ROBOSTEREO: RECURRENT 4D RECONSTRUCTION FROM STEREO VIDEOS FOR EMBODIED INTERACTION

Anonymous authors
Paper under double-blind review

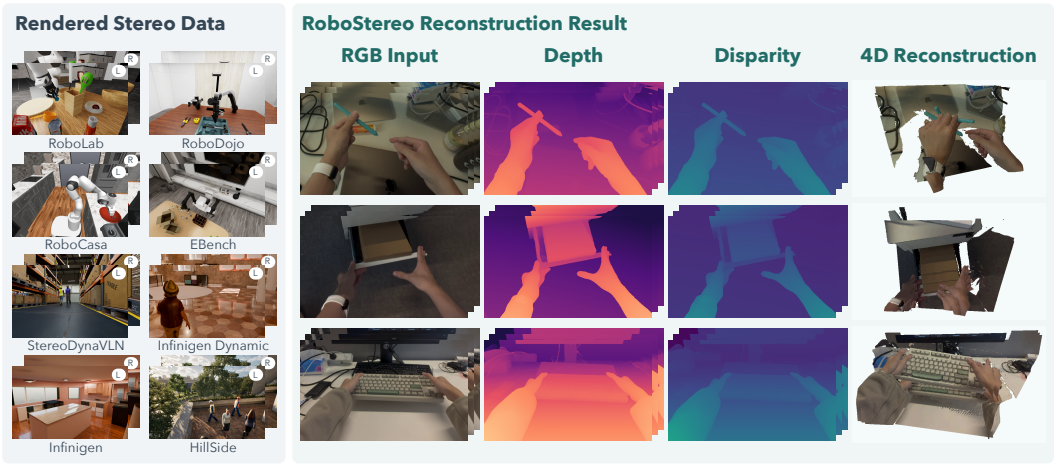

Figure 1: **Rendered stereo data and multi-frame reconstruction. Left:** Eight rendered scenes, each shown as a stacked left–right RGB pair. **Right:** Reconstruction results for three examples, with three selected frames shown per example.

ABSTRACT

Embodied robots require accurate 3D geometry during close-range interactions, yet object motion and limited relative viewpoint changes can leave local geometry weakly constrained in monocular video reconstruction. Synchronized stereo provides geometric evidence, but textureless regions, repetitive patterns, and occlusion can make matching ambiguous. We present RoboStereo, a recurrent model that reconstructs per-frame dense geometry and cameras in a shared coordinate system from synchronized, rectified stereo videos. Within a single inference run, multi-frame representations repeatedly guide stereo matching, while matching features feed back into the reconstruction state. To address the limited scene diversity of existing stereo data, we render annotated stereo videos for training and evaluation and construct the Embodiment Stereo Benchmark from selected renderings, with an emphasis on dynamic scenes and embodied manipulation. Experiments on our benchmark and public datasets demonstrate improved scale-aligned depth estimation over stereo and monocular multi-frame baselines, including on manipulation datasets excluded from joint training. Qualitative comparisons further show clearer local geometry around hand–object interactions.

1 INTRODUCTION

Embodied manipulation requires accurate 4D reconstruction during interactions with moving objects: recovering time-varying 3D scene geometry in a shared coordinate system. Reconstructing these interactions from egocentric and wrist-mounted views is challenging because object motion and occlusion limit the available geometric evidence. Recent monocular multi-frame models, such as VGGT and π^3, recover dense geometry and cameras by aggregating observations across time (Wang et al., 2025a; 2026b), but additional frames do not always resolve local ambiguities. In

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.

particular, limited viewpoint diversity, whether due to restricted camera motion or scene motion, can leave the object's shape and depth weakly constrained across frames.

Synchronized stereo provides depth constraints from the same scene state, without requiring inter-frame object motion to be recovered first (Wang & Deng, 2026). However, textureless regions, repetitive patterns, and occlusion can make correspondence ambiguous. Multi-frame observations can provide complementary structural context, as already exploited by video stereo methods such as DynamicStereo (Karaev et al., 2023) and Stereo Any Video (Jing et al., 2025). These methods use temporal context to refine disparity, rather than jointly updating matching and a reconstruction state that recovers both geometry and cameras. We study this recurrent interaction within a unified model.

We present **RoboStereo**, a unified model for dynamic reconstruction from synchronized, rectified stereo videos. Its core is Geometry-Guided Stereo Fusion, which interleaves stereo refinement with recurrent multi-frame reconstruction. Gated exchanges allow multi-frame representations to guide disparity updates and matching features to refine the reconstruction state. Updated disparities change the right-view features sampled in subsequent matching steps, while stereo-informed representations provide context for later refinement. This recurrent interaction makes correspondence estimation part of the reconstruction process, rather than an independent preprocessing stage.

Developing such models also requires stereo data that capture dynamic embodied interactions. Existing dynamic stereo datasets offer limited coverage of close-range manipulation and egocentric observations. We therefore render a diverse collection of stereo videos from simulation platforms and scene assets, spanning robot manipulation, moving subjects, and varied environments (Figure 1). The collection provides synchronized images, depth, and camera annotations, including head- and wrist-mounted observations, for both training and evaluation.

From selected subsets of these renderings, we establish the **Embodiment Stereo Benchmark** to evaluate dynamic reconstruction with an emphasis on embodied manipulation and wrist-mounted observations. It comprises seven evaluation groups with a common protocol for depth and camera estimation. RoboStereo achieves the lowest scale-aligned depth AbsRel among the compared methods across all seven groups, with additional depth improvements on public datasets (Table 2 and Appendix C, Table 7).

Our contributions are:

- **Model.** We introduce RoboStereo, a unified reconstruction model whose geometry-guided recurrent fusion jointly updates stereo correspondences and multi-frame scene representations to recover dense geometry and cameras.
- **Data and Benchmark.** We construct the Embodiment Stereo Benchmark, a comprehensive stereo video benchmark with synchronized images, depth maps, and camera annotations. It covers diverse dynamic scenes, with a particular focus on embodied interactions and challenging wrist-mounted observations, and supports the evaluation of depth and camera estimation. We hope this benchmark can facilitate the research of stereo matching for embodiment.
- **Performance.** RoboStereo achieves the lowest scale-aligned depth AbsRel among the compared methods on the Embodiment Stereo Benchmark, including evaluation datasets excluded from joint training. It also delivers competitive 3D reconstruction results on both the benchmark and public datasets.

2 Related Work

Multi-view Geometric Reconstruction. DUSt3R (Wang et al., 2024), MASt3R (Leroy et al., 2024), and POMATO (Zhang et al., 2025) recover geometry from image pairs, while VGGT (Wang et al., 2025a), π^3 (Wang et al., 2026b), and VGGT-Ω (Wang et al., 2026a) aggregate multiple views to predict dense geometry and cameras. DVLT, introduced in Déjà View (Burzio et al., 2026), refines per-view representations through a shared recurrent transformer. RoboStereo builds on this backbone to incorporate explicit stereo correspondence updates into the reconstruction loop.

Stereo Matching with Geometric Priors. RAFT-Stereo (Lipson et al., 2021) refines disparities recurrently, while WAFT-Stereo (Wang & Deng, 2026) uses cost-volume-free, warping-based updates

that we adapt for RoboStereo. MonSter (Cheng et al., 2025b) mutually refines monocular depth and stereo predictions; MonSter++ (Cheng et al., 2025a) extends this approach to multi-view stereo using known camera parameters. RoboStereo instead exchanges matching features with a multi-frame state that supports geometry and camera recovery, without requiring camera parameters as network inputs.

Stereo Video and Multi-frame Reconstruction. DynamicStereo (Karaev et al., 2023) uses neighboring frames for temporally consistent disparity estimation, while Stereo Any Video (Jing et al., 2025) incorporates a frozen monocular video depth prior. Stereo4D (Jin et al., 2025) combines camera estimation, disparity, and temporal tracks to reconstruct dynamic point clouds from Internet stereo videos. Pi3X (Wang et al., 2025b) supports external depth conditioning, which we use for a one-way WAFT-Stereo $\rightarrow$ Pi3X baseline. RoboStereo feeds matching features back into its multi-frame reconstruction state during inference, so later correspondence updates use context that has itself incorporated stereo evidence.

Synthetic Data for Stereo Video Geometry. Dynamic Replica (Karaev et al., 2023), Spring (Mehl et al., 2023), and TartanAir (Wang et al., 2020) provide synthetic supervision for stereo and video geometry. We complement these resources with renderings from existing platforms (Raistrick et al., 2023; Nasiriany et al., 2024) focused on close-range embodied interactions and egocentric observations for training and evaluation, and establish the Embodiment Stereo Benchmark for depth and camera evaluation.

3 METHOD

3.1 PROBLEM SETUP

We consider a synchronized, rectified stereo sequence $\mathcal{I} = \{(I_i^L, I_i^R)\}_{i=1}^N$ with $I_i^L, I_i^R \in \mathbb{R}^{H\times W\times 3}$. The rectified views share intrinsics within each pair, and the focal-length–baseline product $f_x b$ is constant across the clip. RoboStereo takes these images as input, with no intrinsics or baseline supplied to the network, and predicts

$$\mathrm{RoboStereo}(\mathcal{I}) = \{D_i^{\mathrm{out}}, \mathcal{C}_i, d_{\mathrm{full},i}, P_i^{\mathrm{out}}\}_{i=1}^N. \tag{1}$$

Dense outputs are per-frame maps on the left-view pixel grid, without explicit cross-frame point correspondences. The default depth $D_i^{\mathrm{out}} \in \mathbb{R}^{H\times W}$ is Z-depth in the left camera. The camera $\mathcal{C}_i = (\widehat{K}_i, \widehat{R}_i, \widehat{t}_i)$ contains predicted intrinsics and a camera-to-world rotation and translation. Disparity $d_{\mathrm{full},i} \in \mathbb{R}^{H\times W}$ denotes horizontal displacement in input-image pixels, mapping each left pixel (u, v) to $(u - d_{\mathrm{full},i}(u, v), v)$ in the right image. The world point map $P_i^{\mathrm{out}} \in \mathbb{R}^{H\times W\times 3}$ is computed analytically from depth and camera:

$$P_i^{\mathrm{out}}(p) = \widehat{R}_i\Big[D_i^{\mathrm{out}}(p)\widehat{K}_i^{-1}\bar{p}\Big] + \widehat{t}_i, \qquad \bar{p} = [u, v, 1]^\top. \tag{2}$$

The predicted depths, camera translations, and 3D points share a common non-metric scale, with the first left-camera frame serving as the world reference. Known focal length and metric baseline enable joint metric rescaling in post-processing (Section 3.5).

3.2 ARCHITECTURE OVERVIEW

RoboStereo couples a multi-frame reconstruction state with per-frame stereo matching states: scene context guides correspondence refinement, and matching features feed back into subsequent reconstruction updates. A shared Loop Transformer follows DVLT's recurrent design (Burzio et al., 2026), with frame-wise and global attention as in VGGT (Wang et al., 2025a). Geometry-Guided Stereo Fusion adopts WAFT-Stereo's disparity-guided refinement (Wang & Deng, 2026).

As shown in Figure 2, a shared frozen image encoder produces the left-view patch tokens Z and features from both views. A trainable dense feature head converts the latter into fine-grained stereo features (F^L, F^R), computed once and reused throughout refinement. At selected backbone steps, stereo fusion reads Z to refine h, d and writes matching features back into Z. Camera and register tokens interact with the stereo-updated patch tokens through subsequent backbone attention.

