X-LENS: REAL-TIME METRIC DEPTH ESTIMATION WITH HETEROGENEOUS CAMERAS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present X-Lens, a compact feed-forward model for metric depth estimation from a variable number of distorted and pinhole views. To support real-time downstream perception, X-Lens is built around a geometry-aware heterogeneous camera formulation with two key components. Learnable calibration tokens provide a coarse alignment between fisheye and pinhole projective spaces, while a Jacobian-parameterized distortion bias injected into cross-attention models local projection changes and promotes cross-camera consistency, enabling robust generalization with only 0.04B parameters and up to 41 FPS. The model predicts dense depth together with a global metric scale, avoiding auxiliary reconstruction targets that increase computation and optimization complexity. To facilitate cross-camera generalization at scale and depth, X-Lens is trained on multiple public datasets and OmniScene, our newly curated large-scale synthetic dataset containing 266K synchronized six-view frames, around 1.6M individual images, and 126 indoor and outdoor scenes. Extensive experiments demonstrate superior heterogeneous-camera metric depth accuracy on our OmniScene dataset, alongside optimal or competitive performance on real-world fisheye and pinhole-only benchmarks. On OmniScene-Full, X-Lens reduces AbsRel by 25.4% over the strongest baseline while using 88.9% fewer parameters.

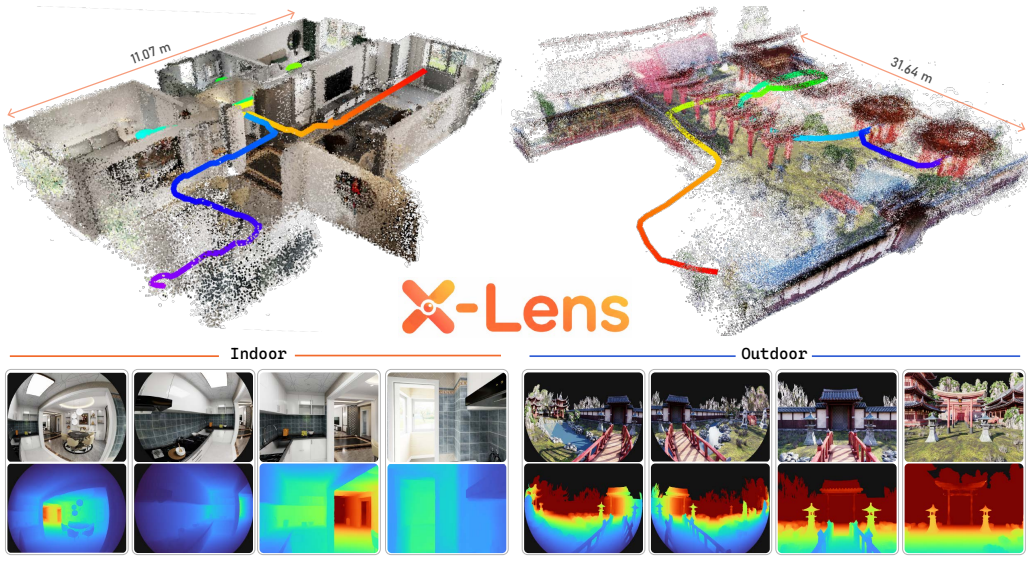


Figure 1: Visualization of X-Lens's prediction. The top row shows metric point clouds projected from calibrated six-view heterogeneous cameras for two scenes with ground-truth extrinsics, demonstrating metric depth estimation across mixed camera rigs. The bottom row shows RGB inputs and predicted depth maps from heterogeneous multi-view pinhole and fisheye cameras.

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.

1 INTRODUCTION

The ability to perceive 3D spatial information is central to spatial intelligence (Zhen et al., 2024; Zheng et al., 2024), enabling persistent world models (Zheng et al., 2024; Bruce et al., 2024), physical reasoning (Wang et al., 2024b; 2025a), and safe action (Wang et al., 2023; Brohan et al., 2023). As autonomous robots (Brohan et al., 2023; Octo Model Team et al., 2024), vision-language-action (VLA) models (Brohan et al., 2023; Kim et al., 2024; Zhen et al., 2024), and physical world models (Zheng et al., 2024; Bruce et al., 2024) increasingly require fine-grained spatial reasoning and long-horizon execution (Zhen et al., 2024; Wang et al., 2023), they demand accurate geometry for interaction (Kim et al., 2024), generalization (Zhen et al., 2024), and robust behavior (Zheng et al., 2024). Multi-camera 3D perception meets this need by expanding coverage, reducing occlusion, and providing complementary cues: robots combine workspace and wrist-mounted cameras, whereas autonomous systems use surround-view rigs for wide-area perception and occupancy reasoning (Zheng et al., 2024; Wang et al., 2023). Pinhole cameras preserve distant detail, while wide-FOV or fisheye cameras capture near-field and peripheral regions.

However, existing 3D and depth foundation models are not suitable for heterogeneous camera rigs. Most depth estimation methods (Bhat et al., 2023; Yin et al., 2023; Piccinelli et al., 2024; Yang et al., 2024) and multi-view geometric methods (Wang et al., 2024b; Leroy et al., 2024; Wang et al., 2025a; 2026c; Keetha et al., 2026; Wang et al., 2026a) assume a single camera model or pinhole-centric geometry, while their reconstruction-driven objectives are often difficult to satisfy in real time. In mixed fisheye–pinhole rigs, view-specific projections and distortions produce heterogeneous ray distributions, hindering cross-view correspondence and metric-scale alignment. Panorama-based approaches (Jiang et al., 2021; Shen et al., 2022; Jiang et al., 2025a; Lin et al., 2026b) simplify omnidirectional processing, but alter native image geometry and underuse co-visible cues from calibrated views. Meanwhile, geometric foundation models (Wang et al., 2024b; Leroy et al., 2024; Wang et al., 2025a; 2026c; Keetha et al., 2026; Wang et al., 2026a) remain computationally demanding for edge-side deployment. These limitations motivate a compact formulation that natively supports heterogeneous multi-view inputs while estimating depth and metric scale.

To address these challenges, we present **X-Lens**, a feed-forward model for metric depth estimation from calibrated heterogeneous camera rigs, with representative predictions shown in Figure 1. Given arbitrary fisheye and pinhole views in their native image domains, X-Lens predicts dense depth maps and a global scale factor through a compact DINOv2 (Oquab et al., 2024) backbone, DPT (Ranftl et al., 2021) depth head, and scale-regression MLP. Rather than depending on large-scale pre-trained backbones (Oquab et al., 2024; Siméoni et al., 2025), the model learns transferable depth representations through staged training on calibrated heterogeneous multi-view data. The training process first establishes robust pinhole depth priors, then adapts them to wide-FOV observations with learnable calibration tokens. To further bridge camera-specific projection geometries, we inject a distortion bias derived from patch-center Jacobians and local ray directions into global cross-attention, allowing attention weights to account for local projection changes across views. This design improves cross-camera feature consistency while preserving each camera's native sampling pattern, avoiding panorama conversion, and eliminating auxiliary reconstruction targets that are unnecessary for metric depth inference.

Furthermore, to overcome the scarcity of heterogeneous-camera depth training corpora, we construct a large-scale synthetic dataset with 266,000 frames across 126 indoor and outdoor scenarios. It provides diverse scene layouts, high-quality rendering, and calibrated camera configurations for omnidirectional and heterogeneous depth perception. X-Lens generalizes to held-out scenes and varying fisheye intrinsics on OmniScene, and demonstrates zero-shot cross-dataset transfer on ETH3D and OmniOcc datasets. Figure 2 summarizes this balance of accuracy and efficiency across fisheye, pinhole, and heterogeneous-camera settings.

In summary, our main contributions are as follows:

- **Efficient Heterogeneous Camera Depth Estimation Model:** We propose X-Lens, a lightweight feed-forward metric depth estimation model with only 0.04B parameters that natively supports multi-view heterogeneous inputs without panoramic stitching.
- **Large-scale Dataset:** We curate a high-quality synthetic heterogeneous dataset comprising 266,000 frames across 126 diverse scenarios, with comprehensive metric depth annotations and enhanced visual diversity and realism.

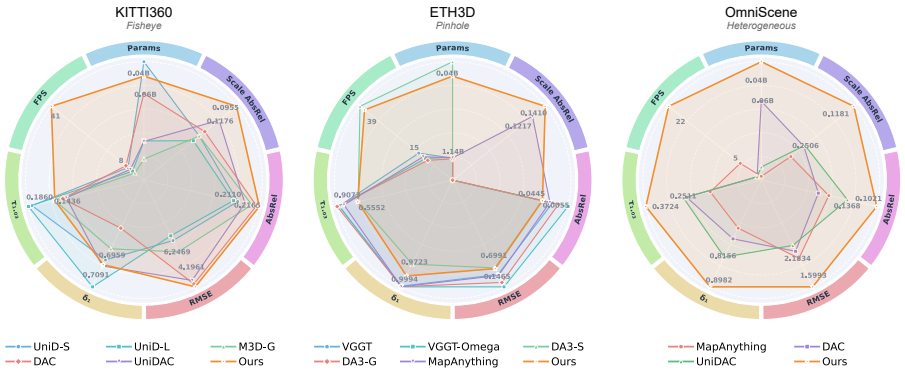


Figure 2: Radar-chart comparison across fisheye, pinhole, and heterogeneous-camera benchmarks. X-Lens consistently delivers strong accuracy and efficiency across all camera settings.

- **Superior Zero-shot Generalization and Real-Time Performance:** Extensive experiments on both synthetic and real-world environments show that X-Lens achieves state-of-the-art accuracy with strong zero-shot generalization. Its compact design also offers high-frame-rate inference with extreme low latency, facilitating edge-side deployment.

2 Related Works

Wide-FOV Depth Estimation. Wide-FOV depth estimation has developed along two main directions, panoramic depth estimation and direct wide-angle processing. Panorama-based methods project inputs into an Equirectangular Projection and estimate depth on the spherical image domain. Early dual-projection methods (Wang et al., 2020; Jiang et al., 2021) fused equirectangular and cubemap features to reduce polar distortion, while HoHoNet (Sun et al., 2021), PanoFormer (Shen et al., 2022), and EGformer (Yun et al., 2023) introduced layout cues, tangent-plane partitioning, and spherical geometry biases for stronger panoramic depth estimation. Recent panoramic foundation models further adapt large monocular depth backbones to wide-FOV inputs, with DepthAnything in 360° (Jiang et al., 2025a), DepthAnything in Any Direction (Li et al., 2025), DepthAnyCamera (Guo et al., 2025), UniDAC (Ganesan et al., 2026), and DepthAnyPanoramas (Lin et al., 2026b) improving scale-invariant, camera-agnostic, and zero-shot metric panoramic depth. Direct wide-angle methods instead avoid full panorama conversion and estimate depth in fisheye or multi-fisheye image spaces, including distortion-aware monocular depth (Tateno et al., 2018; Kumar et al., 2020), spherical cost-volume stereo (Won et al., 2019; Meuleman et al., 2021), and learning-based multi-fisheye systems such as OmniMVS (Won et al., 2019), Unsupervised OmniMVS (Chen et al., 2023), and CasOmniMVS (Wang et al., 2024a). Recent wide-FOV studies also continue to improve fisheye benchmarks (Indyk et al., 2026), camera-model conditioning (Jung et al., 2026), and generalization across unseen camera types (Guo et al., 2025; Lin et al., 2026b). These methods mainly rely on panorama resampling, single-view wide-FOV inference, or hardware-specific stereo volumes, while X-Lens keeps the input in its native camera views and learns cross-view depth fusion without converting the observations into a stitched panorama.

