UMI for Excavators

Michael Yu
Stanford University
yumich@stanford.edu

Harsh Singh
Stanford University
singhh@stanford.edu

Nicole Ma
Stanford University
manicole@stanford.edu

Abstract: Teaching excavators to scoop typically requires collecting demonstrations on the machines themselves, making robot learning dependent on expensive equipment access and skilled operators. We present UMI for Excavators, a framework that instead collects human scooping demonstrations using a handheld bucket, without operating the target robot. The central challenge is translating freely performed bucket motions into actions executable by an excavator's constrained four-joint mechanism. To bridge this gap, we develop a camera–IMU-equipped bucket with visual–inertial motion reconstruction, a scan-derived MuJoCo digital twin for kinematically constrained retargeting, and trajectory retiming that preserves visual–action correspondence. We incorporate twin-rendered CLOAK masking to bridge the visual gap between demonstration collection and deployment. Fine-tuning $\pi_{0.5}$ on the resulting demonstrations reduces held-out-session action prediction error by 51% relative to the pretrained model. On a physical tabletop excavator, our policies achieve 50% scooping success without masking and 60% with CLOAK, compared to 0% for the pretrained model.

Keywords: UMI, excavation, learning from demonstration, visual-inertial odometry, vision-language-action models

1 Introduction

How can we teach excavators to scoop without operating an excavator for every demonstration? Excavators are central to construction, mining, and infrastructure development, yet collecting demonstrations for learning-based control requires access to a machine, a suitable worksite, and an operator skilled at coordinating its articulated controls [15–20].

The Universal Manipulation Interface (UMI) offers an alternative: collecting handheld demonstrations independently of the target robot [1, 7, 9, 10]. Although UMI has been extended to legged and aerial robots, dexterous hands, and humanoids [5, 6, 8, 11], transferring freehand scooping to an excavator introduces distinct challenges. A person can move a bucket freely, whereas an excavator must coordinate four coupled joints—slew, boom, stick, and bucket—to reproduce its path and cutting angle within workspace and speed limits. Low-texture soil and occlusions further complicate motion reconstruction. Transferring these demonstrations therefore requires converting them into physically feasible robot actions aligned with the original observations.

We present UMI for Excavators (Figure 1), a handheld demonstration interface and policy learning pipeline that bridges these gaps. Our system combines a camera–IMU-equipped bucket with visual–inertial motion reconstruction, a scan-derived digital twin for constrained retargeting, and trajectory retiming that preserves image–action correspondence. We additionally investigate twin-rendered CLOAK masking to bridge the visual gap between collection and deployment. Fine-tuning $\pi_{0.5}$ on these demonstrations enables physical scooping on a tabletop excavator, achieving up to 60% success compared to 0% for the pretrained model.