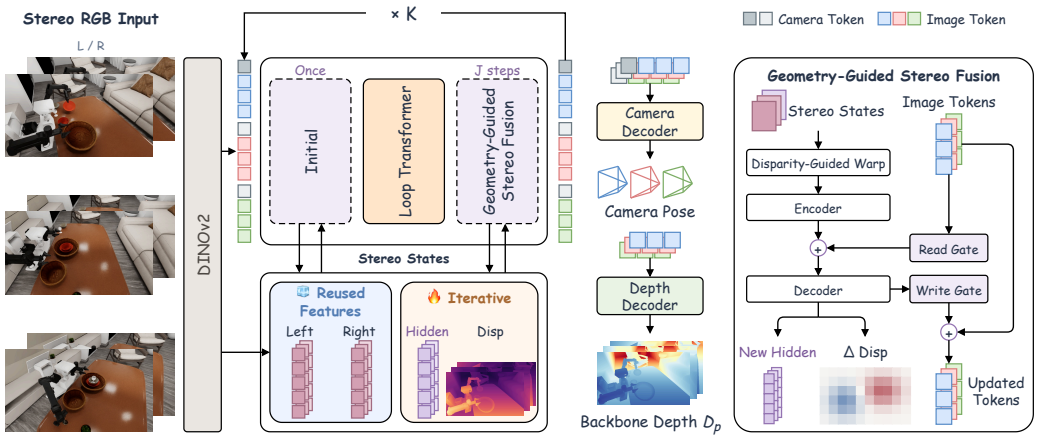

Figure 2: **RoboStereo overview.** RoboStereo interleaves the Loop Transformer with Geometry-Guided Stereo Fusion through gated read/write exchanges. Stereo features F^L, F^R are computed once and reused, while Z, h, d evolve across refinement. The depth decoder predicts D_p; default depth is obtained from disparity using D_p as the scale reference (Section 3.5). Image-token colors distinguish frames; gray blocks denote camera tokens.

We use disparity-derived depth as the default output, aligned to the predicted backbone depth D_p; cameras are recovered from predicted rays (Section 3.5). Depth and cameras determine point maps through Eq. (2).

3.3 Geometry-Guided Stereo Fusion

Disparity-Guided Sampling. Each frame maintains its own matching state h_i and disparity d_i across refinement iterations. Stereo updates operate on individual image pairs and receive cross-frame context through the corresponding backbone tokens Z_i. We omit frame and iteration indices below and denote the incoming stereo state by (h, d). Following WAFT-Stereo's disparity-guided refinement (Wang & Deng, 2026), we sample the right feature map using the current disparity:

$$\widetilde{F}^R(u, v) = F^R(u - d(u, v), v)\,. \tag{3}$$

Here (u, v) and d are measured in matching-grid pixels; Appendix A.1 specifies the grids and unit conversion. Rectification fixes the sampling row, while disparity determines the horizontal sampling position and changes the matching evidence available at the next update.

Bidirectional State Update. The bidirectional update is

$$x = \mathcal{E}\Big([F^L, \widetilde{F}^R, h, d]\Big)\,, \tag{4}$$

$$\widetilde{x} = x + \mathcal{R}(Z, x), \tag{5}$$

$$(h', \Delta d, M) = \mathcal{D}(\widetilde{x}), \tag{6}$$

$$d' = d + \Delta d, \tag{7}$$

$$Z' = Z + \mathcal{W}(M, Z). \tag{8}$$

Here $[\cdot]$ denotes concatenation, and $\mathcal{E}$ encodes the stereo inputs into tokens x. The read adapter $\mathcal{R}$ projects and spatially aligns Z to the stereo Transformer grid, then gates the aligned message before adding it to x. The stereo module $\mathcal{D}$ applies a Transformer and dense decoder to predict h' and Δd. Its write adapter projects and spatially aligns stereo Transformer features with downsampled h' on the backbone grid, then fuses them into M. The write module $\mathcal{W}$ gates M before adding it to Z. Appendix A.1 details the decoding and adapters.

Gated Feature Exchange. Separate gates control the backbone-to-stereo read and stereo-to-backbone write messages, after matching their magnitudes to the receiving features. Refinement

progress (Section 3.4) controls the overall strength of each message and its weights across channels. During refinement, spatial weights are computed from the message, the receiving features, and matching cues derived from predicted disparity distributions and disparity-update magnitudes. Read gates use cues from the preceding stereo call, whereas write gates use the current call's cues. Write gates additionally summarize these cues over each frame to attenuate the entire write-back message. Appendix A.2 provides the gate formulations and initialization details.

Initialization. At the first selected fusion step, a separately parameterized initializer uses the unwarped pair (F^L, F^R) and current backbone context to initialize h, d and write matching information into Z. It follows WAFT-Stereo's classification-based disparity initialization (Wang & Deng, 2026) and uses its own read and write modules.

3.4 Recurrent Refinement

Interaction Schedule. To balance the two modules' refinement budgets and limit high-resolution stereo computation, we select $J < K$ backbone steps for stereo fusion, with $0 \le k_1 < \cdots < k_J = K - 1$ and $J = 5$. Each backbone step applies frame-wise attention followed by global attention, with stereo refinement after attention at the selected steps. At k_1, initialization runs before attention. The backbone and stereo updater each share parameters across calls. The final stereo update skips write-back, and its disparity is convex-upsampled and converted to input-image pixels to obtain d_{full}. At inference, we use $K = 12$ and an approximately uniform schedule, with stereo updates after backbone iterations $1, 4, 7, 10, 12$.

Training Schedule and Progress Conditioning. During training, we vary both the backbone step count and fusion positions, exposing the shared stereo updater to different reconstruction stages. Since the call index alone does not determine the backbone stage or the interval since the previous call, we condition the gates on

$$e_j = [\gamma(\tau_j), \gamma(\Delta\tau_j), E(j)]. \tag{9}$$

Here $j = 1, \ldots, J$ indexes the stereo updates at the selected steps, while $j = 0$ denotes initialization within the first selected step. The normalized backbone progress is τ_j, and $\Delta\tau_j = \tau_j - \tau_{j-1}$ measures the interval since the preceding stereo call. We encode the continuous progress variables with sinusoidal features γ and the call index with a learned embedding E. Appendix A.3 details how backbone step counts and fusion positions are sampled, along with the precise progress indices.

3.5 Readout and Supervision

Geometry Readout. We retain DVLT's depth and ray decoders and auxiliary camera head, initialized from its pretrained weights (Burzio et al., 2026). The backbone depth D_p and camera translations share a common non-metric scale under reconstruction supervision. Following DVLT's default inference procedure, we recover $\mathcal{C}_i$ by fitting cameras to the predicted ray maps and use these cameras for output backprojection in Eq. (2).

Since $f_x b$ is constant across the clip, inverse disparity determines depth up to a single unknown scale factor. We align inverse disparity to the scale of D_p and use the resulting depth as the default output:

$$D_i^{\text{out}} = \eta\, d_{\text{full},i}^{-1}, \qquad \eta = \frac{\mathcal{T}(D_p)}{\mathcal{T}(d_{\text{full}}^{-1})}, \tag{10}$$

where $\mathcal{T}$ is a trimmed mean over the same valid pixel support pooled across the clip. We retain the depth branch both to preserve the pretrained geometric prior through continued reconstruction supervision and to provide this scale reference for stereo-derived depth. When f_x in input-image pixels and the metric baseline b are available, the post-processing factor $s_{\text{m}} = f_x b/\eta$ jointly rescales both depth predictions, camera translations, and output point maps to metric units, with $s_{\text{m}} D_i^{\text{out}} = f_x b / d_{\text{full},i}$. Camera rotations and predicted intrinsics remain unchanged.

Supervision. We retain DVLT's reconstruction losses and WAFT-Stereo's disparity supervision (Burzio et al., 2026; Wang & Deng, 2026), and add a stereo-point term:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{stereo}} + \lambda_{\text{sp}} \mathcal{L}_{\text{stereo-point}}. \tag{11}$$

Table 1: **Stereo data overview.** Scene coverage, moving objects, rendering platforms, and sizes of public datasets and our stereo renderings. Citations identify the datasets or source platforms and assets. All data are synthetic; the sizes of our renderings reflect the data actually rendered.

Dataset	Task	Moving Objects	Rendering Platform	Samples
TartanGround (Patel et al., 2025)	Robot Percep. & Nav.	Limited	Unreal Engine	∼1440k
TartanAir (Wang et al., 2020)	SLAM	Limited	Unreal Engine	∼1000k
TartanAir V2 (TartanAir, n.d.)	SLAM	Limited	Unreal Engine	1427k
Dynamic Replica (Karaev et al., 2023)	General Stereo Matching	Humans, Animals	Blender / PyTorch3D	169k
Sintel (Butler et al., 2012)	Optical Flow Estimation	Humans, Animals	Blender	1k
VKITTI2 (Cabon et al., 2020)	Autonomous Driving	Vehicles	Unity	21k
StereoDataset (stereo-dataset, n.d.)	Indoor/outdoor dynamic scenes	Humans	Unreal Engine	688k
RoboLab (Yang et al., 2026)	Tabletop Manipulation	Robotic Assets	Isaac	45k
RoboDojo (Chen et al., 2026)	Tabletop Manipulation	Robotic Assets	Isaac	100k
RoboCasa (Nasiriany et al., 2026)	Household Manipulation	Robotic Assets	MuJoCo	637k
EBench (Gao et al., 2026)	Household Manipulation	Robotic Assets	Isaac	63k
StereoDynaVLN (Huang et al., 2026)	Robot Navigation	Humans	Isaac	129k
Infinigen (Raistrick et al., 2024)	Indoor Roaming	None	Blender	84k
Infinigen Dynamic (Raistrick et al., 2024)	Indoor Roaming	Humans	Blender	212k
HillSide (Epic Games, 2023)	Outdoor Roaming	Humans	Unreal Engine	102k

Here $\mathcal{L}_{\text{recon}}$ supervises backbone depth, ray maps, camera-token predictions, and depth-derived world points, while $\mathcal{L}_{\text{stereo}}$ supervises the initial disparity classification and subsequent disparity refinements. Training point maps are formed directly as $P = o + Dv$, using predicted world-frame ray origins o, unnormalized directions v, and depth D. For $\mathcal{L}_{\text{stereo-point}}$, we use disparity-derived depth aligned through Eq. (10) and apply the same point loss and normalization scales as for backbone-depth points. This couples disparity and ray geometry through a common 3D error. Appendix A.4 provides the scale derivations, camera recovery procedure, and loss details.

4 Data Construction

High-quality dynamic stereo data remain limited in scene diversity and interaction complexity, particularly for embodied manipulation and egocentric observation. Dynamic Replica (Karaev et al., 2023) animates humans and animals in scanned interiors, while StereoDataset (stereo-dataset, n.d.) combines animated characters with Unreal Engine scenes. TartanGround, TartanAir, and TartanAir V2 contain very few moving objects and have nearly identical scene distributions.

To broaden this coverage, we render stereo sequences spanning robot manipulation, moving subjects in indoor and outdoor environments, and static scenes (Table 1). These include close-range interactions and wrist-mounted observations, with synchronized stereo images, depth, and camera annotations to support both training and evaluation. Platform-specific rendering procedures, camera trajectory generation, and geometric annotation are detailed in Appendix A.5.

From selected subsets of these renderings, we construct the **Embodiment Stereo Benchmark** to evaluate multi-frame 3D reconstruction in dynamic scenes, with an emphasis on embodied manipulation. The benchmark comprises seven evaluation groups, each containing 100 clips of 24 synchronized stereo pairs.

5 Experiments

We evaluate how jointly modeling stereo correspondences and multi-frame context improves depth estimation on the Embodiment Stereo Benchmark, with public datasets and camera pose estimation as complementary evaluations.

5.1 Experimental Setup

Implementation and Training. We initialize the reconstruction backbone from DVLT and partially initialize the Geometry-Guided Stereo Fusion module from WAFT-Stereo, while training its newly introduced components from scratch. We freeze the image encoder and optimize the remaining components with AdamW for 30,000 steps, using a peak learning rate of 2.5×10^{-4} and 2–24 stereo

Table 2: **Scale-aligned depth** on the Embodiment Stereo Benchmark. δ_1 is a fraction; row-best/second-best results are bold/underlined (ranked before rounding). RoboStereo uses disparity-derived depth. Abbreviations: Infinigen-D (Infinigen Dynamic), SDVLN-W (StereoDynaVLN warehouse), WAFT-S (WAFT-Stereo), WS→P (WAFT-Stereo → Pi3X), and SAV (Stereo Any Video).

