evaluation conditions.** Standard uses the reconstructed task without variation, and each of the other eight conditions changes a single factor while keeping the rest fixed. Scene and interaction conditions alter the environment or objects, whereas viewpoint conditions change only the head-camera observation.

Category	Condition	Description
Reference	Standard	Reconstructed task setting without additional variation.
Scene / Interaction	Lighting	Change illumination and compatible room color while retaining the task geometry.
	Background	Replace the room shell and compatible floor or tabletop appearance.
	Object	Vary task-object category, scale, placement, or material while keeping the objective feasible.
	Distractor	Add or rearrange task-irrelevant clutter outside required contacts.
Viewpoint	Target-centered	Translate the head camera and look toward the manipulation region.
	Free View	Translate the head camera without a forced look-at orientation.
	Dynamic View	Move the head camera on a returning path during the rollout.
	Occlusion	Apply a fixed soft-edged partial occluder at a head-view boundary.

arm with an Allegro hand, and the five target embodiments are Franka–ShadowHand, Franka–Wuji-2, Rizon4–Allegro, UR10e–Allegro, and Rizon4–ShadowHand, covering a changed hand, a changed arm, and a change of both. The fingertip-retargeting baseline, used on Rizon4 with ShadowHand, follows EgoEngine (Liu et al., 2026) and solves inverse kinematics to place the target fingertips at the source fingertip positions while constraining the finger orientations, as described in Sec. 4.2. All other editing stages, the number of generated demonstrations, the policy architecture, and the rollout initialization are the same as for WorldZoo. Each target embodiment uses its own URDF joint limits, meshes, collision geometry, and calibration. Each task has 50 source demonstrations. WorldZoo generates 50 demonstrations for every target embodiment, and fingertip retargeting additionally generates 50 for Rizon4 with ShadowHand. The source-embodiment policy is trained on the source demonstrations, and each target policy is trained only on the 50 demonstrations generated for that embodiment and evaluated on it with 50 rollouts per task.

C.4 Fixed-Budget Augmentation Composition

The fixed-budget study in Tab. 4 uses the Hang mug task with a budget of 400 augmented samples, under the same data, training, and evaluation setting as the RoboDojo experiments above. It varies the proportions of scene, object, viewpoint, and trajectory edits while keeping the source demonstrations, the number of augmented samples, the policy initialization, and the training schedule fixed. Uniform allocates 25% of the budget to each category; each heavy setting allocates 50% to one category and about 16.7% to each of the others, and each weak setting removes one category and divides the budget equally among the remaining three. Performance is the mean success rate over the eight generalization conditions, excluding Standard.

D Real Robot Experiment Setting

D.1 Setup and Data Collection

The physical experiments use a dual-arm I2RT YAM robot. Each arm has six joints and a two-finger parallel gripper, and the same robot model is used for rendering in visual editing. The policy observes the shared workspace through a third-person camera and two wrist cameras, all Intel RealSense D405; the third-person camera is the one whose viewpoint changes across the difficulty levels, while each wrist camera moves with its arm under a fixed mounting transform. Camera intrinsics and camera-to-base extrinsics are calibrated before data collection. Demonstrations are collected by leader–follower teleoperation at a control frequency of 30 Hz, and each of the three tasks has 100 demonstrations containing synchronized camera views, arm and gripper states, actions, and the task instruction. Fig. 6 shows the setup beside the initial and successful states of the three tasks under the Standard setting.