Feed-forward Geometric Models. Recent feed-forward geometry foundation models provide a new route for multi-view depth estimation by learning dense 3D structure directly from image collections. DUSt3R (Wang et al., 2024b) introduced pointmap regression in a shared coordinate frame for dense matching, pose recovery, and depth reasoning, and MASt3R (Leroy et al., 2024) further grounded image matching in this 3D representation. Subsequent works extend this feed-forward paradigm across view count and scene dynamics, with Spann3R (Wang & Agapito, 2025) accumulating geometry with spatial memory, MonST3R (Zhang et al., 2025) handling dynamic scenes, and Fast3R (Yang et al., 2025) targeting large image collections. VGGT (Wang et al., 2025a) further unified depth, point maps, intrinsics, and poses in a single visual geometry Transformer. π^3 (Wang et al., 2026c) then studied scalable permutation-equivariant visual geometry learning. MapAnything (Keetha et al., 2026) extended this direction toward universal feed-forward metric 3D reconstruction. DepthAnything3 (Lin et al., 2025) further extends monocular depth priors to

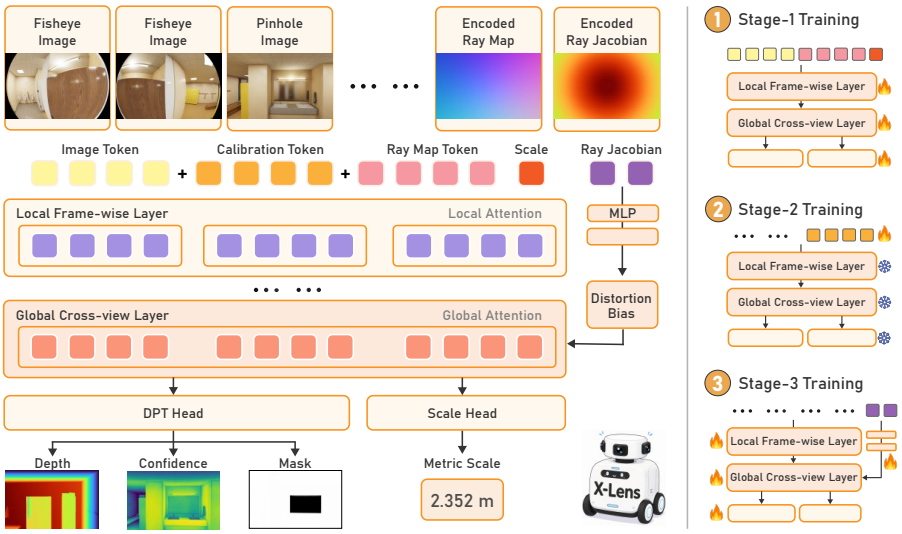


Figure 3: Overview of the proposed X-Lens pipeline. Heterogeneous multi-view inputs are represented through generic camera rays and processed by an omnidirectional heterogeneous projection transformer with calibration tokens and distortion-aware attention, followed by factored prediction heads for normalized depth, confidence, and global metric scale.

any-view visual-space recovery. Recent feed-forward multi-view depth and reconstruction studies also continue to improve generality under sparse-view (Chen et al., 2026; Lin et al., 2026a; Burzio et al., 2026), synthetic-data (Ma et al., 2026), long sequences (Jin et al., 2026; Xie et al., 2026) and large training corpora (Wang et al., 2026a), as well as high-fidelity reconstruction (Jiang et al., 2025b; Zhao et al., 2026) and wide-FOV camera models (Jung et al., 2026). These works provide strong priors for multi-view geometry, but most are still optimized for general reconstruction where depth, pose, intrinsics, and point maps are inferred jointly. For heterogeneous multi-view rigs, such coupling and the common pinhole-centric design make it difficult to simultaneously and robustly support fisheye cameras and mixed sensor configurations. X-Lens instead focuses on heterogeneous multi-view metric depth with a compact perception-oriented design, making real-time deployment practical.

3 Method

We present **X-Lens**, a feed-forward multi-view depth estimation model that predicts an up-to-scale depth field, an associated confidence map, and one global metric scalar. We first formulate heterogeneous cameras with a generic ray representation and define the factored depth output in §3.1. We then describe the heterogeneous projection transformer that combines calibration tokens with distortion-aware cross-view attention in §3.2. Finally, we present the three stage training pipeline and optimization objective in §3.3. Figure 3 provides an overview of the proposed pipeline and training strategy. Additional method details are provided in the appendix.

3.1 Formulation and Design Principles

Generic Heterogeneous Camera Representation Existing 3D depth and foundation models are commonly centered on the pinhole camera model. They ingest a linear intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ and assume a one-to-one correspondence between image coordinates and 3D rays via $\mathbf{r} \propto \mathbf{K}^{-1}[u, v, 1]^\top$. However, this assumption fails under strongly non-linear lenses such as fisheye, where the projection is governed by a parametric distortion vector rather than a few scalars. To handle multi-view heterogeneous configurations, X-Lens abstracts cameras away from specific projection models and replaces the pinhole intrinsic with a generic unprojection map $\mathcal{G}$. Each view s is formally specified as:

$$\mathbf{r}_{s,p} = \mathbf{R}_s\, \mathcal{G}(p; \boldsymbol{\xi}_s, \tau_s), \qquad \mathbf{r}_{s,p} \in \mathbb{S}^2,\ p \in [0, W] \times [0, H], \tag{1}$$

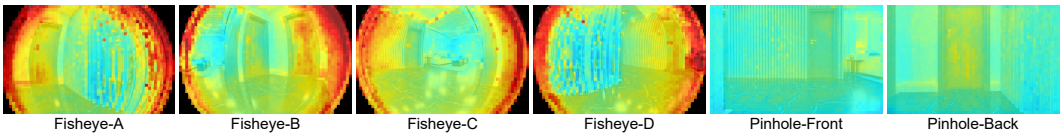


Figure 4: Visualization of the bias-correction magnitude from $\mathbf{B}$ in Equation (4). Fisheye views require larger corrections in highly distorted boundary regions, while pinhole views show smoother and more spatially uniform corrections.

where $\mathcal{G}$ maps a continuous pixel coordinate p to a camera-frame unit ray represented as a column vector. It absorbs both the per-camera calibration $\boldsymbol{\xi}_s$, including focal lengths, principal point, and distortion coefficients, and an explicit camera-type indicator τ_s, with $\tau_s \in \{\text{pinhole}, \text{fisheye}\}$. $\mathbf{R}_s \in SO(3)$ rotates this ray from the camera frame into the shared rig frame.

Crucially, all downstream geometric reasoning inside the network is conducted in the ray space $\{\mathbf{r}_{s,p}\}$ rather than the pixel coordinate space. For cameras without a closed-form projection model, we implement $\mathcal{G}$ using a tabulated radial profile sampled during training. Consequently, $\mathcal{G}$ can be flexibly specified per dataset and per sensor modality. Without any architectural modifications, the same network can directly ingest heterogeneous mixtures of pinhole, fisheye, and 360° cameras.

Factored Depth Prediction Given multiple heterogeneous views $\mathcal{I} = \{I_s \in \mathbb{R}^{3\times H\times W}\}_{s=1}^{S}$ and their corresponding calibration parameters $\{\boldsymbol{\xi}_s, \mathbf{T}_s\}_{s=1}^{S}$, X-Lens predicts a factored triplet:

$$\left[\hat{D},\ \hat{C},\ \hat{m}\right], \quad \hat{D} \in \mathbb{R}_{>0}^{S\times H\times W},\ \hat{C} \in \mathbb{R}_{>1}^{S\times H\times W},\ \hat{m} \in \mathbb{R}_{>0}, \tag{2}$$

where $\hat{D}$ is the z-depth field normalized by its mean over valid pixels across all S views of each sample, independently of other batch samples. $\hat{C}$ is the self-calibrated confidence map, and $\hat{m}$ is one global scalar per sample. Euclidean fisheye ground truth is converted to z-depth for evaluation. Metric depth is reconstructed as $\hat{D}^{\text{m}} = \hat{m} \cdot \hat{D}$. The shared DPT head also predicts a validity mask $\hat{M} \in [0,1]^{S\times H\times W}$, supervised by ground-truth validity labels for sky and pixels outside the physical lens mask. Ground-truth masks, rather than $\hat{M}$, select valid pixels for geometric supervision in eq. (6); $\hat{M}$ is not part of the metric output.

3.2 Heterogeneous Projection Transformer

The core of X-Lens is built upon a Vision Transformer (Dosovitskiy et al., 2021) backbone with alternating within-view layers L_s and cross-view layers L_g for local feature extraction and global context aggregation. Since standard multi-view attention does not model the heterogeneous imaging topologies of pinhole and fisheye cameras, we introduce ***Multi-View Calibration Tokens*** to absorb lens-specific distortion, inspired by calibration-token adaptation (Gangopadhyay et al., 2025), and ***Jacobian Distortion Bias*** to inject cross-lens geometric correspondence priors into global attention.

Multi-View Calibration Tokens Heterogeneous inputs carry distinct per-lens distortion patterns. To keep these from being absorbed indiscriminately by the shared visual tokens, we introduce a learnable calibration-token tensor $\boldsymbol{\Theta} \in \mathbb{R}^{N_L\times T\times K\times C}$ indexed by Transformer layer N_L, camera type T, token count K, and dimension C. At each layer i, the type-specific slice $\boldsymbol{\Theta}[i, \tau_s]$ is appended to view s's token sequence, attends, and is then dropped before the next layer re-injects a fresh slice, forming an inject-attend-drop process. A slice is injected at every layer, including the cross-view layers. Within a cross-view layer, an attention mask keeps each token view-local, so it attends only to its own view and adds no camera-type signal to the global metric fusion. The drop serves a separate purpose: it prevents the tokens from persisting across layers. Tokens are injected only for fisheye views, leaving the pinhole pathway unchanged, and $\boldsymbol{\Theta}$ is zero-initialized so that corrections are learned gradually from an identity start.

Jacobian Distortion Bias Calibration tokens provide view-local adaptation for lens-specific distortions, whereas reliable heterogeneous matching also requires cross-view geometric compatibility.

We therefore introduce the Jacobian Distortion Bias in the cross-view attention layers L_g. The bias is applied to all patch tokens and acts as a geometry-aware prior that modulates attention according to local ray orientation and projection distortion.

JDB reuses the existing ray field $\{\mathbf{r}_{s,p}\}$ from eq. (1) and downsamples it to the ViT patch grid, yielding a unit ray $\mathbf{r}_i \in \mathbb{S}^2$ for each patch i. The local unprojection Jacobian $\mathbf{J} \in \mathbb{R}^{3\times 2} = [\partial \mathbf{r}/\partial u, \partial \mathbf{r}/\partial v]$ is estimated directly by finite differences on this patch grid, avoiding repeated Jacobian computation at the original image resolution. The scalar $s = \|\mathbf{J}\|_F$ captures local spatial expansion, while the full Jacobian retains anisotropic stretching and orientation cues that are important near strongly distorted fisheye regions.

