ROBOEDIT: TURNING HUMAN MANIPULATION VIDEOS INTO SCALABLE ROBOT EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Collecting robot hand-object interaction data is costly and embodiment-specific, yet abundant human-object videos remain unusable for robot training. We present *RoboEdit*, a human-to-robot video editing suite that transforms human manipulation videos into action-consistent, physically plausible robot videos with aligned 3D hand states. To enable scalable supervision, we introduce *RoboEdit-ADC*, an automatic pipeline that reconstructs and retargets 3D interactions from RGB videos across embodiments. This pipeline generates *RoboEdit-14M*, a large-scale dataset of 174K aligned video pairs (14M frames) spanning seven robot embodiments, diverse scenes, and interaction types. The core editing engine, *RoboEdit-Trans*, employs cross-embodiment adaptation modules to preserve temporal coherence while adapting appearance and motion. It further integrates a 3D Robot-State Decoder to recover per-frame hand states for structured motion supervision. Experiments show that RoboEdit achieves state-of-the-art editing quality and supports downstream robot control policies in real-world manipulation tasks. Ultimately, the RoboEdit suite unlocks the vast potential of unlabeled human videos, providing scalable, high-fidelity visual and 3D motion supervision for generalizable robot learning.

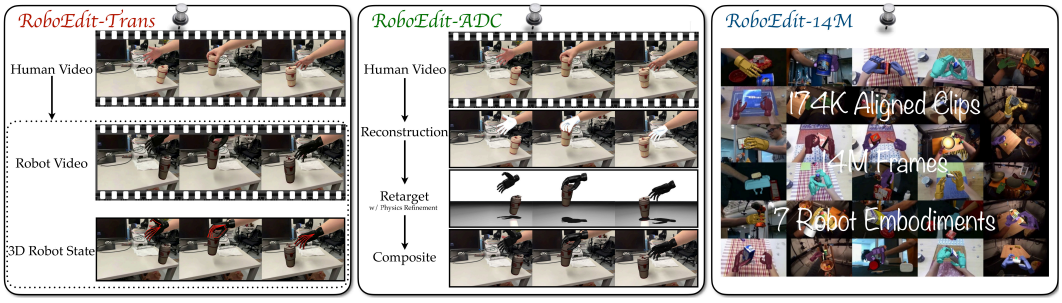

Figure 1: RoboEdit Overview: The framework takes an RGB human video and a target robot, to generate a physically plausible robot interaction video and 3D robot-hand states (RoboEdit-Trans). These states enable motion supervision for downstream robot control, while the RoboEdit-ADC pipeline generates the large-scale paired dataset (RoboEdit-14M) required for training.

1 INTRODUCTION

Video provides rich supervision for robot manipulation by capturing contact, object motion, viewpoint, scene context, and embodiment appearance—details often lost in compact state vectors (Black et al., 2026). However, collecting robot interaction videos is expensive and strictly embodiment-specific: variations in morphology, kinematics, and control prevent direct data transfer across different robot hands (Zhang et al., 2026). Consequently, existing datasets cover only a fraction of the objects, scenes, viewpoints, and behaviors needed for robust, generalizable robot learning.

Human interaction videos offer a scalable alternative, rich with diverse scenes, viewpoints, object motions, and contact-rich behaviors (Chao et al., 2021; Engel et al., 2023; Liu et al., 2024). Yet embodiment mismatches prevent their direct use as robot training data. While prior work has successfully extracted representations or action supervision from human videos (Nair et al., 2022; Wang et al., 2023), directly translating these videos into physically plausible target-robot interaction

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.

videos remains challenging, particularly in preserving scene context, interaction dynamics, and temporal consistency across robot embodiments (Lepert et al., 2025).

This raises a central question: *Can we faithfully transform abundant human videos into scalable, high-fidelity robot interaction videos?* We answer this question through human-to-robot manipulation video editing, which retargets the interaction to a target robot while preserving the original scene context and temporal dynamics.

We introduce *RoboEdit* (Fig. 1), a suite designed to transform human manipulation videos into scalable, high-fidelity robot training data. It comprises three core components: *RoboEdit-Trans*, a robot-conditioned video editor; *RoboEdit-ADC*, an automatic paired-data curation pipeline; and *RoboEdit-14M*, the resulting large-scale dataset of paired supervision.

Generating entire robot interactions de novo can disrupt the observed scene context and produce physically implausible motion. RoboEdit-Trans (Fig. 3) avoids these pitfalls by editing the RGB human interaction video directly, preserving the source scene, camera motion, and object dynamics while retargeting the human embodiment to the target robot. We introduce cross-embodiment adaptation modules that jointly model cross-embodiment appearance, kinematics, and contact dynamics across diverse robots, ensuring temporal coherence in the generated target-robot video. Furthermore, we integrate a 3D Robot-State Decoder to recover per-frame 3D hand states to serve as structured supervision essential for downstream learning and control.

To provide scalable, aligned supervision, we introduce RoboEdit-ADC (Fig. 2), an automatic pipeline that constructs paired human/robot videos from RGB footage. It first reconstructs the 3D hand-object interaction and camera motion, then retargets the action to a target robot while preserving the original object dynamics and viewpoint. To ensure physical consistency, we incorporate depth regularization and physics-based refinement, effectively reducing artifacts such as object penetration, floating contacts, and temporal jitter. The pipeline concludes by inpainting the human-object regions and compositing the rendered robot interaction into the source scene (Ci et al., 2025; Zi et al., 2025). This process yields RoboEdit-14M (Fig. 4), a massive paired dataset with over 174K aligned video clips (over 14M frames) across 7 robot embodiments, diverse real and synthetic scenes, and varied camera perspectives.

Benchmark comparisons show that RoboEdit-Trans achieves state-of-the-art performance in human-to-robot video editing, outperforming strong baselines in reconstruction fidelity, local editing accuracy, background preservation, and perceptual video quality (He et al., 2025). Ablation studies confirm consistent gains from each component, while real-world deployments validate the utility of the recovered 3D hand states for downstream control. Collectively, these results demonstrate that human-to-robot video editing transforms abundant, unlabeled human manipulation videos into scalable, high-fidelity robot experience, providing both visual and structured trajectory supervision for robot learning.

Our key contributions are as follows:

- We introduce RoboEdit, an end-to-end human-to-robot video editing suite that transforms RGB human manipulation videos into cross-embodiment robot interaction videos and corresponding structured 3D hand states.
- We develop RoboEdit-Trans, a novel video editor featuring robot-aware adaptation modules for cross-embodiment synthesis and a 3D Robot-State Decoder for 3D robot kinematic recovery.
- We design RoboEdit-ADC, an automatic data curation pipeline that constructs RoboEdit-14M, a large-scale dataset containing 174K aligned clips (14M frames) across 7 robot embodiments, diverse scenes, and interaction types.
- We demonstrate state-of-the-art video editing performance through rigorous benchmarks and ablations, validating the utility of our recovered 3D states for downstream control in real-robot experiments.

2 Related Work

Robot Learning from Human Videos. Human manipulation videos provide a scalable source of interaction data (Grauman et al., 2022; Goyal et al., 2017). Prior work leverages this abundance to pretrain transferable visual representations (Nair et al., 2022; Ma et al., 2023a), learn value

functions for imitation and reinforcement learning (Ma et al., 2023b; Zakka et al., 2021), or extract structured interaction priors such as affordances and contact regions (Bahl et al., 2023; Shan et al., 2020). Closer to our work, recent methods treat human hands as action interfaces or reconstruct and retarget hand-object motion into robot trajectories (Shaw et al., 2022; Mandikal & Grauman, 2022; Sivakumar et al., 2022). However, these approaches typically convert human videos into intermediate representations or sparse motion supervision. By contrast, RoboEdit directly generates physically-plausible target-robot videos with corresponding 3D hand states, providing rich visual and kinematic supervision in a single framework.

Human-to-Robot Retargeting. Human-to-robot retargeting adapts human demonstrations to robot embodiments with distinct morphology and kinematics. Classical approaches solve geometric, task-space, or joint-space objectives to align human wrist, fingertip, grasp, or object-relative motion, forming the basis of teleoperation and data collection (Handa et al., 2019; Lakshmipathy et al., 2024). Alternatively, learning-based methods train cross-embodiment policies for direct imitation (Arunachalam et al., 2022; Qin et al., 2022; Shaw et al., 2022). For contact-rich manipulation tasks, retargeting further incorporates object dynamics, contact constraints, and physical feasibility through optimization or reinforcement learning (Xin et al., 2025; Pan et al., 2025). Recent monocular pipelines combine 3D hand-object reconstruction with retargeting to derive robot trajectories or policy supervision directly from human videos (Paliwal et al., 2026). However, these methods typically output sparse trajectories, whereas RoboEdit generates full interaction videos alongside dense 3D states.

Human-to-Robot Video Editing. Video editing alters the semantic content of a video while preserving temporal coherence and visual consistency. Diffusion-based approaches have evolved from latent inversion and optimization (Mokady et al., 2023; Qi et al., 2023) to training-free or feed-forward feature manipulation (Geyer et al., 2024; Ku et al., 2024), and increasingly, to multimodal control through references, poses, masks, or trajectories (Liang et al., 2025). Despite the rapid advancement of video diffusion models (Wan et al., 2025), most editing methods assume a fixed agent morphology and motion capabilities. Recent work extends video generation to robotics, synthesizing robot demonstrations from human videos for imitation learning and data generation (Yang et al., 2025; Song et al., 2025). However, human-to-robot video translation presents a unique challenge: it must transform appearance, kinematic structure, and motion dynamics across embodiments while preserving interaction semantics. Recent methods attempt to tackle this using video foundation models and embodiment-aware conditioning (Ci et al., 2025; Xie et al., 2025). Yet significant differences in morphology, kinematics, and physical interaction constraints still limit the accuracy of current cross-embodiment editing.