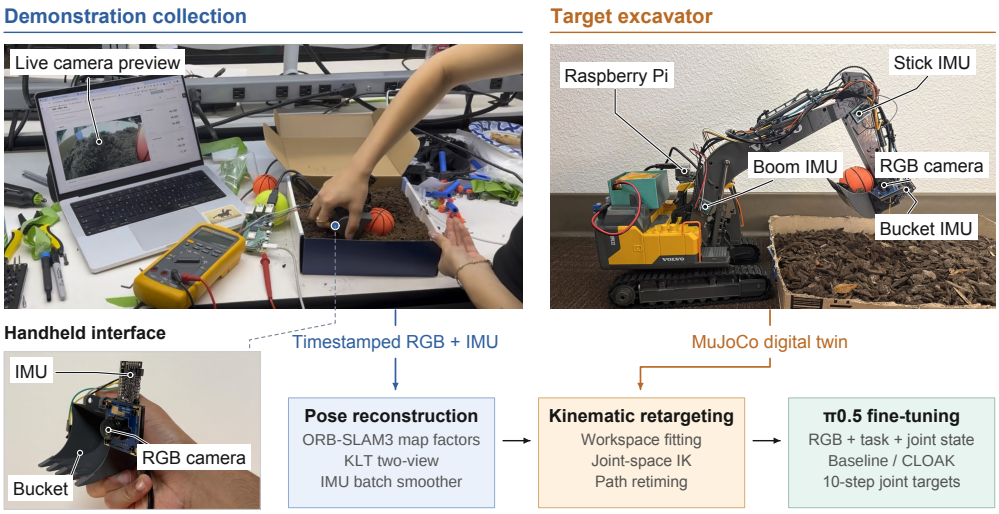


Figure 1: From handheld collection to supervision: the bucket camera and IMU record a demonstration, a batch smoother reconstructs motion, a scan-derived MuJoCo twin fits it to the workspace and retimes it, and the exported RGB frames, joint states, and future targets supervise $\pi_{0.5}$; CLOAK masking (Figure 3) adapts the observation input.

2 Method

2.1 Reconstructing bucket motion

A Raspberry Pi records bucket-camera images and inertial measurements. A batch smoother (Figure 2) reconstructs motion by combining ORB-SLAM3 [3] monocular–inertial map factors, KLT two-view constraints [23–25], and IMU preintegration [26]. Appendix A gives reconstruction details and diagnostics.

2.2 Fitting motion to the excavator

An articulated MuJoCo twin [21] is built using iPhone photogrammetry, LiDAR scanning, 3D Gaussian splatting, and a motion video; measured joint travel and speeds define a calibrated profile (Appendix A). Retargeting over slew, boom, stick, and bucket fits a scale, yaw, and translation of the reconstructed path to the workspace, then inverse kinematics matches tooth-tip position and bucket pitch within joint limits.

Following the bucket-tip path alone is insufficient: bucket pitch near soil contact determines the cutting angle of the scoop. We use proximity to the reconstructed path's lowest point as a contact proxy, prioritizing demonstrated pitch there and clamping it to feasible pitch limits elsewhere while enforcing joint limits throughout.

2.3 Preserving image–action correspondence

Slowing the trajectory without synchronizing its images would misalign observations and supervised actions. Reachability-based retiming [22] slows the joint curve to satisfy direction-dependent speed and acceleration limits, starting and ending at rest. A monotone source-time map associates each execution-time state with the latest corresponding camera frame, so slowing the robot also slows image progression. The aligned data contain 10 Hz RGB observations, reconstructed four-joint states, and one-second chunks of ten future targets.

CLOAK observations. CLOAK-inspired masking hides the target excavator in robot-camera observations (Figure 3, left) [14]. For training, we project the twin silhouette at each reconstructed joint state onto the handheld image and exclude tokens with a strict majority of masked pixels

