Agentic Real2Sim: Physics-based World Modeling with Vision-Language Agents

Guanxiong Chen[1,*,#], **Qianjun Xia**[1,*], **Jiawei Peng**[2,*], **Heng Zhang**[1,*], **Bole Ma**[3,*], **Justin Qian**[1], **Ziyi Jiao**[1], **Bingyang Zhou**[6], **Luoxin Ye**[2], **Kaifeng Zhang**[4], **Kunyi Wang**[1], **Weijia Zeng**[1], **Yunuo Chen**[5], **Pengzhi Yang**[6], **Ziqiu Zeng**[6], **Huamin Wang**[7], **Chao Liu**[1], **Alan Yuille**[2], **Fan Shi**[6], **Changxi Zheng**[4], **Yunzhu Li**[4], **Chenfanfu Jiang**[5] and **Peter Yichen Chen**[1]

[1]University of British Columbia, [2]Johns Hopkins University, [3]Erlangen National High Performance Computing Center (NHR@FAU), Friedrich-Alexander-Universität Erlangen-Nürnberg, [4]Columbia University, [5]University of California, Los Angeles, [6]National University of Singapore, [7]Style3D

Real-to-sim conversion for robotic interaction with objects remains labor-intensive because it requires more than visual reconstruction: a streamlined real2sim process must recover scene geometries and object states, infer physical parameters, and assemble actors, objects, cameras, poses, and trajectories into a runnable physical simulation. Today this process still depends on manual tuning of visual foundation models, mesh cleanup, coordinate-frame alignment, and brittle workflow glue across visual perception tools and simulators. We introduce *Agentic Real2Sim*, a framework for generalized physical world modeling with vision-language agents, converting a real-world recording of object-robot interaction into a simulatable episodic twin which preserves observations, geometries, robot interactions, and object states. We evaluate Agentic Real2Sim on rigid-object manipulation, deformable-object interaction, and humanoid motion scenes, spanning domains that are usually handled by separate Real2Sim pipelines, marking a first step toward scalable conversion. The framework's agentic decisions can be driven by an open-weight VLM backend at a small fraction of the cost of frontier models, while attaining comparable conversion success rate. We aim to use the resulting real-world-aligned twins for downstream robotics tasks, specifically policy learning and evaluation. The project site is available at https://agentic-real2sim.github.io/.

1. Introduction

Large real-world robot datasets and increasingly capable simulators have made it tempting to treat real-to-sim conversion as a solved plumbing problem: collect observations, reconstruct assets, drop them into a physics engine, and train or evaluate policies. In practice, physical interaction episodes resist this simplification. A manipulation recording is not just a static scene: it contains a robot, manipulated objects, camera setup, object states, contacts, physical parameters, as well as a task description which dictates how the robot behaves across space and time. Accumulating these elements into a *simulatable* twin episode still requires a significant amount of manual labor: tuning visual foundation models, cleaning up meshes, aligning coordinate frames, and gluing together brittle workflows across visual perception tools and simulators.

Recent work has made strong progress on parts of this problem. Automated Real2Sim frameworks recover simulation-ready assets and physical parameters from robotic interaction [15], while Real2Sim2Real frameworks align visual and dynamic properties for manipulation policy development [4, 5, 9]. At the same time, state-of-the-art VLM-based agentic data-generation frameworks use the generate-critic-improve paradigm to build simulation-ready scenes [14, 16, 20, 26] or object-level assets [25]. These lines of work point to a missing capability: scalable conversion of recorded physical interaction episodes into physically simulatable digital twins. Given a real-world recording of an

[*] Equal technical contribution. [#] Project lead.

interaction, can a framework produce a simulatable artifact that replays the episode, with physically plausible scene setup and physical parameters that support downstream applications such as robotic policy learning?

We frame this problem as *Agentic Real2Sim*: Given a recorded physical interaction episode, our goal is to establish a framework that automatically produces a physically realistic digital twin, preserving the actors, objects, trajectories, physical parameters, and replay artifacts needed for downstream robotic learning and evaluation—see Fig. 1. Our contributions are threefold: (1) First, we introduce an Agentic Real2Sim pipeline for converting DROID robot manipulation episodes [12] into MuJoCo-simulated episodic twins [18]. We view this as a first step toward scalable interaction conversion: the system transforms recorded robot-object interactions into simulatable artifacts through visual processing, physical-prior inference, scene preparation, and simulator-in-the-loop refinement. (2) Second, we demonstrate that Agentic Real2Sim supports interchangeable VLM backends. Across four backends evaluated on DROID-100, an open 31B VLM achieves comparable observed replay-success outcomes to proprietary backends while reducing model cost by up to 31.4×. (3) Third, we generalize the framework beyond the DROID manipulation dataset, by enabling the same conversion process for deformable manipulation in a PhysTwin-style setting [9] and for humanoid motion with BFM-Zero-style motion context [13].