3 THE ROBOEDIT SUITE

Problem Formulation. Given an RGB human hand-object interaction video $v^h_{1:T}$ (where T is the number of frames and h denotes the human domain) and a target robot embodiment e, RoboEdit aims to synthesize a corresponding robot interaction video $\hat{v}^{r,e}_{1:T}$ in the same scene (r denotes the robot domain) and predict the corresponding robot hand state trajectory $\hat{q}^e_{1:T}$.

Since aligned human/robot video pairs are scarce, we utilize RoboEdit-ADC to construct necessary paired supervision. From the source video, the pipeline reconstructs a 3D representation $\mathcal{Z}_{1:T} = (H_{1:T}, O_{1:T}, M_o, m_{1:T}, C_{1:T})$, where H and O are the human hand and object pose trajectories, M_o is the object mesh, m is the edit mask, and C represents the source camera states. The pipeline then retargets the human trajectory $H_{1:T}$ relative to the object $O_{1:T}$ into a robot-hand state trajectory $q^e_{1:T}$, inpaints the human hand and object regions, and renders the robot-object interaction under the original camera motion $C_{1:T}$ to produce the target video $v^{r,e}_{1:T}$.

Each curated sample is defined as

$$\mathcal{D}_i = (v^h_{1:T},\, m_{1:T},\, v^{r,e}_{1:T},\, q^e_{1:T}).$$

These samples constitute the RoboEdit-14M dataset (detailed in Section 3.3). Finally, RoboEdit-Trans learns a mapping $F_\theta(v^h_{1:T}, e) \to (\hat{v}^{r,e}_{1:T}, \hat{q}^e_{1:T})$, preserving the source scene, camera motion, and object dynamics while transforming the human interaction to the target embodiment e.

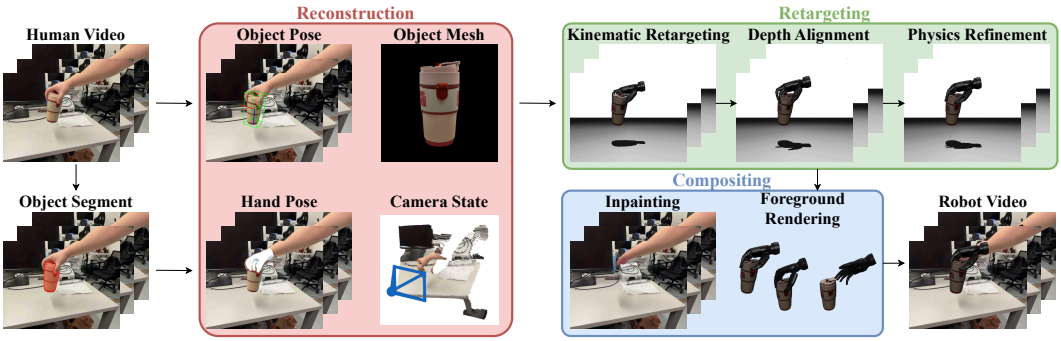

Figure 2: RoboEdit-ADC reconstructs 3D interaction from RGB video, performs depth/physics-refined retargeting, and composites the foreground into the inpainted scene.

3.1 RoboEdit-ADC: Automatic Data Curation

3D Interaction Reconstruction. As illustrated in Fig. 2, RoboEdit-ADC reconstructs the full 3D human-object interaction from a single RGB video. We estimate the articulated hand trajectory $H_{1:T}$ using HaMeR (Pavlakos et al., 2024) and segment the manipulated object using SAM 2 (Ravi et al., 2024). The edit masks $m_{1:T}$ are then derived by combining the object segmentation with the hand mask. For object geometry, we employ TRELLIS (Xiang et al., 2024) to reconstruct the mesh M_o and FoundationPose (Wen et al., 2024) to track its 6D pose trajectory $O_{1:T}$. Finally, VGGT (Wang et al., 2025) estimates the source camera state $C_{1:T}$, including intrinsics, extrinsics, and depth. To ensure physical consistency, we rectify the reconstruction via coordinate-frame and scale calibration before retargeting (see Appendix A.1).

Monocular HaMeR estimates often suffer from depth and scale ambiguities, such that even an image-aligned hand can be misplaced in 3D space, leading to missed contacts, object penetration, or instability after retargeting. To address this, we leverage the aligned depth to correct HaMeR's camera-space scale and depth prior to retargeting. We project the recovered palm joints into the depth map and back-project the depth observations into metric 3D palm anchors to estimate a scale factor. We then rescale the camera-space hand reconstruction about the camera center to align its palm depth with the depth observations, while keeping the relative hand articulation fixed. This correction improves the hand-object contact geometry for subsequent retargeting. Finally, the reconstruction stage outputs the unified hand-object-camera representation:

$$\mathcal{Z}_{1:T} = (H_{1:T}, O_{1:T}, M_o, m_{1:T}, C_{1:T}) .$$

Human-Robot Retargeting. Given $\mathcal{Z}_{1:T}$ and a target embodiment e, RoboEdit-ADC retargets the human hand motion using a strategy inspired by SPIDER (Pan et al., 2025). For anthropomorphic hands, we use SPIDER's wrist-and-fingertip IK in the MuJoCo (Todorov et al., 2012) simulator. For morphologically distinct two- and three-finger grippers, we apply specialized strategies: a two-finger gripper derives its jaw pose and opening from wrist and thumb-index geometry, while a three-finger gripper estimates palm pose and optimizes joint angles to match the thumb, index, and middle fingertips. This yields an initial kinematic trajectory $q^{e,\mathrm{IK}}_{1:T}$.

To mitigate penetration, floating contacts, and motion artifacts caused by reconstruction noise and morphological mismatches, we refine this trajectory via a physics-guided refinement: $q^{e}_{1:T} = \arg\min_{\{q^e_t \in \mathcal{Q}_e\}_{t=1}^T} \mathcal{L}_{\text{ret}}$, where $\mathcal{Q}_e$ denotes the set of joint-limit-feasible configurations for e and

$$\mathcal{L}_{\text{ret}} = \lambda_{\text{track}}\mathcal{L}_{\text{track}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}}.$$

The terms of this objective function are defined as follows:
Tracking Loss: $\mathcal{L}_{\text{track}} = \sum_{t=1}^{T} \|q^e_t - q^{e,\mathrm{IK}}_t\|_2^2$ ensures fidelity to the kinematic reference.
Temporal Loss: $\mathcal{L}_{\text{temp}} = \sum_{t=2}^{T} \|q^e_t - q^e_{t-1}\|_2^2$ smooths motion and reduces jitter.
Geometry Loss: $\mathcal{L}_{\text{geo}}$ is an L_2 penalty on robot-object penetration detected via MuJoCo collision checks and visual-mesh geometry (see Appendix A.2).
Contact Loss: $\mathcal{L}_{\text{contact}} = \sum_{t=1}^{T} \sum_{k \in \mathcal{C}_t} \|p^e_k(q^e_t) - a_{t,k}\|_2^2$ preserves valid contacts, where $\mathcal{C}_t$ is the set of fingertip indices whose reconstructed fingertip-to-surface distance is below a threshold consistently

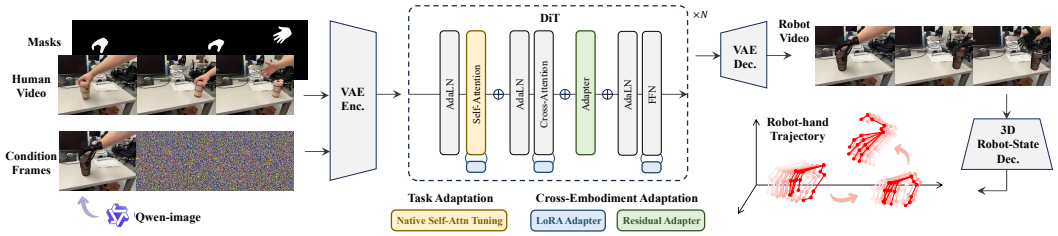

Figure 3: RoboEdit-Trans uses LoRA and residual adapter for editing and 3D Robot-State Decoder for state recovery.

across neighboring frames, $a_{t,k}$ is the closest 3D object-surface point for each $k \in \mathcal{C}_t$, and $p_k^e(q_t^e)$ is the 3D position of robot fingertip obtained from FK $p_k^e(\cdot)$.

The optimized trajectory $q_{1:T}^e$ is then used for rendering the target robot interaction.

Robot Interaction Compositing. Given the source video $v_{1:T}^h$, edit masks $m_{1:T}$, robot and object trajectories $q_{1:T}^e$ and $O_{1:T}$, and aligned camera states $C_{1:T}$, RoboEdit-ADC constructs the paired robot video in three steps: (1) MiniMax-Remover (Zi et al., 2025) inpaints the human hand and object regions within $m_{1:T}$, producing a clean background $b_{1:T}$; (2) we render the robot hand and object under C_t as foreground r_t; and (3) we composite r_t onto b_t to obtain the target frame $v_t^{r,e}$ and the full video $v_{1:T}^{r,e}$. This process preserves the source background and camera motion while seamlessly substituting the hand-object interaction with the target robot.