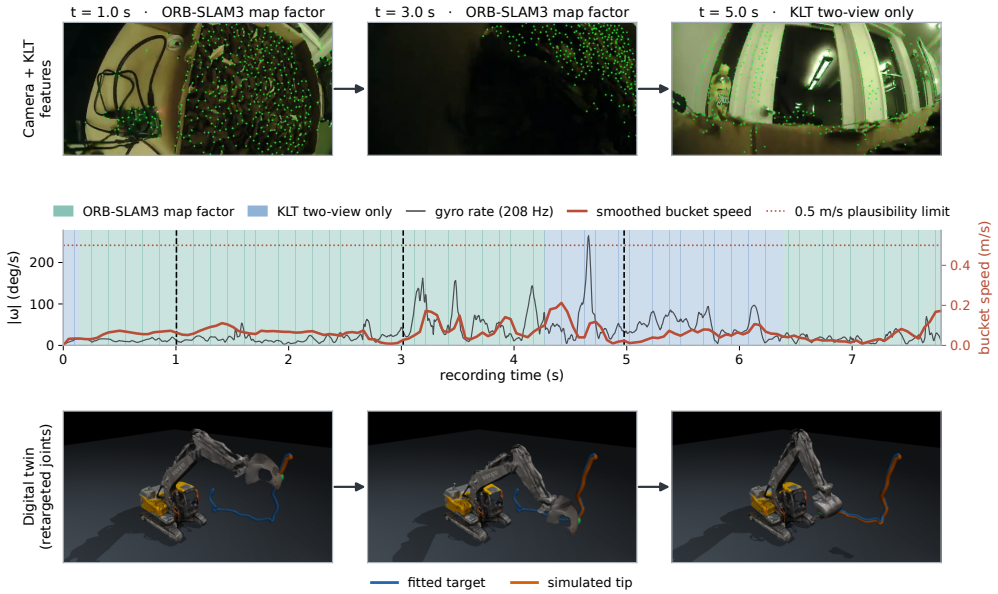


Figure 2: One dirt demonstration: KLT camera tracks (top); gyroscope rate, bucket speed, visual-factor type, and plausibility threshold (middle); retargeted motion in the calibrated twin (bottom).

(Figure 3, right). Since the excavator is absent from these recordings, this removes scene context where the robot will appear at deployment.

3 Experiments and Results

3.1 Dataset and evaluation

We fine-tune the pretrained $\pi_{0.5}$ base [4, 12, 13] on 42 reviewed dirt demonstrations for 1,000 steps, using state-relative actions and training-set normalization. Evaluation uses 42 demonstrations from an earlier, held-out session (504 one-second windows), with normalized joint-action MAE averaged equally over episodes. Appendix C gives the screening procedure and additional evaluation details.

$\pi_{0.5}$ receives bucket-camera RGB, four-joint state, and the instruction "scoop the dirt." It predicts ten state-relative joint targets sampled at 10 Hz, converted to absolute angles at inference. The runner estimates boom, stick, and bucket angles using calibrated link IMUs with pulse-based dead reckoning as fallback; slew stays fixed. Clipped, speed-limited targets become timed ESP32 motor pulses. Each cycle skips targets made stale by inference latency, executes the remaining chunk, and requests new actions from an updated observation.

Policy	Held-out action MAE ↓	Scooping success ↑
Pretrained $\pi_{0.5}$ base	0.529	0% (0/20)
Fine-tuned, no mask	0.258	50% (10/20)
Fine-tuned, CLOAK	0.280	60% (12/20)

Table 1: Held-out-session action prediction and physical scooping. Fine-tuned policies use step-1,000 checkpoints; each policy is evaluated in 20 physical attempts.

3.2 Policy learning and physical execution

Fine-tuning $\pi_{0.5}$ on the reconstructed demonstrations reduces held-out-session action error by 51%, from 0.529 to 0.258 (Table 1). The CLOAK-trained policy reaches 0.280, a 47% reduction relative to

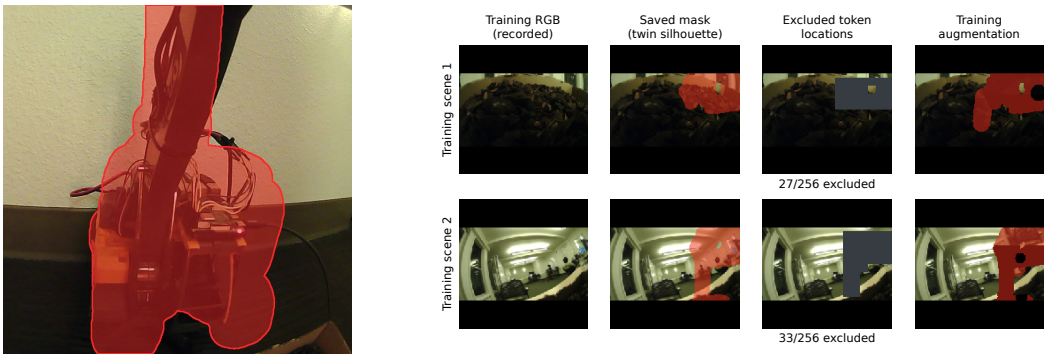


Figure 3: CLOAK masking during deployment (left) and training (right). Red marks the robot silhouette. The training examples show recorded RGB, twin-rendered masks, excluded image tokens (gray), and augmented observations.

the pretrained base. Both policies learn from handheld recordings collected without operating the target excavator.