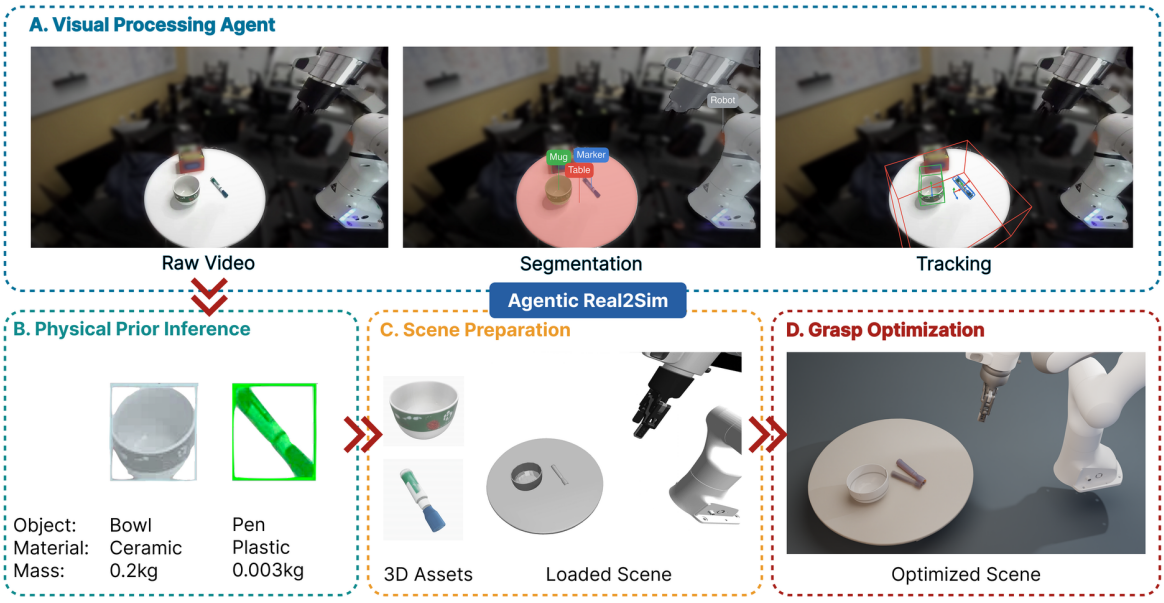

Figure 1 | **Agentic Real2Sim architecture.** A recorded DROID episode is converted into a simulatable digital twin through four linked agents: The visual processing agent from raw video extracts segmentation and depth masks, geometries and object poses. The physical-prior inference agent attaches object identity, material class, mass hints, and other constitutive properties needed by the simulator. The scene preparation agent initializes a scene with the robot's initial state, cameras, and reconstructed geometry, and places objects of interest in the correct locations. Simulator-in-the-loop grasp optimization further optimizes object placements to enable successful grasping. The same episode contract supplies the inputs and feedback channels used by the rigid, deformable, and humanoid adapters.