For patch tokens indexed by i and j, we form a relative descriptor $\boldsymbol{\phi}_{ij} \in \mathbb{R}^9$:

$$\boldsymbol{\phi}_{ij} = \Big[\mathbf{r}_i \cdot \mathbf{r}_j,\ \log\|\mathbf{J}_i\|_F - \log\|\mathbf{J}_j\|_F,\ (\mathbf{r}_i - \mathbf{r}_j)^\top,\ \mathrm{vec}(\mathbf{J}_i^\top \mathbf{J}_j)^\top\Big]^\top, \tag{3}$$

which combines angular consistency, relative local scale, ray displacement, and Jacobian correlation. A lightweight head-specific MLP converts this descriptor into an additive bias for the k-th attention head:

$$\mathrm{Attention}^{(k)}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}^{(k)}(\mathbf{K}^{(k)})^\top}{\sqrt{d_k}} + \mathbf{B}^{(k)}\right)\mathbf{V}^{(k)}, \tag{4}$$

$$\mathbf{B}_{ij}^{(k)} = \mathrm{MLP}^{(k)}(\boldsymbol{\phi}_{ij}), \tag{5}$$

where $\mathbf{B}^{(k)}$ is added before Softmax normalization. This formulation encourages cross-view attention to favor patches with compatible 3D ray geometry and local projection structure, rather than relying solely on appearance similarity in distorted image coordinates.

3.3 Multi-Stage Heterogeneous Training Strategy

X-Lens is trained with a progressive three-stage pipeline that transitions from homogeneous pinhole geometry to fisheye adaptation and, finally, heterogeneous multi-camera optimization. In Stage 1, we pre-train the base network on multi-view pinhole data to learn a stable geometric representation for depth, confidence, and global-scale estimation. In Stage 2, we adapt only the calibration tokens $\boldsymbol{\Theta}$ to omnidirectional fisheye inputs while freezing the shared backbone and prediction heads. This controlled adaptation captures radial compression and nonlinear projection effects without overwriting the geometric priors learned from pinhole data. In Stage 3, we activate the Jacobian Distortion Bias and jointly fine-tune the model on mixed fisheye–pinhole data together with pure-pinhole samples. This final stage learns to reconcile heterogeneous image-plane distortions in the shared ray-space representation while preserving pinhole multi-view performance. Detailed training schedules, parameter-freezing strategies, and dataset compositions are provided in the appendix.

Across all stages, we optimize a multi-task objective that separates scale-invariant geometric learning from metric-scale supervision:

$$\mathcal{L} = \lambda_{\text{depth}}\mathcal{L}_D + \lambda_{\text{grad}} \sum_{k\in\{x,y\}} \left\| \nabla_k \frac{\hat{D}}{\tilde{D}} - \nabla_k \frac{D}{\tilde{D}} \right\|_1 + \lambda_{\text{local}}\mathcal{L}_{\text{local}} + \lambda_{\text{scale}} \frac{|\hat{m} - m|}{m} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}, \tag{6}$$

where Ω denotes pixels valid under the ground-truth mask, ∇_x, ∇_y represent horizontal and vertical finite difference operators. The scaling factor m provides the absolute physical metric anchor. The loss hyper-parameters are balanced via coefficients $\lambda_{\text{depth}} = 1.0$, $\lambda_{\text{grad}} = 1.0$, $\lambda_{\text{local}} = 0.5$, $\lambda_{\text{scale}} = 1.0$, and $\lambda_{\text{mask}} = 0.2$. The full definitions of losses, loss weights, and optimization settings are deferred to the appendix.

4 The OmniScene Dataset

To bridge the gap in multi-camera geometric perception, we introduce **OmniScene**, a large-scale synthetic dataset for multi-view metric depth estimation with heterogeneous cameras. OmniScene is rendered using a calibrated six-camera rig with four fisheye and two pinhole views, providing synchronized observations across complementary projection models. It addresses the limited availability of wide-FOV fisheye data and the lack of heterogeneous multi-camera benchmarks with

Table 1: Heterogeneous-camera evaluation OmniScene-Full with six input views (4 fisheye + 2 pinhole). FPS is measured at 504×798 resolution. Colors indicate best, second, and third; ties share a color.

Dataset	Views	Method	Params ↓	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
OmniScene-Full	6	MapAnything	1.23B	0.3701	0.1746	2.1834	0.7357	0.1647	5
		DepthAnyCamera	0.06B	0.2571	0.2066	2.3981	0.7653	0.2411	1
		UniDAC	0.36B	0.2506	0.1368	2.6125	0.8156	0.2511	1
		Ours	0.04B	0.1181	0.1021	1.5993	0.8982	0.3724	22

dense metric supervision. Built from professionally authored Kujiale[1] and Unreal Engine assets, OmniScene covers diverse indoor and outdoor environments.

OmniScene contains **126** complex scenes, **774** randomized motion sequences, and approximately **266K** synchronized multi-view frames, corresponding to approximately **1.6M** images, as shown in Figure 7. Four $180°$ fisheye cameras following the Kannala–Brandt model capture overlapping surround context, while front- and rear-facing pinhole cameras provide complementary perspective observations. All cameras are rendered synchronously at 504×798 resolution and are accompanied by calibrated intrinsics and extrinsics under the OpenCV convention. Each frame provides dense, noise-free metric depth, validity masks, sky indicators, and camera poses, enabling supervised learning and evaluation without manual rectification. Training and validation use different sequences from shared scenes, while test scenes are held out from both splits to assess generalization to unseen environments. Further details on data construction, trajectory generation, annotation processing, and quality control are provided in the appendix.

5 Experiments

Main-paper baselines receive no additional fine-tuning; implementation details are in the appendix, including fine-tuning comparisons in Section D.5.

Datasets & Metric. We evaluate heterogeneous cameras on synthetic OmniScene and homogeneous fisheye cameras on real and synthetic data. KITTI360 (Liao et al., 2022) is evaluated at 504×504 in monocular and two-view settings; the latter uses frames separated by 6 timestamps along one trajectory. At 504×798, OmniScene-Quad uses four fisheye views, while OmniScene-Full combines four fisheye and two pinhole views. We report Scale AbsRel, AbsRel, RMSE, δ_1, $\tau_{1.03}$, and FPS. Scale AbsRel measures the error of the predicted global scale, while AbsRel, RMSE, and δ_1 follow standard depth-estimation protocols. $\tau_{1.03}$ measures the fraction of pixels whose relative depth error is within 3%. OmniScene test scenes are excluded from both training and validation; benchmark-specific splits are detailed in Section C.3. All main-paper FPS measurements use a single NVIDIA H100 GPU, with monocular baselines processing views sequentially at batch size 1. Full-view batched inference for these baselines is detailed in Section D.7, and RDK S600 edge deployment in Section D.6.

5.1 Heterogeneous Cameras

We first evaluate X-Lens under the heterogeneous setting, where fisheye and pinhole cameras are jointly used as input. This setting assesses whether a compact feed-forward model can fuse mixed camera views while preserving metric consistency. As shown in Table 1, X-Lens achieves the best performance on OmniScene-Full across all reported metrics, with the lowest AbsRel and RMSE. Compared with MapAnything (Keetha et al., 2026), X-Lens lowers Scale AbsRel by 68.1%, uses 96.7% fewer parameters, and runs more than four times faster. DepthAnyCamera (Guo et al., 2025) and UniDAC (Ganesan et al., 2026) are monocular any-camera methods and are therefore applied independently to each view. Without cross-view fusion, they lag behind X-Lens in both accuracy and speed. These results demonstrate that explicitly modeling heterogeneous multi-view depth provides a practical path toward real-time metric perception on mixed fisheye and pinhole rigs. Figure 9 visualizes one fisheye view and one pinhole view from the same six-view OmniScene-Full sample,

[1] https://www.kujiale.com/

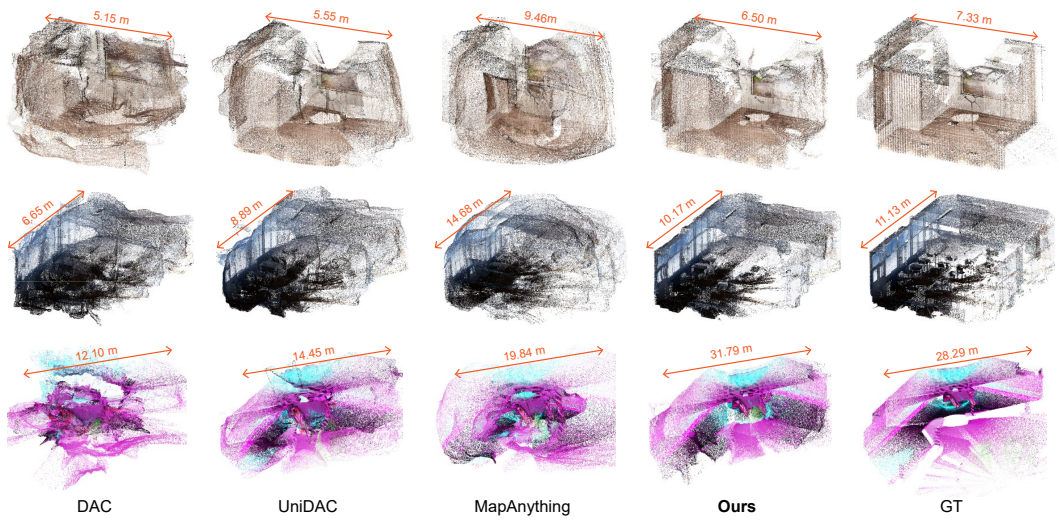


Figure 5: Point-cloud comparison on the synthetic OmniScene-Full setting with six multi-view inputs, consisting of four fisheye and two pinhole views. Compared with DepthAnyCamera (Guo et al., 2025), UniDAC (Ganesan et al., 2026), and MapAnything (Keetha et al., 2026), X-Lens reconstructs more accurate geometry and preserves metric scale consistently.

Table 2: Fisheye evaluation on one- and two-view KITTI360 (Liao et al., 2022) and four-view OmniScene-Quad. FPS is reported at the native evaluation resolution of each dataset. Colors follow Table 1, ranked within each input setting.

Dataset	Views	Method	Params ↓	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
KITTI360	1	UniDepthv2-Small	0.03B	0.1597	0.2597	7.4067	0.6931	0.1803	7
		UniDepthv2-Large	0.35B	0.1787	0.2718	8.0432	0.7091	0.1860	5
		Metric3Dv2-Small	0.03B	0.2631	0.2690	6.4410	0.6132	0.1477	8
		Metric3Dv2-Giant	1.38B	0.1598	0.2258	6.2469	0.6865	0.1434	4
		DepthAnyCamera	0.06B	0.1453	0.2165	4.3103	0.6742	0.1307	8
		UniDAC	0.36B	0.1176	0.2413	4.4671	0.6967	0.1251	6
		Ours	0.04B	0.0955	0.2110	4.1961	0.6959	0.1436	41
KITTI360	2	UniDepthv2-Small	0.03B	0.1592	0.2595	7.4066	0.6942	0.1807	3
		UniDepthv2-Large	0.35B	0.1781	0.2699	8.0433	0.7097	0.1868	2
		Metric3Dv2-Small	0.03B	0.2633	0.2691	6.4408	0.6132	0.1435	4
		Metric3Dv2-Giant	1.38B	0.1598	0.2257	6.2469	0.6866	0.1434	2
		DepthAnyCamera	0.06B	0.1454	0.2163	4.3103	0.6743	0.1307	4
		UniDAC	0.36B	0.1175	0.2413	4.4673	0.6968	0.1254	3
		Ours	0.04B	0.0832	0.2069	4.0433	0.7014	0.1557	41
OmniScene-Quad	4	MapAnything	1.23B	1.0455	0.1918	2.1602	0.6827	0.1178	6
		DepthAnyCamera	0.06B	0.2681	0.1963	2.8911	0.7950	0.2771	2
		UniDAC	0.36B	0.2719	0.1869	2.8903	0.7693	0.2691	1
		Ours	0.04B	0.1268	0.1138	1.6481	0.8822	0.3328	24

showing cross-camera depth consistency across near-field wide-FOV observations and long-range pinhole observations. Figure 5 further compares multi-view point clouds on OmniScene-Full, where X-Lens preserves the scene geometry and metric scale more faithfully than DepthAnyCamera (Guo et al., 2025), UniDAC (Ganesan et al., 2026), and MapAnything (Keetha et al., 2026).