Real rollouts. In twenty operator-scored physical scooping attempts per policy, the unmasked policy succeeds in 50% (10/20) of trials and the CLOAK policy in 60% (12/20), compared with 0% (0/20) for the pretrained base. These results demonstrate transfer from handheld bucket demonstrations to physical excavation. The CLOAK policy uses twin-rendered masks to remove the target robot from its camera observations at deployment and achieves the highest success rate in these trials.

4 Discussion and Conclusion

UMI [1] allows human tool motions to be recorded independently of the target robot, but usable supervision must preserve bucket-tip motion and pitch near soil contact while satisfying joint and timing constraints. Retargeting and temporal alignment therefore determine the supervision presented to the policy.

Our physical results also distinguish offline action prediction from closed-loop task performance, consistent with prior findings in imitation learning [32]. Although the unmasked policy achieves lower held-out action error (0.258 versus 0.280), CLOAK achieves higher observed scooping success (60% versus 50%; Table 1). Both fine-tuned policies scoop where the pretrained model fails, demonstrating useful supervision despite differences between collection and deployment. These findings motivate evaluating the complete collection-to-control pipeline: extending UMI to excavators requires translating task-relevant human motions into physically executable, temporally aligned robot supervision.

References

[1] Cheng Chi, Zhenjia Xu, Chuer Pan, et al. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. *RSS*, 2024. https://roboticsproceedings.org/rss20/p045.html.

[2] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A Research Platform for Visual-Inertial Estimation. *ICRA*, 2020. https://pgeneva.com/downloads/papers/Geneva2020ICRA.pdf.

[3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *IEEE Transactions on Robotics*, 2021. https://arxiv.org/abs/2007.11898.

[4] Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *CoRL; PMLR 305*, 2025. https://proceedings.mlr.press/v305/black25a.html.

[5] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. UMI-on-Legs: Making Manipulation Policies Mobile with Manipulation-Centric Whole-body Controllers. *CoRL 2024; PMLR 270*, 2025. https://proceedings.mlr.press/v270/ha25a.html.

[6] Harsh Gupta, Xiaofeng Guo, Huy Ha, et al. UMI-on-Air: Embodiment-Aware Guidance for Embodiment-Agnostic Visuomotor Policies. *arXiv:2510.02614*, 2025. https://arxiv.org/abs/2510.02614.

[7] Zhaxizhuoma, Kehui Liu, Chuyue Guan, et al. FastUMI: A Scalable and Hardware-Independent Universal Manipulation Interface with Dataset. *CoRL; PMLR 305*, 2025. https://proceedings.mlr.press/v305/zhaxizhuoma25a.html.

[8] Mengda Xu, Han Zhang, Yifan Hou, et al. DexUMI: Using Human Hand as the Universal Manipulation Interface for Dexterous Manipulation. *CoRL; PMLR 305*, 2025. https://arxiv.org/abs/2505.21864.

[9] Simple AI, Yuteng Wei, Jinming Ma, et al. HiFi-UMI: Learning Deployable Manipulation Policies from High-Fidelity UMI Data Alone. *arXiv preprint arXiv:2607.25895*, 2026. https://arxiv.org/abs/2607.25895.

[10] Zimu Han, Yiming Zeng, Jiyao Zhang, et al. HIL-UMI: Bringing Human-in-the-Loop Post-Training of Vision-Language-Action Models to Universal Manipulation Interface. *arXiv preprint arXiv:2609.20659*, 2026. https://arxiv.org/abs/2609.20659.

