EGOTACO: EGOCENTRIC TASK ABSTRACTION FOR CROSS-EMBODIMENT OBJECT MANIPULATION IN EDITABLE WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Egocentric video reveals what a manipulation task changes in the world, but its observed motion is specific to a human operator and one set of object properties. This makes direct trajectory transfer brittle when robot morphology, object geometry, or dynamics change. We introduce EgoTACO, a framework that represents a task through its goal, ordered events, object-state transition, affordances, and contact roles. A three-layer decomposition separates this task evidence from scene appearance. Independent object and embodiment conditions then support task-contact replanning and condition full-frame robot-video generation. A verification harness checks geometric feasibility, physical constraints, and task completion, and recompiles failed phases. In 60 constructed task–embodiment cases with supplied physical properties, task-contact replanning achieves 100% terminal success versus 0% for framewise inverse kinematics, despite deviating further from the human path. Controlled object interventions also show complementary effects of geometry and physics adaptation. These results support task-state abstraction as a representation for cross-embodiment compilation.

1 INTRODUCTION

Egocentric recordings expose everyday manipulation across objects, environments, and activities at a scale that is difficult to match with robot demonstrations (Damen et al., 2022; Grauman et al., 2024; Kwon et al., 2021). They show the interaction that matters: which object changes state, which contacts enable that change, and in what order the relevant events occur. Transferring this evidence to a robot requires a representation that survives changes in the agent and its physical environment.

A human trajectory is one realization of a task, rather than a complete specification of it. Replacing a small object with a larger or heavier one can invalidate the demonstrated grasp, clearance, speed, or support configuration. A new robot may have a different workspace, aperture, payload, or number of arms. Matching the observed wrist path can therefore preserve visual similarity while preventing the intended object-state transition. Conversely, a robot can complete the same task through motion that differs substantially from the demonstration.

We study this distinction through *task-state abstraction*. EgoTACO represents the transferable content of an egocentric video as a terminal goal, an ordered event graph, the required object-state change, affordances, and contact roles. The representation separates what must happen from the geometry and dynamics of a particular execution. Object properties and embodiment descriptors remain independent conditions, so an edit can trigger a new contact topology and robot motion while preserving the task goal.

The framework joins this representation with visual generation. It decomposes an observation into a completed background, human-operation evidence, and manipulated-object evidence. The latter two define the task latent; the background supplies the scene context. A frozen video backbone with trainable conditioning adapters predicts the full robot video. A constraint harness tests recovered states for reachability, contact consistency, physical feasibility, and task progress, and recompiles failed temporal phases. Figure 1 illustrates the intended separation of task semantics from their robot realization.

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.

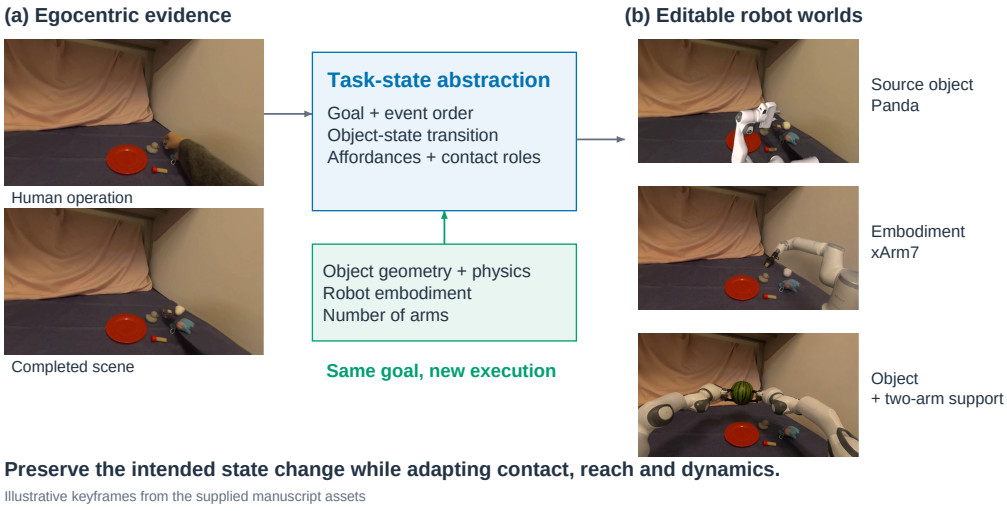

Figure 1: **Task semantics persist across editable robot worlds.** Egocentric evidence defines the intended state change. Object and embodiment edits alter its physical realization. The keyframes are illustrative examples from the supplied visual assets, including a completed scene, robot replacement, and two-arm object support; they do not by themselves establish closed-loop success.

Our contribution is an explicit factorization for editable human-to-robot generation, together with a task-contact compilation and verification mechanism. We evaluate the mechanism using controlled object interventions and 60 constructed cross-embodiment cases. The resulting comparison isolates why path similarity, IK feasibility, and task completion must be measured separately. A fixed-learner downstream comparison and conditional video-prediction metrics organize the evaluation of the resulting robot-training data.