3.2 RoboEdit-Trans: Human-Robot Video Editing

Model Overview and Robot Conditioning. While RoboEdit-ADC produces physically plausible robot interaction videos, its multi-stage pipeline is computationally expensive. This motivates RoboEdit-Trans, which aims to generate robot videos more efficiently. RoboEdit-Trans extends the NovaEdit (Pan et al., 2026b) architecture to enable cross-embodiment robot editing and 3D robot-state prediction (Fig. 3). We introduce adaptation modules supporting diverse robot embodiments and a 3D Robot-State Decoder to recover the robot hand trajectory. The editor takes a masked human video, providing the scene, camera motion, and object dynamics, and sparse target-robot condition frames specifying the target embodiment and representative hand-object configurations. Let z^h and $z^{c,e}$ denote their respective latent representations. The editor predicts target-robot latent $\hat{z}^{r,e} = G_\theta(z^h, z^{c,e})$, where G_θ is the robot-conditioned editor (c denotes conditioning) optimized via flow matching.

Cross-Embodiment Adaptation. The shared editor learns a common human-to-robot task adaptation, but variations in robot appearance, morphology, kinematics, and contact patterns require cross-embodiment adaptation. We introduce two complementary modules: LoRA adapters (Hu et al., 2021) adapt spatiotemporal representations to diverse robot appearance and motion, while residual bottleneck adapters refine embodiment-specific hand geometry and interaction patterns. For input feature x and intermediate feature h, LoRA and residual-adapter mappings f_{LoRA} and f_{Ada} are:

$$f_{\text{LoRA}}(x) = W_0 x + \tfrac{\alpha}{r} BAx,$$
$$f_{\text{Ada}}(h) = h + W_2^{\text{Ada}} \phi\big(W_1^{\text{Ada}} \text{LayerNorm}(h)\big)\,.$$

Here, W_0 is the frozen pretrained weight, A and B are trainable rank-r matrices, and α scales the LoRA update. W_1^{Ada} and W_2^{Ada} are the trainable down- and up-projection matrices; LayerNorm denotes layer normalization, and ϕ is a nonlinear activation. Together, these modules adapt one shared editor across target embodiments while retaining its human-to-robot editing capability.

3D Robot-State Decoder. Edited videos do not explicitly encode metric 3D motion, so we leverage a 3D Robot-State Decoder to recover camera-space robot states. Our 3D Robot-State Decoder first estimates robot states in each frame and then refines them over the full sequence. For framewise spatial estimation, inspired by heatmap-based 2D hand pose estimation (Iqbal et al., 2018), we use a shared mask-aware image encoder (He et al., 2016) and design robot-specific prediction heads to estimate per-frame 2D palm anchors and fingertips, palm-frame fingertip coordinates, and joint configurations across diverse embodiments. A shared camera head predicts camera intrinsics, while

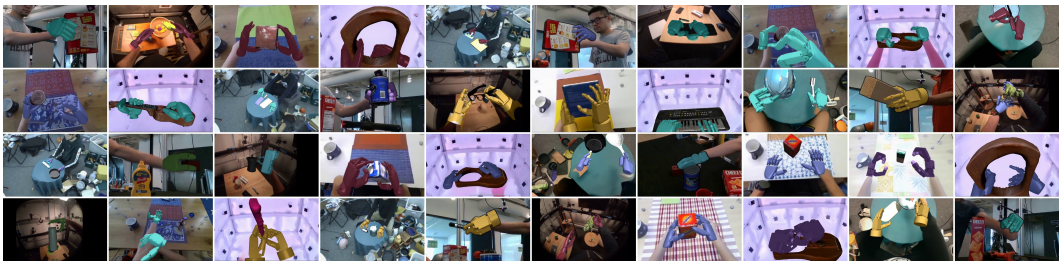

Figure 4: RoboEdit-14M spans diverse everyday manipulation tasks across a wide range of robot embodiments.

Table 1: Comparison of representative human-to-robot manipulation datasets. Frames follow each work's reported scale; Emb. denotes the number of robot embodiments; Auto denotes curation without per-sample human intervention.

Dataset	Frames	View	Camera	Scene	Emb.	RGB Pair	Robot State	Auto
H&R (Xie et al., 2025)	∼1.2M	Third	Static	Real	1	✓	✓	×
H2R (Li et al., 2026)	∼1M	Ego	Moving	Real	3	✓	×	✓
UniDex (Zhang et al., 2026)	9M	Ego	Moving	Real	8	×	✓	×
X-Humanoid (Yang et al., 2025)	2.8M	Third	Moving	Synth.	1	✓	×	×
RoboEdit-14M (Ours)	14.1M	Ego+Third	Both	Both	7	✓	✓	✓

the predicted 2D palm anchors are matched with embodiment-specific 3D palm geometry to recover the 3D wrist pose through Perspective-n-Point (PnP). A temporal Transformer then jointly refines the framewise states over the full sequence to enforce temporal consistency and reduce frame-to-frame jitter, after which forward kinematics produces the complete camera-space robot-hand trajectory. The decoded states provide structured motion guidance for downstream robot learning and control.

We supervise per-frame 2D localization, 3D geometry, articulation, and camera estimation together with temporal state consistency. More details are in Appendix B.

3.3 RoboEdit-14M: Paired Human-Robot Dataset

Dataset Overview. As shown in Fig. 4, RoboEdit-14M contains 174,547 aligned human/robot video pairs totaling over 14.1M paired frames, constructed from 24,197 human-interaction clips. Each pair includes a target robot video and its retargeted 3D hand trajectory, providing large-scale visual and robot-state supervision. Table 1 shows that RoboEdit-14M uniquely combines automatic curation, paired RGB video, and robot-state labels at 14.1M-frame scale. More details are in Appendix C.

Source Diversity. We source human videos from DexYCB (Chao et al., 2021), HOT3D (Banerjee et al., 2024), H2O (Kwon et al., 2021), GigaHands (Fu et al., 2024), and TACO (Liu et al., 2024), covering static and moving egocentric views, single- and bimanual manipulation, and diverse objects and tools. RoboEdit-ADC converts them into 145,459 real-scene pairs spanning 7 embodiments: Inspire, XHand, Ability, SCHUNK SVH, Allegro, Unitree Dex3, and the Franka Panda gripper.

Synthetic Scene Augmentation. To expand visual diversity, we construct 29,088 synthetic pairs by rendering aligned human and retargeted scene under varied cameras and lighting. We composite both foregrounds onto the same background generated by RoboEngine (Yuan et al., 2025), preserving source–target alignment while diversifying scene context.

4 Experiments

4.1 Experimental Setup

Training and Inference Details. RoboEdit-Trans builds on NovaEdit's Wan2.1-VACE-1.3B backbone (Wan et al., 2025; Jiang et al., 2025). We fine-tuned the backbone on paired human/robot clips for task adaptation, then froze it and trained LoRA and residual adapters jointly for cross-embodiment adaptation. Following NovaEdit, we used 81-frame clips with sparse target-robot keyframes at indices $\{0, 10, \ldots, 80\}$. At inference, we used Qwen-Image-Edit (Wu et al., 2025), fine-tuned on RoboEdit-14M, to generate these keyframes; RoboEdit-Trans then used them to produce the edited video.



Figure 5: Qualitative comparison across four human-object interactions and four robot embodiments. The left two columns show RoboEdit-ADC results, while the remaining columns compare RoboEdit-Trans with baseline video editing models.

We trained the 3D Robot-State Decoder in two stages: the framewise spatial estimator on rendered and composited robot frames, followed by the temporal Transformer on full 81-frame sequences with the spatial estimator frozen. At inference, it recovered the camera-space robot-hand trajectory from the generated video. Appendix D provides the details.

Benchmark and Metrics. We evaluated RoboEdit-Trans on a fixed benchmark of 300 cases spanning diverse source domains, cameras, interactions, and embodiments. See Appendix H for benchmark cases distribution. To evaluate generalization to unseen human demonstrations, we split the dataset by source human video, keeping all robot videos curated from the same human video across different embodiments in the same split. We report SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018) against the paired reference, edit-region LPIPS against the same reference, and background SSIM against the source outside the edit mask. Additionally, we report VBench (Huang et al., 2024) measuring visual quality and temporal consistency, and OpenVE-Bench local-replacement protocol (He et al., 2025) assessing prompt compliance, visual naturalness, temporal stability, and physical and motion integrity.

Baselines and Protocols. We compared RoboEdit-Trans with VACE (Jiang et al., 2025), UniVideo (Wei et al., 2026), VINO (Chen et al., 2026a), Kiwi-Edit (Lin et al., 2026), OmniWeaving (Pan et al., 2026a), EditCtrl (Litman et al., 2026), AnyV2V (Ku et al., 2024), and ReCo (Zhang et al., 2025). Following each method's supported interface, single-reference methods received one robot frame. In

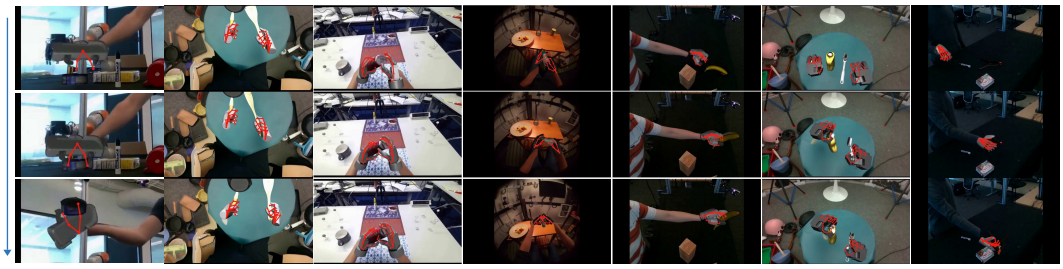

Figure 6: 3D robot-state predictions on RoboEdit-Trans edited videos across seven embodiments.