[11] Yuxuan Nai, Leixin Chang, Liangjing Yang, Shuo Yang, and Zhongyu Li. Whole-Body UMI: Transferring UMI Manipulation Skills to Humanoid Whole-Body Manipulation via Real-Time Motion Generation. *arXiv preprint arXiv:2609.22829*, 2026. https://arxiv.org/abs/2609.22829.

[12] Kevin Black, Noah Brown, Danny Driess, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *RSS*, 2025. https://roboticsproceedings.org/rss21/p010.html.

[13] Physical Intelligence. OpenPI. *Software repository, accessed October 5, 2026*, 2026. https://github.com/Physical-Intelligence/openpi.

[14] Michael Piseno, Guy Tevet, and C. Karen Liu. Cloak: Zero-Shot Cross-Embodiment Manipulation by Masking the End-Effector from the VLA. *arXiv preprint, v2 arXiv:2606.22836*, 2026. https://arxiv.org/abs/2606.22836.

[15] Liangjun Zhang, Jinxin Zhao, Pinxin Long, et al. An autonomous excavator system for material loading tasks. *Science Robotics, 6(55):eabc3164*, 2021. https://doi.org/10.1126/scirobotics.abc3164.

[16] Ryan Luke Johns, Martin Wermelinger, Ruben Mascaro, et al. A framework for robotic excavation and dry stone construction using on-site materials. *Science Robotics, 8(84):eabp9758*, 2023. https://doi.org/10.1126/scirobotics.abp9758.

[17] Yifan Zhai, Lorenzo Terenzi, Patrick Frey, Diego Garcia Soto, Pascal Egli, and Marco Hutter. ExT: Towards Scalable Autonomous Excavation via Large-Scale Multi-Task Pretraining and Fine-Tuning. *arXiv preprint arXiv:2509.14992*, 2025. https://arxiv.org/abs/2509.14992.

[18] Lennart Werner, Pol Eyschen, Sean Costello, Pierluigi Micarelli, Andrei Cramariuc, and Marco Hutter. Size Doesn't Matter: Material-State Reinforcement Learning for Excavator Transferable Soil Manipulation. *arXiv preprint arXiv:2609.12677*, 2026. https://arxiv.org/abs/2609.12677.

[19] Daoxiong Gong, Yu Wang, Jianjun Yu, and Guoyu Zuo. Motion Mapping from a Human Arm to a Heterogeneous Excavator-like Robotic Arm for Intuitive Teleoperation. *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.

[20] Lane Burgett, Shashi Shekhar Verma, Feras Sawan, and Raza Malik. A VLA Policy for an Excavator. *Actor Labs research report, March 7*, 2026. https://labs.actor/research/vla-excavator.

[21] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *IROS*, 2012. https://roboti.us/lab/papers/TodorovIROS12.pdf.

[22] Hung Pham and Quang-Cuong Pham. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. *IEEE Transactions on Robotics, 34(3):645–659*, 2018. https://doi.org/10.1109/TRO.2018.2819195.

[23] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *IJCAI*, 1981. https://www.ijcai.org/Proceedings/81-2/Papers/017.pdf.

[24] Jianbo Shi and Carlo Tomasi. Good Features to Track. *CVPR*, 1994. https://doi.org/10.1109/CVPR.1994.323794.

[25] David Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):756–770*, 2004. https://doi.org/10.1109/TPAMI.2004.17.

[26] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Transactions on Robotics, 33(1):1–21*, 2017. https://doi.org/10.1109/TRO.2016.2597321.

[27] Anastasios I. Mourikis and Stergios I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. *ICRA*, 2007. https://doi.org/10.1109/ROBOT.2007.364024.

[28] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17:261–272*, 2020. https://doi.org/10.1038/s41592-019-0686-2.

[29] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993. https://doi.org/10.1201/9780429246593.

[30] Double Eagle. Remote Control Volvo Excavator Toy E568 (E568-003). *Manufacturer specifications. EC160E identification in distributor specifications*. Accessed October 15, 2026.