2 Related Work

Learning from human video. Egocentric datasets provide observations of human–object interaction (Damen et al., 2022; Grauman et al., 2024; Kwon et al., 2021). Visual representations such as R3M and affordance-based approaches connect human experience to robot manipulation (Nair et al., 2022; Bahl et al., 2023). Human-to-robot generation methods further study paired supervision, motion anchors, and alignment (Xie et al., 2026; Ye et al., 2025; Li et al., 2025). EgoTACO focuses on the variable transferred between embodiments: event and object-state semantics, with the robot execution recompiled under edited physical conditions.

Task structure and cross-embodiment learning. Task programs and relational constraints expose manipulation semantics in executable form (Luo, 2026; Huang et al., 2024). Heterogeneous robot datasets and generalist policies broaden the range of embodiments represented during learning (Collaboration et al., 2023; Kim et al., 2024). These directions motivate separating task information from embodiment-specific realization. EgoTACO expresses that separation through independently editable task, object, and robot conditions, and evaluates contact replanning under controlled changes in those conditions.

Conditional video generation. Video diffusion provides a model family for temporal visual synthesis (Ho et al., 2022; Blattmann et al., 2023). Segmentation and video completion provide the observation layers needed to recover the scene (Ravi et al., 2024; Suvorov et al., 2022; Zhou et al., 2023). EgoTACO uses scene tokens and task-dependent adapters for full-frame prediction, while a separate verification stage evaluates whether the predicted interaction realizes the task. This distinguishes visual prediction quality from execution validity.

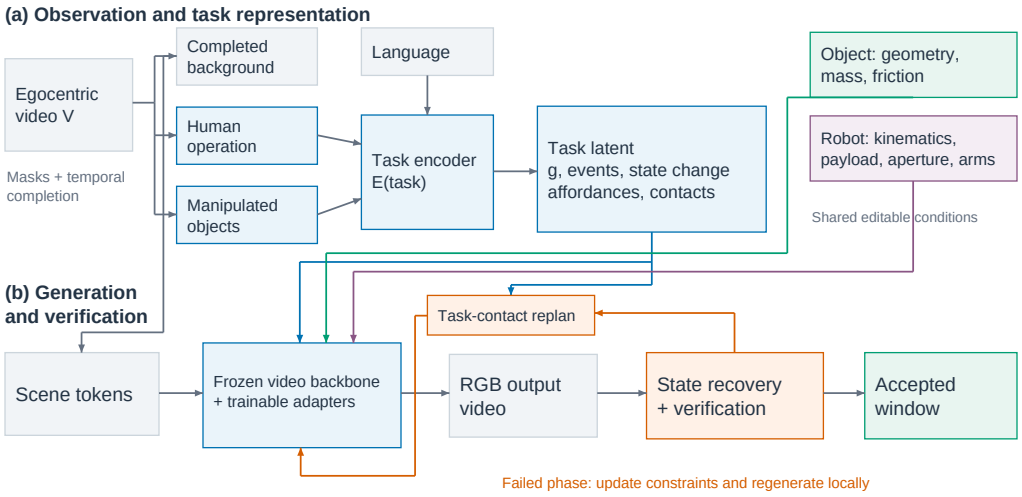

Figure 2: **EgoTACO computation graph.** Human-operation and object layers determine the task latent; the completed background supplies scene tokens. Object and robot conditions enter the video backbone through trainable adapters. Recovered states are checked against task, geometry, and physics constraints. Failed phases trigger contact replanning and local regeneration. The framework does not supply a framewise human wrist path to the video generator.

3 METHOD

3.1 PROBLEM FORMULATION

Given an egocentric human-operation video $V = \{I_t\}_{t=1}^{T}$, and an optional language instruction l, our objective is to generate a robot execution video $\hat{V}_R$ under target object condition c and robot embodiment e. The generated video should preserve the original scene semantics and realize the same task-state transition through motion executable by the target robot.

Unlike framewise retargeting, EgoTACO gives the generator no human wrist trajectory or IK projection. It instead extracts a path-independent task representation and recompiles execution for the target object, embodiment, and physical conditions.

The observation is first decomposed into visual layers. The task encoder receives the operation and object evidence:

$$(V_B, V_H^{\text{op}}, V_O) = \mathcal{D}(V), \qquad z_\tau = E_\tau(V_H^{\text{op}}, V_O, l), \tag{1}$$

followed by conditional generation

$$\hat{V}_R = G_{\phi,\psi}(V_B, z_\tau, z_O, z_e, l). \tag{2}$$

Here V_B is the completed background video with the human operator removed, V_H^{op} is the visible human-operation layer, and V_O is the manipulated-object layer. The task encoder uses the operation and object layers together with l, while the background provides scene context. Parameters ϕ belong to the frozen video backbone and ψ denotes the trainable conditioning adapter.