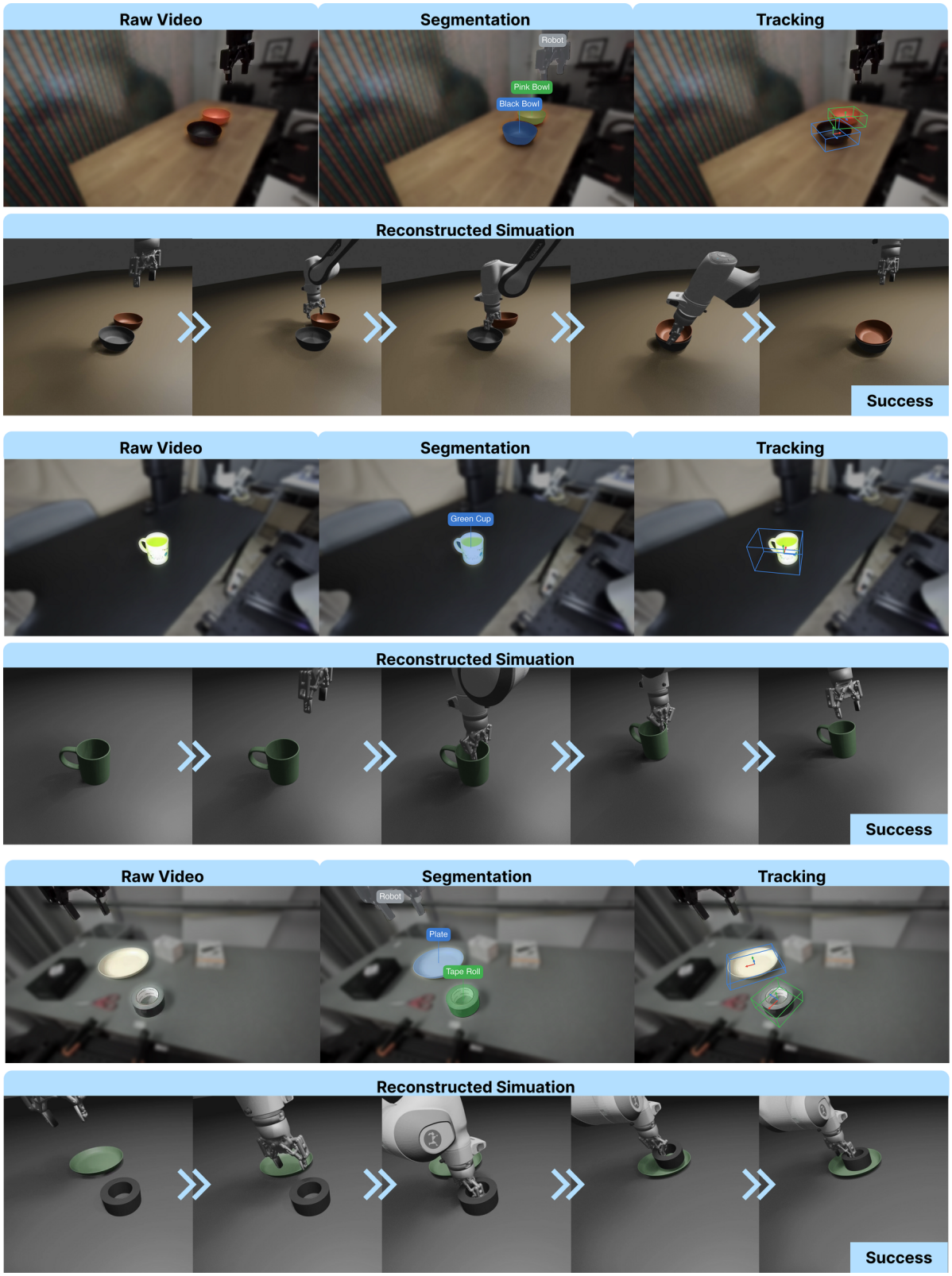

Figure 2 | **DROID episode conversion at scale.** Representative episodes from **DROID-100** spanning the sampled axes, each pairing the real DROID observation with its MuJoCo episode twin. Intermediate pipeline artifacts are exposed for selected scenes: object segmentation and FoundationPose pose-tracking overlays.

2. Related Work

Automated real-to-sim alignment. Real2Sim frameworks have been gradually shifting from manually authored simulator scenes toward automated asset and parameter recovery: scalable Real2Sim reconstructs simulation-ready object assets through robotic pick-and-place interaction, estimating visual geometry, collision geometry, and inertial properties with a dedicated acquisition setup [15]. TwinAligner similarly targets physics-aware Real2Sim2Real for robotic manipulation, combining visual alignment with rigid-body system identification and evaluating whether aligned simulators improve policy development [5]. These works establish that reconstructing photorealistic scenes alone is insufficient for robotic learning: simulators used for manipulation must also recover dynamics, contacts, and task-relevant behavior. Agentic Real2Sim shares this emphasis on physical alignment, but differs in its unit of conversion. Rather than scanning isolated objects or constructing a scene for a new policy-training setup, it converts recorded interaction episodes into simulatable twin episodes that preserve the observed actors, manipulated objects, trajectories, contacts, and ensures visual alignment.

VLM-driven asset generation. Recent works leverage state-of-the-art visual-language models for generating visually realistic and physically plausible assets for downstream applications. While earlier frameworks primarily rely on one-off VLM API calls embedded in larger scene-reconstruction and task-generation pipelines [11, 17, 23, 26], more recent works such as SceneSmith [16], SceneWeaver [24], PhyScensis [20], SimWorld Studio [10], and LychSim [14] have shown the strong potential of agentic VLM pipelines in reconstructing realistic scenes from real-world data, or building articulated assets through agentic code and testing loops [25]. These agentic frameworks enable VLMs to operate directly over intermediate generation artifacts, using structured, artifact-level feedback to identify omissions or implausibilities and to iteratively refine subsequent generation decisions. Our work follows this paradigm of using VLM-driven agents for synthetic scene generation and real-to-sim construction, yet places greater emphasis on reconstructing full manipulation behaviors from real-world recordings, as well as generating physical parameters, in addition to placing assets appropriately to recreate visually realistic scenes.