Table 2: Quantitative results on RoboEdit-Trans benchmark. AQ/DD/MS denote VBench Aesthetic Quality/Dynamic Degree/Motion Smoothness; OpenVE is the overall OpenVE-Bench score. S/M/T: single reference / multi-keyframe / text instruction. (Best results: bold. Second-best: underlined.)

Method	Params	Cond.	Recon.		Local edit		VBench			OpenVE
			SSIM ↑	LPIPS ↓	Edit LPIPS ↓	BG SSIM ↑	AQ ↑	DD ↑	MS ↑	Overall ↑
VACE	1.3B	S	0.8764	0.1070	0.0497	**0.9487**	0.4673	0.4867	0.9952	3.2270
UniVideo	14B	S	0.3249	0.6515	0.1043	0.4026	0.3871	0.4133	0.9890	2.2455
VINO	13B	S	0.5564	0.3348	0.0679	0.6041	0.4488	0.4433	**0.9967**	**3.2711**
Kiwi-Edit	5B	S	0.7415	0.2059	0.0627	0.8137	0.4307	0.4600	0.9950	3.2386
OmniWeaving	13B	S	0.8107	0.1590	0.0625	0.8855	0.4478	**0.6900**	0.9955	3.1634
EditCtrl	1.3B	T	0.5452	0.4295	0.0809	0.6154	0.4319	0.4933	0.9951	3.0319
AnyV2V	1.3B	S	0.5060	0.4837	0.0978	0.5753	0.3891	0.1567	0.9756	2.9023
ReCo	1.3B	S	0.8433	0.1366	0.0546	0.8869	0.4827	0.4967	0.9939	3.1255
VACE	1.3B	M	0.8996	0.0778	0.0258	0.9419	0.4768	0.5667	0.9923	3.1446
UniVideo	14B	M	0.3364	0.6464	0.1032	0.4125	0.3836	0.4233	0.9900	2.2089
VINO	13B	M	0.5539	0.3380	0.0541	0.5897	0.4396	0.2733	0.9944	3.1447
Ours	1.3B	M	**0.9282**	**0.0470**	**0.0171**	0.9188	**0.4832**	0.6300	0.9956	3.2511

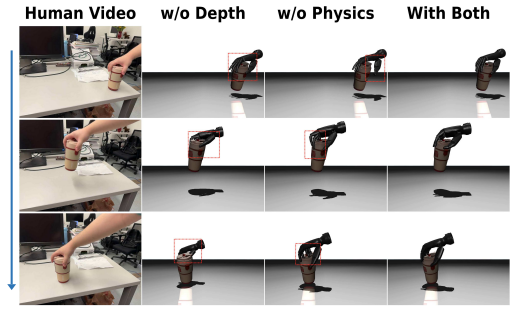

Figure 7: Qualitative ablation of RoboEdit-ADC retargeting. Depth regularization improves hand-object alignment, while physics refinement reduces floating and penetration.

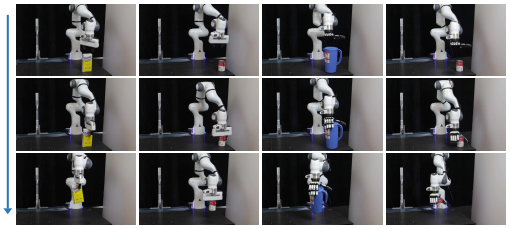

Figure 8: Real-robot deployment with 3D Robot-State Decoder trajectories across four YCB-object tasks.

the multi-reference setting, RoboEdit-Trans, VACE, UniVideo, and VINO used the same number of keyframes as reference, sampled at indices $\{0, 10, \ldots, 80\}$ for fair comparison.

4.2 Qualitative Results

Fig. 5 and Fig. 6 show cases from distinct sources and robot hands, using only midpoint non-keyframes such as 15 and 25 to best show editing and 3D decoding ability. Fig. 5 compares all methods at matched timestamps and crops. RoboEdit-Trans consistently synthesized the target embodiment while preserving the scene and object motion, whereas the baselines often retained human hands or generated inconsistent hand geometry. Fig. 6 shows that the 3D Robot-State Decoder tracks the generated robot hands and recovered their corresponding 3D motion. More results are in Appendix F.

Table 3: Ablation of adaptation modules on 300 cases using identical sparse-keyframe conditioning.

LoRA	Adapter	SSIM ↑	LPIPS ↓	Edit LPIPS ↓	BG SSIM ↑
×	×	0.9255	0.0494	0.0179	0.9174
✓	×	0.9264	0.0490	0.0176	0.9181
×	✓	0.9276	0.0478	0.0173	0.9186
✓	✓	**0.9282**	**0.0470**	**0.0171**	**0.9188**

Table 4: Decoder accuracy and ablations (↓). Shared: shared prediction heads; Temp.: temporal Transformer.

Metric	Shared	w/o Temp.	Full
Wrist (mm)	43.15	47.21	**42.03**
Rotation (°)	25.27	33.23	**23.28**
Fingertip (mm)	46.27	55.85	**43.26**
Joint (norm.)	0.2533	0.2402	**0.2260**

4.3 Quantitative Comparison

Table 2 shows that RoboEdit-Trans achieves SOTA performance, with stronger reconstruction and local-editing while remaining competitive in visual quality and motion consistency. Its slightly lower BG SSIM reflects the differing spatial extent of robot and human hands, which causes valid changes near the edit-mask boundary to count as background errors.

Table 4 reports 3D Robot-State Decoder's wrist and fingertip errors in camera space, together with wrist-rotation and normalized joint errors, evaluated on the same 300 cases. Our decoder achieves robot hand estimation errors comparable to those reported for camera-space human hand pose estimation (Ren et al., 2026; Valassakis & Garcia-Hernando, 2024).

We also measured runtime for RoboEdit-ADC curation and RoboEdit-Trans inference on a same NVIDIA H100 GPU: RoboEdit-ADC takes approximately 20 minutes across its stages, while RoboEdit-Trans takes approximately 8 minutes including reference keyframes generation, highlighting the efficiency of learned RoboEdit-Trans. A detailed runtime breakdown is in the Appendix E.

4.4 Ablation Study

Video Editing Components. Table 3 reports the contributions of the two cross-embodiment adaptation modules. Both LoRA and the residual adapter improve all metrics over the backbone-only variant; the residual adapter provides the larger individual gain, while combining both performs best.

3D Robot-State Decoder. The variants in Table 4 replace robot-specific prediction heads with shared heads (Shared) or remove temporal Transformer refinement (w/o Temp.), isolating embodiment-specific prediction and temporal context. Robot-specific heads improve joint estimation, while temporal refinement particularly benefits wrist orientation and fingertip localization.

Retargeting Quality. Fig. 7 presents the two refinements in RoboEdit-ADC. Without depth regularization, monocular depth and scale errors propagate to retargeting and displace the robot hand in the scene. Without physics refinement, kinematic matching produces floating or penetrating fingers.

4.5 Real-Robot Deployment

To evaluate our 3D Robot-State Decoder for downstream control, we trained a residual PPO controller (Schulman et al., 2017) in Genesis (Genesis Authors, 2024) to track robot-hand trajectories decoded from edited videos. Across 512 randomized simulation environments, it achieves trajectory-reproduction success rates of 71% with the Panda gripper and 62% with XHand.

We further deploy the controller on a 7-DoF Franka Panda with a parallel-jaw gripper or 12-DoF XHand for four YCB-object tasks (Fig. 8). Under RoboDojo's evaluation protocol (Chen et al., 2026b), each task–end-effector setting achieves at least 50% success over 10 trials. Appendix G details the evaluation protocol and controller.

5 Conclusions

We have introduced RoboEdit, a comprehensive suite that transforms human manipulation videos into physically plausible robot interaction videos and structured 3D hand states. RoboEdit-ADC automatically curates the RoboEdit-14M dataset to provide scalable paired supervision, while RoboEdit-Trans performs cross-embodiment editing and 3D robot state decoding. Our experiments have demonstrated state-of-the-art editing quality, consistent gains from each component, and effective motion guidance for downstream control. Collectively, they prove that RoboEdit unlocks the vast potential of abundant human videos, converting them into scalable, high-fidelity robot experience for generalizable learning.

AI USE STATEMENT

Generative AI tools assisted with language editing, literature lookup, and LaTeX formatting during manuscript preparation. The authors are responsible for reviewing and verifying all manuscript text, code, figures, references, and experimental outputs, and take full responsibility for the final content. Generative models used in the research pipeline, including synthetic data augmentation, are documented in the method and implementation details.

REFERENCES

Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation, 2022. URL https://arxiv.org/abs/2203.13251.

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, et al. HOT3D: Hand and object tracking in 3d from egocentric multi-view videos, 2024. URL https://arxiv.org/abs/2411.19167.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Yu-Wei Chao, Wei Yang, Yu Xiang, et al. DexYCB: A benchmark for capturing hand grasping of objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. Vino: A unified visual generator with interleaved omnimodal context, 2026a. URL https://arxiv.org/abs/2601.02358.

Tianxing Chen, Yue Chen, Zixuan Li, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies, 2026b. URL https://arxiv.org/abs/2607.04434.