5.2 Homogeneous Cameras

We evaluate homogeneous inputs to verify that X-Lens remains robust beyond its primary heterogeneous setting. Pinhole results and complete fisheye-only and pinhole-only ablations are provided in Appendix Table 12. Figure 6 visualizes real fisheye-only and pinhole-only predictions.

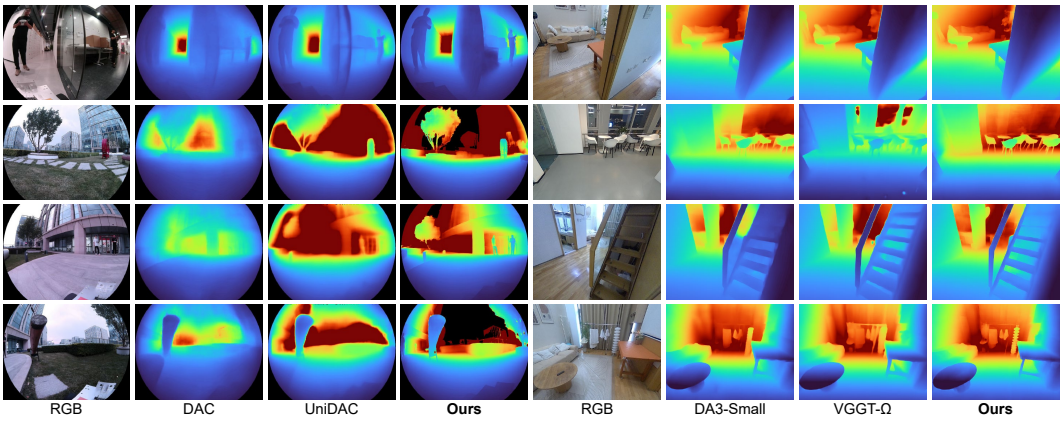


Figure 6: Qualitative results on the real-world pinhole scenes and real-world fisheye scenes using X-Lens. The figure shows fisheye-only and pinhole-only results across outdoor and indoor scenes.

Table 3: Ablation of training stages and geometry modules on the primary heterogeneous setting, OmniScene-Full (four fisheye and two pinhole cameras). Additional fisheye-only and pinhole-only ablations are reported in Appendix.

Variant	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑
S1: pinhole pre-training	0.3026	0.7563	2.4716	0.4811	0.1156
S1→S2: calibration adaptation	0.3308	0.3952	2.2159	0.6264	0.1331
S1→S2→S3, frozen	0.2886	0.2333	2.6006	0.7436	0.1743
Full model w/o JDB	0.1801	0.1912	2.5437	0.7812	0.2102
Full X-Lens	0.1181	0.1021	1.5993	0.8982	0.3724

Table 2 reports monocular and two-view KITTI360 (Liao et al., 2022) results and four-view OmniScene-Quad. On KITTI360, X-Lens obtains the best Scale AbsRel, AbsRel, and RMSE while running at 41 FPS; two views reduce Scale AbsRel by 12.9% and RMSE by 3.6% over one view. On OmniScene-Quad, X-Lens leads all reported accuracy metrics at 24 FPS and uses 96.7% fewer parameters than MapAnything (Keetha et al., 2026). These results support efficient raw-fisheye processing without panorama stitching or hardware-specific stereo volumes.

6 Ablation Study

We ablate the three-stage curriculum and the Jacobian Distortion Bias on the primary heterogeneous OmniScene-Full setting. Our loss terms follow DepthAnything3 (Lin et al., 2025), with the local depth consistency term adopted from MoGe (Wang et al., 2025b). Here, S1–S3 follow the training stages in Section 3.3; *frozen* applies to the backbone and DPT decoder in S3.

Table 3 shows that calibration adaptation alone is insufficient; joint optimization with JDB yields the best heterogeneous results. Removing JDB also degrades pinhole scale accuracy relative to S1 (Table 12), suggesting that projection-aware attention helps preserve metric geometry during heterogeneous training.

7 Conclusion

We present X-Lens, a compact feed-forward model for real-time metric depth estimation from heterogeneous fisheye and pinhole views. Calibration tokens and a Jacobian-parameterized bias reconcile projection differences while preserving native image geometry. This 0.04B-parameter model predicts dense depth and global scale, reducing AbsRel by 25.4% over the strongest baseline with 88.9% fewer parameters on OmniScene-Full, while remaining effective in homogeneous settings.

Ethics Statement

This work studies metric depth estimation for robotic perception. Our synthetic OmniScene dataset reduces reliance on collecting real-world imagery, but simulation and benchmark coverage cannot represent every deployment environment. Depth errors, calibration drift, and domain shifts can compromise downstream robot safety; practical deployment therefore requires validation under the intended operating conditions and independent safety mechanisms. Real-world data collection and release should protect identifiable people and private spaces and respect applicable dataset licenses. The reported perception and preliminary policy results should not be interpreted as a guarantee of safe autonomous operation.

AI Use Statement

AI tools were used to assist with manuscript writing, translation, and reference checking. They were not used to develop research ideas, implement model or experimental code, generate data or images, or analyze experimental results. The authors are responsible for the final manuscript and its accuracy.

References

Rahul Ahuja, Mudit Jain, Bala Murali Manoghar Sai Sudhakar, Venkatraman Narayanan, Pratik Likhar, Varun Ravi Kumar, and Senthil Yogamani. FishRoPE: Projective rotary position embeddings for omnidirectional visual perception. *arXiv preprint arXiv:2604.10391*, 2026.

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024.

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

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.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Youyu Chen, Junjun Jiang, Yueru Luo, Kui Jiang, Xianming Liu, Xu Yan, and Dave Zhenyu Chen. Reliev3R: Relieving feed-forward reconstruction from multi-view geometric annotations. *arXiv preprint arXiv:2604.00548*, 2026.

Zisong Chen, Chunyu Lin, Lang Nie, Kang Liao, and Yao Zhao. Unsupervised OmniMVS: Efficient omnidirectional depth inference via establishing pseudo-stereo supervision. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10873–10879. IEEE, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Girish Chandar Ganesan, Yuliang Guo, Liu Ren, and Xiaoming Liu. UniDAC: Universal metric depth estimation for any camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26953–26963, 2026.

Rit Gangopadhyay, Jung-Hee Kim, Xien Chen, Patrick Rim, Hyoungseob Park, and Alex Wong. Extending foundational monocular depth estimators to fisheye cameras with calibration tokens. In *ICCV*, 2025.

Yuliang Guo, Sparsh Garg, S Mahdi H Miangoleh, Xinyu Huang, and Liu Ren. Depth any camera: Zero-shot metric depth estimation from any camera. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26996–27006. IEEE, 2025.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Kaixuan Wang, Hao Chen, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024.

Yuan-Ting Hu, Jiahong Wang, Raymond A. Yeh, and Alexander G. Schwing. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ilia Indyk, Ignat Penshin, Ivan Sosin, Maxim Monastyrny, Aleksei Valenkov, and Ilya Makarov. WideDepth: Millimeter-accurate benchmark for fisheye depth estimation. *arXiv preprint arXiv:2605.24074*, 2026.

Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. UniFuse: Unidirectional fusion for 360 panorama depth estimation. *IEEE Robotics and Automation Letters*, 6(2):1519–1526, 2021.

Hualie Jiang, Ziyang Song, Zhiqiang Lou, Rui Xu, and Minglang Tan. Depth anything in 360: Towards scale invariance in the wild. *arXiv preprint arXiv:2512.22819*, 2025a.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. AnySplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025b.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T Barron, Noah Snavely, and Aleksander Hołyński. Zipmap: Linear-time stateful 3d reconstruction via test-time training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21748–21759, 2026.

Dongki Jung, Jaehoon Choi, Adil Qureshi, Somi Jeong, Dinesh Manocha, and Suyong Yeon. Wid3R: Wide field-of-view 3d reconstruction via camera model conditioning. *arXiv preprint arXiv:2602.05321*, 2026.

Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. *CVPR*, 2023.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. MapAnything: Universal feed-forward metric 3d reconstruction. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

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

Varun Ravi Kumar, Sandesh Athni Hiremath, Markus Bach, Stefan Milz, Christian Witt, Clément Pinard, Senthil Yogamani, and Patrick Mäder. Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 574–581. IEEE, 2020.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Haodong Li, Wangguangdong Zheng, Jing He, Yuhao Liu, Xin Lin, Xin Yang, Ying-Cong Chen, and Chunchao Guo. DA2: Depth anything in any direction. *arXiv preprint arXiv:2509.26618*, 2025.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Siyou Lin, Zhou Xue, Hongwen Zhang, Liang An, Dongping Li, Shaohui Jiao, and Yebin Liu. Mix3R: Mixing feed-forward reconstruction and generative 3d priors for joint multi-view aligned 3d reconstruction and pose estimation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026a.

Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26835–26844, 2026b.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *European Conference on Computer Vision*, pp. 589–604. Springer, 2020.

Zeyu Ma, Alexander Raistrick, and Jia Deng. SimpleProc: Fully procedural synthetic data from simple rules for multi-view stereo. *arXiv preprint arXiv:2604.04925*, 2026.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4981–4991. IEEE, 2023.

Andreas Meuleman, Hyeonjoong Jang, Daniel S Jeon, and Min H Kim. Real-time sphere sweeping stereo from multiview fisheye images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11418–11427. IEEE, 2021.

Zehao Ni, Yonghao He, Lingfeng Qian, Jilei Mao, Fa Fu, Wei Sui, Hu Su, Junran Peng, Zhipeng Wang, and Bin He. VO-DP: Semantic-geometric adaptive diffusion policy for vision-only robotic manipulation. *arXiv preprint arXiv:2510.15530*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 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. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10106–10116. IEEE, 2024.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168. IEEE, 2021.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. PanoFormer: Panorama transformer for indoor 360 depth estimation. In *European Conference on Computer Vision*, pp. 195–211. Springer, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. HoHoNet: 360 indoor holistic understanding with latent horizontal features. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2573–2582. IEEE, 2021.

Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *European conference on computer vision*, pp. 732–750. Springer, 2018.

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision (ECCV)*, 2024.

Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. BiFuse: Monocular 360 depth estimation via bi-projection fusion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 459–468. IEEE, 2020.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

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

Pinzhi Wang, Ming Li, Jinghao Cao, Sidan Du, and Yang Li. CasOmniMVS: Cascade omnidirectional depth estimation with dynamic spherical sweeping. *Applied Sciences*, 14(2):517, 2024a.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271. IEEE, 2025b.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026b.

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

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*, volume 2026, pp. 10481–10497, 2026c.

Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Changhee Won, Jongbin Ryu, and Jongwoo Lim. Omnimvs: End-to-end learning for omnidirectional stereo matching. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8986–8995. IEEE, 2019.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3R: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1787–1796. IEEE, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22. IEEE, 2023.

Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3D: Towards zero-shot metric 3d prediction from a single image. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9009–9019. IEEE, 2023.

Ilwi Yun, Chanyong Shin, Hyunku Lee, Hyuk-Jae Lee, and Chae Eun Rhee. EGformer: Equirectangular geometry-biased transformer for 360 depth estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6078–6089. IEEE, 2023.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025.

Jia-Chen Zhao, Beiqi Chen, Xinyang Chen, Guangcong Wang, and Liqiang Nie. StructSplat: Generalizable 3d gaussian splatting from uncalibrated sparse views. *arXiv preprint arXiv:2606.28321*, 2026.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3d vision-language-action generative world model. In *ICML*, 2024.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, 2024.

APPENDIX

This appendix provides method details (Section A), dataset descriptions (Section B), additional experimental details (Section C), additional experiments (Section D), ablations (Section E), robotics downstream evaluation (Section F), and further discussion (Section G).

A METHOD DETAILS

A.1 FACTORED METRIC DEPTH PREDICTION

The depth variable in Equation (2) denotes optical-axis z-depth, not Euclidean distance along a camera ray. Fisheye ground truth expressed as Euclidean ray distance is converted to z-depth before evaluation. The model separates normalized dense shape $\hat{D}$ from a single positive metric scale $\hat{m}$, and reconstructs metric depth as $\hat{D}^{\mathrm{m}} = \hat{m}\hat{D}$. The shared DPT (Ranftl et al., 2021) head also predicts confidence $\hat{C}$ and a validity-mask channel $\hat{M}$, supervised by ground-truth validity labels. Ground-truth masks determine the pixels used for geometric supervision independently of $\hat{M}$, which is not an additional metric prediction target.

This factorization places calibration and pose information on the input side, where it is available for calibrated rigs, and reserves the output capacity for depth, uncertainty, and scale. Absolute scale is isolated in a one-dimensional pathway, while the remaining geometric losses operate on mean-normalized depth. Geometric quantities are expressed relative to a canonical reference camera, avoiding dependence on an arbitrary world coordinate frame. This design also reduces the coupling among depth, camera parameters, and ray-map prediction, as tested in Section E.2.

A.2 GEOMETRIC ENCODING FOR HETEROGENEOUS CAMERAS

The geometric input combines the viewing direction of each pixel with the relative position of its camera. Let $\mathbf{T}_s$ denote the camera-to-world pose of view s. Taking the first view as the reference, we canonicalize the poses as

$$\widetilde{\mathbf{T}}_s = \mathbf{T}_0^{-1}\mathbf{T}_s = \begin{bmatrix} \widetilde{\mathbf{R}}_s & \mathbf{c}_s \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \widetilde{\mathbf{T}}_0 = \mathbf{I}, \tag{7}$$

where $\mathbf{c}_s$ is the camera center in the reference frame. With column-vector notation, the camera-frame unit ray $\mathbf{d}_{s,p}$ obtained from $\mathcal{G}$ becomes $\mathbf{r}_{s,p} = \widetilde{\mathbf{R}}_s\mathbf{d}_{s,p}$. A point on this ray satisfies $\mathbf{X} = \mathbf{c}_s + \lambda\mathbf{r}_{s,p}$, where λ is distance along the ray. Thus, viewing direction and ray origin jointly describe the multi-camera geometry.

To give different rigs a consistent input magnitude, we normalize camera centers within each sample. Let $\mathcal{V}_+$ contain its valid non-reference views. We compute

$$\ell = \max\left(\frac{\sum_{s\in\mathcal{V}_+}\|\mathbf{c}_s\|_2}{\max(1,|\mathcal{V}_+|)}, 10^{-6}\right), \qquad \bar{\mathbf{c}}_s = \frac{\mathbf{c}_s}{\ell}. \tag{8}$$

The reference view has zero translation and is excluded from the mean, as are padded views. Broadcasting each normalized center over its image gives the six-channel ray map

$$\mathbf{U}_{s,p} = [\mathbf{r}_{s,p}^\top, \bar{\mathbf{c}}_s^\top]^\top \in \mathbb{R}^6. \tag{9}$$

This representation retains relative camera placement: $\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_i = (\mathbf{c}_j - \mathbf{c}_i)/\ell$. Canonicalization removes the arbitrary world frame, since $(\mathbf{A}\mathbf{T}_0)^{-1}(\mathbf{A}\mathbf{T}_s) = \mathbf{T}_0^{-1}\mathbf{T}_s$ for a common rigid transform $\mathbf{A}$. Away from the numerical floor in Equation (8), uniformly scaling all camera centers leaves their normalized values unchanged. The geometric pathway therefore describes the rig layout up to a common scale; metric scale is learned through the separately supervised scalar prediction in Section A.3.

A convolutional ray-map encoder E_{ray} converts $\mathbf{U}$ into features aligned with the image patch grid. For the DINOv2 ViT-S (Oquab et al., 2024) model, pixel unshuffle with patch size 14 is followed by a 3×3 convolution to 256 channels, a residual convolutional block to 384 channels, a 1×1

projection, and token-wise LayerNorm. Before positional embeddings and special tokens are added, each image patch token receives the corresponding geometric feature:

$$\mathbf{h}_{s,i}^{(0)} = E_{\text{img}}(I_s)_i + E_{\text{ray}}(\mathbf{U}_s)_i + \mathbf{e}_{\tau_s}, \tag{10}$$

where $\mathbf{e}_{\tau_s}$ is the camera-type embedding. The final ray-encoder projection is zero-initialized, allowing geometric conditioning to be learned from an initially unchanged image representation. Camera translation consequently enters the patch features used to form queries, keys, and values throughout the subsequent within-view and cross-view attention layers.

The calibration-token mechanism in Section 3.2 differs from the monocular adaptation of Gangopadhyay et al. (2025). That method inserts a single token set at the input to align fisheye features with perspective features. Here, the token slices are specific to each layer and camera type and are re-injected at every layer. Even in a global cross-view layer, the attention mask confines each calibration token to its own view. Dropping these tokens after attention prevents their state from propagating across layers and keeps lens-specific adaptation separate from cross-view metric fusion. We use 16 tokens per layer and activate them only for fisheye views.

FishRoPE (Ahuja et al., 2026) supplies angular position information from camera-frame unit rays, while the Jacobian Distortion Bias characterizes local projection variation on the patch grid. These direction-based mechanisms complement the ray-map features carrying camera origins. Together, the geometric input, view-local calibration tokens, and attention bias provide camera layout, lens adaptation, and local distortion cues within the shared backbone.

A.3 Confidence-Weighted Scale Prediction

Fisheye boundary regions are radially compressed and can carry greater geometric uncertainty. Uniformly pooling their features together with more reliable central regions can therefore impair global-scale regression. Scale Attention discards the lowest-confidence 25% of pixels and pools the remaining features using confidence weights:

$$\hat{m} = \text{MLP}\left(\frac{\sum_{p\in\Omega_{\text{core}}} \hat{C}_p \mathbf{F}_p}{\sum_{p\in\Omega_{\text{core}}} \hat{C}_p}\right), \tag{11}$$

where Ω_{core} is the retained pixel set and $\mathbf{F}_p$ is the spatial feature at pixel p. This pooling rule reduces the influence of low-confidence boundary features on the metric scale.

A.4 Depth Training Objectives

The total objective and its weights are given in Equation (6). The confidence-weighted dense-depth term is

$$\mathcal{L}_D = \frac{1}{|\Omega|}\sum_{p\in\Omega}\left(\hat{C}_p\left|\frac{\hat{D}_p}{\bar{\hat{D}}} - \frac{D_p}{\bar{D}}\right| - \lambda_{\text{conf}}\log\hat{C}_p\right), \tag{12}$$

where Ω is defined by the ground-truth validity mask and $\lambda_{\text{conf}} = 1.0$.

The confidence is parameterized as $\hat{C}_p = 1+\exp(\min(z_p, 10))$, where z_p is the raw confidence logit. This parameterization keeps confidence positive and bounded, avoiding a logarithmic singularity and unbounded exponential growth. Confidence acts as a learned error weight rather than a probability: increasing it amplifies the depth residual, while the negative log term discourages low confidence. The upper bound also limits confidence when the residual approaches zero.

For sample b, let $\Omega_b = \{(v,p) : M^{\text{gt}}_{b,v,p} = 1\}$ collect all valid pixels across its S input views. Predicted and ground-truth depth are normalized by their respective means over this same set:

$$\bar{\hat{D}}_b = \frac{1}{|\Omega_b|}\sum_{(v,p)\in\Omega_b}\hat{D}_{b,v,p}, \qquad \bar{D}_b = \frac{1}{|\Omega_b|}\sum_{(v,p)\in\Omega_b} D_{b,v,p}.$$

Each mean is computed independently for one complete multi-view sample; no averaging occurs across different samples in the batch. The predicted mask $\hat{M}$ does not select pixels for these means or for geometric supervision. This normalization removes absolute scale from shape supervision.

The gradient loss uses horizontal and vertical finite differences of $\hat{D}/\bar{\hat{D}}$ and $D/\bar{D}$, keeping it consistent with the scale-invariant depth term. The local loss $\mathcal{L}_{\text{local}}$ follows the multi-scale local consistency objective of MoGe (Wang et al., 2025b; 2026b) and is also evaluated on mean-normalized depth. The scale loss $|\hat{m} - m|/m$ provides the metric anchor. The mask loss $\mathcal{L}_{\text{mask}}$ supervises the shared DPT mask channel using ground-truth validity labels for sky and pixels outside the physical lens mask, following DepthAnything3 (Lin et al., 2025). The weights for depth, gradient, local consistency, scale, and mask supervision are 1.0, 1.0, 0.5, 1.0, and 0.2, respectively.

B Datasets

B.1 OmniScene: Synthetic Heterogeneous-Camera Data

OmniScene contains 126 scenes and 774 sequences, spanning residential interiors, commercial offices, shopping malls, industrial warehouses, urban scenes, sci-fi complexes, and stylized period interiors. Professionally authored Kujiale and Unreal Engine assets provide diverse appearance and scene geometry. Figure 7 illustrates the scene diversity beyond the compact description in the main paper.



Figure 7: Scene diversity in OmniScene. The six rows show the calibrated camera views across apartment, bazaar, gym, museum, office, palace, parking, and railway environments.

Camera Rig. Four fisheye cameras surround the platform with overlapping fields of view. Each follows the Kannala–Brandt projection model (Kannala & Brandt, 2006) with a 180° field of view. Two perspective cameras face forward and backward to complement the surround coverage. All six cameras render synchronously at 504×798, share a rig coordinate frame, and provide per-frame intrinsics and extrinsics in the OpenCV convention. Models can therefore use the original images and calibration directly without manual rectification.

Trajectory Generation. We derive navigable waypoints from scene occupancy, retaining locations with valid geometry and sufficient obstacle clearance. Constrained random walks limit horizontal displacement, vertical variation, and heading changes while allowing three-dimensional exploration. When progress becomes insufficient, the sampler expands its search range, applies a large-angle reorientation, or relocates the rig to a more open region. A sliding-window spatial-extent check rejects trajectories with limited coverage. Each scene has up to 10 independently generated trajectories, with 500–1000 frames per trajectory. Synchronized rendering keeps RGB, depth, semantic annotations, and camera poses temporally and geometrically aligned.