		Multi-frame reconstruction			Stereo / depth-conditioned			RoboStereo
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WAFT-S	WS→P	SAV	Ours
AbsRel ↓	RoboCasa†	0.0668	0.0418	0.0559	0.0439	<u>0.0284</u>	0.2445	**0.0053**
	RoboCasa wrist†	0.3594	0.4136	0.6551	<u>0.1275</u>	0.1629	0.1754	**0.0296**
	Infinigen-D†	0.1047	0.0916	<u>0.0709</u>	0.0984	0.1251	0.2195	**0.0149**
	EBench	0.0684	0.0707	0.0979	0.0303	<u>0.0269</u>	0.0697	**0.0221**
	EBench wrist	0.3107	0.3223	0.5326	<u>0.1717</u>	0.2498	0.2902	**0.1371**
	SDVLN-W	0.0745	0.0998	0.1625	<u>0.0186</u>	0.0199	0.0430	**0.0185**
	RoboLab	0.0525	0.0377	0.0894	<u>0.0184</u>	0.0244	0.0273	**0.0106**
δ_1 ↑	RoboCasa†	0.8970	0.9500	0.9418	0.9668	<u>0.9803</u>	0.7704	**0.9970**
	RoboCasa wrist†	0.6517	0.6697	0.5503	0.8708	<u>0.8731</u>	0.7977	**0.9723**
	Infinigen-D†	0.9704	0.9700	0.9677	0.9679	<u>0.9764</u>	0.9628	**0.9875**
	EBench	0.9291	0.9108	0.8847	**0.9891**	0.9875	0.9316	<u>0.9887</u>
	EBench wrist	0.5818	0.6491	0.5187	<u>0.7759</u>	0.7363	0.6381	**0.8054**
	SDVLN-W	0.9574	0.9134	0.8208	**0.9903**	0.9870	0.9748	<u>0.9878</u>
	RoboLab	0.9407	0.9752	0.8946	<u>0.9897</u>	0.9877	0.9780	**0.9936**
RMSE ↓	RoboCasa†	0.2117	<u>0.1177</u>	0.1400	0.5974	0.2455	1.2747	**0.0503**
	RoboCasa wrist†	<u>0.1769</u>	0.1975	0.2877	0.6794	0.2969	0.4536	**0.0532**
	Infinigen-D†	0.1669	<u>0.1326</u>	0.1636	0.5225	0.2539	0.2115	**0.0785**
	EBench	0.1386	0.1455	0.2184	0.1279	**0.0932**	0.2024	<u>0.0942</u>
	EBench wrist	0.1501	<u>0.1216</u>	0.2328	0.2012	0.1838	0.2810	**0.1215**
	SDVLN-W	1.4801	1.5760	2.5891	**0.9379**	<u>1.0136</u>	1.4554	1.0322
	RoboLab	0.1625	0.1556	0.2408	0.4148	0.1602	<u>0.1466</u>	**0.0895**

pairs per training clip. Training combines our renderings with public stereo datasets; Appendix B summarizes data sampling, optimization, hardware, and input preprocessing.

Evaluation Datasets. We evaluate 100 clips from each of the seven groups of the Embodiment Stereo Benchmark (Section 4) and each of the six public datasets: Sintel, VKITTI2, Dynamic Replica, TartanGround, TartanAir, and TartanAir V2. Dynamic Replica and the three Tartan datasets use test splits separate from joint training; Sintel and VKITTI2 are excluded from joint training. Training and test sets are strictly disjoint at the sequence level.

Baselines. Pi3X (Wang et al., 2025b), VGGT-Ω (Wang et al., 2026a), and DVLT (Burzio et al., 2026) estimate depth and cameras from left-image sequences. WAFT-Stereo (Wang & Deng, 2026) and Stereo Any Video (Jing et al., 2025) use both views for single-pair and video stereo, respectively. WAFT-Stereo → Pi3X converts the disparity predicted by WAFT-Stereo to depth and feeds it into the Pi3X depth encoder as a one-way conditioning baseline. Throughout the result tables in the main text and appendix, † marks datasets used for RoboStereo joint training with held-out test splits; unmarked datasets are excluded from joint training. A ‡ on a baseline result indicates that the baseline was trained on that dataset (at least one component for pipelines).

Evaluation Protocol. We sample 24 timestamps from windows of up to 48 frames, reuse the same samples across methods, and evaluate on a 512×512 grid. We report depth AbsRel, δ_1, and RMSE with an 80 m cutoff and one scale alignment per clip. Camera ATE and translational/rotational RPE use an independent clip-level Sim(3) alignment. RoboStereo uses disparity-derived depth and ray-map cameras; auxiliary outputs and scale-and-shift depth results are reported in Appendix C. Further protocol details are given in Appendix B.2.

5.2 Depth Estimation

Results on the Embodiment Stereo Benchmark. RoboStereo achieves the lowest scale-aligned depth AbsRel across all seven evaluation groups of the Embodiment Stereo Benchmark (Table 2). It also achieves the highest δ_1 and lowest RMSE on five of the seven groups. On the two wrist-view groups, RoboStereo reduces AbsRel relative to WAFT-Stereo, the strongest baseline on this metric, by 76.7% on RoboCasa wrist and 20.1% on EBench wrist. The latter is excluded from joint

Table 3: **Camera estimation** on the Embodiment Stereo Benchmark (DVLT: ray maps). Row-best/second-best results are bold/underlined (ranked before rounding). RoboStereo uses ray-map cameras. Abbreviations: Infinigen-D (Infinigen Dynamic), SDVLN-W (StereoDynaVLN warehouse), and WS→P (WAFT-Stereo → Pi3X).

		Baselines				RoboStereo
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WS→P	Ours
ATE ↓	RoboCasa†	0.0219	0.0225	0.0388	0.0302	**0.0160**
	RoboCasa wrist†	0.1867	0.2061	0.2491	0.2215	**0.0935**
	Infinigen-D†	0.0129	**0.0070**	0.0096	0.0114	0.0085
	EBench	0.0093	**0.0036**	0.0059	0.0102	0.0066
	EBench wrist	0.0380	0.0302	0.0445	0.0456	**0.0113**
	SDVLN-W	0.0443	0.0387	0.0474	0.0476	**0.0334**
	RoboLab	0.0052	**0.0020**	0.0066	0.0039	0.0027
Translational RPE ↓	RoboCasa†	0.0260	0.0265	0.0392	0.0289	**0.0150**
	RoboCasa wrist†	0.1011	0.1126	0.1472	0.1173	**0.0594**
	Infinigen-D†	0.0108	**0.0065**	0.0085	0.0105	0.0077
	EBench	0.0068	**0.0032**	0.0046	0.0087	0.0061
	EBench wrist	0.0172	0.0160	0.0222	0.0298	**0.0093**
	SDVLN-W	0.0495	0.0436	0.0460	0.0543	**0.0380**
	RoboLab	0.0041	**0.0021**	0.0042	0.0036	0.0024
Rotational RPE ↓	RoboCasa†	**0.7339**	0.8030	2.3522	0.8339	0.8946
	RoboCasa wrist†	6.8213	6.4061	20.0309	8.0951	**6.3703**
	Infinigen-D†	0.1234	**0.0955**	0.2638	0.1760	0.3342
	EBench	0.1720	**0.1236**	0.2167	0.2655	0.2126
	EBench wrist	2.2990	2.4244	3.7001	3.4968	**1.9010**
	SDVLN-W	0.8777	**0.7838**	0.9022	0.8507	0.9186
	RoboLab	0.1104	**0.0733**	0.1632	0.1334	0.1047

training, showing that the improvement also extends to manipulation data beyond those used for joint training.

On Infinigen Dynamic, RoboStereo also improves upon the multi-frame model DVLT, which achieves the lowest scale-aligned AbsRel among the baselines. Together with the wrist-view results, this shows that the joint model improves depth in settings favoring either stereo or multi-frame reconstruction. RoboStereo also outperforms the one-way WAFT-Stereo → Pi3X pipeline in AbsRel on all seven groups; Section 5.4 examines the exchange directions through controlled ablations.

Evaluation on Public Datasets. Depth improvements extend to public datasets, where RoboStereo achieves the lowest scale-aligned AbsRel on four of the six evaluated datasets: Sintel, Dynamic Replica, TartanAir, and TartanAir V2. Complete results across all six datasets and three depth metrics are provided in Table 7 in Appendix C.

5.3 Camera Pose Estimation

Alongside depth, RoboStereo achieves the lowest ATE and translational RPE on four of the seven groups of the Embodiment Stereo Benchmark (Table 3). On both wrist-view groups, RoboStereo achieves the best results across all three pose metrics. On EBench wrist, it reduces ATE by 62.6% relative to VGGT-Ω, the strongest baseline on this metric. On public datasets, RoboStereo achieves the lowest ATE on Sintel, TartanGround, and TartanAir (Table 8 in Appendix C), while relative pose accuracy varies across datasets.

3D Reconstruction. We additionally evaluate reconstructed point clouds using one shared Sim(3) alignment per clip, with comparisons restricted to matching timestamps. Across thirteen evaluation groups, RoboStereo achieves the best average F-score (50.41%) and normalized Chamfer distance (4.98%) among the compared methods, supporting improved reconstruction accuracy. Full results and evaluation details are provided in Appendix C.4.

5.4 Ablation Studies

Bidirectional Feature Exchange. Table 4 compares RoboStereo exchange variants with a separately trained backbone-only DVLT under identical data, training configuration, budget, and evaluation (Appendix B.1). The RoboStereo variants differ only in read/write injection, sharing architecture and joint supervision.

Adding read to write-only lowers disparity-derived depth error on both datasets, supporting reconstruction context for stereo refinement.

Adding write to read-only improves both depth outputs, supporting stereo feature feedback. These comparisons support each direction when the other is enabled. Bidirectional exchange performs best for both outputs on these two domains under this training budget. Backbone-depth gains over DVLT combine the stereo branch, joint supervision, and feature exchange.

Table 4: **Read/write ablation.** Scale-aligned AbsRel ↓. Best/second-best: bold/underlined.

Variant	RoboLab		EBench	
	Backbone	Disparity	Backbone	Disparity
DVLT	0.0306	—	0.0697	—
Read only	0.0316	<u>0.0135</u>	0.0704	<u>0.0310</u>
Write only	<u>0.0245</u>	0.0172	<u>0.0380</u>	0.0321
Read + write	**0.0197**	**0.0122**	**0.0358**	**0.0285**

5.5 Qualitative Comparisons

Figure 3 shows clearer contours and fewer nearby scattered points with RoboStereo in the inspected regions. Further comparisons appear in Appendices C.5–C.7.

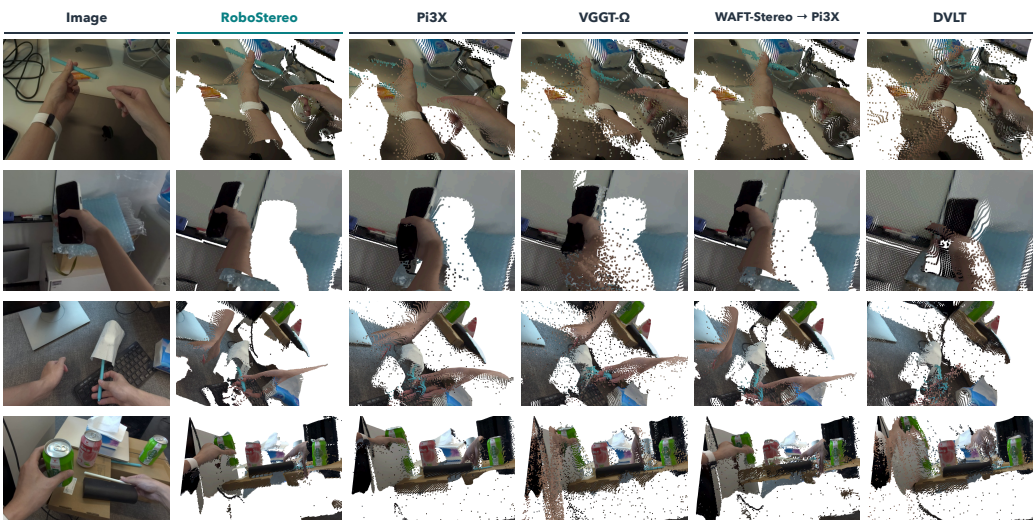

Figure 3: **Qualitative comparison of hand–object reconstruction.** From top to bottom, inspect the fingers holding the thin object, the hand–phone boundary, the hands around the thin rod and tissue, and the surfaces around the hands and cans. All methods' point clouds are shown without filtering; crops retain the original rendering viewpoints.