[31] Volvo Construction Equipment. EC160E, EC180E Crawler Excavators. *Technical brochure, ref. 20046094-B. Manufacturer specifications*. Accessed October 15, 2026.

[32] Ajay Mandlekar, Danfei Xu, Josiah Wong, et al. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. *CoRL 2021; PMLR 164:1678–1690*, 2022. https://proceedings.mlr.press/v164/mandlekar22a.html.

A Implementation and Evaluation Details

Excavator platform. The target is a Double Eagle (Double E) E568-003 electric tabletop excavator, marketed at 1:16 scale and modeled after the Volvo EC160E crawler excavator [30]; the full-size machine is documented in Volvo's technical brochure [31]. The 1:16 ratio denotes the manufacturer's nominal model scale. Our digital twin uses the tabletop machine's scanned geometry and calibrated joint limits, while the trajectory-fitting scale s is estimated separately for each demonstration.

Capture and reconstruction. The handheld rig uses an InnoMaker U20CAM-720P camera (1280×720, up to 20 Hz), BerryIMUv3 LSM6DSL IMU (208 Hz), and Raspberry Pi 4 timestamps. The 82-recording estimator cohort includes 54 dirt recordings; the later dirt session contains 85 recordings. Fisheye intrinsics, camera–IMU rotation, and a 7.63 ms timing offset were fitted on 21 recordings. Camera–IMU translation is zero and the camera–bucket transform is identity by assumption.

OpenVINS (`69488123`) uses KLT/MSCKF [27], 250 features, 3-pixel image noise, and 2-second dynamic initialization, without persistent landmarks or zero-velocity updates. ORB-SLAM3 (`4452a3c4`) uses monocular–inertial mode. Coverage weights initialized poses by midpoint-bounded frame exposure; initialization and missing intervals stay in the denominator. ORB-SLAM3 requires inertial initialization and `OK` or `RECENTLY_LOST` status. The smoother uses sparse Levenberg–Marquardt updates implemented with SciPy numerical differentiation and sparse linear solves [28]. It retries initialization and combines map segments with KLT and IMU factors; it rejects recordings lacking a usable map segment or paths exceeding 0.5 m/s speed or 0.5 m extent.

OpenVINS comparison. A historical cohort of 82 recordings separates pose availability from visual support. Raw ORB-SLAM3 initializes on 54 and OpenVINS [2] on 81, covering 8.51% and 57.88% of recording time (paired difference 49.37 percentage points, bootstrap 95% interval [29] [44.72, 53.59]). OpenVINS's own motion warnings (5 m/s or 3 m span) flag 45/82 recordings, and its available poses include intervals supported only by IMU propagation. Table 2 compares 71 paired dirt reconstructions under matched nominal-twin replay settings: OpenVINS paths have a median 2.17 m extent for scoops that span about 0.2 m, which is propagation drift rather than motion, and shrink to 9% scale when fitted to the workspace, so the retargeted motion no longer reflects the demonstration. Smoother paths have a median 0.20 m extent and replay at approximately unit scale; 65/71 pass its plausibility check. Median model-space tip-fitting residuals are 2.51 mm for ours and 3.18 mm for OpenVINS, so the twin residual alone would not have revealed the failure. These comparisons use the nominal twin, which differs from the calibrated profile used for policy targets. Extent is the largest axis span; the normalized residual is the median of per-recording residual/scale ratios.

Metric	ORB-SLAM3 + smoother	OpenVINS
Median path extent, m	0.20	2.17
Retarget scale	1.01×	0.09×
Tip-fitting residual / retarget scale, mm	3.25	37.67

Table 2: Medians over 71 paired dirt recordings in the nominal twin, separate from the policy evaluations. The final row measures per-recording retargeting fit divided by scale, not ground-truth reconstruction accuracy.

Twin and retargeting. The scan-derived twin has boom/stick/tooth lengths of 283/152/64 mm, assuming metric mesh scale, and an effective bucket hinge. Nominal speeds come from motion video; calibrated travel and speeds come from link IMUs (Table 3). Absolute homing-stop angles remain nominal. Simulation uses 5 ms physics steps and 20 Hz control.