Hai Ci, Xiaokang Liu, Pei Yang, Yiren Song, and Mike Zheng Shou. H2r-grounder: A paired-data-free paradigm for translating human interaction videos into physically grounded robot videos. *arXiv preprint arXiv:2512.09406*, 2025.

Jakob Engel, Kiran Somasundaram, Michael Goesele, et al. Project aria: A new tool for egocentric multi-modal ai research, 2023. URL https://arxiv.org/abs/2308.13561.

Rao Fu, Dingxi Zhang, Alex Jiang, et al. Gigahands: A massive annotated dataset of bimanual hand activities, 2024. URL https://arxiv.org/abs/2412.04244.

Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/genesis-world.

Michal Geyer, Omer Bar Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, volume 2024, pp. 1608–1620, 2024.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

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

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision based teleoperation of dexterous robotic hand-arm system, 2019. URL https://arxiv.org/abs/1910.03135.

Haoyang He, Jie Wang, Jiangning Zhang, et al. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing, 2025. URL https://arxiv.org/abs/2512.07826.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Ziqi Huang, Yinan He, Jiashuo Yu, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression, 2018. URL https://arxiv.org/abs/1804.09534.

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

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. URL https://arxiv.org/abs/2403.14468.

Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10138–10148, October 2021.

Arjun S. Lakshmipathy, Jessica K. Hodgins, and Nancy S. Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations, 2024. URL https://arxiv.org/abs/2402.04820.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4545–4565. PMLR, 2025. URL https://proceedings.mlr.press/v305/lepert25a.html.

Guangrun Li, Yaoxu Lyu, Zhuoyang Liu, Chengkai Hou, Jieyu Zhang, and Shanghang Zhang. H2r: A human-to-robot data augmentation for robot pre-training from videos, 2026. URL https://arxiv.org/abs/2505.11920.

Sen Liang, Zhentao Yu, Zhengguang Zhou, Teng Hu, Hongmei Wang, Yi Chen, Qin Lin, Yuan Zhou, Xin Li, Qinglin Lu, et al. Omniv2v: Versatile video generation and editing via dynamic content manipulation. *arXiv preprint arXiv:2506.01801*, 2025.

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-edit: Versatile video editing via instruction and reference guidance, 2026. URL https://arxiv.org/abs/2603.02175.

Yehonathan Litman, Shikun Liu, Dario Seyb, Nicholas Milef, Yang Zhou, Carl Marshall, Shubham Tulsiani, and Caleb Leak. Editctrl: Disentangled local and global control for real-time generative video editing, 2026. URL https://arxiv.org/abs/2602.15031.

Yun Liu, Haolin Yang, Xu Si, et al. Taco: Benchmarking generalizable bimanual tool-action-object understanding. *arXiv preprint arXiv:2401.08399*, 2024.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control, 2023a. URL https://arxiv.org/abs/2306.00958.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023b.

Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video, 2022. URL https://arxiv.org/abs/2202.00164.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6038–6047, 2023.

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

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos, 2026. URL https://arxiv.org/abs/2606.19333.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. Spider: Scalable physics-informed dexterous retargeting, 2025. URL https://arxiv.org/abs/2511.09484.

Kaihang Pan, Qi Tian, Jianwei Zhang, et al. Omniweaving: Towards unified video generation with free-form composition and reasoning, 2026a. URL https://arxiv.org/abs/2603.24458.

Tianlin Pan, Jiayi Dai, Chenpu Yuan, Zhengyao Lv, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Caifeng Shan, and Chenyang Si. Nova: Sparse control, dense synthesis for pair-free video editing, 2026b. URL https://arxiv.org/abs/2603.02802.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2022. URL https://arxiv.org/abs/2108.05877.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Kaiwen Ren, Yiran Jiang, Yongjing Ye, and Shihong Xia. Estimating accurate hand pose in camera space with vision transformer, 2026. URL https://arxiv.org/abs/2609.24424.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos, 2022. URL https://arxiv.org/abs/2212.04498.

Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube, 2022. URL https://arxiv.org/abs/2202.10448.

Yiren Song, Cheng Liu, Weijia Mao, and Mike Zheng Shou. Mitty: Diffusion-based human-to-robot video generation. *arXiv preprint arXiv:2512.17253*, 2025.

George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pp. 478–494, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58451-1. doi: 10.1007/978-3-030-58452-8_28. URL https://doi.org/10.1007/978-3-030-58452-8_28.

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. doi: 10.1109/IROS.2012.6386109.

Eugene Valassakis and Guillermo Garcia-Hernando. HandDGP: Camera-space hand mesh prediction with differentiable global positioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning (CoRL)*, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. URL https://arxiv.org/abs/2503.11651.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos, 2026. URL https://arxiv.org/abs/2510.08377.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

Chenfei Wu, Jiahao Li, Jingren Zhou, et al. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

Sicheng Xie, Haidong Cao, Zejia Weng, Zhen Xing, Haoran Chen, Shiwei Shen, Jiaqi Leng, Zuxuan Wu, and Yu-Gang Jiang. Human2robot: Learning robot actions from paired human-robot videos, 2025. URL https://arxiv.org/abs/2502.16587.

Chendong Xin, Mingrui Yu, Yongpeng Jiang, Zhefeng Zhang, and Xiang Li. Analyzing key objectives in human-to-robot retargeting for dexterous manipulation, 2025. URL https://arxiv.org/abs/2506.09384.

Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. X-humanoid: Robotize human videos to generate humanoid videos at scale, 2025. URL https://arxiv.org/abs/2512.04537.

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, 2025. URL https://arxiv.org/abs/2503.18738.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. XIRL: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2021.

Gu Zhang, Qicheng Xu, Haozhe Zhang, et al. Unidex: A robot foundation suite for universal dexterous hand control from egocentric human videos, 2026. URL https://arxiv.org/abs/2603.22264.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Zhongwei Zhang, Fuchen Long, Wei Li, Zhaofan Qiu, Wu Liu, Ting Yao, and Tao Mei. Region-constraint in-context generation for instructional video editing. *https://arxiv.org/abs/2512.17650*, 2025.

Bojia Zi, Weixuan Peng, Xianbiao Qi, Jianan Wang, Shihao Zhao, Rong Xiao, and Kam-Fai Wong. Minimax-remover: Taming bad noise helps video object removal. *arXiv preprint arXiv:2505.24873*, 2025.

APPENDIX

A ROBOEDIT-ADC DETAILS

A.1 CAMERA ALIGNMENT AND DEPTH REGULARIZATION

Camera Alignment. VGGT predicts camera-from-world extrinsics $E_t = [R_t^{\mathrm{cw}} \mid t_t^{\mathrm{cw}}]$, camera intrinsics K_t, and depth maps D_t. We convert each extrinsic into camera center $c_t = -R_t^{\mathrm{cw}\top} t_t^{\mathrm{cw}}$ and orientation $R_t^{\mathrm{wc}} = R_t^{\mathrm{cw}\top}$. Because the VGGT reconstruction has an arbitrary scale and world frame, we align it to the hand-object rendering frame with a global similarity transform (s, R_a, t_a). We estimate this transform using standard closed-form SVD alignment of sparse corresponding camera centers between the VGGT and rendering coordinate systems. The aligned camera and depth are $c'_t = sR_a c_t + t_a$, $R'_t = R_a R_t^{\mathrm{wc}}$, and $\widetilde{D}_t = sD_t$, yielding $C_t = (K_t, c'_t, R'_t)$.

Depth Regularization. For depth regularization, we use the aligned depth map to correct the scale and depth of the monocular HaMeR hand reconstruction before retargeting. We select stable wrist and finger-base landmarks from MANO (Romero et al., 2017) as palm anchors and denote their set by $\mathcal{A}$. For anchor j, we project its HaMeR camera-space position $H_{t,j}$ with K_t, map the resulting pixel $\mathbf{u}_{t,j} = (u_{t,j}, v_{t,j})$ to the resolution of $\widetilde{D}_t$, and take the median positive, finite depth $d_{t,j}$ within a 5×5 neighborhood. We then back-project this observation into a metric 3D anchor:

$$p_{t,j}^D = d_{t,j} K_t^{-1} \big[u_{t,j}, v_{t,j}, 1\big]^\top.$$

Invalid samples are excluded. From the valid set $\mathcal{A}_t$, we compute $\bar{z}_t^D = \mathrm{median}_{j \in \mathcal{A}_t}(p_{t,j}^D)_z$ and $\bar{z}_t^H = \mathrm{median}_{j \in \mathcal{A}_t}(H_{t,j})_z$, giving $\alpha_t = \bar{z}_t^D / \bar{z}_t^H$. We rescale the complete camera-space hand about the camera center as $\widetilde{H}_t = \alpha_t H_t$, applying the same factor to its global translation while leaving the MANO pose, shape, and relative articulation unchanged. This camera-centered isotropic scaling aligns the metric palm depth while preserving the original image projection.