6 Conclusion and Limitations

RoboStereo integrates stereo matching and multi-frame reconstruction through recurrent bidirectional feature exchange. We complement the model with diverse rendered stereo videos and the Embodiment Stereo Benchmark for evaluating dynamic-scene reconstruction, with an emphasis on embodied manipulation. The results show lower scale-aligned depth errors in several settings, with strong manipulation results, while camera accuracy and depth improvements vary across datasets and metrics.

Quantitative evaluation is limited to synthetic videos, with real-world performance illustrated qualitatively. We do not separately isolate the effects of joint training and the rendered data.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, for code generation; drafting, translation, and revision of the manuscript; literature and reference checks; and interpretation and presentation of results. AI assistance also included discussion of the method and feedback on experimental design and comparisons, as well as drafting and checking mathematical derivations concerning geometry readout and scale alignment. Additional uses included auxiliary scripts for document checks, table formatting, schematic figure preparation, and the layout and annotation of visualizations from existing data and experimental outputs. Recorded checks include comparisons with source papers and implementation code, consistency checks between result tables and supplied numerical records, and compilation and visual inspection of the manuscript. The authors have fully reviewed and verified the code and manuscript and take responsibility for the final content, including all AI-assisted text, claims, and artifacts.

ETHICS STATEMENT

All rendered data were generated and used in compliance with the applicable licenses for non-commercial academic research. Real-world videos were recorded by the authors and processed to protect privacy. Only the hands of individuals are visible, and no faces or other personally identifiable information are included. This work involves neither human-subject studies nor real-robot experiments.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model, recurrent interaction schedule, geometry readout, and supervision. Appendix A provides feature resolutions, fusion and gating formulations, scale handling, and rendering procedures. Appendix B documents training data and sampling weights, optimization, preprocessing, controlled ablations, and evaluation metrics and alignment. Appendix C reports additional depth and camera results, point-cloud evaluation details and results, and qualitative comparisons.

REFERENCES

Alessandro Burzio, Tobias Fischer, Sven Elflein, Qunjie Zhou, Riccardo de Lutio, Jiawei Ren, Jiahui Huang, Shengyu Huang, Marc Pollefeys, Laura Leal-Taixé, Zan Gojcic, and Haithem Turki. Déjà View: Looping transformers for multi-view 3D reconstruction. *arXiv preprint arXiv:2605.30215*, 2026.

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, pp. 611–625, 2012.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

Junda Cheng, Wenjing Liao, Zhipeng Cai, Longliang Liu, Gangwei Xu, Xianqi Wang, Yuzhou Wang, Zikang Yuan, Yong Deng, Jinliang Zang, Yangyang Shi, Jinhui Tang, and Xin Yang. MonSter++: Unified stereo matching, multi-view stereo, and real-time stereo with monodepth priors. *arXiv preprint arXiv:2501.08643v2*, 2025a.

Junda Cheng, Longliang Liu, Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Yong Deng, Jinliang Zang, Yurui Chen, Zhipeng Cai, and Xin Yang. MonSter: Marry monodepth to stereo unleashes power. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6273–6282, 2025b.

Epic Games. Dream builders: Recreating the original vision for Habitat 67 with an Unreal Engine and RealityCapture sample project. Epic Games Spotlight, project introduction, 2023. Official introduction to the Hillside sample project, May 30, 2023. Accessed September 26, 2026.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, Xuekun Jiang, Zihou Zhu, Xinyu Li, Shuai Wang, Hao Li, Wenzhe Cai, Yuqiang Yang, Xudong Xu, Zhaoyang Lyu, Yao Mu, Tai Wang, Jiangmiao Pang, Jia Zeng, Weinan Zhang, and Chunhua Shen. EBench: Elemental diagnosis of generalist mobile manipulation policies. *arXiv preprint arXiv:2606.18239*, 2026.

Zhiyu Huang, Yun Zhang, Johnson Liu, Rui Song, Chen Tang, and Jiaqi Ma. TIC-VLA: A think-in-control vision-language-action model for robot navigation in dynamic environments. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4D: Learning how things move in 3D from internet stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10497–10509, 2025.

Junpeng Jing, Weixun Luo, Ye Mao, and Krystian Mikolajczyk. Stereo Any Video: Temporally consistent stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20836–20846, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13229–13239, 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision (ECCV)*, 2024.

Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4981–4991, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. TartanGround: A large-scale dataset for ground robot perception and navigation. *arXiv preprint arXiv:2505.10696*, 2025.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12630–12641, 2023.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen Indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, 2024.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12179–12188, 2021.

Mike Roberts, Renhan Wang, Rushikesh Zawar, Rachith Dey-Prakash, Quentin Leboutet, Stephan R. Richter, Matthias Müller, German Ros, Rui Tang, Stefan Leutenegger, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Vladlen Koltun. SPEAR: A simulator for photorealistic embodied AI research. In *European Conference on Computer Vision (ECCV)*, 2026.

stereo-dataset. StereoDataset: Dataset card. `https://huggingface.co/datasets/stereo-dataset/stereo-dataset`, n.d. Accessed September 26, 2026.

TartanAir. TartanAir V2: Dataset documentation. `https://tartanair.org/`, n.d. Accessed September 26, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3X: Engineering update to Pi3. Code repository, `https://github.com/yyfz/Pi3`, 2025b. Released December 28, 2025. Accessed September 26, 2026.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations (ICLR)*, 2026b.

Yihan Wang and Jia Deng. WAFT-Stereo: Warping-alone field transforms for stereo matching. *arXiv preprint arXiv:2603.24836*, 2026.

Xuning Yang, Rishit Dagli, Alex Zook, Hugo Hadfield, Ankit Goyal, Stan Birchfield, Fabio Ramos, and Jonathan Tremblay. RoboLab: A high-fidelity simulation benchmark for analysis of task generalist policies. In *Robotics: Science and Systems (RSS)*, 2026. doi: 10.15607/RSS.2026.XXII.096.

Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. Pomato: Marrying pointmap matching with temporal motions for dynamic 3d reconstruction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5680–5689. IEEE, 2025.

A Additional Details

This appendix expands the fusion operators, schedule defaults, scale alignment, and supervision used in Section 3, followed by the rendering pipelines used to construct the data in Section 4.

A.1 Stereo Fusion Details

Shared Image Features. Both views are processed by the same frozen DINOv2 ViT-B/14 encoder (Oquab et al., 2024). We extract its normalized patch features at layers $3, 6, 9, 12$ and initialize the backbone patch tokens Z with the last-layer left-view features. A shared, trainable dense feature head uses a DPT-style top-down decoder (Ranftl et al., 2021) to fuse the four feature maps into matching features F^L, F^R at half input resolution. These features are computed once and reused throughout refinement. Table 5 summarizes the model configuration. Disparity d is refined on the same half-resolution grid as h, in matching-grid pixels.

Table 5: Stereo fusion configuration. Layer indices in this table are one-based.

Component	Configuration
Backbone patch tokens Z	768 channels, $H/14 \times W/14$
Loop Transformer attention	12 heads, feed-forward expansion $4\times$
Matching features F^L, F^R and state h	128 channels, $H/2 \times W/2$
Dense feature head projection widths	$128, 256, 512, 1024$
Dense feature head upsampling factors	$8, 4, 2, 1$
Initializer / updater Transformer	12 layers, 384 channels, 6 heads
Stereo Transformer grid	$\lceil H/16 \rceil \times \lceil W/16 \rceil$
Transformer feed-forward expansion	$4\times$
Stereo decoder feature taps	Layers $3, 6, 9, 12$
High-resolution bypass	4 residual convolutional blocks
Initialization bins	41 centers, 0–800 input-image pixels
Convex upsampling	$2\times$, 3×3 neighborhood, 36 mask channels

Initialization and Matching. The initializer predicts a distribution over disparity bins and uses its expectation as $d_i^{(0)}$, following WAFT-Stereo (Wang & Deng, 2026). It reads the unwarped feature pair and current backbone tokens, initializes the dense matching state, and produces a write-back message. For matching-grid bin logits $l_r^{(0)}$ and full-resolution centers $a_r = 20r$, $r = 0, \ldots, 40$, initialization is

$$d^{(0)} = \sum_{r=0}^{40} \mathrm{softmax}(l^{(0)})_r \, \frac{a_r}{2}. \tag{12}$$

During subsequent updates, the disparity-guided warp in Eq. (3) samples fractional horizontal coordinates with bilinear interpolation and zero extension outside the right feature map.

Stereo Encoder and Decoder. The convolutional encoder projects the concatenated input and progressively downsamples it to the stereo Transformer grid, where the backbone read residual is injected. The stereo Transformer processes each image pair separately. A four-level decoder combines its intermediate features with high-resolution encoder bypass features to recover h'; these bypass connections are implicit in Eq. (6). The initializer uses the unwarped feature pair as its encoder input and has separate parameters from the updater; the updater's parameters are shared across all five calls.

Read and Dual-Source Write Adapters. The read adapter normalizes and projects Z to 384 channels and aligns it with the stereo Transformer input. For writing, the current call's final stereo tokens are projected and spatially aligned, while h' is downsampled by learned convolutions. Both paths produce 384-channel features on the backbone grid, which are separately normalized, concatenated, fused, and projected to the 768-channel message M. Reads use the preceding call's matching cues, while writes use the current call's cues; Appendix A.2 defines the separately parameterized gates and the initializer's cue handling.

Prediction Heads and Full-Resolution Outputs. The initializer predicts disparity-bin logits and an upsampling mask from its dense state; the updater predicts a disparity increment, mixture-distribution parameters, and an upsampling mask from h'. Let Up_m denote learned convex upsampling, which uses normalized mask weights to combine neighboring matching-grid values. At update j, full-resolution disparity and distribution parameters are

$$d_{\mathrm{full}}^{(j)} = \mathrm{Up}_{m^{(j)}}(2d^{(j)}), \qquad \xi_{\mathrm{full}}^{(j)} = \mathrm{Up}_{m^{(j)}}(\xi^{(j)}). \tag{13}$$

The factor of two converts disparity from matching-grid pixels to input-image pixels. The outputs $\xi^{(j)}$ contain two mixture logits and one predicted Laplace log-scale for the loss in Appendix A.4; gate cues use matching-grid predictions. For initialization, full-resolution probabilities are obtained as $\pi^{(0)} = \mathrm{softmax}(\mathrm{Up}_{m^{(0)}}(l^{(0)}))$ and used for classification supervision with centers a_r.

A.2 Residual Gate Details

For an aligned message M and receiving features X with D channels, the read or write gate returns the residual

$$\mathcal{G}(X, M; e, U) = G(X, M, e, U) \odot \widetilde{M}, \tag{14}$$

where e is the progress encoding in Eq. (9), U contains matching-state cues, and $\widetilde{M}$ is magnitude-matched to X. The gate factorizes as

$$G(X, M, e, U) = A(e)Q(U)\big[C(e) \odot S(X, M, U)\big]. \tag{15}$$

Here A is a positive scalar amplitude, C is a channel vector, S is a spatial map shared across channels, and Q is a per-view write attenuation factor. All read gates set $Q = 1$; the initializer's read gate also sets $S = 1$ and requires no matching cues.

Magnitude Matching. After adapter projection and spatial alignment, we compute

$$\widetilde{M} = \mathrm{sg}\left[\min\left(r_{\max}, \frac{\mathrm{RMS}_w(X)}{\mathrm{RMS}_w(M) + \epsilon}\right)\right] M, \tag{16}$$

where sg stops gradients through the scale ratio. For each view, let $\langle f \rangle_w = \sum_p w_p f(p) / \sum_p w_p$ denote a coverage-weighted spatial mean. Then $\mathrm{RMS}_w(X) = [D^{-1} \sum_c \langle X_c^2 \rangle_w]^{1/2}$. The weights w_p measure the fraction of each spatial cell covered by the input image; they are one throughout the backbone grid.

Progress and Channel Factors. Each of τ and $\Delta\tau$ is encoded with sine and cosine features at eight frequencies $2^k\pi$, $k = 0, \ldots, 7$. Together with a learned 64-dimensional embedding of the call index, these features form e. A progress MLP produces one amplitude logit a and D channel logits:

$$(a, c) = f_{\mathrm{progress}}(e), \qquad A = \mathrm{softplus}(a), \qquad C_c = \frac{\mathrm{softplus}(c_c) + \epsilon}{D^{-1} \sum_{c'=1}^{D} [\mathrm{softplus}(c_{c'}) + \epsilon]}. \tag{17}$$

Thus A controls the overall strength, while C distributes it across channels with unit channel mean.