Robotic interaction datasets. DROID provides large-scale in-the-wild robot demonstrations with synchronized RGB streams, depth, calibration, language instructions, and robot trajectories, with a primary focus on rigid-object manipulation [12]. PointWorld builds on this style of in-the-wild manipulation data for scalable 3D world modeling and further refines camera extrinsic estimates as part of its annotation pipeline [8]. These datasets make DROID-style manipulation a natural substrate for episode-level Real2Sim, while also exposing the practical difficulty of turning raw demonstrations into physically and visually realistic simulated demonstrations. Beyond rigid object manipulation, PhysTwin emphasizes deformable manipulation by reconstructing physics-informed deformable digital twins from videos of interacted objects [9]. For humanoid motion, BFM-Zero studies promptable behavioral foundation models and motion-context retrieval for humanoid control [13]. Together, these lines motivate a conversion problem that spans rigid robot-object interaction, deformable-object interaction, and whole-body motion rather than a single dataset format.

Visual processing and simulation tools. Our pipeline also relies on replaceable visual and simulation components. SAM 3 supplies open-vocabulary segmentation for selecting and tracking relevant objects or regions [2]. SAM 3D is used as a geometry-extraction component, not as the object-discovery mechanism, by recovering 3D object assets from image evidence [3]. FoundationStereo provides stereo depth estimation from rectified camera pairs [22], and FoundationPose provides 6D

object pose estimation and tracking for novel objects [21]. Deformable simulation tools such as PhysTwin and EMPM further show how visual evidence can be coupled with simulator rollouts and parameter optimization to recover material or connectivity parameters [4, 9]. These components are important building blocks, but they are not the paper's main contribution. Agentic Real2Sim contributes the episode contract and alignment loop that compose visual extraction, simulator execution, critic feedback, and domain-specific repair into executable episode twins.

Downstream use of simulator episodes. Simulation-ready episodes are useful only if they support physical queries beyond visual reconstruction. Prior work on demonstration amplification and simulation data generation studies how converted or synthesized demonstrations can support skill learning and deployment [6, 19]. We aim to use our converted synthetic DROID episodes as simulation assets for downstream robotics tasks, specifically policy learning and evaluation.

3. Method

Agentic Real2Sim converts a real physical interaction episode into a simulatable *episode twin*: a simulator artifact that preserves the observation streams, actors, manipulated entities, geometry, temporal state, physical parameters, and evaluation evidence needed to replay or query the original episode. An episode twin is represented as

$$\mathcal{T} = (\mathcal{O}, \mathcal{A}, \mathcal{G}, \mathcal{S}_{1:T}, \Theta, \mathcal{B}, \mathcal{M}), \tag{1}$$

where $\mathcal{O}$ denotes real observations, $\mathcal{A}$ actors or end effectors, $\mathcal{G}$ geometry and appearance assets, $\mathcal{S}_{1:T}$ simulator states over time, Θ physical and alignment parameters, $\mathcal{B}$ the simulator backend, and $\mathcal{M}$ the metrics and traces used to decide whether conversion succeeded. The design goal is to share the pipeline stages, artifact contract, replay loop, and critic interface across domains while allowing skill and tool invocations to differ between rigid-object manipulation, deformable-object interaction, and whole-body humanoid motion.

3.1. System Overview

Fig. 1 summarizes the architecture through a rigid object manipulation example. The *visual processing agent* reads a recorded demonstration, discovers relevant objects, selects keyframes, segments actors and objects, reconstructs or imports geometry, estimates scale, tracks poses, and emits a canonical episode folder. The *physical-prior inference agent* converts visual evidence and task context into simulator-facing priors such as object identity, material class, mass hints, and contact-relevant attributes. The *scene preparation agent* converts those artifacts into an initialized simulator scene by aligning robot base pose, object pose, camera frame, assets, trajectories, and ground reference. The final *simulator-in-the-loop grasp optimization stage* optimizes the position of the manipulated object and identifies the optimal object placement which enables successful grasping.

The architecture deliberately separates deterministic tools from agentic decisions. Agentic nodes make choices that depend on ambiguous visual or physical evidence: which objects matter, which frame provides a clean mask, whether a segmentation or track is acceptable, which object defines the ground plane, which episode subtype should be routed to which adapter, and which feedback channel should drive the next refinement. Deterministic tools invoked by agents perform extraction, rendering, pose optimization and grasp optimization. This separation makes the system easier to ablate: removing a critic or replacing a VLM backend changes the decision layer without rewriting the underlying visual processing and simulation utilities.

3.2. Converting DROID Episodes

Our framework primarily focuses on DROID-style robot-object interactions [12]. The input contains synchronized camera streams, calibration data, a robot trajectory, depth or stereo-derived geometry, and language or task context. The visual pipeline first selects the primary external camera from the synchronized streams, then extracts frames from its recording and builds a rectified video. A VLM-driven object discovery node proposes scene entities, after which a keyframe selector chooses a frame for segmentation. A mask critic can reject poor segmentations and route the graph back to keyframe selection with a bounded retry budget. Accepted masks feed mesh recovery, while full-image depth reconstruction [22] further enables mesh scaling and FoundationPose-style object tracking [21]. A tracking critic records pose-quality judgments and can request a new initialization frame. Finally, two VLM queries identify the object the robot manipulates and the scene entity that serves as the ground reference.