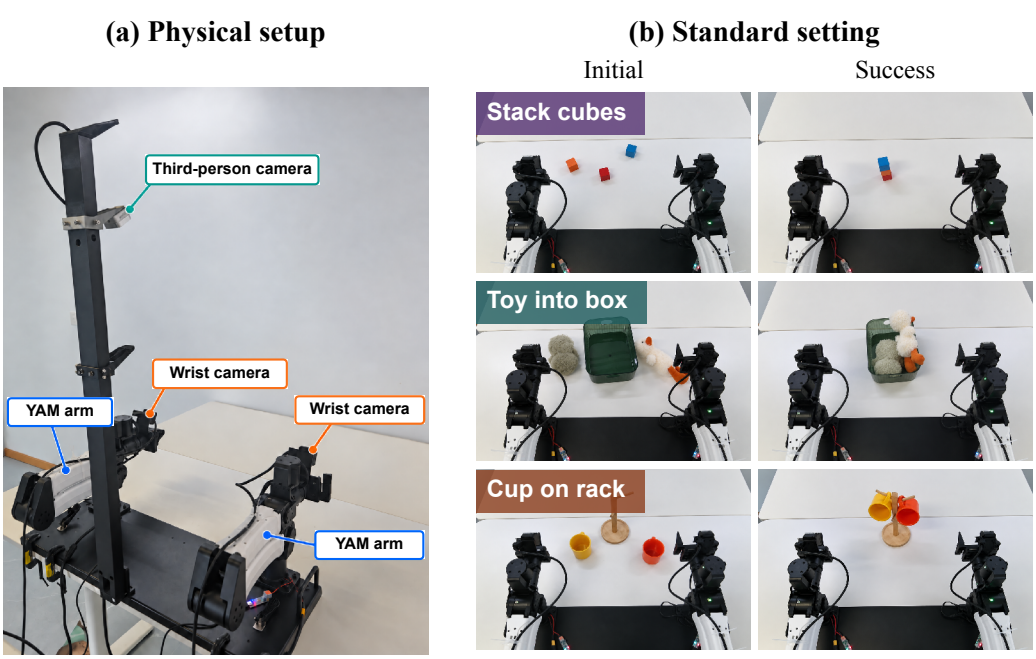


Figure 6: **Real-world setup and Standard task states.** (a) The dual-arm I2RT YAM robot with its third-person and two wrist cameras. (b) Representative initial (left) and successful terminal (right) states for Stack cubes, Toy into box, and Cup on rack.

D.2 Tasks

We evaluate three tasks. In *Stack cubes*, the robot grasps one cube, places it on the target cube, and releases it; a trial succeeds when the upper cube remains supported by the lower cube after release. In *Toy into box*, the robot grasps the toy, transports it above the open box, lowers it into the box, and releases it; a trial succeeds when the toy remains inside the box. In *Cup on rack*, the robot grasps the cup, moves it to the designated rack support, and releases it; a trial succeeds when the cup remains on the rack without gripper support. The same criterion is applied to every policy and condition. Fig. 6 shows Standard task states; Fig. 7 shows the task scenes at Levels 1–3.

D.3 Evaluation Protocol and Difficulty Levels

The Standard setting keeps the appearance and camera configuration of the demonstrations and varies only the initial object placements. Each task is evaluated on 50 predetermined object configurations under each condition; the configurations are sampled once before any evaluation and reused for Original, Uniform, and UniAug, so that all training sets face the same trials. The success rate is the fraction of the 50 configurations completed, and the Avg. column of Tab. 5 is the unweighted mean of Standard and Levels 1–3.

Levels 1–3 progressively broaden the visual and object distribution, as listed in Tab. 10. At each level, changes in lighting, third-person viewpoint, tabletop, distractors, and object identity are applied together within a rollout. Level 1 keeps the third-person camera unchanged and applies mild appearance changes, while Levels 2 and 3 add camera shifts and stronger changes to the remaining factors. The wrist cameras remain fixed on the arms at all levels.

D.4 Training Sets for the Real-World Comparison

Original uses the 100 teleoperated demonstrations of each task. Single-axis adds single-dimension edits from the background, lighting, viewpoint, object, and trajectory categories, dividing the augmentation budget evenly among them, so that each sample changes only one dimension while all other factors keep their source settings. UniAug adds samples in which several dimensions are com-