Matching-State Cues. The gate cues summarize classification ambiguity, mixture predictions, and update magnitude on the matching grid:

$$\begin{aligned} U = [&\mathcal{H}, \pi_1, \pi_2, \ell_1, \ell_2, \\ &\log(1 + |\Delta d|), m_{\mathcal{H}}, m_{\mathrm{mix}}]. \end{aligned} \tag{18}$$

Initialization provides the normalized entropy $\mathcal{H} = -\sum_b p_b \log(p_b + \epsilon) / \log B$ of the B-bin disparity distribution. Updates provide mixture weights (π_1, π_2), a predicted Laplace log-scale ℓ_1 clamped to $[0, 10]$, a fixed $\ell_2 = 0$, and $\log(1 + |\Delta d|)$ in matching-grid units. The binary indicators $m_{\mathcal{H}}, m_{\mathrm{mix}}$ identify the available cue family; unavailable cues are zero. Cues are detached from the matching predictions and follow the read/write timing specified in Appendix A.1.

Spatial Factor. The message M and receiving features X are separately normalized and projected, then concatenated with cues aligned to the receiving grid. A convolutional head maps these features to spatial logits s:

$$S(p) = \frac{\mathrm{softplus}(s(p)) + \epsilon}{\langle \mathrm{softplus}(s) + \epsilon \rangle_w}. \tag{19}$$

The unit coverage-weighted mean redistributes the contribution within each view while retaining gradients through X and M.

View-Wise Write Attenuation. Write gates summarize each cue channel by its spatial mean μ_U and standard deviation σ_U over the input image. A separate MLP maps these statistics to

$$Q = \mathrm{sigmoid}(f_{\mathrm{view}}([\mu_U, \sigma_U])) \,. \tag{20}$$

Unlike the unit-mean spatial factor, Q can attenuate the entire write message.

Configuration and Initialization. The progress and view MLPs have two layers with hidden width 256; the spatial head uses 32-channel projections of X and M. We use $\epsilon = 10^{-6}$ and $r_{\max} = 10$. Output biases and small zero-mean Gaussian weights (standard deviation 10^{-3}) initialize A near 0.1 for reads and 10^{-3} for writes, C, S near one, and Q near 0.9. The small initial write amplitude limits perturbations of pretrained backbone tokens.

A.3 Training Schedule and Default Inference Schedule

This section specifies the progress indices and training sampler for one initializer call and $J = 5$ updater calls in Section 3.4.

Fusion Progress Indices. We define

$$\begin{aligned} \tau_0 &= \frac{k_1}{K}, & \tau_{-1} &= 0, \\ \tau_j &= \frac{k_j + 1}{K}, & 1 \le j \le J, \\ \Delta\tau_j &= \tau_j - \tau_{j-1}, & 0 \le j \le J. \end{aligned} \tag{21}$$

Here $j = 0$ denotes initialization before backbone step k_1; update $j \ge 1$ follows step k_j.

Training. The backbone update count is sampled using the inherited DVLT sampler (Burzio et al., 2026), with $r \sim \mathrm{Beta}(2, 1)$ and $K = 8 + \mathrm{round}(4r)$, giving $K \in \{8, 9, 10, 11, 12\}$. Conditional on K, the position sampler chooses between two branches:

- With probability 0.25, use the last five steps, $(K - 5, \ldots, K - 1)$.
- With probability 0.75, draw four distinct positions without replacement from $\{0, \ldots, K - 2\}$ using weights
$$w_k = (k + 1)^\alpha, \qquad \alpha = 1.5, \tag{22}$$
then sort them and append the mandatory final step $K - 1$.

The power weights favor later backbone updates while retaining support for every non-final position.

Inference. By default, $K = 12$ and the five updater calls follow zero-based backbone steps $(0, 3, 6, 9, 11)$, with initialization before step 0. The initializer and the first four updater calls write back to the backbone; the final updater call skips write-back.

Backbone Updates. The backbone follows $\mathcal{Z} \leftarrow \mathcal{F}_\theta(\mathcal{Z}; t_k, t_{k+1})$, conditioned on its inference interval. Its time grid is $t_k = k/(K - 1)$ for $0 \le k < K$, with $t_K = 1$, distinct from the fusion progress τ_j. During training, gradients through the input disparity are stopped before every updater call, including the first after initialization; gradients through the matching state and backbone tokens are retained.

A.4 Readout, Scale Alignment, and Supervision

We detail the camera recovery, scale conversion, and losses in Section 3.5; evaluation alignments are specified separately in Appendix B.2.

Inherited Depth and Camera Readout. Following DVLT (Burzio et al., 2026), the final tokens are decoded into a depth map D_p and dense ray origins and directions (o_i, v_i). The directions are unnormalized and paired with Z-depth, so the depth-decoder point map is $P_{p,i} = o_i + D_{p,i} v_i$. The default cameras are $\mathcal{C}_i^{\text{ray}} = \text{FitCamera}(o_i, v_i)$; the inherited camera-token head provides an auxiliary $\mathcal{C}_i^{\text{token}}$ in the same reconstruction scale. Output backprojection uses the ray-fitted cameras, while point supervision uses raw rays.

Camera Recovery from Rays. We fit cameras from ray maps downsampled to $H/14 \times W/14$, retaining their input-image coordinates. The camera model is

$$v_i(p) \;\propto\; \widehat{R}_i \widehat{K}_i^{-1} \bar{p}, \qquad o_i(p) \;\approx\; \widehat{t}_i. \tag{23}$$

The direction relation defines a homography in projected ray coordinates, which we estimate using RANSAC and refit on its inliers. A QR decomposition separates the camera-to-world rotation and intrinsic factor; intrinsics are expressed in input-image pixels. Translation is the spatial mean of the downsampled ray origins.

Disparity to the Backbone Scale. Write $q_i = d_{\text{full},i}^{-1}$, and let Ω be the common support of finite, positive D_p and q, excluding image padding and pooling all frames of a clip. For a field a restricted to these $n = |\Omega|$ pixels, let $a_{(1)} \le \cdots \le a_{(n)}$ be its sorted values. With a trim fraction $\rho = 0.05$ per tail and $m = \lfloor \rho n \rfloor$, define

$$\mathcal{T}_\Omega(a) = \frac{1}{n - 2m} \sum_{r=m+1}^{n-m} a_{(r)}. \tag{24}$$

Both fields use the same support and trimming rule, with separate sorting; Eq. (10) suppresses the subscript Ω.

We use rectified pairs with common left/right intrinsics and a clip-constant product $c = f_x b > 0$ measured in the input-image coordinate system. Metric stereo depth is then $D_{s,i}^{\text{metric}} = c q_i$. Positive scaling preserves the order and trimmed indices on unchanged support, giving $\mathcal{T}_\Omega(cq) = c\mathcal{T}_\Omega(q)$ and therefore

$$\begin{aligned} \frac{\mathcal{T}_\Omega(D_p)}{\mathcal{T}_\Omega(D_s^{\text{metric}})} D_{s,i}^{\text{metric}} &= \frac{\mathcal{T}_\Omega(D_p)}{c\mathcal{T}_\Omega(q)} c q_i \\ &= \eta q_i = D_i^{\text{out}}. \end{aligned} \tag{25}$$

Depth and Cameras in Metric Units. With known $c = f_x b$, the joint rescaling in Section 3.5 uses

$$s_{\text{m}} = \frac{\mathcal{T}_\Omega(D_s^{\text{metric}})}{\mathcal{T}_\Omega(D_p)} = \frac{c}{\eta}, \qquad P_i^{\text{out,metric}} = s_{\text{m}} P_i^{\text{out}}. \tag{26}$$

Scaling depth and camera translation together gives this point-map relation for either camera branch; raw-ray points follow the same relation when both origins and depth are scaled.

Reconstruction and Stereo Losses. The terms in Eq. (11) expand as

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{ray}} + \mathcal{L}_{\text{point-prior}} + \mathcal{L}_{\text{camera}}, \tag{27}$$

$$\mathcal{L}_{\text{stereo}} = \mathcal{L}_{\text{cls}} + \sum_{j=1}^{J} 0.9^{J-j} \mathcal{L}_{\text{disp}}^{(j)}. \tag{28}$$

The reconstruction terms retain the inherited depth, ray, point, and camera losses (Burzio et al., 2026); stereo uses soft-label classification at initialization and mixture-of-Laplace disparity supervision (Wang & Deng, 2026). For the full-resolution bin centers a_r and initialization probabilities

$\pi_r^{(0)}$ defined in Appendix A.1, and target disparity d^{gt}, the per-pixel classification loss is

$$y_r = \frac{\exp(-|a_r - d^{\mathrm{gt}}|)}{\sum_s \exp(-|a_s - d^{\mathrm{gt}}|)}, \qquad \ell_{\mathrm{cls}} = -\sum_r y_r \log \pi_r^{(0)}. \tag{29}$$

At update j, the predicted mixture weights $\pi_{j,1}, \pi_{j,2}$ are obtained by softmax, and the per-pixel disparity loss is

$$\ell_{\mathrm{disp}}^{(j)} = -\log\left[\sum_{a=1}^{2} \frac{\pi_{j,a}}{2b_{j,a}} \exp\left(-\frac{|d_{\mathrm{full}}^{(j)} - d^{\mathrm{gt}}|}{b_{j,a}}\right)\right]. \tag{30}$$

Here $b_{j,1} = \exp(\mathrm{clip}(\beta_j, 0, 10))$ for predicted log-scale β_j, and $b_{j,2} = 1$; disparities and Laplace scales use input-image pixels. The corresponding $\mathcal{L}_{\mathrm{cls}}$ and $\mathcal{L}_{\mathrm{disp}}^{(j)}$ average these errors over valid supervised pixels within each clip and then over clips. Stereo supervision uses pixels with valid depth and disparity annotations and target disparity in $(0, 800)$ input-image pixels.

Stereo-Point Loss and Its Two Scale Operations. Training restricts alignment to $\Omega_{\mathrm{align}}^{\mathrm{sup}} = \Omega \cap \Omega_{\mathrm{depth}}^{\mathrm{gt}}$, where $\Omega_{\mathrm{depth}}^{\mathrm{gt}}$ denotes valid ground-truth depth pixels. Using predicted D_p and q on this support gives

$$D_{s,i}^{\mathrm{sup}} = \frac{\mathcal{T}_{\Omega_{\mathrm{align}}^{\mathrm{sup}}}(D_p)}{\mathcal{T}_{\Omega_{\mathrm{align}}^{\mathrm{sup}}}(q)} q_i, \qquad P_{s,i}^{\mathrm{sup}}(p) = o_i(p) + D_{s,i}^{\mathrm{sup}}(p) v_i(p). \tag{31}$$

This aligns stereo depth to the backbone scale before applying the inherited point-loss normalization over valid points Ω_{rec}. For prior points $P_p = o + D_p v$ and ground-truth points P^{gt}, define the stabilized mean distance from the world origin:

$$a(P; \Omega_{\mathrm{rec}}) = \mathrm{clip}\left(\frac{\sum_{(i,p)\in\Omega_{\mathrm{rec}}} \|P_i(p)\|_2}{|\Omega_{\mathrm{rec}}| + \epsilon_s}, \epsilon_s, a_{\max}\right), \tag{32}$$

$$\widehat{s}_p = a(P_p; \Omega_{\mathrm{rec}})^{-1}, \qquad s_{\mathrm{gt}} = a(P^{\mathrm{gt}}; \Omega_{\mathrm{rec}})^{-1}. \tag{33}$$

We use $\epsilon_s = 10^{-3}$ and $a_{\max} = 10^3$. The point terms are

$$\mathcal{L}_{\text{point-prior}} = \ell_{\mathrm{point}}\left(\widehat{s}_p P_p, s_{\mathrm{gt}} P^{\mathrm{gt}}\right), \tag{34}$$

$$\mathcal{L}_{\text{stereo-point}} = \ell_{\mathrm{point}}\left(\widehat{s}_p P_s^{\mathrm{sup}}, s_{\mathrm{gt}} P^{\mathrm{gt}}\right). \tag{35}$$

Both use the same predicted rays, point error, and inverse scales estimated from the backbone point map and ground-truth points. For each clip and its valid finite supervised point set $\mathcal{V}$, we use the unweighted point error

$$\ell_{\mathrm{point}}(A, B; \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{(i,p)\in\mathcal{V}} \min(\|A_i(p) - B_i(p)\|_2, 100). \tag{36}$$

Point losses are averaged over clips after the within-clip reduction. Stereo-point supervision propagates through ray origins, ray directions, and aligned depth.