A.2 PHYSICS-GUIDED REFINEMENT

Refinement Objective. For each retargeted trajectory, we initialize from the kinematic trajectory $q_{1:T}^{e,\mathrm{IK}}$ and optimize the complete temporal window under its kinematics, joint limits, palm geometry, and fingertip correspondences. We retain only trajectories that respect robot joint limits and maintain consistent hand-object contact over time. The main paper defines $\mathcal{L}_{\mathrm{ret}}$; here we expand its four terms. Let $[x]_+ = \max(x, 0)$.

$$\mathcal{L}_{\mathrm{track}} = \sum_{t=1}^{T} \left\| q_t^e - q_t^{e,\mathrm{IK}} \right\|_2^2,$$

$$\mathcal{L}_{\mathrm{geo}} = \sum_{t=1}^{T} \sum_{c \in \mathcal{P}_t} [-d_{t,c}]_+^2 + \beta_{\mathrm{geo}} \sum_{t=1}^{T} \sum_{u \in \mathcal{V}_t} [\delta - s_{t,u}]_+^2,$$

$$\mathcal{L}_{\mathrm{contact}} = \sum_{t=1}^{T} \sum_{k \in \mathcal{C}_t} w_{t,k} \left\| p_k^e(q_t^e) - a_{t,k} \right\|_2^2,$$

$$\mathcal{L}_{\mathrm{temp}} = \sum_{t=2}^{T} \left\| q_t^e - q_{t-1}^e \right\|_2^2.$$

Geometry Loss. The first term of $\mathcal{L}_{\mathrm{geo}}$ detects coarse penetration with the simplified MuJoCo collision geometry: $\mathcal{P}_t$ contains collision pairs with signed distance $d_{t,c}$, where $d_{t,c} < 0$ indicates penetration. The second term checks the robot and object visual meshes to capture fine-scale

penetration. Here, $\mathcal{V}_t$ contains samples from the robot visual mesh near the object, and $s_{t,u}$ is the signed distance from sample u to the object surface. This term penalizes both negative $s_{t,u}$, which indicates penetration, and positive distances smaller than the clearance threshold $\delta = 0.005\,\mathrm{m}$. The weight β_{geo} balances this fine-scale visual-mesh penalty with the MuJoCo collision penalty.

Contact Loss. We construct the valid-contact set $\mathcal{C}_t$ from the reconstructed human fingertips. At each frame, fingertip k is transformed into the object coordinate frame, and $\rho_{t,k}$ is its Euclidean distance to the nearest point on the reconstructed object surface. Contact starts when $\rho_{t,k} \le \tau_k$. We select $\tau_k \in [0.02\,\mathrm{m}, 0.06\,\mathrm{m}]$ according to the target embodiment and contact geometry. Once active, a contact remains active until $\rho_{t,k} > \tau_k + 0.01\,\mathrm{m}$ where we use a $1\,\mathrm{cm}$ wider exit threshold to prevent contact labels from toggling when the measured distance fluctuates near τ_k. We discard contact segments shorter than 10 consecutive frames. For each $k \in \mathcal{C}_t$, $a_{t,k}$ is that nearest object-surface point, $w_{t,k} \ge 0$ is the contact-confidence weight obtained from the temporally filtered contact segment, and $p_k^e(q_t^e)$ is the corresponding robot-fingertip position.

B 3D Robot-State Decoder Details

Decoder Architecture. A shared ResNet-34 (He et al., 2016) feature pyramid extracts local and global features. For each robot hand, a palm-anchor head predicts heatmaps and visibility for eight predefined 2D palm anchors, a spatial fingertip heatmap head localizes up to five fingertips, and an MLP state head predicts palm-frame 3D fingertip coordinates and normalized joint configurations. A shared camera head predicts intrinsics. SQPnP (Terzakis & Lourakis, 2020) recovers the camera-space wrist pose from the predicted 2D palm anchors and predefined 3D palm geometry. A temporal Transformer refines the framewise states over all 81 frames, and embodiment-specific forward kinematics produces the complete camera-space robot-hand trajectory.

Loss Functions. The decoder is trained with three objectives:

$$\begin{aligned}
\mathcal{L}_{\mathrm{anchor}} &= \lambda^a_{\mathrm{hm}}\mathcal{L}^a_{\mathrm{hm}} + \lambda^a_{\mathrm{uv}}\mathcal{L}^a_{\mathrm{uv}} + \lambda^a_{\mathrm{vis}}\mathcal{L}^a_{\mathrm{vis}},\\
\mathcal{L}_{\mathrm{spa}} &= \lambda_{\mathrm{2D}}\mathcal{L}_{\mathrm{2D}} + \lambda^s_{\mathrm{palm}}\mathcal{L}_{\mathrm{palm}} + \lambda_{\mathrm{cam}}\mathcal{L}_{\mathrm{cam}}\\
&\quad + \lambda^s_{\mathrm{qpos}}\mathcal{L}_{\mathrm{qpos}} + \lambda_K\mathcal{L}_K,\\
\mathcal{L}_{\mathrm{tmp}} &= \lambda^t_{\mathrm{palm}}\mathcal{L}_{\mathrm{palm}} + \lambda^t_{\mathrm{qpos}}\mathcal{L}_{\mathrm{qpos}} + \lambda_{\Delta\mathrm{palm}}\mathcal{L}_{\Delta\mathrm{palm}}\\
&\quad + \lambda_{\Delta\mathrm{qpos}}\mathcal{L}_{\Delta\mathrm{qpos}} + \lambda_{\mathrm{res}}\mathcal{L}_{\mathrm{res}}.
\end{aligned}$$

Here, each λ is a scalar weight for its supervision term; a, s, and t distinguish anchor, spatial, and temporal supervision, respectively. The individual terms are defined as follows:

Anchor Heatmap Loss: $\mathcal{L}^a_{\mathrm{hm}}$ uses a heatmap-distribution loss to supervise the rigid palm-anchor heatmaps.

Anchor Coordinate Loss: $\mathcal{L}^a_{\mathrm{uv}}$ uses Smooth-L1 loss on the 2D image coordinates of the palm anchors.

Anchor Visibility Loss: $\mathcal{L}^a_{\mathrm{vis}}$ uses binary cross-entropy to supervise anchor visibility.

2D Fingertip Loss: $\mathcal{L}_{\mathrm{2D}}$ combines a heatmap-distribution loss with Smooth-L1 loss on the 2D fingertip coordinates.

3D Geometry Losses: $\mathcal{L}_{\mathrm{palm}}$ and $\mathcal{L}_{\mathrm{cam}}$ use Smooth-L1 losses on 3D fingertip coordinates in the palm and camera frames, respectively.

Joint-State Loss: $\mathcal{L}_{\mathrm{qpos}}$ uses Smooth-L1 loss on normalized robot joint states.

Camera-Intrinsics Loss: $\mathcal{L}_K$ uses Smooth-L1 loss on the predicted camera intrinsics.

Temporal Motion Losses: $\mathcal{L}_{\Delta\mathrm{palm}}$ and $\mathcal{L}_{\Delta\mathrm{qpos}}$ use Smooth-L1 losses to match adjacent-frame fingertip and joint-state motion.

Temporal Residual Loss: $\mathcal{L}_{\mathrm{res}}$ applies an L2 penalty to the temporal corrections.

Unavailable fingertips, joints, and frames are excluded using validity masks. $\mathcal{L}_{\mathrm{anchor}}$, $\mathcal{L}_{\mathrm{spa}}$, and $\mathcal{L}_{\mathrm{tmp}}$ are the weighted combinations shown above.

Table S1: RoboEdit-14M source distribution, with duration computed at 30 FPS.

Source dataset	Paired clips	Paired frames	Duration (h)
DexYCB	67,200	5,443,200	50.40
GigaHands	11,715	948,915	8.79
H2O	7,955	644,355	5.97
HOT3D	35,591	2,882,871	26.69
TACO	52,086	4,218,966	39.06
Total	174,547	14,138,307	130.91

Table S2: RoboEdit-14M distribution by target embodiment.

Embodiment	Real	Synthetic	Total
Ability	24,197	4,839	29,036
Allegro	24,197	4,839	29,036
Unitree Dex3	12,237	2,447	14,684
Inspire	24,197	4,839	29,036
Panda gripper	12,237	2,447	14,684
SCHUNK SVH	24,197	4,838	29,035
XHand	24,197	4,839	29,036
Total	145,459	29,088	174,547

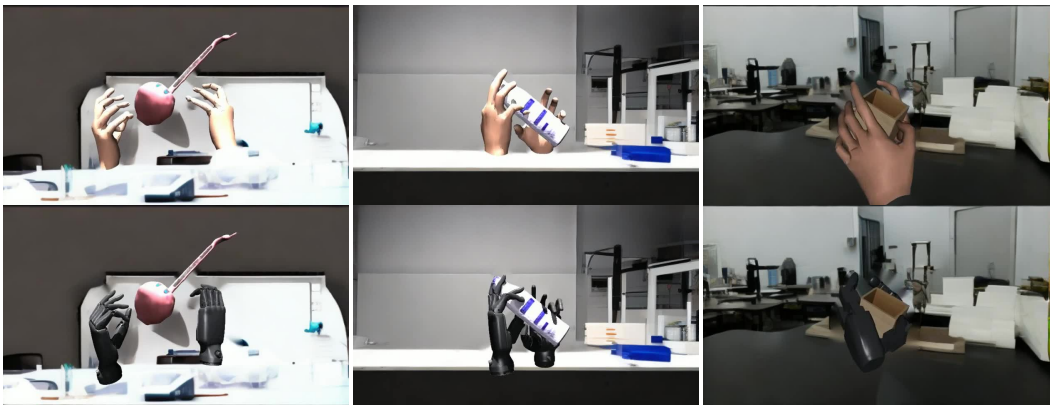

Figure S1: Synthetic paired-video examples. Each column shows an aligned synthetic human frame and its robot target under the same scene, object state, and camera view.

C RoboEdit-14M Statistics

Tables S1 and S2 summarize RoboEdit-14M by source and target embodiment, while Fig. S1 shows representative synthetic human/robot pairs.