Workspace fitting uses $\widetilde{p} = sR_z(\psi)p + b$ and one pitch offset, choosing 90% of the maximum feasible scale. Bounded IK minimizes squared tip error plus squared pitch error weighted by 0.2 m/rad; handheld roll/yaw are not reproduced. Pitch is enforced within 5 cm of the path minimum and saturated elsewhere; 14/120 retained dirt recordings still require clipping. Refinement inserts samples

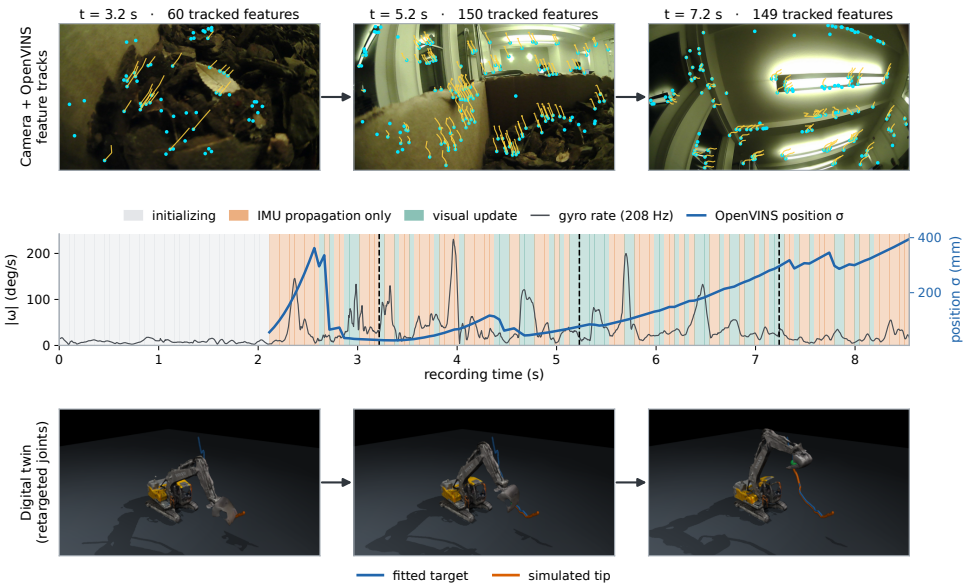


Figure 4: OpenVINS reconstruction and twin replay of a dirt demonstration: camera frames with OpenVINS feature tracks (top); gyroscope rate with visual updates marked and the intervals in which the pose is propagated from the IMU alone shaded (middle); calibrated-twin replay of the retargeted OpenVINS trajectory (bottom).

	Nominal		Calibrated	
Joint	Range	Speed +/−	Range	Speed +/−
Slew	$\pm 165^\circ$	.80/.80	continuous	1.00/1.02
Boom	-48 to 30°	.52/.68	-14.4 to 30°	.68/.63
Stick	-68 to 58°	.40/.40	-9.8 to 58°	.57/.58
Bucket	-40 to 140°	1.20/1.20	-40 to 116.8°	2.08/2.21

Table 3: Twin limits; speeds are rad/s. Calibrated travel is anchored at nominal homing angles.

for interpolation errors above 0.5 mm or 0.25°. Retiming starts/ends at rest, never accelerates source time, and uses $a_j = \max(v_j^-, v_j^+)/0.1$ s. Source-frame age is at most 75 ms. Manual screening compares synchronized video and twin motion, rejecting altered scoops rather than applying a hand-rotation threshold (Figure 5).

B Limitations

The dataset is small: 42 usable demonstrations from one session train each fine-tuned policy and 42 from a second session evaluate them, so both the cross-session gap and the CLOAK penalty may change with more data. There is no independent trajectory ground truth, so reconstruction is judged by consistency and plausibility rather than accuracy, and manual screening may favor easier motions. A feasible twin replay can still alter the intended scoop, motivating visual screening of reconstructed demonstrations.