Shared Reconstruction Normalization. The same inverse scales $\widehat{s}_p$ and s_{gt} normalize predicted and ground-truth depth, ray origins, and camera translations, respectively. Ray directions, rotations, and fields of view retain their original parameterization. Ground-truth rays are $o_i^{\mathrm{gt}} = t_i^{\mathrm{gt}}$ and $v_i^{\mathrm{gt}}(p) = R_i^{\mathrm{gt}}(K_i^{\mathrm{gt}})^{-1}\bar{p}$. Following DVLT (Burzio et al., 2026), depth uses absolute error and a four-scale edge term (strides $1, 2, 4, 8$) on synthetic data; ray and camera terms use component-wise absolute errors. Camera translation, quaternion, and field-of-view terms have weights 1, 1, and 0.5. Depth and camera-translation errors are capped at 100 per component, and ray errors at 100 per frame mean, with inherited valid-data masks and reductions.

Loss Configuration. All loss terms have unit outer weights; depth regression and its edge term also have unit weights, and depth and point regression use no confidence weighting. Reconstruction terms are averaged over clips after their inherited within-clip reductions. Stereo-point supervision starts after the first 10% of training; Appendix B gives the main and ablation training schedules. Gradients propagate through the depth-alignment statistics and the predicted point-normalization scale; the recurrent disparity detachment is specified in Appendix A.3.

A.5 Data Rendering

We build on existing simulation platforms and assets, adding stereo observation, camera motion, and geometric annotation pipelines. For StereoDynaVLN, we directly reuse the official DynaNav pipeline (Huang et al., 2026) to re-render stereo sequences.

Collection Sizes. Table 1 counts TartanGround, TartanAir V2, and StereoDataset by synchronized sampling instants, without expanding the available camera-pair combinations. For our renderings, the table reports the actual rendered data sizes, counting the frame pairs from each retained stereo view.

Stereo Geometry and Synchronization. The custom pipelines use parallel stereo cameras with shared intrinsics and orientation, separated along the local horizontal camera axis by a baseline fixed within each sequence. Both eyes and their annotations are captured from the same scene state, holding physical time fixed across views. All pipelines export RGB, depth, intrinsics, and per-frame camera poses.

RoboDojo: Action Replay. We replay policy-generated manipulation actions from the same initial layout in RoboDojo/Isaac Sim (Chen et al., 2026), annotating the resulting replay states. Six stereo rigs cover the head, both wrists, and three external views; external trajectories use smooth random motion subject to workspace, collision, and stereo-visibility constraints. We also render optical flow. We exclude incomplete or invalid captures and trim prolonged inactive tails based on wrist motion, without requiring task success.

RoboLab: Replay with Geometric Environments. We use a replay-based rendering workflow analogous to RoboDojo, replaying manipulation episodes in RoboLab (Yang et al., 2026) with our custom camera trajectory generation algorithm. We replace the texture-only environment representation with explicit 3D scene geometry, so that the surroundings also provide geometrically defined depth. During replay, we capture synchronized stereo RGB and depth and save camera intrinsics and per-frame poses.

RoboCasa: Demonstration-State Replay. We restore recorded demonstration states in RoboCasa365/MuJoCo (Nasiriany et al., 2026), using a wrist rig, a head rig, and four candidate external rigs. External paths combine orbiting, smooth random walks, and robot or target following; selection enforces trajectory continuity, clearance, stereo visibility, and complementary coverage. View-specific image and depth checks determine the retained views. We also generate optical flow for head and external views.

EBench: Frozen-State Re-rendering. We replay EBench episodes (Gao et al., 2026) in GenManip/Isaac Sim, preserving recorded states and sampling times. Up to six stereo views cover the wrist, a robot-following reference, and four external trajectories selected for coverage, visibility, clearance, and diversity. We also render optical flow. Views failing image, synchronization, or geometric checks are rejected.

Infinigen: Procedural Scenes and Animated Subjects. Static scenes follow the official Infinigen Indoors pipeline (Raistrick et al., 2024) with moving cameras and fixed geometry. Dynamic scenes add an articulated subject following a smoothed, collision-aware A* route through a fixed procedural room, with local animation adjusted to its speed. We render a first-person view with the path-carrying subject hidden, and external views that either alternate attention between subject and environment or follow the subject with exploratory motion. Camera paths obey distance and stereo-rig clearance constraints. Cycles renders RGB; a separate geometry pass produces depth and optical flow. Filters check completeness, geometric validity, and framing; external views require the projected subject proxy to intersect the image in at least 30% of frames, without guaranteeing unoccluded visibility.

HillSide: Architectural Exploration and Walking Subjects. We render HillSide architectural assets (Epic Games, 2023) in Unreal Engine through SPEAR (Roberts et al., 2026). Static captures use smooth, collision-checked camera paths; dynamic captures add walking characters and a camera

planner constrained by clearance, line of sight, and turning limits. We reject unsafe motion, poor depth coverage, and synchronization failures, and verify stereo geometry by depth-aware cross-eye reprojection.

Annotation Scope. Simulator geometry supplies depth and camera supervision; invalid geometric measurements are masked rather than treated as valid surfaces. The rendering filters above govern capture quality; benchmark selection and train/test partitioning are separate steps.

B Experimental Details

Training Data. We train on RoboDojo, Spring, Infinigen, StereoDataset, and HillSide, together with the training splits of RoboCasa, Infinigen Dynamic, Dynamic Replica, TartanGround, TartanAir, and TartanAir V2. The relative clip-sampling weights for these eleven sources, in the same order, are $8:3:5:15:5:10:15:10:15:15:25$. RoboLab, EBench (including wrist views), StereoDynaVLN warehouse, Sintel, and VKITTI2 are excluded from joint training.

Optimization. Reconstruction components are initialized from DVLT, while the Geometry-Guided Stereo Fusion module is partially initialized from WAFT-Stereo and its newly introduced components are randomly initialized. We freeze the image encoder and jointly optimize the remaining components for 30,000 optimizer updates on 16 NVIDIA A100 GPUs with 80 GB memory each, using BF16 precision and no gradient accumulation. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, and gradient-norm clipping at 1.0. The one-cycle schedule spans all 30,000 updates, with a peak learning rate of 2.5×10^{-4}, a 3,000-update warmup followed by cosine decay, and a 0.1 learning-rate multiplier for the recurrent backbone, camera/register tokens, depth/ray decoders, and camera head. Stereo-point supervision has weight 1 and starts at step 3,000.

Inputs and Preprocessing. Training clips contain 2–24 stereo pairs, with dynamic batches capped at 48 pairs per device. We sample image sizes with areas of 200,000–255,000 pixels and aspect ratios between 0.5 and 2.0. The left and right images share isotropic resizing and principal-point-centered cropping, with corresponding updates to intrinsics and geometric targets. No additional photometric augmentation, rotation, or flipping is applied.

B.1 Read/Write Ablation Controls

Data and Training Budget. All four variants in Table 4 use the same 30% subsets of the eleven training sources and the main recipe's sampling, preprocessing, AdamW settings, peak learning rate, and parameter-group multipliers. Each run completes a 7,500-update one-cycle schedule with 750 warmup updates, using seed 42, BF16, eight GPUs, and two gradient-accumulation steps per update. The per-device budget is at most 48 stereo pairs for RoboStereo and 48 left views for DVLT. All variants share the reconstruction initialization; the three RoboStereo variants also share the stereo-module initialization procedure.

Feature Exchange and Supervision. The three RoboStereo variants share the same architecture and losses. Disabling a direction fixes its injection amplitude A to zero throughout training and inference, in both the initializer and updater; the final updater always omits write-back. Adapters and gates remain in the computation graph. All four variants retain the unit-weight reconstruction losses in Eq. (27). The three RoboStereo variants retain the stereo losses in Eq. (28) and enable unit-weight stereo-point supervision after 750 updates. No extra gradient detachment is introduced: stereo-point supervision propagates through the backbone's predicted rays and depth-dependent scale statistics (Appendix A.4). DVLT uses only left images and reconstruction losses, with no stereo branch.

Evaluation. All variants use their final 7,500-update checkpoint and their respective main-table inference settings, including the approximately uniform fusion schedule for RoboStereo. Evaluation uses the same RoboLab and EBench data and scale-aligned AbsRel, with backbone and disparity-derived depth reported separately.

B.2 Evaluation Protocol

Clip Sampling and Inputs. Each clip contains 24 timestamps sampled uniformly without replacement from a temporal window of up to 48 frames and sorted chronologically. Windows are shortened for sequences with fewer than 48 available frames; at least 24 frames are required. The same sampled frames are used across methods. We evaluate 100 clips from each of the seven groups of the Embodiment Stereo Benchmark, totaling 700 clips. Each of the six public datasets also contributes 100 clips, giving 1,300 clips across all thirteen evaluation groups. RoboLab clips are drawn from camera00–camera03 with seed 6657, without motion filtering or camera balancing. The evaluation grid is 512×512; model adapters resize or pad inputs as needed to satisfy patch-size requirements. VGGT-Ω uses the reproduced checkpoint with 416×416 inference inputs. Pi3X, VGGT-Ω, and DVLT receive only the left images. WAFT-Stereo, Stereo Any Video, RoboStereo, and the WAFT-Stereo $\rightarrow$ Pi3X pipeline also use the corresponding right images. Depth predictions are evaluated on the left-view grid, using bilinear resizing when their output resolution differs.

Depth Alignment and Metrics. The evaluation mask intersects the dataset validity mask with finite, positive ground-truth depths below 80 m and finite predictions. We fit a single scale, or scale and shift, using valid pixels pooled over all 24 frames of a clip. Scale-only alignment uses ten iteratively reweighted least-squares updates to approximate an absolute-error fit, initialized by the ratio of mean ground-truth and predicted depths. Scale-and-shift alignment minimizes the sum of absolute depth errors, initialized by the ratio of median depths and zero shift. Predicted depths are clamped to at least 10^{-6} for fitting, and aligned depths are clamped to the same minimum for scoring. On the valid support Ω of a clip, with aligned prediction $\widetilde{D}$ and ground truth D^{gt}, the reported metrics are

$$\mathrm{AbsRel} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \frac{|\widetilde{D}(p) - D^{\mathrm{gt}}(p)|}{D^{\mathrm{gt}}(p)}, \tag{37}$$

$$\delta_1 = \frac{1}{|\Omega|} \sum_{p \in \Omega} \mathbf{1}\left[\max\left(\frac{\widetilde{D}(p)}{D^{\mathrm{gt}}(p)}, \frac{D^{\mathrm{gt}}(p)}{\widetilde{D}(p)} \right) < 1.25 \right], \tag{38}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{p \in \Omega} (\widetilde{D}(p) - D^{\mathrm{gt}}(p))^2}. \tag{39}$$

Here p indexes both frame and pixel; δ_1 is reported as a fraction. Each RoboStereo depth output is aligned and scored separately.

Camera Alignment and Metrics. We retain frames with finite predicted and ground-truth camera-to-world poses and fit one Umeyama Sim(3) transform between their camera centers per clip. This transform aligns both camera centers and rotations; its scale is estimated independently of depth alignment. ATE is the root mean square distance between aligned and ground-truth camera centers. For consecutive retained timestamps in the sampled clip, let $\widehat{T}_{i \to i+1} = \widehat{T}_i^{-1} \widehat{T}_{i+1}$ and $T^{\mathrm{gt}}_{i \to i+1} = (T^{\mathrm{gt}}_i)^{-1} T^{\mathrm{gt}}_{i+1}$. Translational RPE is the root mean square Euclidean difference between their translation vectors; rotational RPE is the root mean square geodesic rotation difference, in degrees. Thus, RPE uses adjacent sampled observations, which need not be consecutive frames in the source video. ATE and translational RPE are expressed in meters after alignment to the ground-truth trajectory. Ray-map and camera-token trajectories are aligned and evaluated separately.