The resolved visual state is written into an episode folder containing object meshes, scales, pose tracks, robot trajectory, camera metadata, first-frame observations, masks, and task semantics. Next, the scene preparation agent proceeds: it invokes a deterministic calibration stage, optimizes the robot base pose against the robot mask, and estimates the ground plane's orientation and offset beneath the ground reference chosen upstream. The agent then loads the calibrated scene into MuJoCo [18], and makes a decision: either it performs a deterministic sweep over object shift parameters, or it calls an LLM-assisted loop that proposes refinements from replay evidence. The sweep path evaluates a batch of candidate shifts against contact and grasp outcomes; the loop path exposes rendered keyframes and structured replay summaries to the agent before each refinement.

Tab. 1 lists the tools exposed to each stage in our conversion process, separating the tools an agent uses to interact with and build artifacts from the skills that describe the high-level decision-making subprocesses an agent must follow within each stage.

3.3. Converting PhysTwin and BFM-Zero Episodes

The deformable and humanoid adapters extend the same episode contract beyond rigid DROID manipulation. The deformable adapter follows the PhysTwin/EMPM setting, replacing rigid object pose tracking with tracked geometry, point, particle, spring, or material state and using simulator rollouts to check whether recovered parameters reproduce observed deformation [4, 9]. The humanoid adapter replaces object-centric scene preparation with motion-context retrieval, embodiment-specific initialization, and closed-loop replay against retargeted reference motion, using motion priors from the humanoid/video domain [7]. Both adapters reuse the shared folder contract, visual evidence, structured replay metrics, and repair interface, but expose domain-specific states and refinement variables instead of forcing all episodes into a single rigid-body representation.

4. Experiments

4.1. Evaluation Metric

We evaluate the rigid DROID conversions with replay success, an episode-level metric that measures consistency between the simulated episode and the real episode. Specifically, following paradigm adopted in Cao et al. [1], we compute the metric following a structured, VLM-based process: a evaluator first prepares a fixed set of real and simulated keyframes with the labels `start`, `middle_1`, `middle_2`, and `end`. Next, the evaluator selects reconstruction candidates deterministically from the latest grasp sweep: a sample is eligible only if the probe marks it as grasped, its replay video exists, and its lift and displacement statistics are finite and within broad sanity bounds. Among the

Block	**Tool**	**Skill**	**Purpose**
Object discovery	—	`object_discovery`, `object_relevance`, `pickup_object`, `support_tree`	List the objects in the scene, keep the ones the task involves, pick out the one being manipulated, and infer what rests on what.
Segmentation	`segment` *(SAM 3)*	`segment`: keyframe + mask critic	Pick a clean frame, segment each object, and retry with a new frame if a mask is rejected.
Geometry	`mesh_recover` *(SAM 3D)*, `mesh_scale`	`scale_critic`	Reconstruct each object's mesh and set its real-world size; the critic checks the size is stable.
Pose tracking	`pose_tracking` *(FoundationPose)*	`tracking_critic`	Track each object's 6-DoF pose over time; the critic checks the track and can restart it from a new frame.
Physical prior	—	`material_classify`	Infer each object's material and mass so the simulator has physical parameters to use.
Scene prep	`calibration`, base-pose opt.	`camera_select`, `ground_ref`	Pick the primary camera view (done once at data preprocessing), calibrate the cameras, place the ground and robot base, and load the scene into MuJoCo.
Grasp opt.	grasp sweep	—	Try many small shifts of the object and keep the one that leads to a successful grasp.

Table 1 | **Tools and skills exposed to each conversion stage.** Each stage wraps fixed perception and simulation backends as *tools* (backend in *italics*), and defers high-level decisions to *skills*: bounded, schema-constrained VLM queries. Monospace names are the exact invoked tool functions; a dash (—) means the stage has no tool or skill to be invoked.

eligible samples, the evaluator sends at most five candidates to the judges, ranked by peak object displacement with sample id as the deterministic tie-breaker. Three judges, each with a distinct VLM backend, then independently compare the episode's real keyframes with each candidate's simulated keyframes, using the same structured schema; each judge scores the quality of a candidate based on target-object identity, the target object's final location, action similarity, and end-gripper location, while recording start-pose drift as context. The rubric is defined as follows: a judge scores each candidate out of 10, subtracting up to 4 points for identifying the wrong target object, 3 for a wrong final object location, 2 for a wrong action performed, and 1 for a wrong final gripper location. Starting-pose drift is recorded as context rather than directly subtracted; scores of 8 or higher count as pass, 7 as partial, and 6 or lower as fail. Each judge acts as a success finder by nominating its own highest-scored candidate. We define $r_e \in \{0, 1\}$ as the episode replay-success indicator, with $r_e = 1$ if at least one of the three judges assigns a best-candidate score of at least 8 out of 10; otherwise $r_e = 0$. Episodes without a valid judge record are also assigned $r_e = 0$. The reported replay score for the episode is the maximum of the three judge-best scores.