3.2 OBSERVATION FACTORIZATION AND SPATIAL GROUNDING

We separate the egocentric observation into scene, human-operation, and manipulated-object layers. SAM2 temporal mask propagation (Ravi et al., 2024) estimates temporally consistent human and object supports M_H and M_O. The visible human and object regions are kept disjoint to prevent human pixels at occlusions from contaminating the object appearance. Temporal video completion (Suvorov et al., 2022; Zhou et al., 2023) reconstructs the background regions occluded by the operator, giving

$$V_H^{\text{op}} = V \odot M_H, \qquad V_O = V \odot M_O. \tag{3}$$

The decomposition preserves scene context without copying the operator and exposes object identity and state for independent editing.

Across time, we extract visible object-centroid displacement, principal-axis rotation, support-relation changes, and human–object interaction intervals. These measurements describe the object-state change induced by the task rather than a robot end-effector trajectory. The resulting execution can therefore differ from human motion when robot kinematics, object dimensions, or physical properties change.

Camera–robot calibration. To relate recovered object and contact geometry to executable robot space, we establish a correspondence between the camera frame $\mathcal{F}_C$ and robot base frame $\mathcal{F}_R$. Camera–robot calibration provides

$$ {}^{R}\mathbf{T}_C = \begin{bmatrix} {}^{R}\mathbf{R}_C & {}^{R}\mathbf{t}_C \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{4} $$

which maps a camera-frame point to robot coordinates:

$$ \tilde{\mathbf{p}}_R = {}^{R}\mathbf{T}_C \tilde{\mathbf{p}}_C. \tag{5} $$

For a moving camera, the mapping is evaluated at the relevant pose and places target regions, contacts, and obstacles in robot coordinates; it establishes reachability but does not prescribe arm trajectories.

3.3 Structured task abstraction

Figure 4 illustrates how event semantics become robot-specific constraints. The task encoder extracts a structured representation from the operation layer, object layer, and language:

$$ \begin{aligned} z_\tau &= E_\tau(V_H^{\mathrm{op}}, V_O, l) \\ &= \{g, \mathcal{E}, \Delta s_O, \mathcal{A}, \mathcal{C}\}, \end{aligned} \tag{6} $$

where g is a terminal predicate, $\mathcal{E}$ an ordered event graph, Δs_O the required object-state transition, and $\mathcal{A}$ and $\mathcal{C}$ the affordance and contact-role sets.

Rather than prescribing an action sequence, $\mathcal{E}$ represents the events and ordering dependencies required to complete the task. For pouring, it distinguishes the source vessel from the receiver and recovers approach, support, grasp, tilt, transfer, and release phases without recording the human wrist samples that realized them. The representation specifies

$$ s_O^0 \xrightarrow{\mathcal{E}} s_O^T, \qquad g(s_O^T) = 1, \tag{7} $$

without fixing a Cartesian path. Different embodiments can therefore realize the same task-state transition through different robot motions.

Editable conditions. The object condition is represented as

$$ z_O = \{o, \mathcal{G}, m, \mu, \mathcal{R}\}, \tag{8} $$

where o denotes identity, $\mathcal{G}$ geometry, m mass and inertia-related properties, μ friction and contact properties, and $\mathcal{R}$ candidate interaction regions. The embodiment condition is

$$ z_e = \{\mathcal{K}, \mathcal{W}, p_{\max}, a_g, n_a\}, \tag{9} $$

where $\mathcal{K}$ is the kinematic chain, $\mathcal{W}$ the reachable workspace, $p_{\max}$ the payload constraint, a_g the gripper aperture range, and n_a the number of available arms.

Separating these factors from z_τ permits independent object and embodiment edits:

$$ (z_\tau, z_O, z_e) \xrightarrow{u} (z'_\tau, z'_O, z'_e), \qquad g' = g. \tag{10} $$

The terminal predicate remains unchanged unless the instruction explicitly edits the goal. Replacing a lightweight cup with a larger or heavier object may alter grasp locations, route, speed, support relations, and the number of participating arms while preserving the relocation goal.

3.4 CONSTRAINT COMPILATION AND CONTACT REPLANNING

Given (z_τ, z_O, z_e), the execution harness recompiles the abstract task into robot-specific constraints. For event i, it constructs

$$\Omega_i = \Omega_i^{\text{goal}} \cap \Omega_i^{\text{reach}} \cap \Omega_i^{\text{contact}} \cap \Omega_i^{\text{physics}}, \tag{11}$$

where the four terms encode the required object-state change, kinematic reachability, contact topology, and dynamic constraints.