Aggregation. Within each dataset, clip-level depth metrics are averaged with weights equal to the number of evaluated pixels in each clip. In particular, the reported RMSE is a weighted average of clip RMSE values, rather than a square root taken after pooling squared errors across clips. Camera metrics are averaged uniformly over clips with finite scores.

C Additional Experimental Results

C.1 Auxiliary Outputs

Table 6 reports the auxiliary backbone-depth and camera-token outputs on the Embodiment Stereo Benchmark, complementing the default outputs in Tables 2 and 3. On Infinigen Dynamic, auxiliary backbone depth achieves an AbsRel of 0.0134, compared with 0.0149 for the default disparity-derived depth. We use disparity-derived depth as the default depth output, reflecting the central role of our stereo module, and cameras fitted to the predicted ray maps as the default camera output. These defaults are fixed across datasets and metrics, even when auxiliary backbone-depth or camera-token outputs score better; we do not select outputs based on evaluation scores.

Table 6: **Auxiliary outputs on the Embodiment Stereo Benchmark.** Backbone depth uses clip-level scale alignment; camera-token poses use independent Sim(3) alignment. Dataset abbreviations follow Table 2.

	Backbone depth			Camera-token poses		
Dataset	AbsRel ↓	δ_1 ↑	RMSE ↓	ATE ↓	Trans. RPE ↓	Rot. RPE ↓
RoboCasa†	0.0093	0.9965	0.0521	0.0166	0.0152	0.7918
RoboCasa wrist†	0.0465	0.9683	0.0556	0.0943	0.0598	4.5325
Infinigen-D†	0.0134	0.9896	0.0790	0.0096	0.0083	0.1794
EBench	0.0251	0.9882	0.0943	0.0075	0.0067	0.1816
EBench wrist	0.1404	0.7962	0.1195	0.0112	0.0093	1.5091
SDVLN-W	0.0303	0.9858	1.0807	0.0341	0.0382	0.8620
RoboLab	0.0155	0.9936	0.0916	0.0032	0.0027	0.1052

C.2 Public Dataset Results

Tables 7 and 8 report the complete scale-aligned depth and camera results for the six public datasets evaluated in Section 5. The default disparity-derived depth achieves the lowest AbsRel on four of the six datasets: Sintel, Dynamic Replica, TartanAir, and TartanAir V2. On VKITTI2 and TartanGround, it obtains 0.0219 and 0.0901, respectively, compared with 0.0187 and 0.0872 for WAFT-Stereo → Pi3X. The default ray-map cameras achieve the lowest ATE on Sintel, TartanGround, and

Table 7: **Scale-aligned depth** on public datasets. δ_1 is a fraction; row-best/second-best results are bold/underlined (ranked before rounding). Disparity is the default output; backbone depth is auxiliary. Abbreviations follow Table 2.

		Multi-frame reconstruction			Stereo / depth-conditioned			RoboStereo	
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WAFT-S	WS→P	SAV	Backbone (aux.)	Disparity (default)
AbsRel ↓	Sintel	0.2196	0.2323	0.2611	0.0309‡	0.0706‡	0.0398	0.0463	**0.0295**
	VKITTI2	0.0235	0.0279‡	0.0358‡	0.0196‡	**0.0187**‡	0.0236‡	0.0290	0.0219
	Dynamic Replica†	0.0150	0.0167‡	0.0174‡	0.0077	0.0086	0.0116‡	0.0071	**0.0053**
	TartanGround†	0.1728	0.1059‡	0.1933	0.1114‡	**0.0872**‡	0.1799	0.1020	0.0901
	TartanAir†	0.0492‡	0.0606	0.0784‡	0.0504‡	0.0410‡	0.0769	0.0448	**0.0381**
	TartanAir V2†	0.1119	0.0883‡	0.0934‡	0.0748	0.0606	0.1133	0.0566	**0.0511**
δ_1 ↑	Sintel	0.7636	0.7291	0.7364	0.9802‡	0.9032‡	0.9752	0.9649	**0.9866**
	VKITTI2	0.9838	0.9811‡	0.9784‡	0.9865‡	**0.9865**‡	0.9822‡	0.9841	0.9851
	Dynamic Replica†	0.9941	0.9941‡	0.9946‡	**0.9981**	0.9969	0.9957‡	0.9975	0.9981
	TartanGround†	0.8131	0.9143‡	0.8737	0.9282‡	0.9201‡	0.8393	0.9253	**0.9320**
	TartanAir†	0.9613‡	0.9514	0.9503‡	0.9708‡	0.9658‡	0.9304	0.9662	**0.9714**
	TartanAir V2†	0.8887	0.9247‡	0.9356‡	0.9506	0.9520	0.9026	0.9581	**0.9632**
RMSE ↓	Sintel	3.0874	3.3012	3.3389	1.2156‡	**1.0453**‡	1.6038	1.5265	1.3815
	VKITTI2	3.0746	3.5947‡	4.6567‡	2.7014‡	**2.2179**‡	2.8390‡	3.1187	3.2165
	Dynamic Replica†	0.1202	0.1357‡	0.1348‡	0.0997	0.1089	0.1326‡	0.0959	**0.0942**
	TartanGround†	4.5677	3.9115‡	4.6661	4.1437‡	3.6614‡	4.7709	3.1102	**2.9942**
	TartanAir†	2.4888‡	3.4879	3.9169‡	2.8766‡	2.3901‡	3.7740	2.1978	**2.1196**
	TartanAir V2†	3.0724	2.9843‡	3.9205‡	3.0322	2.4500	3.8338	2.0903	**2.0062**

TartanAir; VGGT-Ω leads on VKITTI2, and DVLT leads on Dynamic Replica and TartanAir V2.

The full translational and rotational RPE results further show that the ranking depends on the dataset and camera metric.

Table 8: **Camera estimation** on public datasets (DVLT: ray maps). Row-best/second-best results are bold/underlined (ranked before rounding). Ray maps give default cameras; camera tokens are diagnostic. Abbreviations follow Table 3.

		Baselines				RoboStereo	
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WS→P	Ray map (default)	Camera token (aux.)
ATE ↓	Sintel	0.0771	0.0665	0.0817	0.0620‡	**0.0407**	0.0438
	VKITTI2	0.0554	**0.0284**‡	0.0619‡	0.0566‡	0.0725	0.0857
	Dynamic Replica†	0.0028	0.0016‡	**0.0015**‡	0.0031	0.0018	0.0020
	TartanGround†	0.1553	0.0298‡	0.0922	0.1731‡	**0.0268**	0.0308
	TartanAir†	0.0318‡	0.0270	0.0361‡	0.0315‡	**0.0227**	0.0268
	TartanAir V2†	0.2125	0.1081‡	**0.0394**‡	0.1621	0.0417	0.0486
Translational RPE ↓	Sintel	0.0443	**0.0365**	0.0426	0.0494‡	0.0413	0.0411
	VKITTI2	0.0507	**0.0304**‡	0.0603‡	0.0494‡	0.0760	0.0720
	Dynamic Replica†	0.0026	0.0016‡	**0.0016**‡	0.0033	0.0019	0.0020
	TartanGround†	0.0970	**0.0238**‡	0.0517	0.0955‡	0.0265	0.0290
	TartanAir†	0.0247‡	**0.0195**	0.0229‡	0.0261‡	0.0195	0.0223
	TartanAir V2†	0.1569	0.0926‡	**0.0379**‡	0.1335	0.0383	0.0438
Rotational RPE ↓	Sintel	0.5191	**0.2631**	1.0940	0.5699‡	0.9215	0.4062
	VKITTI2	0.0475	**0.0400**‡	0.2846‡	0.0530‡	0.3817	0.0930
	Dynamic Replica†	0.0358	**0.0299**‡	0.0487‡	0.0679	0.0666	0.0430
	TartanGround†	0.3512	**0.0887**‡	0.1817	0.3328‡	0.2109	0.1477
	TartanAir†	0.1352‡	**0.1272**	0.2028‡	0.1584‡	0.2127	0.1804
	TartanAir V2†	1.0056	0.4979‡	**0.3081**‡	0.8870	0.3314	0.3339

C.3 SCALE-AND-SHIFT DEPTH ALIGNMENT

Table 9 reports scale-and-shift depth results for the same model evaluated in Section 5. This evaluation examines how allowing an additive depth offset affects the errors, complementing the scale-only results. The alignment fits absolute depth error, so AbsRel, RMSE, and δ_1 need not all improve when a shift is allowed. For example, the default output's AbsRel on TartanGround changes from 0.0901 under scale-only alignment to 0.0906 under scale-and-shift alignment. We report both the default disparity-derived depth and auxiliary backbone depth under scale-and-shift alignment.

Table 9: **Scale-and-shift-aligned depth**. δ_1 is a fraction; row-best/second-best results are bold/underlined (ranked before rounding). Disparity is the default output; backbone depth is auxiliary. Abbreviations follow Table 2.

		Multi-frame reconstruction			Stereo / depth-conditioned			RoboStereo	
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WAFT-S	WS→P	SAV	Backbone (aux.)	Disparity (default)
AbsRel ↓	EBench	0.0623	0.0671	0.0924	0.0284	0.0260	0.0652	0.0241	**0.0214**
	EBench wrist	0.1990	0.2215	0.3354	0.1347	0.2084	0.3140	0.1049	**0.1010**
	SDVLN-W	0.0504	0.0628	0.1024	0.0154	0.0169	0.0325	0.0222	**0.0153**
	RoboLab	0.0336	0.0271	0.0495	0.0172	0.0239	0.0260	0.0130	**0.0098**
	Sintel	0.1464	0.1725	0.1653	0.0297‡	0.0836‡	0.0360	0.0396	**0.0238**
	VKITTI2	0.0208	0.0253‡	0.0332‡	0.0186‡	**0.0172**‡	0.0222‡	0.0273	0.0200
	RoboCasa†	0.0626	0.0388	0.0529	0.0530	0.0274	0.2594	0.0090	**0.0053**
	RoboCasa wrist†	0.2845	0.3153	0.5140	0.1414	0.1670	0.1908	0.0420	**0.0277**
	Infinigen-D†	0.0919	0.0658	0.0437	0.0845	0.1075	0.1641	**0.0134**	0.0141
	Dynamic Replica†	0.0109	0.0142‡	0.0130‡	0.0075	0.0083	0.0113‡	0.0064	**0.0052**
	TartanGround†	0.1300	0.1012‡	0.1681	0.1072‡	**0.0758**‡	0.1667	0.1040	0.0906
	TartanAir†	0.0450‡	0.0593	0.0737‡	0.0492‡	0.0372‡	0.0704	0.0433	**0.0368**
	TartanAir V2†	0.0933	0.0841‡	0.0915‡	0.0722	0.0548	0.1085	0.0555	**0.0496**
δ_1 ↑	EBench	0.9305	0.9161	0.8885	**0.9887**	0.9873	0.9374	0.9879	0.9884
	EBench wrist	0.6980	0.7073	0.6060	0.8192	0.7785	0.6831	**0.8538**	0.8494
	SDVLN-W	0.9626	0.9365	0.8596	**0.9903**	0.9869	0.9741	0.9856	0.9877
	RoboLab	0.9763	0.9794	0.9563	0.9901	0.9873	0.9783	0.9934	**0.9936**
	Sintel	0.8320	0.8018	0.8020	0.9786‡	0.8962‡	0.9752	0.9641	**0.9860**
	VKITTI2	0.9836	0.9810‡	0.9784‡	**0.9864**‡	0.9864‡	0.9821‡	0.9840	0.9850
	RoboCasa†	0.9030	0.9532	0.9426	0.9846	0.9802	0.9377	0.9965	**0.9970**
	RoboCasa wrist†	0.7244	0.7100	0.5906	0.8864	0.8689	0.8356	0.9701	**0.9730**
	Infinigen-D†	0.9744	0.9670	0.9672	0.9729	0.9729	0.9616	0.9877	**0.9881**
	Dynamic Replica†	0.9955	0.9940‡	0.9946‡	**0.9981**	0.9969	0.9956‡	0.9975	0.9981
	TartanGround†	0.8478	0.9145‡	0.8707	0.9284‡	0.9205‡	0.8527	0.9233	**0.9309**
	TartanAir†	0.9569‡	0.9515	0.9502‡	0.9709‡	0.9657‡	0.9302	0.9663	**0.9714**
	TartanAir V2†	0.9088	0.9260‡	0.9391‡	0.9562	0.9514	0.9089	0.9583	**0.9631**
RMSE ↓	EBench	0.1378	0.1432	0.2187	0.1232	**0.0922**	0.2050	0.0938	0.0939
	EBench wrist	0.1525	0.1410	0.2425	0.2065	0.1853	0.3688	**0.1295**	0.1405
	SDVLN-W	1.5259	1.8127	2.9058	**0.9407**	1.0193	1.4755	1.0839	1.0343
	RoboLab	0.1645	0.1584	0.2476	0.4151	0.1604	0.1471	0.0909	**0.0895**
	Sintel	3.8795	3.8690	3.9631	1.2198‡	**1.0507**‡	1.6147	1.5563	1.3971
	VKITTI2	3.0816	3.6157‡	4.7006‡	2.7076‡	**2.2209**‡	2.8463‡	3.1368	3.2257
	RoboCasa†	0.2077	0.1160	0.1394	0.7275	0.2445	2.0443	0.0520	**0.0503**
	RoboCasa wrist†	0.2180	0.2296	0.3107	1.0064	0.3523	0.7051	0.0570	**0.0547**
	Infinigen-D†	0.1740	0.1536	0.1846	0.6228	0.2652	0.2287	0.0797	**0.0790**
	Dynamic Replica†	0.1170	0.1356‡	0.1322‡	0.0997	0.1087	0.1328‡	0.0958	**0.0942**
	TartanGround†	4.8083	3.9590‡	4.9192	4.2078‡	3.7055‡	5.1009	3.2235	**3.0526**
	TartanAir†	2.5293‡	3.5147	3.9724‡	2.8880‡	2.3964‡	3.7988	2.1946	**2.1195**
	TartanAir V2†	3.1745	3.0581‡	4.0439‡	3.0660	2.4702	3.8985	2.0902	**2.0066**