4.2. DROID Episode Conversion at Scale

We evaluate the rigid object manipulation setting on **DROID-100**, a randomly selected set of 100 manipulation episodes sampled to span objects, camera viewpoints, occlusion patterns, and manipulation verbs (pick / place / push / insert). All sampled episodes remain in the replay-success denominator, including runs that stop before producing a valid replay record.

Quantitative. Using Gemma 4 31B as the VLM backend, Agentic Real2Sim produces 48 successful, 8 partial, and 44 failed replay outcomes over DROID-100, with a total model bill of $2.62. Fig. 3 reports the corresponding outcomes for three additional VLM backends and compares their model costs.

Qualitative. Fig. 2 pairs real DROID observations with their MuJoCo replays for representative episodes spanning the sampled axes, exposing intermediate pipeline artifacts for at least one scene: recovered object meshes, FoundationPose-style pose-tracking overlays, and the calibrated robot/ground state. We include at least one informative failure (e.g. a segmentation or pose-tracking miss) so the panel is honest rather than cherry-picked.

4.3. Multi-VLM Support and Cost Efficiency

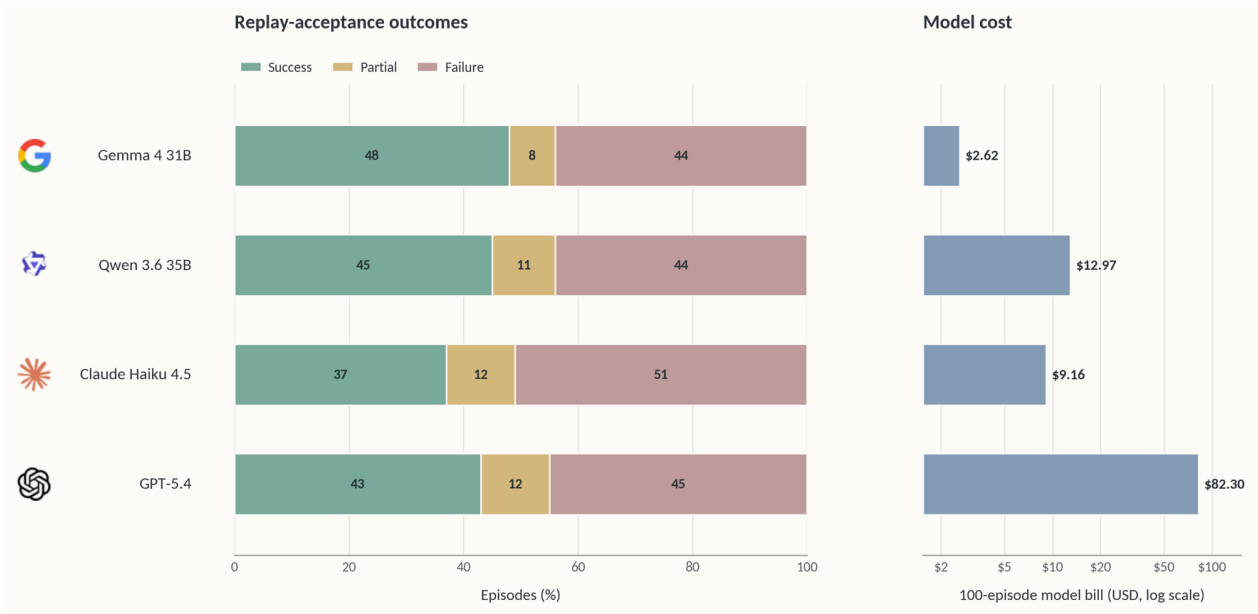

Figure 3 | **Replay-acceptance outcomes and model cost across VLM backends.** The left panel decomposes all 100 attempted DROID episodes into replay success, partial, and failure outcomes for each backend. Replay success denotes a best judge score of at least 8/10, partial a score of 7/10, and failure a score of 6/10 or lower or the absence of a valid judge record. The right panel reports the corresponding 100-episode model bill on a logarithmic scale. Observed replay-success counts range from 37 to 48 episodes, while model bills range from 2.62 to 82.30 USD.