Geometry adaptation updates candidate grasps, collision boundaries, swept volumes, and gripper-aperture constraints according to the edited object and embodiment. A source path that causes penetration or collision after replacement is not replayed; the edit instead triggers a new spatial solution. Physics adaptation further adjusts execution under joint, velocity, acceleration, jerk, mass, inertia, friction, and support limits. Thus, even a collision-free motion is rejected if it exceeds joint-load limits or destabilizes contact.

Object geometry primarily updates Ω_i^{reach} and $\Omega_i^{\text{contact}}$, whereas mass, friction, and dynamics primarily update $\Omega_i^{\text{physics}}$. This factorization enables local counterfactual edits without redefining the task goal.

Contact selection and robot motion. EgoTACO selects the contact relations required by task events before solving for robot motion, rather than applying framewise inverse kinematics to the complete human trajectory. Affordances $\mathcal{A}$ and contact roles $\mathcal{C}$ determine whether the robot must grasp, support, push, pour, or stabilize. Candidate contact states are sampled in Ω_i and tested with inverse kinematics; the compiler does not minimize the distance between robot and human trajectories.

A candidate execution sequence

$$X = \{x_i\}_{i=1}^{|\mathcal{E}|} \tag{12}$$

is valid only when

$$x_i \in \Omega_i, \qquad \forall i, \tag{13}$$

connecting motions remain feasible, and the object state advances according to $\mathcal{E}$ until $g(s_O^T) = 1$. Multiple robot trajectories may therefore realize one task, so EgoTACO preserves task-state consistency rather than pointwise Cartesian agreement with human motion.

Arm-count selection follows the same constraint logic. A single-arm candidate is rejected when object span exceeds the effective gripper aperture, predicted joint load exceeds single-arm capacity, or one contact cannot provide stable support. If the embodiment provides two arms, the compiler evaluates dual-arm topologies. For pouring, one arm may stabilize the receiver while the other controls the source vessel; for bulky or heavy objects, both arms can share support.

3.5 FACTORIZED FULL-FRAME VIDEO GENERATION

The completed background is encoded into spatially aligned scene tokens. Task, object, and embodiment conditions enter the frozen video denoising backbone through trainable attention adapters (Figure 2). The model predicts all output latent tokens and produces the complete robot video through joint RGB decoding. No source pixels, robot renderings, or robot masks are composited after decoding. Robot masks are used only to define training-region loss weights, not to prescribe the robot position or motion.

Training pairs consist of robot-removed backgrounds and corresponding real robot-video targets. Since static background occupies most image regions, a uniform objective may neglect the dynamic regions that determine manipulation quality. We therefore use region-balanced flow matching:

$$\begin{aligned} \mathcal{L}_{\text{flow}} &= \lambda_M \mathcal{L}_M + (1 - \lambda_M)\mathcal{L}_{\bar{M}}, \\ \lambda_M &= 0.7, \end{aligned} \tag{14}$$

where M denotes dynamic latent support and $\bar{M}$ the remaining scene support. A temporal consistency term compares predicted and target latent changes between adjacent time tokens.

During training, task, object, or embodiment conditions are shuffled to construct condition negatives. A change in one factor should primarily affect manipulation-related regions while leaving unrelated background stable. This encourages condition-sensitive interaction prediction instead of simple background reconstruction.

3.6 Execution verification and local recompilation

After generation, EgoTACO recovers robot and object states from the RGB output and verifies the interaction. Checks cover robot identity, link proportions, joint feasibility, collision, contact order, object-state progress, slip, support relations, and the terminal predicate. For a temporal window W, let $\mathcal{V}_{\text{task}}(W)$, $\mathcal{V}_{\text{geo}}(W)$, and $\mathcal{V}_{\text{phys}}(W)$ denote binary task, geometry, and physics checks. A window is accepted only if

$$\begin{aligned}\mathcal{V}_{\text{task}}(W) &= \mathcal{V}_{\text{geo}}(W) \\ &= \mathcal{V}_{\text{phys}}(W) = 1.\end{aligned} \tag{15}$$

Verification distinguishes visual fidelity from task validity. Recovered object poses and event timing test whether Δs_O is realized in the correct order, while kinematic, collision, and contact checks test physical feasibility. A preserved scene without a robot, a smooth but static output, or an arm that fails to change the object state is not counted as successful manipulation.

When an event fails verification, the harness identifies the violated factor and updates the corresponding constraints. Collision failures revise Ω_i^{reach} or $\Omega_i^{\text{contact}}$, whereas mass- or dynamics-related failures revise $\Omega_i^{\text{physics}}$. The affected phase then undergoes contact replanning and local regeneration instead of regenerating the entire video.

Long-horizon state. Long egocentric operations are generated in overlapping temporal windows. Window k maintains the compact state

$$q_k = \{p_k, s_{O,k}, e, j_k, c_k, h_k\}, \tag{16}$$

where p_k is the task phase, $s_{O,k}$ the object state, e the robot identity, j_k a joint-state hypothesis, c_k the contact state, and h_k the visual history. The joint hypothesis is used only for verification; adjacent windows are accepted when robot identity, object pose, contact state, and task phase agree in overlap.