Annotations. Every view has noise-free metric z-depth, validity masks, and sky indicators. Post-processing removes frames affected by camera–geometry collisions or uninformative views, masks pixels outside the physical fisheye boundary, isolates sky with unbounded depth, and verifies cross-view depth consistency. Figure 8 illustrates why validity masking is necessary for sky and out-of-lens regions.

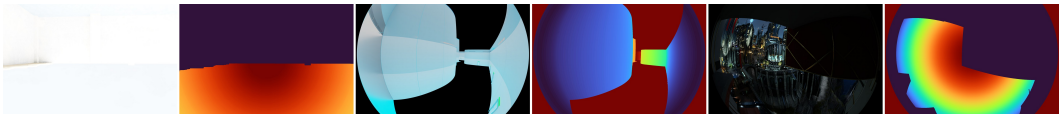


Figure 8: Example RGB/depth pairs illustrating image regions relevant to validity masking, including sky and the boundary of fisheye observations. Invalid pixels are excluded from geometric supervision and evaluation.

B.2 OmniOcc: Real-World Pinhole-Only Multi-Camera Data

OmniOcc is our publicly released real-world pinhole-only multi-camera dataset. It contains 25 indoor scenes, including offices, hotel rooms, corridors, and elevator lobbies. Each scene is represented by a synchronized keyframe snapshot. The capture rig has four stereo pairs, cam_0–cam_3, providing eight synchronized RGB views at 1280×1088, together with a registered LiDAR sensor in the same metric coordinate frame.

Supervision is obtained by projecting registered LiDAR points onto each camera and retaining valid observations after projection and visibility filtering. The resulting supervision is sparse and visibility-limited, rather than dense or pixel-perfect.

C Additional Experimental Details

C.1 Training Dataset Composition

Stage 1 uses 13 pinhole datasets covering indoor, outdoor, synthetic, and in-the-wild scenes: BlendedMVS (Yao et al., 2020), Mapillary Planet-Scale Depth (López Antequera et al., 2020), ScanNet++ v2 (Yeshwanth et al., 2023), Spring (Mehl et al., 2023), TartanAirV2-WB (Patel et al., 2025), UnrealStereo4K (Tosi et al., 2021), Aria Synthetic Environments (Avetisyan et al., 2024), DL3DV (Ling et al., 2024), Dynamic Replica (Karaev et al., 2023), MegaDepth (Li & Snavely, 2018), MVS-Synth (Huang et al., 2018), ParallelDomain-4D (Van Hoorick et al., 2024), and SAIL-VOS 3D (Hu et al., 2021). Stage 2 uses only the fisheye subset of OmniScene. Stage 3 combines the Stage-1 pinhole datasets, heterogeneous six-camera OmniScene data, and KITTI360 (Liao et al., 2022). Thus, the data sources comprise 14 public datasets together with OmniScene.

C.2 Three-Stage Optimization Schedule

Table 4 summarizes the optimization schedule. Stage 1 learns pinhole geometric priors with the calibration tokens and Jacobian Distortion Bias disabled. Stage 2 freezes the backbone, ray encoders, and prediction heads, updating only the calibration tokens. Stage 3 enables the Jacobian Distortion Bias and jointly fine-tunes the model. Pure-pinhole samples remain in Stage 3 because the front- and rear-facing pinhole views in OmniScene have little overlap; the replay ablation in Section E.3 examines this choice.

Stage 1 runs on 64 NVIDIA H100 GPUs. It samples the input resolution from 504×504, 504×378, 504×336, 504×280, 336×504, 896×504, 756×504, and 672×504. For square inputs, the view count is sampled uniformly from $[2, 16]$. Stage 2 uses four fisheye views at a fixed resolution and isolates lens adaptation in the token pathway.

Stage 3 also uses 64 H100 GPUs, with peak learning rates of 3×10^{-5} for the MLP, 10^{-4} for the Jacobian-bias module, and 3×10^{-6} for the backbone and DPT decoder. Pose conditioning is

Table 4: Optimization schedule. Learning rates are peak values; the Stage-3 module-specific rates are detailed below.

Setting	Stage 1	Stage 2	Stage 3
Optimization steps	150K	10K	50K
Warm-up steps	5K	500	1K
Peak learning rate	10^{-4}	5×10^{-5}	Per-module rates below
Input resolution	Randomized	504×798	504×798
Input views	2–16 (square inputs)	4	6–8
Calibration tokens	Disabled	16 per layer	16 per layer
Jacobian Distortion Bias	Disabled	Disabled	Enabled

activated with probability 0.7. Camera views are sampled during training instead of always using the canonical dataset layout, exposing the model to different multi-camera configurations.

C.3 Held-Out Evaluation Splits

For OmniScene, the training split contains 115 scenes and 690 sequences. The validation split contains 62 distinct sequences from 44 of the training scenes, while the test split contains 22 sequences from 11 held-out scenes. Thus, training and validation share scenes but use different sequences, and test scenes are excluded from both splits. Validation sequences are used for convergence monitoring and checkpoint selection. OmniScene-Single, OmniScene-Quad, and OmniScene-Full are evaluated on the same held-out test-scene partition, using one fisheye view, four fisheye views, and the complete six-camera rig, respectively.

ETH3D (Schöps et al., 2017) and OmniOcc are used only for evaluation. ScanNet++ v2 (Yeshwanth et al., 2023) and KITTI360 (Liao et al., 2022) appear in both training and evaluation, with their evaluation scenes or sequences excluded from training and validation. The fixed two-view ScanNet++ v2 protocol uses the following 10 held-out scene identifiers:

```
062e5a23a6  13b4efaf62  3c8d535d49  6c14d5fd01  95b9971d01
9bfbc75700  a4c043ac48  cba701332a  e3b3b0d0c7  e667e09fe6
```

For KITTI360, sequence 0 is the test set and the remaining sequences are used for training and validation, following DepthAnyCamera (Guo et al., 2025) and UniDAC (Ganesan et al., 2026).

C.4 Depth Evaluation Protocol

For OmniOcc, all 25 scenes are reserved for evaluation and never used for training. The six-view input comprises the left and right images from cam_0, cam_1, and cam_2. The fourth stereo pair is mounted at the rear and has limited field-of-view overlap with the four cameras on the left and right sides of the rig; it is therefore excluded from this evaluation. Depth errors are evaluated only on valid LiDAR-observed pixels.

Table 5 collects the input settings. The two-view KITTI360 input consists of frames separated by six timestamps on the same trajectory. OmniScene benchmarks differ only in the selected camera subset, while ETH3D and ScanNet++ v2 use two pinhole views and OmniOcc uses six.

Table 5: Benchmark input configurations. All runtime measurements use a single NVIDIA H100 GPU at the listed resolution.

Benchmark	Camera setting	Views	Resolution
KITTI360	Fisheye	1 or 2	504×504
OmniScene-Single	Fisheye	1	504×798
OmniScene-Quad	Fisheye	4	504×798
OmniScene-Full	4 fisheye + 2 pinhole	6	504×798
ETH3D	Pinhole	2	504×588
ScanNet++ v2	Pinhole	2	504×588
OmniOcc	Pinhole	6	504×798

The fisheye and heterogeneous comparisons include UniDepthv2 (Piccinelli et al., 2025), Metric3Dv2 (Hu et al., 2024), DepthAnyCamera (Guo et al., 2025), UniDAC (Ganesan et al., 2026), and MapAnything (Keetha et al., 2026), as applicable to each benchmark. Pinhole comparisons additionally include VGGT (Wang et al., 2025a), VGGT-Omega (Wang et al., 2026a), and DepthAnything3 (Lin et al., 2025). DepthAnyCamera and UniDAC process each image independently and do not fuse views. For fisheye views, MapAnything receives the original images, per-pixel camera-frame rays computed from the fisheye calibration, and the corresponding camera poses through its released interface. We retain the native fisheye projection without rectifying the images to a pinhole model. Available intrinsics and extrinsics are supplied to DepthAnything3. A dash in the scale column indicates that a method does not explicitly predict a global scale term.

Main-paper baselines are evaluated without additional fine-tuning on OmniScene or KITTI360, whereas X-Lens uses their training splits as described above. The main comparisons therefore differ in target-domain supervision. To examine this factor, Section D.5 additionally evaluates DepthAnyCamera and UniDAC fine-tuned on OmniScene. Their gains narrow the gap but do not eliminate it; this comparison complements the zero-shot transfer evaluation of X-Lens on ETH3D and OmniOcc.

We report Scale AbsRel for the predicted global scalar, standard depth AbsRel, RMSE, and δ_1, and the fraction $\tau_{1.03}$ of valid pixels whose relative depth error is within 3%. Higher δ_1, $\tau_{1.03}$, and FPS are better; lower error metrics are better. Runtime is reported for the complete benchmark input configuration. Since the resolution and view count vary, FPS should be compared within the same benchmark setting.

D Additional Experiments

D.1 Monocular Fisheye Results

The main paper reports KITTI360 and OmniScene-Quad results. Table 6 completes the fisheye evaluation with OmniScene-Single. X-Lens achieves the lowest Scale AbsRel, AbsRel, and RMSE, and the highest throughput, at 34 FPS. UniDepthv2-Large (Piccinelli et al., 2025) obtains higher δ_1 and $\tau_{1.03}$, so the improvement is not uniform across all accuracy metrics. The comparison complements the four-view results by separating monocular performance from the benefit of joint multi-view inference.

Table 6: Monocular fisheye evaluation on OmniScene-Single at 504×798. Bold indicates the best value in each error, accuracy, or runtime column.

Method	Params	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
UniDepthv2-Small	0.03B	0.3793	0.1341	4.3389	0.8771	0.2876	6
UniDepthv2-Large	0.35B	0.3263	0.1583	12.6289	**0.9102**	**0.4775**	5
Metric3Dv2-Small	0.03B	0.3607	0.2409	2.6357	0.6931	0.1698	6
Metric3Dv2-Giant	1.38B	0.2811	0.1352	2.2612	0.8453	0.3541	3
DepthAnyCamera	0.06B	0.2681	0.1963	2.8911	0.7950	0.2771	8
UniDAC	0.36B	0.2719	0.1869	2.8903	0.7693	0.2691	5
X-Lens	0.04B	**0.2287**	**0.1191**	**1.9801**	0.8868	0.3357	**34**

D.2 Multi-View Pinhole Results

Table 7 summarizes the multi-view pinhole results. Among the reported methods, X-Lens obtains the lowest explicitly predicted scale error on ETH3D and OmniOcc, while MapAnything (Keetha et al., 2026) has a lower Scale AbsRel on ScanNet++ v2. Larger models remain stronger on several dense-depth metrics. X-Lens reaches 39 FPS for two-view inputs and 26 FPS for six-view OmniOcc, with 0.04B parameters. Its parameter count is 97.1% lower than DA3-Giant (Lin et al., 2025), although DA3-Small is both smaller and slightly faster. These results characterize the trade-off between compact metric-depth perception and larger reconstruction models.

Table 7: Multi-view pinhole evaluation on ETH3D, ScanNet++V2, and OmniOcc six-view dataset. Bold marks the best result on each dataset. A dash indicates that the corresponding baseline does not explicitly predict a global scale term.