Across all four backends, we use the same pipeline configuration, with reasoning enabled for selected agentic steps, and vary only the VLM. The reported model bills include all model usage under this configuration. Fig. 3 shows that the four backends have broadly similar observed outcome compositions: Gemma 4 31B, Qwen 3.6 35B, GPT-5.4, and Claude Haiku 4.5 obtain 48/100, 45/100,

43/100, and 37/100 replay successes, respectively. The clearer separation is cost: relative to Gemma's 2.62 model bill, Claude costs 3.5× as much, Qwen 5.0×, and GPT-5.4 31.4×.

We attribute the similar outcomes across backends to how the pipeline scopes the VLM's role. The VLM does not perform geometry or physics itself: it orchestrates deterministic specialist components for segmentation, stereo depth, and pose tracking, together with a deterministic grasp sweep, and is queried only for bounded, schema-constrained decisions such as object discovery, keyframe and mask selection under a capped retry budget, and replay-refinement choices. Because these queries are narrowly scoped, they lie well within the capability of modern reasoning VLMs, and the pipeline transfers across backends without per-model tuning: even the 31B open model attains the highest observed success count while using roughly 3% of GPT-5.4's model bill. Backend choice therefore reduces largely to cost, making open-weight models a practical default for large conversion batches. That absolute replay success stays below 50% for every backend further suggests that the remaining headroom lies mostly in the upstream visual and simulation components rather than in the choice of VLM alone.

4.4. Deformable and Humanoid Episode Conversion

We use the deformable and humanoid settings as qualitative stress tests for the same conversion contract under non-rigid dynamics and closed-loop humanoid control. These domains are presented through representative visual comparisons rather than aggregate scores.

Deformable. These are PhysTwin-style episodes with assets such as rope, cloth, plush objects, soft packages, and elastoplastic materials. We compare the real interaction video with the recovered geometry and simulated deformation, and report representative real-sim matches together with failure cases.

Humanoid. These are Unitree G1 clips of locomotion and short whole-body motions, retargeted from LAFAN1. We compare the simulated humanoid against the reference motion in closed-loop simulation and report qualitative side-by-side motion comparisons.

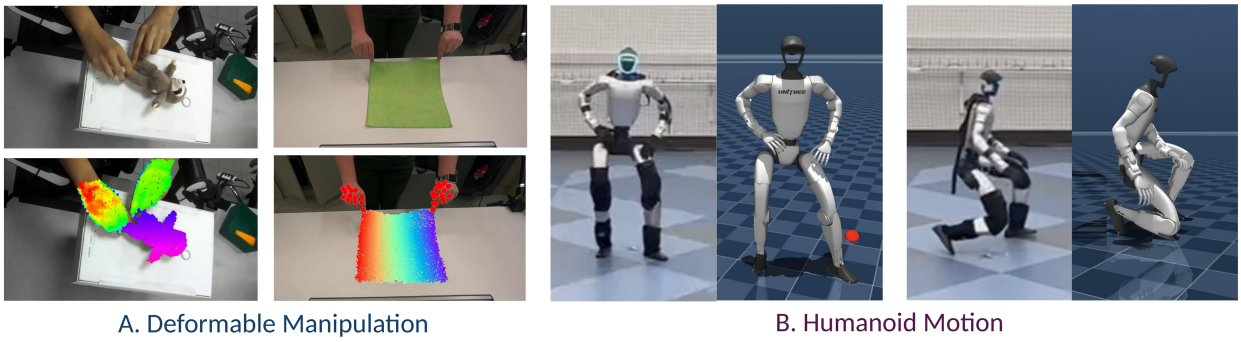

Figure 4 | **Deformable and humanoid episode conversion.** Two non-rigid object manipulation settings that stress the same conversion contract beyond rigid manipulation. *A:* PhysTwin-style deformable episodes pair real interaction video with the recovered geometry and a deformation overlay. *B:* Unitree G1 humanoid clips of locomotion and short whole-body motions, retargeted from LAFAN1, where the simulated humanoid is shown against the real frames after the controller's joint-level gains are tuned to match the reference trajectory in closed-loop simulation.

Qualitative. Fig. 4 shows both domains. In Fig. 4 (a), deformable cases pair real interaction video with tracked geometry and a deformation overlay, emphasizing whether the recovered material response follows the visible bending, stretching, and contact sequence. In Fig. 4 (b), we show a simulated humanoid against its reference motion for standing, kneeling, and short-locomotion clips, emphasizing gross pose, balance, and motion-phase agreement.

5. Conclusions, Limitations and Future Work

We introduced *Agentic Real2Sim*, a framework for converting real-world physical interaction recordings into simulatable digital twins. The framework must preserve the actors, manipulated entities, geometry, physical parameters, camera setup and objects along with end effector trajectories needed for simulation. For the task of converting DROID episodes, Agentic Real2Sim converts robot-object manipulation recordings into MuJoCo episode twins through visual processing, physical-prior inference, scene preparation, and simulator-in-the-loop refinement. The same episode contract also provides a common interface for PhysTwin-style deformable manipulation and BFM-Zero-style humanoid motion, allowing the pipeline to span settings that are usually handled by separate Real2Sim systems. Also, our framework decouples agentic decisions from any single VLM provider: on DROID-100, an open 31B backend attains comparable observed replay-success outcomes to proprietary alternatives while using only about 3% of GPT-5.4's model cost. Finally, we aim to use the aligned episode twins as simulator assets for downstream robotics tasks, specifically policy learning and evaluation.