4 Experiments

4.1 Experimental setup

Data and evaluation protocols. Controlled videos from RoboTwin, CALVIN, and LIBERO provide task configurations, robot states, actions, and success predicates. The broader data protocol uses approximately 100 source videos to produce 500 synthetic videos, with approximately 50 held-out videos for generalization. EgoDex, Ego4D, and EPIC-KITCHENS 100 provide natural egocentric observations. A separate controlled video split comprises 36 tasks, divided into 24 synthesis, six validation, and six out-of-distribution (OOD) tasks, rendered at 8 fps.

Baselines and metrics. For downstream data utility, native simulator demonstrations, raw egocentric videos, synthetic videos without adaptation, and EgoTACO videos are compared with fixed ReKap architecture, training budget, data volume, and evaluation. Prediction accuracy and task success measure downstream utility. Relative-GT uses the simulator as a 100% reference. Executability, IK feasibility, geometry and physics violations, and human-path deviation characterize the generated interaction. Object-state, robot-identity, and temporal consistency further distinguish visual fidelity from task validity. The intended reporting protocol uses five runs.

4.2 Main downstream comparison

Table 1 presents the main comparison of training data sources. The simulator reference supplies native robot supervision, while raw egocentric video supplies human visual evidence. Synthetic

video without adaptation isolates visual conversion, and EgoTACO adds task abstraction and object–embodiment adaptation. Holding the downstream learner fixed makes this comparison a test of the training data rather than a change in policy architecture.

Table 1: **Main downstream comparison under fixed ReKap settings.** Relative-GT is normalized to the simulator reference.

Training data	Accuracy ↑	Task success ↑	Relative-GT ↑	Executability ↑
Simulator	94.8%	91.6%	100%	100%
Raw egocentric	68.7%	51.3%	63.9%	57.8%
Synthetic, no adaptation	79.6%	68.4%	77.2%	73.6%
EgoTACO	89.1%	84.7%	91.8%	92.4%

4.3 Object editing and cross-embodiment transfer

Table 2 presents the original object-editing and cross-embodiment comparison. The object-editing protocol uses 100 sources and contrasts source replay, geometry adaptation, and joint geometry–physics adaptation. The transfer protocol evaluates 20 tasks on three robot morphologies, giving 60 matched task–embodiment cases with supplied physical properties and goals.

Table 2: **Object editing and cross-embodiment generalization.** * marks constructed goals with supplied physical properties.

Setting	N	Success ↑	Geometry violations ↓	Physics violations ↓
Object: source replay	100	2.0%	97.5%	100%
Object: geometry	100	61.0%	0%	100%
Object: full	100	94.0%	0%	0%
Cross-embodiment: IK	60	0%*	15.8%	26.7%
Cross-embodiment: EgoTACO	60	100%*	0%	0%

Source replay has a reported geometry-violation rate of 97.5% and physics-violation rate of 100%. Geometry adaptation reduces geometric violations to zero, but leaves physics violations unchanged. Joint adaptation reduces both to zero. The comparison separates spatial clearance from the effects of altered mass, inertia, and dynamic limits: a collision-free motion can still exceed load limits or destabilize support. Object-edit task success remains a separate metric in Table 2.

4.4 Component ablations

Table 3 isolates task structure, geometry, physics adaptation, and task-contact replanning. Removing physics adaptation retains 0% geometric violations but yields 100% physics violations. Removing contact replanning yields 84.2% IK feasibility and 0% terminal success on the 60 constructed cross-embodiment cases; the full method reaches 100% on both measures.

Table 3: **Component ablation under matched settings.** *: 60 cross-embodiment tasks. †: 120 object interventions in the original ablation annotation.

Variant	Success ↑	IK feasibility ↑	Geo. violations ↓	Phys. violations ↓
Without task structure	31.7%*	71.7%*	12.5%†	18.3%†
Without geometry	43.3%*	66.7%*	92.5%†	25.0%†
Without physics adaptation	16.7%*	100%*	0%	100%
Without contact replanning	0%*	84.2%*	15.8%†	21.7%†
Full EgoTACO	100%*	100%*	0%†	0%†

4.5 TASK COMPLETION VERSUS TRAJECTORY SIMILARITY

Table 4 compares the same cross-embodiment strategies through path deviation as well as execution. Framewise IK stays closer to the human path, with a mean deviation of 6.1 cm, but reaches only 84.2% IK feasibility and 0% terminal success. Task-contact replanning increases deviation to 17.9 cm while achieving 100% feasibility and success. Thus, matching local human poses does not ensure an ordered sequence of valid interactions. The task representation permits robot-specific motion that realizes the required object-state transition.

Table 4: **Human-path similarity and execution on 60 constructed cases.** Physical properties and goals are supplied. Path deviation measures similarity to the demonstration, rather than an optimization objective for EgoTACO.