C.4 POINT-CLOUD RECONSTRUCTION

We evaluate each method's predicted world-space point maps, fitting one Sim(3) transform from corresponding points across the clip and applying it to all frames. Depth and cameras are not independently aligned to ground truth, and nearest-neighbor comparisons are restricted to matching timestamps. RoboStereo and WAFT-Stereo → Pi3X use known intrinsics and baseline for stereo-depth conversion, with predicted geometry and poses determining their world-space point maps.

We sample up to 8,192 pixels per frame once from the shared ground-truth validity mask with $0 < D^{\mathrm{gt}} < 80\,\mathrm{m}$ and reuse these locations across all methods. Predicted point maps are mapped to the common ground-truth grid before reading the 3D coordinates at these pixels, without method-specific confidence or error filtering.

Let m denote the median valid ground-truth depth of a clip. F-score@1% uses a distance threshold of $0.01m$ and is reported as a percentage. Normalized Chamfer distance is the mean of the two directional mean nearest-neighbor Euclidean distances, divided by m and multiplied by 100; distances are not squared. Table 10 reports both metrics, with the mean giving equal weight to the thirteen evaluation groups.

Table 10: **Point-cloud reconstruction** across thirteen evaluation groups under one shared Sim(3) alignment per clip. F-score@1% and normalized Chamfer distance are reported in percent; row-best/second-best results are bold/underlined. The mean weights all groups equally. Abbreviations follow Table 2.

		Baselines				RoboStereo
Metric	Dataset	Pi3X	VGGT-Ω	DVLT	WS→P	Ours
F-score@1% ↑	RoboCasa†	53.30	49.90	41.50	<u>69.34</u>	**82.15**
	RoboCasa wrist†	9.49	11.54	8.44	<u>14.86</u>	**31.87**
	Infinigen-D†	83.04	80.61	76.51	<u>84.12</u>	**89.09**
	EBench	40.56	42.25	36.40	**63.34**	<u>50.95</u>
	EBench wrist	6.28	<u>9.15</u>	6.90	4.76	**10.64**
	SDVLN-W	30.25	30.11	22.88	**37.07**	<u>36.40</u>
	RoboLab	51.16	56.78	35.26	**72.12**	<u>67.52</u>
	Sintel	31.69	33.42	22.38	<u>38.24</u>‡	**44.31**
	VKITTI2	63.19	<u>64.14</u>‡	60.09‡	**66.46**‡	52.79
	Dynamic Replica†	80.51	70.95‡	74.10‡	<u>82.80</u>	**87.76**
	TartanGround†	10.03	**30.12**‡	23.46	13.24‡	<u>24.92</u>
	TartanAir†	43.61‡	39.12	30.33‡	**48.46**‡	<u>44.92</u>
	TartanAir V2†	12.91	23.00‡	**32.19**‡	18.66	<u>31.96</u>
	Mean	39.69	41.62	36.19	<u>47.19</u>	**50.41**
Chamfer ↓	RoboCasa†	2.96	<u>2.28</u>	3.96	2.57	**0.94**
	RoboCasa wrist†	<u>35.35</u>	36.37	54.42	42.86	**13.85**
	Infinigen-D†	4.74	<u>1.44</u>	1.88	2.06	**0.67**
	EBench	3.15	2.91	3.68	**1.62**	<u>1.95</u>
	EBench wrist	<u>19.01</u>	22.43	30.64	26.64	**16.71**
	SDVLN-W	4.78	4.95	8.19	**3.15**	<u>3.24</u>
	RoboLab	4.07	3.58	5.72	<u>1.84</u>	**1.44**
	Sintel	13.52	15.93	16.43	<u>5.25</u>‡	**5.24**
	VKITTI2	<u>1.70</u>	1.87‡	2.29‡	**1.46**‡	2.18
	Dynamic Replica†	0.76	0.93‡	0.89‡	<u>0.70</u>	**0.57**
	TartanGround†	22.49	<u>11.98</u>‡	16.50	16.35‡	**9.95**
	TartanAir†	3.71‡	9.66	5.24‡	<u>3.59</u>‡	**3.29**
	TartanAir V2†	8.94	6.94‡	<u>6.43</u>‡	6.76	**4.68**
	Mean	9.63	9.33	12.02	<u>8.83</u>	**4.98**

C.5 Additional Multi-Model Qualitative Comparisons

Figure 4 complements Figure 3 with four additional hand–object interaction examples using the same model order. These views focus on local surfaces and boundaries around the hands and manipulated objects.

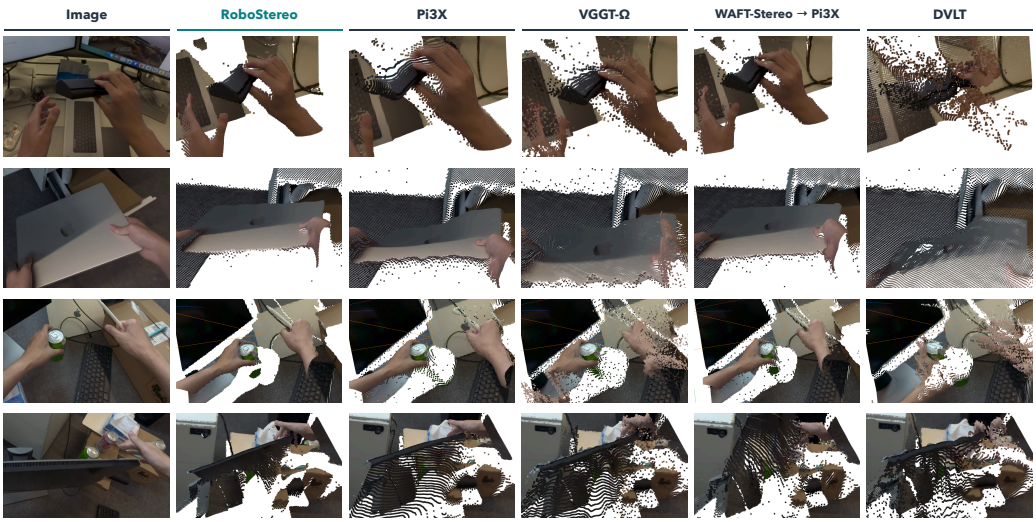

Figure 4: **Additional qualitative comparisons of hand–object reconstruction.** From top to bottom: fingers around a black object, a laptop edge, hands holding a can and a thin object, and the hand and monitor edge. Within each row, all methods use the same point-cloud crop window and display size, retaining their original rendering viewpoints without additional point filtering or registration.

C.6 Depth Accuracy and Reconstruction Consistency

Disparity predicted by WAFT-Stereo is converted to depth and supplied to Pi3X through its depth encoder. These predicted depths differ in distribution from the encoder's training inputs, and temporal inconsistencies in this conditioning depth affect Pi3X's reconstruction. Despite the aggregate depth improvements in Table 2, the reconstructed point clouds still exhibit cross-frame inconsistency, visible as layered background screen surfaces and duplicated edges in Figure 5.

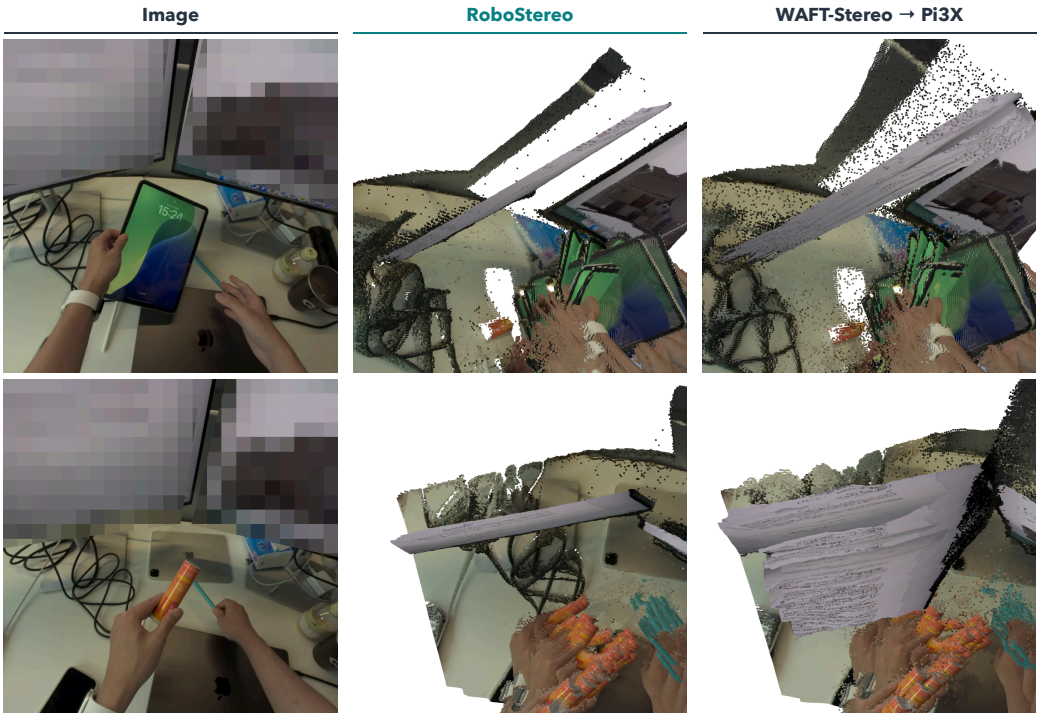

Figure 5: **Reconstruction consistency under stereo-depth conditioning.** From left to right: RGB reference, RoboStereo, and WAFT-Stereo → Pi3X. Close-ups show layered surfaces and separated edges of the left background monitor. Screen content in the RGB references is pixelated for privacy; point-cloud crops retain their original pixels. Both methods use the same crop window and display size within each row, retaining their original rendering viewpoints without additional point filtering or registration.

C.7 Multi-Frame Context for Stereo Geometry

WAFT-Stereo processes each stereo pair independently, without jointly using multi-frame observations or explicitly enforcing temporal consistency (Wang & Deng, 2026). Weakly textured surfaces provide few distinctive correspondence cues, leaving pairwise matching susceptible to local geometric errors. In Figure 6, WAFT-Stereo predicts incorrect depth for part of the cabinet (top), the brown cardboard box (middle), and the white pen (bottom).



Figure 6: **Stereo geometry with multi-frame context.** From left to right: RGB reference, RoboStereo, and WAFT-Stereo. Close-ups highlight depth errors in part of the cabinet (top), the brown cardboard box (middle), and the white pen (bottom). Both methods use the same crop window and display size within each row, retaining their original rendering viewpoints without additional point filtering or registration.