The excavator has no joint encoders. During deployment, slew is held fixed and its policy input carries the last requested slew angle. Link-IMU angles are anchored at nominal homing stops, leaving unmeasured offsets in deployment state and twin calibration. Metric scale and camera mounting are approximate; a few degrees of boom-angle error can move the tooth tip by centimetres.

Each policy has one seed and one held-out session. Physical evaluation uses one machine and dirt setup; twenty attempts cannot separate the fine-tuned policies. Soil loads, multi-step excavation, and

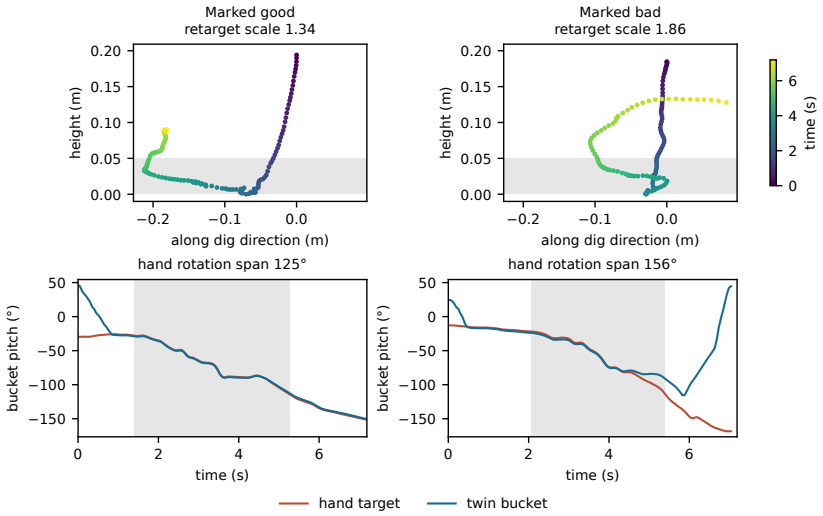


Figure 5: Manual screening: a demonstration marked good (left) and one marked bad (right). Top: the ORB-SLAM3 hand path in the vertical plane of the dig direction, colored by time, with the fitted retarget scale. Bottom: the hand's bucket pitch against the twin's actuated bucket pitch; the shaded band is the contact window in which the bucket angle is enforced. The rejected demonstration rotates the hand through a larger span and its replay departs from the intended scoop.

full-size transfer are untested, and nominal 1:16 scale does not establish dynamic similarity to the hydraulic Volvo EC160E.

These limitations motivate systematic testing on full-size excavators across soil conditions and workloads, an important next step toward extending handheld demonstration learning to practical earthmoving.

C Additional Evaluation Details

Dataset screening. Of 85 dirt recordings from the training session, replay review against the twin marks 43 as good scoops and 42 of those pass the calibrated twin's timing planner; these are the 42 training episodes. The same review marks 42 of the 54 dirt recordings from an earlier session good, and those form the held-out session.

Additional policy evaluation. The unmasked fine-tuned policy reaches a normalized held-out-session action error of 0.258 at step 1,000; its in-sample error is 0.062. The matched CLOAK run fits the training set equally well (0.061), but reaches 0.280 on the held-out session, a paired difference of $+0.022$ $[0.013, 0.034]$ relative to the unmasked policy, with lower error on 5 of 42 recordings and most of the increase in slew.

Physical evaluation. For twenty operator-scored attempts per policy, exact 95% success intervals are 0–17% for the pretrained base, 27–73% for the unmasked policy, and 36–81% for CLOAK. These trials establish successful scooping by both fine-tuned policies but do not establish a reliable difference between them. Handheld evaluation frames contain no excavator, so CLOAK masking removes scene context there; at deployment it masks the robot itself. This difference in observation content motivates evaluating masking through physical execution as well as offline prediction.