Method	Terminal success ↑	IK feasibility ↑	Path deviation (cm)
Framewise IK	0%	84.2%	6.1
Task-contact replanning	100%	100%	17.9

4.6 CONDITIONAL VISUAL PREDICTION

Table 5 makes the available prediction results explicit. Task conditioning improves motion-region PSNR by 3.79 dB over an equal-capacity unconditioned predictor. A lightweight adapter reduces held-out BridgeData flow loss by 39.50%. The table separates the two metrics.

Table 5: **Conditional prediction results.** Flow-loss scores are rounded to three decimals.

Metric	Baseline	Conditioned	Change
Motion-region PSNR (dB) ↑	24.86	28.65	+3.79 dB
Held-out BridgeData flow loss ↓	0.286	0.173	−39.50%

Figure 3 shows favorable and unfavorable OOD examples. The first row depicts UR5e predictions and the second xArm7 predictions. These selected examples expose the predicted robot and object regions alongside the target frame, complementing the aggregate prediction metrics.

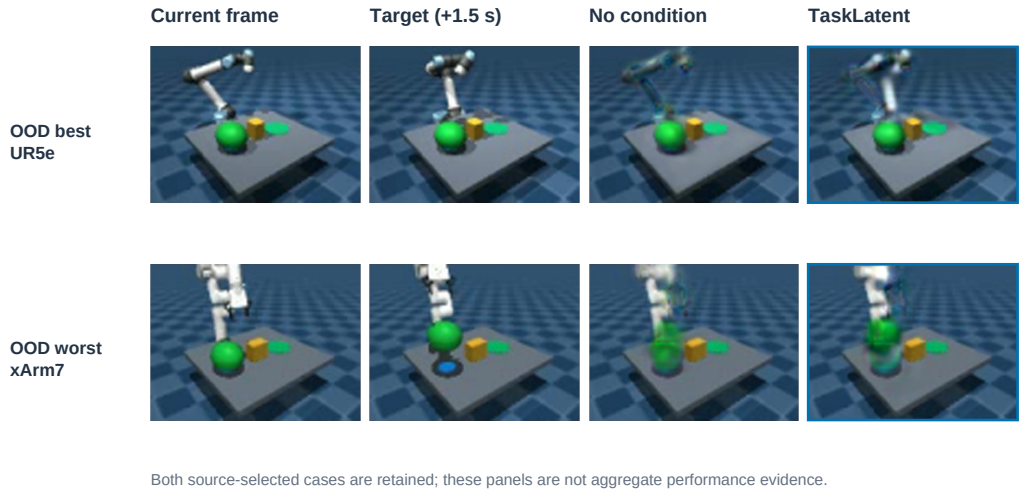

Figure 3: **Future-frame prediction on OOD examples.** Current frame, target after 1.5 seconds, unconditioned prediction, and TaskLatent prediction. The examples retain the original best-UR5e and worst-xArm7 selections.

5 DISCUSSION AND CONCLUSION

EgoTACO represents manipulation through object-state transitions, ordered events, and contact semantics. Independent object and robot conditions determine their physical realization, while verification links generated video to task and execution constraints. The controlled results separate task completion from human-path similarity and show complementary effects of geometry and physics adaptation. Conditional prediction further tests the representation's visual utility, while the fixed-ReKap main comparison evaluates its value as robot-training data.

AI USE STATEMENT

Generative AI tools assisted with restructuring and editing this manuscript, interpreting the supplied result descriptions, checking venue formatting requirements, and creating editable scientific diagrams from the supplied method description. Experimental images were reused from the original manuscript, and numerical results were provided by the authors. No new experiment was run as part of this manuscript revision. The authors remain responsible for the scientific claims and final submission. This statement describes assistance in preparing this revision and does not establish the AI-use history of the underlying research.

REPRODUCIBILITY STATEMENT

Section 3 defines the task representation, editable conditions, constraint sets, and generation objective. Section 4 describes the available comparisons, and Appendix B collects supplied protocol settings. The current record does not include executable training and evaluation code, model checkpoints, or complete per-run results. These materials are needed to reproduce the reported measurements and resolve the protocol discrepancies.

REFERENCES

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13778–13790, 2023.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023. URL https://arxiv.org/abs/2311.15127.

OX-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Vincent Cartillier, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Devansh Kukreja, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *IEEE transactions on pattern analysis and machine intelligence*, 2024.

Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *ICLR workshop on deep generative models for highly structured data*, 2022.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. URL https://arxiv.org/abs/2409.01652.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

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*, pp. 10138–10148, 2021.

Haoyun Li, Ivan Zhang, Runqi Ouyang, Xiaofeng Wang, Zheng Zhu, Zhiqin Yang, Zhentao Zhang, Boyuan Wang, Chaojun Ni, Wenkang Qin, Xinze Chen, Yun Ye, Guan Huang, Zhenbo Song, and Xingang Wang. Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. *arXiv preprint arXiv:2509.22199*, 2025. URL https://arxiv.org/abs/2509.22199.