Method	Params	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
ETH3D (2 views, 504×588)							
VGGT	1.26B	–	0.0184	0.4073	0.9974	0.8693	15
VGGT-Omega	1.14B	–	**0.0055**	**0.1465**	**0.9994**	0.8693	12
DA3-Small	0.03B	–	0.0454	0.7438	0.9501	0.5502	**41**
DA3-Giant	1.36B	–	0.0113	0.2081	0.9989	**0.9073**	11
MapAnything	1.23B	0.1410	0.0228	0.3743	0.9992	0.7800	13
Ours	0.04B	**0.1217**	0.0445	0.6991	0.9723	0.5552	39
ScanNet++V2 (2 views, 504×588)							
VGGT	1.26B	–	0.0361	0.1393	0.9753	0.7282	15
VGGT-Omega	1.14B	–	**0.0332**	0.1390	0.9756	**0.7824**	12
DA3-Small	0.03B	–	0.0618	0.1754	0.9586	0.4606	**41**
DA3-Giant	1.36B	–	0.0354	**0.1348**	0.9768	0.7379	11
MapAnything	1.23B	**0.0462**	0.0548	0.1542	**0.9773**	0.4523	13
Ours	0.04B	0.0544	0.0549	0.1621	0.9626	0.5423	39
OmniOcc (6 views, 504×798)							
VGGT	1.26B	–	0.0458	**0.2031**	**0.9721**	0.6717	9
VGGT-Omega	1.14B	–	0.0482	0.2227	0.9703	0.6822	7
DA3-Small	0.03B	–	0.0655	0.2736	0.9311	0.4937	**29**
DA3-Giant	1.36B	–	**0.0437**	0.2272	0.9662	**0.7302**	6
MapAnything	1.23B	0.2050	0.0553	0.2305	0.9628	0.5851	8
Ours	0.04B	**0.0670**	0.0656	0.2584	0.9494	0.5550	26

D.3 Qualitative Depth and Point-Cloud Comparisons

We compare X-Lens with DepthAnyCamera (DAC) (Guo et al., 2025), UniDAC (Ganesan et al., 2026), and MapAnything (Keetha et al., 2026). Figure 9 compares depth predictions for one fisheye and one pinhole observation from the same six-view OmniScene-Full input. Figure 10 presents point-cloud reconstructions for two additional scenes, complementing the examples in Figure 5. Both comparisons use four fisheye and two pinhole views and show how the methods recover depth structure and metric scale across camera types. The full-rig comparisons in Figures 13 and 14, placed at the end of this appendix, show all six input views across additional environments.

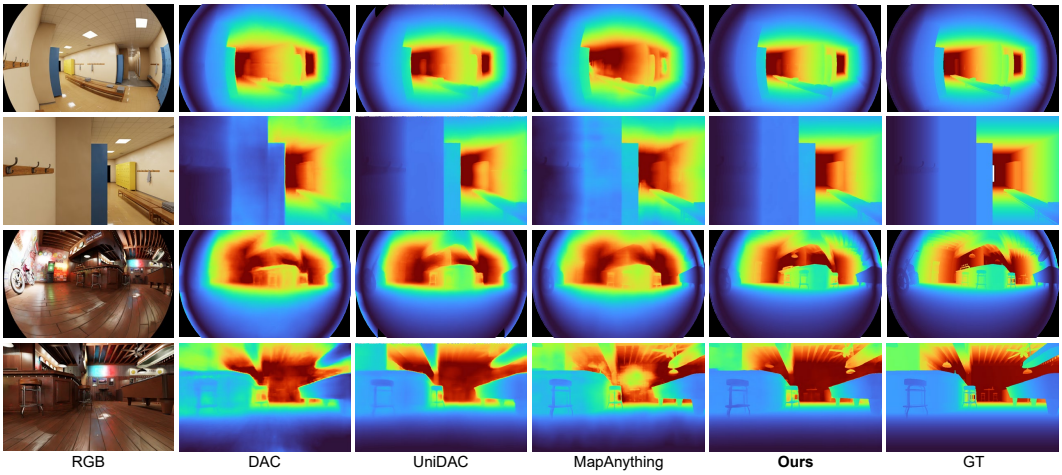


Figure 9: Heterogeneous depth comparison on OmniScene-Full. Each pair presents one fisheye and one pinhole observation from a calibrated six-view input. Columns compare RGB, DAC, UniDAC, MapAnything, X-Lens, and ground truth.

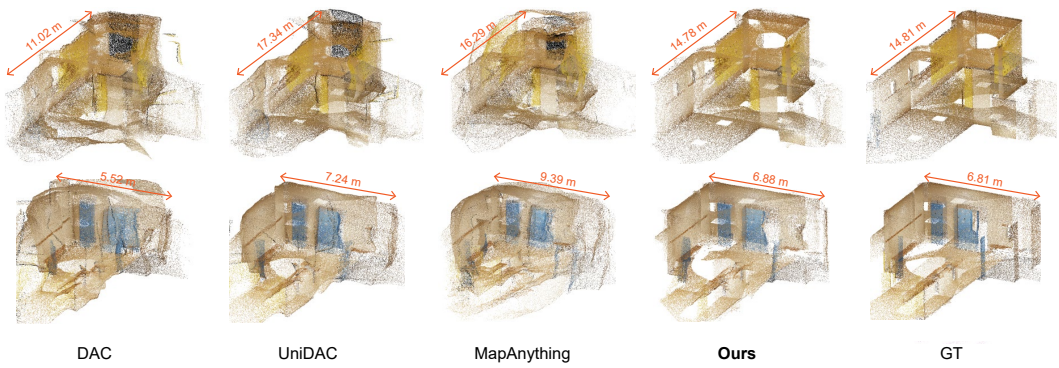


Figure 10: Additional point-cloud comparisons on two OmniScene-Full scenes not shown in Figure 5. Each reconstruction uses four fisheye and two pinhole inputs. Columns compare DAC, UniDAC, MapAnything, X-Lens, and ground truth; the annotated distances provide metric-scale references.

D.4 Robustness to Fisheye Intrinsic Variations

The standard OmniScene benchmark uses a fixed intrinsic configuration for each camera in the rig. To evaluate sensitivity to fisheye calibration changes, we collect additional observations in four scenes from the OmniScene test split: three indoor scenes and one outdoor scene. Pinhole intrinsics remain fixed, while each scene is captured under five different fisheye intrinsic configurations, with 40 frames per configuration. This produces $4 \times 5 \times 40 = 800$ evaluation frames.

We retain the six-view OmniScene-Full camera layout, comprising four fisheye and two pinhole views, at 504×798 resolution. The corresponding camera calibrations are provided for each configuration. Table 8 follows the metrics and comparison methods in Table 1 and reports performance aggregated over the 800-frame evaluation set.

Table 8: Heterogeneous-camera evaluation with varied fisheye intrinsics in four OmniScene test scenes. Pinhole intrinsics are fixed. Results are aggregated over 800 frames from five fisheye intrinsic configurations per scene.

Dataset	Views	Method	Params	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
OmniScene-Full (varied intrinsics)	6	MapAnything	1.23B	0.3623	0.1849	2.1433	0.7139	0.1555	5
		DepthAnyCamera	0.06B	0.2759	0.1966	2.1573	0.7791	0.2684	1
		UniDAC	0.36B	0.2546	0.1338	2.6511	0.8046	0.2671	1
		Ours	0.04B	0.1482	0.1221	1.8321	0.8591	0.3641	22

As shown in Table 8, X-Lens achieves the lowest scale and depth errors and the highest threshold accuracies on the evaluation set with varied fisheye intrinsics. It also provides the highest throughput with the smallest parameter count among the compared methods. These results support the ability of X-Lens to handle the tested intrinsic variations when the corresponding calibrations are provided, extending its heterogeneous depth estimation beyond the fixed camera configuration of the standard OmniScene benchmark.

D.5 Fine-Tuning Monocular Baselines on OmniScene

To evaluate in-domain adaptation, we fine-tune DepthAnyCamera (DAC) (Guo et al., 2025) and UniDAC (Ganesan et al., 2026) on the OmniScene training split, each for 50,000 steps on 32 NVIDIA H100 GPUs using AdamW with a learning rate of 10^{-5}. For each baseline, we select the checkpoint with the best performance on the OmniScene validation split. Evaluation follows the six-view OmniScene-Full protocol in Table 1. Table 9 compares the original baselines, their fine-tuned variants (FT), and X-Lens.

As shown in Table 9, fine-tuning on OmniScene improves scale accuracy, RMSE, and threshold accuracies for both monocular baselines, demonstrating the benefit of adapting to the target domain.

Table 9: OmniScene-Full evaluation of original and fine-tuned monocular baselines. Original baseline and X-Lens results are reproduced from Table 1.

Dataset	Views	Method	Params	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
OmniScene-Full	6	DAC	0.06B	0.2571	0.2066	2.3981	0.7653	0.2411	1
		DAC + FT	0.06B	0.2107	0.2391	2.1782	0.7966	0.2618	1
		UniDAC	0.36B	0.2506	0.1368	2.6125	0.8156	0.2511	1
		UniDAC + FT	0.36B	0.1969	0.1228	2.0194	0.8545	0.2971	1
		Ours	0.04B	0.1181	0.1021	1.5993	0.8982	0.3724	22

Table 10: Deployment comparison on OmniScene-Full with six input views (4 fisheye + 2 pinhole) at 504×798 resolution. BF16 results on the H100 are reproduced from Table 1; INT16 and INT8 results are measured on the S600 after quantization. FPS counts complete six-view predictions per second on the indicated hardware.

Dataset	Views	Precision	Params	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑	FPS ↑
OmniScene-Full	6	BF16 (H100)	0.04B	0.1181	0.1021	1.5993	0.8982	0.3724	22
		INT16 (S600)	0.04B	0.1302	0.1145	1.8224	0.8764	0.3682	12
		INT8 (S600)	0.04B	0.1532	0.1337	2.1266	0.8127	0.3149	14

The gains are not uniform for DAC, whose AbsRel increases after fine-tuning. Nevertheless, X-Lens retains a clear advantage over both fine-tuned baselines across all reported accuracy metrics, with particularly substantial gaps in scale error and RMSE, while remaining smaller and faster. These results suggest that in-domain fine-tuning alone does not close the performance gap to joint heterogeneous multi-view depth estimation.

D.6 Edge Deployment on RDK S600

We evaluate edge deployment on the D-Robotics RDK S600 platform, powered by the S600 system-on-chip. Its four Nash BPU cores provide a manufacturer-rated peak compute capacity of 560 TOPS at INT8 precision, alongside 18 Arm Cortex-A78AE CPU cores at 2.0 GHz. Our device uses 32 GB of LPDDR5 memory with a specified bandwidth of 204.8 GB/s.[2] The memory capacity and bandwidth are relevant to storing model weights and processing feature maps from multiple input views. The platform also provides six MIPI CSI-2 camera interfaces for multi-camera integration.

Our real-world platform uses four fisheye cameras and one pinhole camera, together with an onboard RDK S600, mounted on an AgileX Robotics mobile robot chassis (Figure 11). We provide a supplementary video demonstrating real-time X-Lens inference on this S600-equipped platform. This five-camera configuration is used for the physical demonstration, while the quantitative benchmark below retains the six-view OmniScene-Full protocol from Table 1 for comparison with the original model.



Figure 11: Five-camera mobile rig (4 fisheye + 1 pinhole) on an AgileX Robotics chassis.

The original X-Lens model uses BF16 precision and is evaluated on an NVIDIA H100 GPU. In our deployment setup, BF16 is used only for the H100 reference; the S600 runs the INT16 and INT8 variants quantized from this model. Table 10 reproduces the BF16 accuracy and throughput from Table 1 and reports the measured accuracy and throughput of the quantized variants on the RDK S600, using six-view inputs at 504×798 resolution. Each row identifies its execution hardware, so the FPS values describe the respective deployment configurations rather than a comparison of numerical precision on the same device.