D Training Details

Computing Infrastructure. Experiments were conducted on Ubuntu 22.04.5 workstations equipped with two AMD EPYC 9354 CPUs, 1.5 TiB of host memory, and eight NVIDIA H100 NVL GPUs, each with 94 GiB of memory.

Backbone and Cross-Embodiment Adaptation. We first train the video-editing backbone for approximately 10K steps, updating 283.4M parameters with AdamW at a learning rate of 5×10^{-5} and an effective batch size of 12 across eight GPUs. We then freeze the backbone and jointly train LoRA adapters on selected attention and feed-forward projections together with residual bottleneck adapters in the final ten transformer blocks, using an effective batch size of 16 across four GPUs. The LoRA and residual-adapter branches contain 2.5M and 7.9M trainable parameters, respectively.

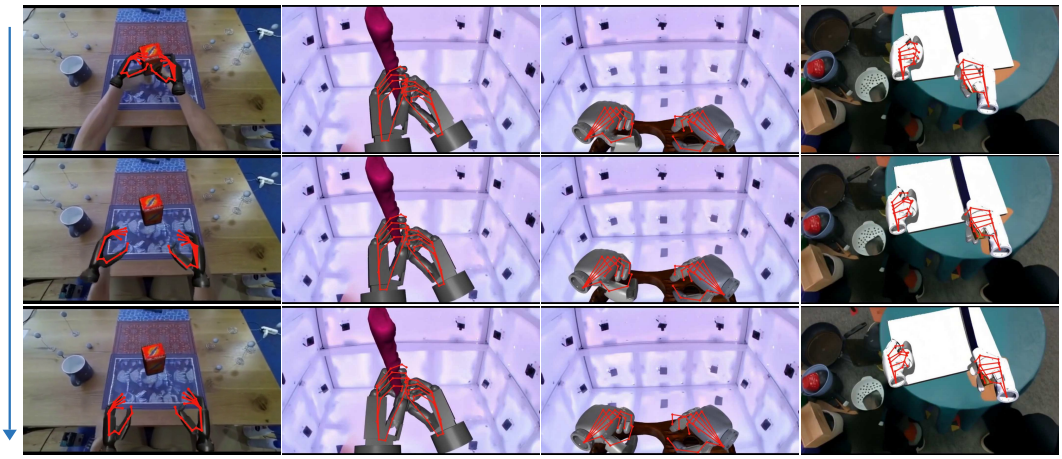

Figure S2: Additional 3D Robot-State Decoder results on RoboEdit-Trans edited videos. Red overlays denote predicted camera-space hand states.

Baseline Backbone Configuration. The VACE baselines use pretrained `Wan2.1-VACE-1.3B`, with UMT5-XXL and the Wan2.1 VAE. Inference uses 81 frames at 480p with 20-step UniPC sampling, guidance scale 5.0, and flow shift 16. RoboEdit-Trans initializes its diffusion transformer and image encoder from `Wan2.1-Fun-1.3B-InP` and its text encoder and VAE from `Wan2.1-T2V-1.3B` before the task adaptation described above.

3D Robot-State Decoder. We first train the palm detector and framewise spatial estimator on rendered and composited robot frames. We then freeze both components and train temporal refinement on complete 81-frame sequences. We train the spatial and temporal stages for approximately 8.9K and 1.5K optimization steps, respectively; both stages use AdamW with a learning rate of 2×10^{-5}.

Qwen-Image-Edit. We fine-tune Qwen-Image-Edit for approximately 4.9K optimization steps on aligned image pairs sampled from RoboEdit-14M. At inference, it generates robot references at indices $\{0, 10, \ldots, 80\}$, after which RoboEdit-Trans produces an 81-frame video.

Randomness Control. For reproducibility, we fix the random seeds used for data sampling, decoder training, and model inference. RoboEdit-Trans data sampling initializes its NumPy generator with 20260531 plus the epoch index. Decoder training initializes the Python, NumPy, and PyTorch generators with 20260705 plus the distributed rank. RoboEdit-Trans, Qwen-Image-Edit, and baseline inference use deterministic per-sample PyTorch seeds derived from 42, 20260629, and 20260624, respectively.

E Runtime Analysis

Table S3 reports the stage-by-stage runtime of RoboEdit-ADC curation, Qwen reference keyframe generation, and RoboEdit-Trans inference averaged over 300 benchmark cases on one NVIDIA H100 NVL GPU. Generating nine Qwen references with 40 steps per image takes 267.71 s. Together with RoboEdit-Trans inference, reference generation, video editing, and 3D state decoding take 498.99 s (8.3 min) in total, substantially more efficient than the 20.0 min required by RoboEdit-ADC.

F Additional Qualitative Results

Figure S3 extends the main-paper comparison with five additional interaction examples. We again use midpoint non-keyframes at indices 5, 35, and 65. Figure S2 shows additional decoded robot states from edited videos, and Figure S5 presents additional real-robot results on YCB-object manipulation with both end effectors.

Table S3: Runtime breakdown for RoboEdit-ADC and RoboEdit-Trans on one H100 GPU.

Method	Stage	Time (s)
RoboEdit-ADC	Object Segment	48.17
	Object Pose	59.25
	Object Mesh	125.58
	Hand Pose	251.92
	Camera State	104.43
	Kinematic Retargeting	111.50
	Depth Alignment	18.66
	Physics Refinement	308.28
	Inpainting	129.08
	Foreground Rendering	43.06
	RoboEdit-ADC total	**1199.94**
Qwen-Image-Edit	Reference generation (9 frames)	267.71
RoboEdit-Trans	Input preparation and encoding	14.05
	Video denoising (20 steps)	134.22
	VAE decoding	7.92
	3D Robot-State Decoder	75.09
	RoboEdit-Trans subtotal	**231.28**
Qwen-Image-Edit + RoboEdit-Trans	**Total**	**498.99**

G Trajectory-Conditioned Control and Real-Robot Deployment

We evaluate whether the robot-hand trajectories recovered by the 3D Robot-State Decoder are physically executable and can supervise downstream control. We first train a trajectory-conditioned controller in simulation and then execute simulation-validated trajectories on a physical Franka Panda.

Trajectory-conditioned residual control. For each interaction, the 3D Robot-State Decoder provides a decoded robot-hand trajectory $\tau^h = \{h_t^\star\}_{t=1}^T$, which serves as the control reference; $h_t^\star$ denotes the reference palm or end-effector state. We represent the trajectory relative to its first frame and resample it to the control horizon. Given the current task state s_t and the robot-hand tracking error $h_t^\star - p_t$, the residual policy π_ϕ predicts a bounded correction to a nominal joint command computed by inverse kinematics (IK):

$$a_t = \pi_\phi(s_t, h_t^\star - p_t)\,,$$
$$q_t^{\mathrm{cmd}} = \mathrm{clip}\big(q_t^{\mathrm{IK}} + \rho_{\mathrm{act}} a_t, q_{\min}, q_{\max}\big)\,.$$

Here, π_ϕ is the residual policy with parameters ϕ, s_t is the current robot and object task state, and p_t is the current palm or end-effector state. q_t^{IK} is the nominal arm configuration computed by IK, and q_t^{cmd} is the resulting command sent to the low-level arm controller; $a_t \in [-1, 1]^7$ is the residual arm action, and $\rho_{\mathrm{act}} = 0.02\,\mathrm{rad}$ is its scale. $q_{\min}$ and $q_{\max}$ denote the Franka arm joint limits. The operator clip enforces these limits elementwise. The low-level controller executes q_t^{cmd}, whose outcome is evaluated through palm tracking and task success.

Control objective. The reward supervises tracking of the decoded robot-hand trajectory while encouraging successful and smooth task execution:

$$r_t = w_{\mathrm{hand}} \exp(-\beta \left\| p_t - h_t^\star \right\|_2) + r_t^{\mathrm{task}} - w_{\mathrm{act}} \left\| a_t \right\|_2^2 .$$

Here, r_t is the total reward; w_{hand} and w_{act} weight trajectory tracking and action regularization, respectively; β controls sensitivity to the Euclidean tracking error; and r_t^{task} contains rewards for grasp maintenance, object-motion consistency, and lift completion. The decoder trajectory therefore conditions both the policy action and its training objective.

Simulation setup. We train the residual policy using PPO (Schulman et al., 2017) in Genesis (Genesis Authors, 2024), with 512 parallel environments spanning 18 YCB objects. A single policy is trained across all objects, without per-object fitting. During training, the initial object position and yaw are randomized by up to 4 cm and 23°, respectively.