Current limitations are the focus on rigid-object DROID episodes and the pipeline's sensitivity to errors in upstream perception and simulator feedback. Future work will extend automated conversion to deformable episodes and systematically evaluate agentic components and VLM backends for reliability, replay stability, and enable policy learning and evaluation for the robotics community.

Acknowledgments

This work was supported in part by the NVIDIA Academic Grant Program. We thank NVIDIA for supporting academic research and for providing resources that helped facilitate this work. We thank Yixian Cheng for episode rendering and video production. We also thank Beichen Li of OpenAI for support with computational resources.

References

[1] Ziang Cao, Yinghao Liu, Haitian Li, Runmao Yao, Fangzhou Hong, Zhaoxi Chen, Liang Pan, and Ziwei Liu. Physx-omni: Unified simulation-ready physical 3d generation for rigid, deformable, and articulated objects. *arXiv preprint arXiv:2605.21572*, 2026.

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[3] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[4] Yunuo Chen, Yafei Hu, Lingfeng Sun, Tushar Kusnur, Laura Herlant, and Chenfanfu Jiang. Empm: Embodied mpm for modeling and simulation of deformable objects. *IEEE Robotics and Automation Letters*, 11(4):4179–4186, 2026.

[5] Hongwei Fan, Hang Dai, Jiyao Zhang, Jinzhou Li, Qiyang Yan, Yujie Zhao, Mingju Gao, Jinghang Wu, Hao Tang, and Hao Dong. Twinaligner: Visual-dynamic alignment empowers physics-aware real2sim2real for robotic manipulation. *arXiv preprint arXiv:2512.19390*, 2025.

[6] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. In *8th Annual Conference on Robot Learning*, 2024.

[7] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics*, 39(4), 2020.

[8] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

[9] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 7219–7230. IEEE, 2025.

[10] Haoqiang Kang, Xiaokang Ye, Yuhan Liu, Siddhant Hitesh Mantri, Lingjun Mao, James Fleming, Drishti Regmi, and Lianhui Qin. Simworld studio: Automatic environment generation with evolving coding agent for embodied agent learning. *arXiv preprint arXiv:2605.09423*, 2026.

[11] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, page 6672–6679. IEEE, 2024.

[12] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Srirama, Lawrence Chen, Kirsty Ellis, Peter Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Ma, Patrick Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Walke, Blake Wulfe, Ted Xiao, Jonathan Yang, Arefeh Yavary, Tony Zhao, Christopher Agia, Rohan Baijal, Mateo Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Foster, Jensen Gao, David Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, 2024.

[13] Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning, 2025.

[14] Wufei Ma, Chloe Wang, Siyi Chen, Jiawei Peng, Patrick Li, and Alan Yuille. Lychsim: A controllable and interactive simulation framework for vision research. *arXiv preprint arXiv:2605.12449*, 2026.

[15] Nicholas Pfaff, Evelyn Fu, Jeremy Binagia, Phillip Isola, and Russ Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 6296–6303. IEEE, 2025.

[16] Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. *arXiv preprint arXiv:2602.09153*, 2026.

[17] Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, et al. Simfoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

[18] 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*, page 5026–5033. IEEE, 2012.

[19] Weikang Wan, Jiawei Fu, Xiaodi Yuan, Yifeng Zhu, and Hao Su. Lodestar: Long-horizon dexterity via synthetic data augmentation from human demonstrations. In *9th Annual Conference on Robot Learning*, 2025.

[20] Yian Wang, Han Yang, Minghao Guo, Xiaowen Qiu, Tsun-Hsuan Wang, Wojciech Matusik, Joshua B Tenenbaum, and Chuang Gan. Physcensis: Physics-augmented llm agents for complex physical scene arrangement. *arXiv preprint arXiv:2602.14968*, 2026.

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

[22] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. *CVPR*, 2025.

[23] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli Vanderbilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 16277–16287. IEEE, 2024.

[24] Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. *Advances in neural information processing systems*, 38: 140319–140351, 2026.

[25] Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation. *arXiv preprint arXiv:2605.15187*, 2026.

[26] Alex Zook, Fan-Yun Sun, Josef Spjut, Valts Blukis, Stan Birchfield, and Jonathan Tremblay. Grs: Generating robotic simulation tasks from real-world images. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, page 594–603. IEEE, 2025.