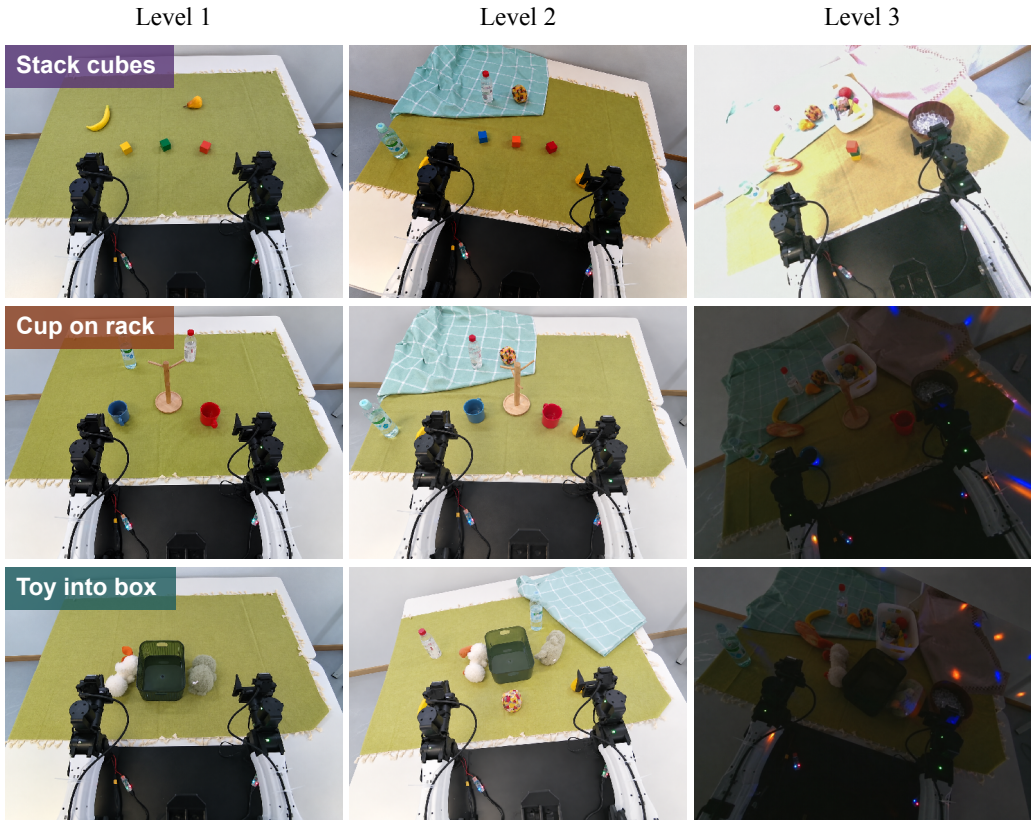


Figure 7: **Real-world tasks under three shifted evaluation levels.** Rows show Stack cubes, Cup on rack, and Toy into box; columns show Levels 1–3. The levels jointly vary lighting, the third-person viewpoint, the tabletop, distractors, and object identity as specified in Tab. 10. Standard initial and successful states are shown in Fig. 6.

Table 10: **Real-world difficulty levels.** All listed factors are applied together within a rollout, and each level broadens the variation relative to the previous one. Level 1 keeps the third-person viewpoint unchanged, and viewpoint shifts apply only to the third-person camera.

Factor	**Level 1**	**Level 2**	**Level 3**
Lighting	Brightness change	Large color-temperature and brightness change	Colored light sources
Viewpoint	Unchanged	Small change	Large change
Table surface	Slight tabletop change	Large tabletop change	Fully replaced tabletop
Distractor	2–4 objects	4–8 objects	9+ objects
Object	Few objects replaced	Most objects replaced	All objects replaced

posed within one edited demonstration according to the sampled edit chain. All three training sets share the same source demonstrations and the same $\pi_{0.5}$ fine-tuning protocol.

D.5 From Egocentric Video to Real-Robot Execution

The experiment in Sec. 4.4 trains a *Toy into box* policy from egocentric human videos alone. The human videos are recorded with an Intel RealSense D455 camera, and the Ego2Robot module (Sec. A.2) converts each video into a YAM demonstration through hand reconstruction, contact-aware retargeting, removal of the human hand, and robot rendering. The comparison trains on the same human videos either as Ego Video, which keeps the original frames showing the human hand, or converted by WorldZoo; both use the same retargeted action labels (Sec. E). Both $\pi_{0.5}$ and

LingBot-VLA are evaluated on the physical YAM robot with 50 rollouts per policy and training condition.

E Policy Training Details

E.1 Training Setup

All policies are fine-tuned on four NVIDIA B300 GPUs for 30,000 steps with a batch size of 256, a learning rate of 2.5×10^{-5}, and the AdamW optimizer, starting from their pretrained checkpoints. Within each comparison, the initialization, observation views, language instructions, action representation, optimizer settings, number of training steps, and batch size are fixed, so the training data is the only difference.

Each policy and training-set combination in the main experiments is fine-tuned separately. On RoboDojo, $\pi_{0.5}$ (Intelligence et al., 2025) and LingBot-VLA (Wu et al., 2026) are each fine-tuned on the Original, MiniMax H3, and WorldZoo training sets. In the DexJoco study, $\pi_{0.5}$ is fine-tuned on the source demonstrations for the source embodiment and on WorldZoo-generated data for each target embodiment, with an additional run on fingertip-retargeted data for Rizon4 with ShadowHand. For the three physical tasks, $\pi_{0.5}$ is fine-tuned on the Original, Uniform, and UniAug training sets. In the egocentric study, $\pi_{0.5}$ and LingBot-VLA are each fine-tuned on two versions of the same human videos, Ego Video and WorldZoo. In both versions, the action labels are the robot joint trajectories obtained from our hand detection, MANO reconstruction, and retargeting pipeline (Sec. A.2).

E.2 Inputs, Actions, and Deployment

Each training example contains the task instruction, the camera views, the robot state, and future action targets aligned with the observations. Embodiment, object, and trajectory variants use their edited action trajectories, scene variants reuse the source actions at the same frame indices, and variants that change the clip length use the source-index map to pair action targets with video frames. RoboDojo and YAM policies use three camera views, a head or third-person camera and two wrist cameras, while DexJoco policies use a single third-person camera; all images are 640×480. Actions are absolute joint positions, and when the target embodiment changes the joint dimension, the action normalization is computed for that embodiment. Appearance edits and object edits keep the task instruction, since the replacement object preserves the task semantics, and trajectory edits that reach the same goal along a different route also keep it, with the action sequence given by the resolved joint trajectory.

At test time, the policy receives only its camera views, the task instruction, and the robot state; object poses, masks, and other quantities used during augmentation are not provided. We evaluate the final checkpoint of each run. The policy predicts a chunk of 60 actions and executes the first 30 before predicting the next chunk. In simulation, success is computed by the benchmark environment, and on the physical YAM robot, policy outputs are executed through the platform controller.