Jia Luo. From passive video to editable experience: Physically grounded experience synthesis for embodied intelligence. *arXiv preprint arXiv:2607.26903*, 2026. URL https://arxiv.org/abs/2607.26903.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022. doi: 10.1109/WACV51458.2022.00323.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11078–11086, 2026. doi: 10.1609/AAAI.V40I13.38086.

Junjie Ye, Rong Xue, Basile Van Hoorick, Pavel Tokmakov, Muhammad Zubair Irshad, Yue Wang, and Vitor Guizilini. Anchordream: Repurposing video diffusion for embodiment-aware robot data synthesis. *arXiv preprint arXiv:2512.11797*, 2025. URL https://arxiv.org/abs/2512.11797.

Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10477–10486, 2023.

A MECHANISM AND TRAINING DIAGRAMS

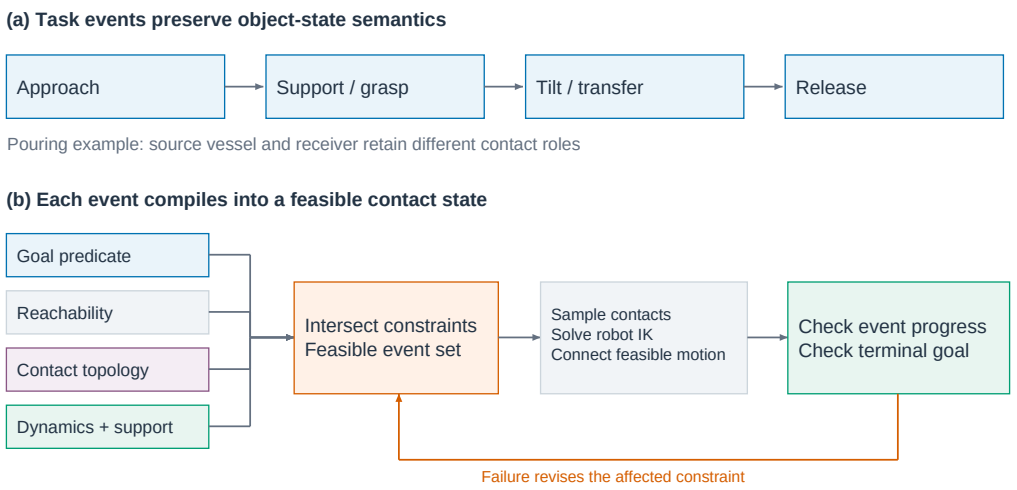

Figure 4: **Task-contact compilation.** The pouring event sequence illustrates task semantics, not a measured rollout. Goal, reach, contact, and physics constraints restrict admissible event states. Contacts are sampled before solving robot IK and connecting the resulting motion. Verification failures revise the relevant constraints.

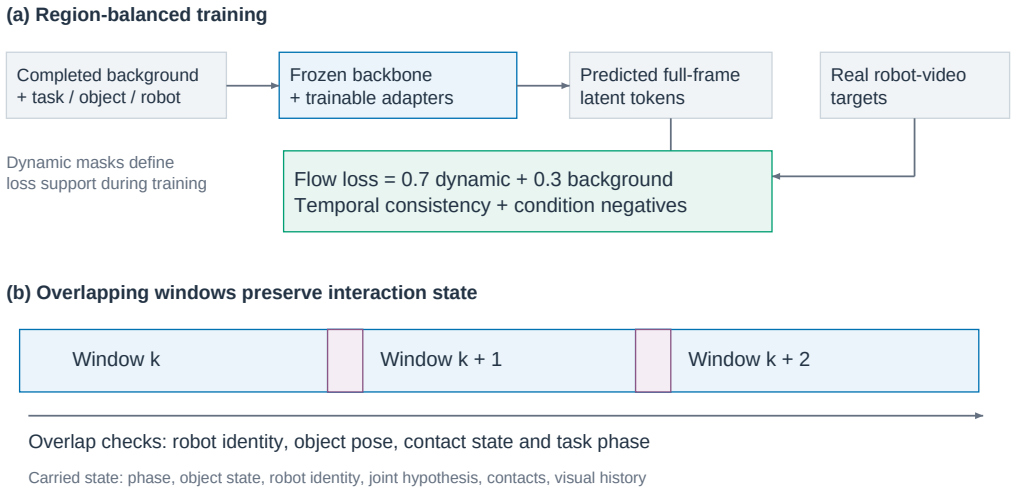

Figure 5: **Training and temporal state.** Dynamic support receives flow-loss weight 0.7, while the remaining scene receives 0.3. Temporal consistency and shuffled-condition negatives supplement this objective. Overlapping generation windows maintain robot identity, object pose, contacts, and task phase. The window lengths shown are schematic.