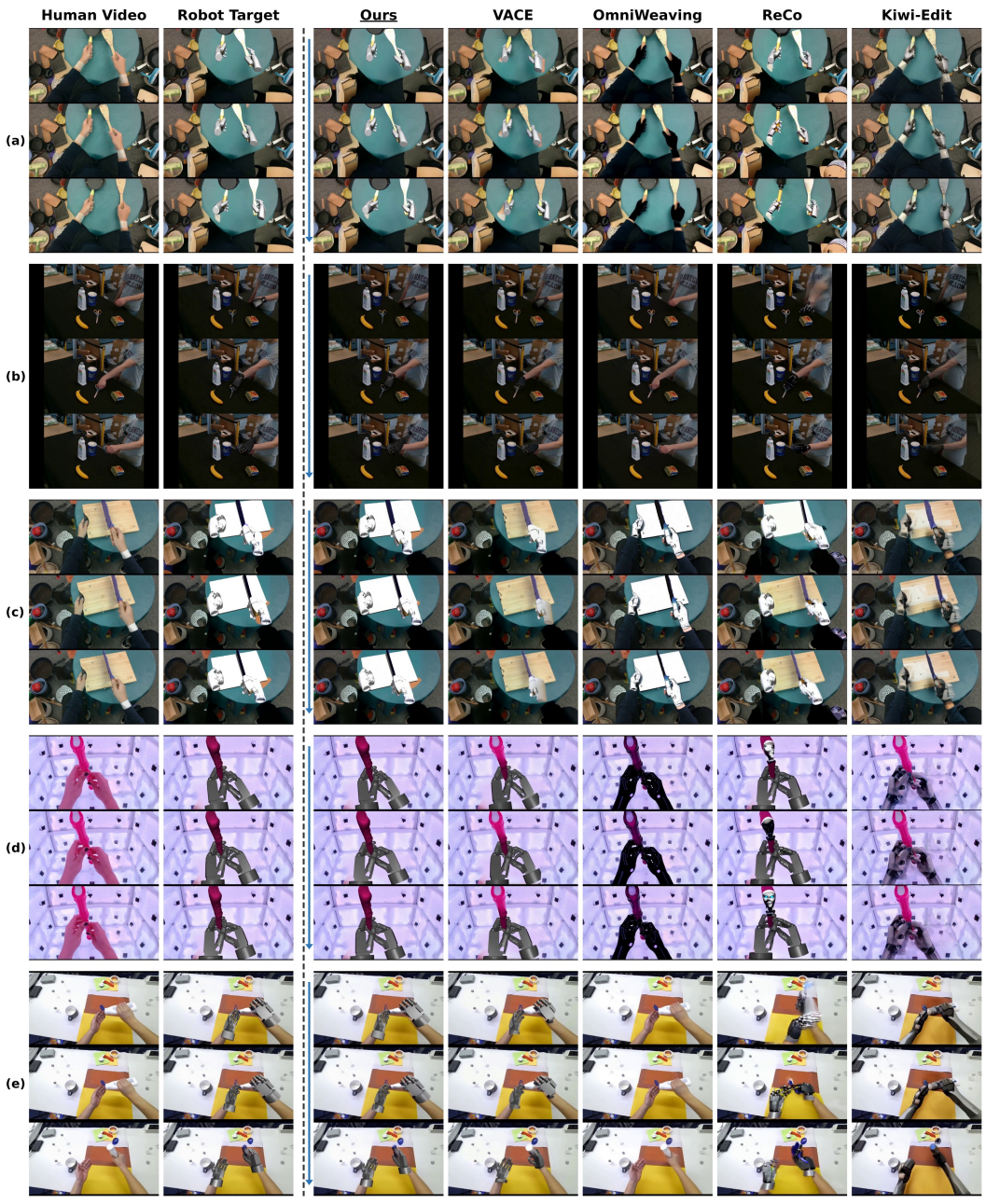

Figure S3: Additional qualitative comparisons on five human-object interactions.

A rollout succeeds if the object remains grasped and its terminal position is within 8 cm of the demonstrated target. Across the randomized environments, the controller achieves trajectory-reproduction success rates of 71% for the Panda gripper and 62% for XHand. Representative simulation rollouts are shown in Fig. S4.

Real-robot deployment. We evaluate four YCB-object tasks on a 7-DoF Franka Panda with a parallel-jaw gripper or 12-DoF XHand, using decoded trajectories as control references. Following RoboDojo (Chen et al., 2026b), each task–end-effector setting undergoes 10 trials with predefined initial layouts replayed before each trial. Success requires task completion within 1.5 times the 90th-percentile demonstration duration; timeouts and safety-triggered terminations count as failures.

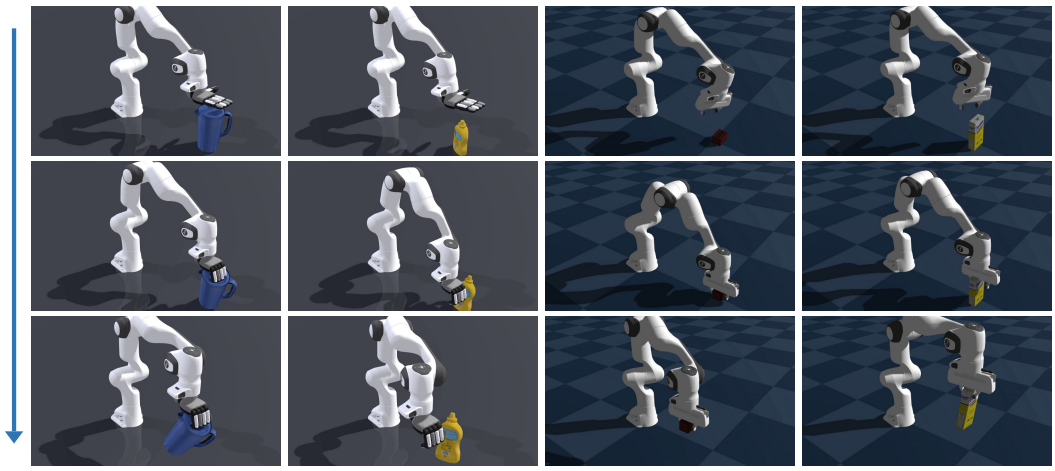

Figure S4: Simulation rollouts of the trajectory-conditioned controller across four YCB-object manipulation tasks.

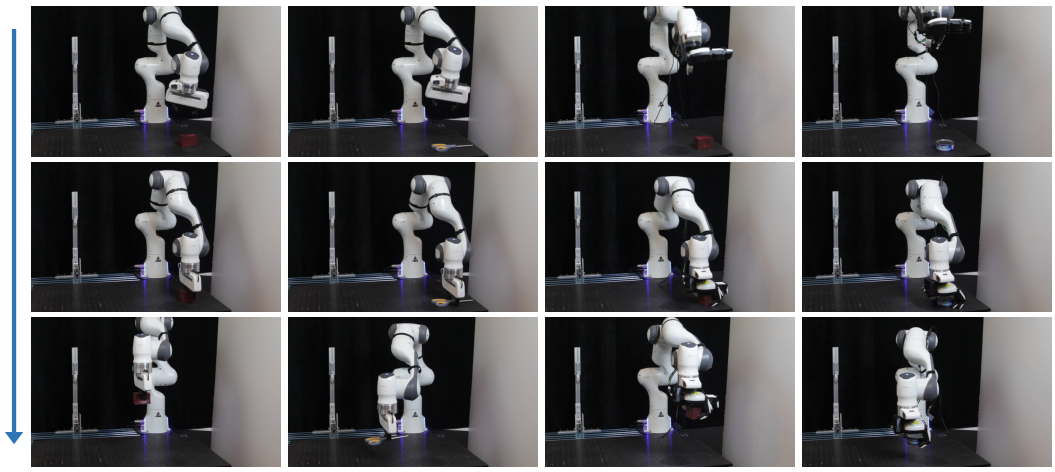

Figure S5: Additional real-robot deployment results across four YCB-object tasks.

H BENCHMARK DISTRIBUTION

Each of the 300 benchmark cases pairs a human-video clip with a target robot embodiment. Figure S6 summarizes their distribution by source dataset, target embodiment, camera setup, and task family.

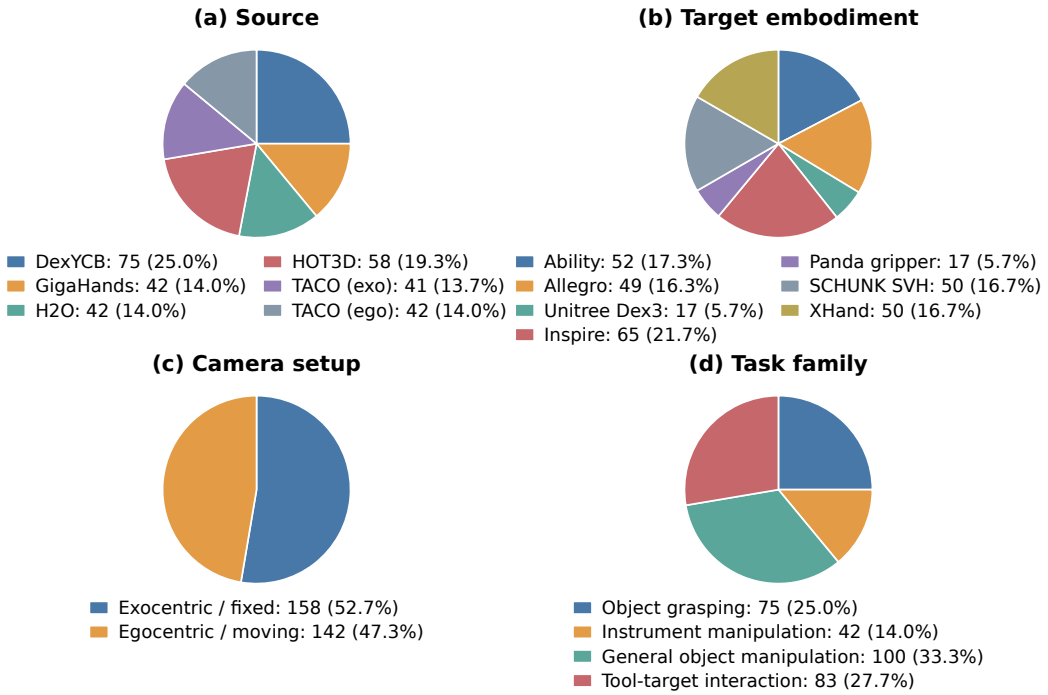

Figure S6: Distribution of the 300-case benchmark. Legends show case counts and percentages; task families are coarse, source-based activity groups.