B PROTOCOL DETAILS AND EVIDENCE SCOPE

Table 6: Settings explicitly supplied in the source manuscript. The video protocols are listed separately because their relationship is not specified.

Component	Supplied setting
Cross-embodiment cases	20 tasks, three robot morphologies, 60 constructed cases
Controlled video split	36 tasks; 24 synthesis, six validation, six OOD; 8 fps
Broader video protocol	Approximately 100 source, 500 synthetic, and 50 held-out videos
Generation training	Frozen video backbone; trainable attention adapters; robot-removed background and real robot-video target pairs
Region-balanced objective	Dynamic support weight 0.7; remaining support weight 0.3
Temporal conditioning	Overlapping windows with phase, object state, identity, joint hypothesis, contacts, and visual history

The generalization table retains the 100-source protocol and the ablation table retains the 120-intervention annotation from the original tables. These entries are reported separately rather than pooled. The source results paragraph also mentions a 60-case object-edit subset; the relationship among these protocols remains to be specified with the completed experimental record.

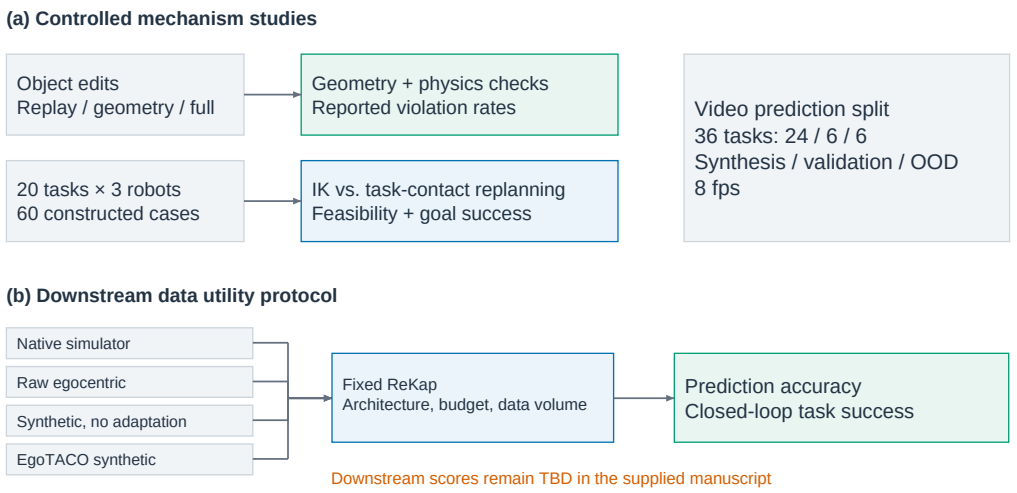

Figure 6: **Evaluation organization.** Controlled studies isolate contact replanning and geometry–physics adaptation. The separate downstream protocol changes the video training source under fixed ReKap settings. Its scores remain unreported.

C ADDITIONAL RESULT VISUALIZATION

The following visualization summarizes the numerical comparisons already presented in Tables 2 and 4 of the main text.

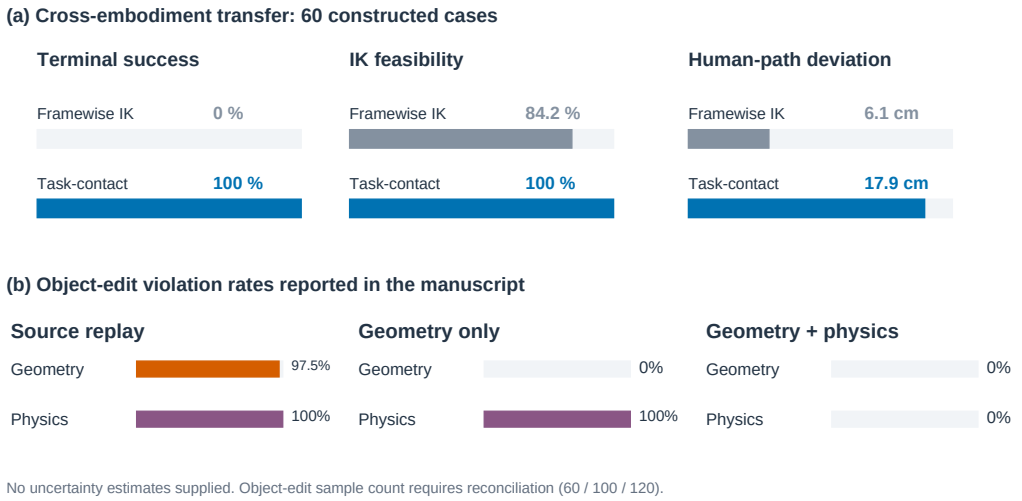

Figure 7: **Visualization of controlled results.** Bar lengths reproduce the percentages and trajectory deviations presented in the main-text tables.