As shown in Table 10, INT16 largely preserves the depth accuracy of the BF16 reference with only a small degradation, while INT8 provides higher throughput at the cost of a larger accuracy drop. Both quantized variants meet the real-time inference requirements of our onboard deployment on

[2]Official RDK S600 specifications: `https://developer.d-robotics.cc/rdks600`.

the RDK S600. These results demonstrate that the compact design of X-Lens supports practical real-time metric depth estimation on a mobile robot platform.

D.7 Full-View Batched Inference for Monocular Baselines

The main-paper runtime evaluation applies DepthAnyCamera (Guo et al., 2025) and UniDAC (Ganesan et al., 2026) sequentially with batch size 1. This additional comparison batches all six views of each OmniScene-Full input set together, using batch size 6 on a single NVIDIA H100 GPU at 504×798 resolution. Each view is still predicted independently, without cross-view feature fusion. FPS counts complete six-view input sets rather than individual images.

Table 11: Six-view FPS on OmniScene-Full (H100). X-Lens uses joint multi-view inference.

Method	Sequential FPS ↑	Batched FPS ↑
DepthAnyCamera	1	4
UniDAC	1	3
X-Lens	–	22

The batched protocol retains the model weights, numerical precision, preprocessing, and timing scope of the sequential evaluation. Sequential baseline FPS is reproduced from Table 1. The X-Lens entry corresponds to joint inference on one complete six-view input set.

As shown in Table 11, batching all views improves the throughput of both monocular baselines, but X-Lens remains substantially faster. A major source of overhead in these baselines is conversion to an equirectangular projection (ERP), which accounts for a substantial portion of their runtime. X-Lens directly processes the native fisheye and pinhole images without ERP conversion, avoiding this overhead while jointly estimating depth across views. These results show that the speed advantage of X-Lens persists when the monocular baselines use full-view batching.

E Additional Ablation Studies

E.1 Training-Stage and Geometry Ablations

The main-paper heterogeneous ablation is complemented by the fisheye-only and pinhole-only results in Table 12. S1 denotes pinhole pre-training; S1→S2 adds calibration-token adaptation; and S1→S2→S3, frozen freezes the backbone and DPT decoder during Stage 3. The full model without JDB is jointly trained without the Jacobian Distortion Bias. These definitions also apply to the qualitative comparison in Figure 12.

Table 12: Additional training-stage and geometry ablations on four-view OmniScene-Quad and six-view OmniOcc. Variant names follow the main paper; bold marks the best result within each camera setting.

Cameras	Views	Variant	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑
Fisheye	4	S1	0.7325	0.2769	2.9182	0.6719	0.1711
		S1→S2	0.2126	0.2083	2.5082	0.7560	0.1990
		S1→S2→S3, frozen	0.4018	0.2946	2.8792	0.6805	0.1436
		Full w/o JDB	0.1854	0.2115	2.8174	0.7717	0.1998
		Full X-Lens	**0.1268**	**0.1138**	**1.6481**	**0.8822**	**0.3328**
Pinhole	6	S1	0.1052	0.0717	0.2801	0.9439	0.5433
		S1→S2	0.1052	0.0717	0.2801	0.9439	0.5433
		S1→S2→S3, frozen	0.1916	0.0718	0.2774	0.9442	0.5025
		Full w/o JDB	0.5631	0.2844	0.8767	0.5332	0.0859
		Full X-Lens	**0.0670**	**0.0656**	**0.2584**	**0.9494**	**0.5550**

Calibration adaptation reduces fisheye Scale AbsRel from 0.7325 to 0.2126, while the pinhole results remain unchanged from S1 to S2, consistent with freezing the shared model and adapting only fisheye tokens. The effect of JDB extends beyond fisheye adaptation. On the pinhole-only OmniOcc benchmark, removing JDB increases scale error to more than five times the S1 value and worsens all dense-depth metrics. This deterioration suggests interference with previously learned pinhole geometry during heterogeneous training. The full model instead improves on S1 and achieves the

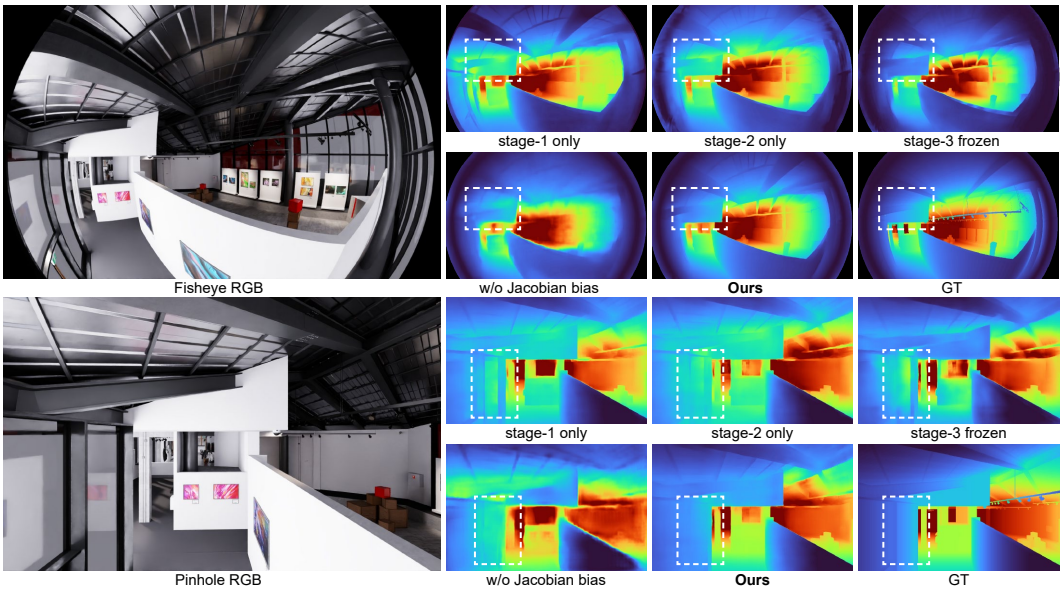


Figure 12: Qualitative ablation on a six-view OmniScene-Full input, showing one fisheye and one pinhole view. The labels "stage-1 only", "stage-2 only", and "stage-3 frozen" correspond to S1, S1→S2, and S1→S2→S3 with a frozen backbone and decoder, respectively. The other variants remove the Jacobian bias or use the full model.

best values across all five metrics in both settings, supporting the role of projection-aware attention in accommodating fisheye distortion while preserving pinhole metric geometry. Figure 4 in the main paper provides a complementary view of the learned correction: fisheye boundary regions require stronger spatial variation than pinhole regions.

Runtime Overhead of JDB. Table 13 reports the measured latency, throughput, and peak GPU memory with JDB enabled or disabled on an H100. The benchmark uses BF16 and six-view OmniScene-Full inputs at 504×798, and FPS counts complete six-view predictions. JDB reuses the ray field to estimate local Jacobians on the patch grid, then constructs pairwise descriptors and predicts the attention bias. Enabling JDB increases latency by 25% and peak memory by 3 GB, while retaining a throughput of 22 FPS. Together with the accuracy ablations above, these results characterize the computational cost of incorporating projection geometry into cross-view attention.

Table 13: JDB runtime on OmniScene-Full (H100).

JDB	Latency (ms) ↓	FPS ↑	Peak memory (GB) ↓
Disabled	36	28	6
Enabled	45	22	9

E.2 Effect of Auxiliary Camera Prediction

We test whether auxiliary camera-parameter prediction benefits the depth-focused model. The variant follows the Stage-1 pinhole setup and adds learnable camera tokens and a head for intrinsics and extrinsics. A second DPT head predicts ray maps, which are optimized jointly with depth through a point loss. Training data, schedule, and parameter budget otherwise remain aligned with the Stage-1 setup.

Table 14: Auxiliary camera-head experiment on six-view OmniOcc after Stage-1 training. Adding explicit camera and ray-map prediction hurts metric depth accuracy.

Variant	Scale AbsRel ↓	AbsRel ↓	RMSE ↓	δ_1 ↑	$\tau_{1.03}$ ↑
With camera head	0.1816	0.0702	0.2861	0.8961	0.4910
Ours after Stage-1 training	**0.0692**	**0.0636**	**0.2499**	**0.9481**	**0.5442**

Table 14 shows higher scale, AbsRel, and RMSE errors and lower threshold accuracy with the auxiliary head. Under this configuration, adding camera and ray-map prediction does not improve depth estimation. The result supports concentrating the compact model on depth, confidence, and scale when calibration is already supplied as input.

E.3 Effect of Pinhole Replay in Stage 3

We train Stage-3 models with and without pure-pinhole data while keeping other settings unchanged, then evaluate them on heterogeneous OmniScene-Full, fisheye OmniScene-Quad, and real-world pinhole OmniOcc.

Table 15: Stage-3 pure-pinhole replay ablation. We compare Stage-3 training with and without pure-pinhole data on heterogeneous (OmniScene-Full), fisheye (OmniScene-Quad), and real-world pinhole (OmniOcc) benchmarks.

Benchmark	Stage-3 variant	Scale AbsRel ↓	AbsRel ↓	RMSE ↓
OmniScene-Full	w/o pinhole replay	0.1057	0.1041	1.5028
	Ours	0.1181	0.1021	1.5993
OmniScene-Quad	w/o pinhole replay	0.1241	0.1033	1.6819
	Ours	0.1268	0.1138	1.6481
OmniOcc	w/o pinhole replay	0.2061	0.0883	0.2881
	Ours	0.0670	0.0656	0.2584

Including pure-pinhole data lowers OmniOcc Scale AbsRel from 0.2061 to 0.0670, AbsRel from 0.0883 to 0.0656, and RMSE from 0.2881 to 0.2584. The other camera settings show mixed changes: replay improves AbsRel on OmniScene-Full and RMSE on OmniScene-Quad, while the remaining reported errors increase. Thus, replay substantially restores pinhole performance with a measurable trade-off on some synthetic fisheye and heterogeneous metrics, rather than uniformly improving every benchmark.

F Robotics Downstream Evaluation

Visual encoders such as CLIP (Radford et al., 2021) and DINOv3 (Siméoni et al., 2025) provide strong semantic features, but their standard 2D pre-training does not directly supervise absolute metric scale or depth continuity. A downstream policy may consequently need to learn task-relevant spatial relationships from robot trajectories. We explore X-Lens as a visual encoder whose pre-training explicitly encourages multi-view geometry. Intermediate attention tokens are passed to the policy instead of converting the observations to rigid point clouds or explicit depth maps, retaining appearance features alongside geometric information.

F.1 Visual Feature Integration for Robot Policies

At each control step, the robot receives V images $\mathcal{I} = \{I_v\}_{v=1}^{V}$, with $I_v \in \mathbb{R}^{3\times H\times W}$, and a 14-dimensional proprioceptive state $\mathbf{q}$. A visual encoder $\mathcal{E}$ and fusion head g produce a conditioning representation for a conditional flow-matching (Lipman et al., 2023) DiT policy (Peebles & Xie, 2023), $\pi_\theta(\mathbf{a} \mid \mathbf{z}, \mathbf{q})$.

Each VO-DPP configuration uses a single visual encoder, selected from the alternatives in Table 16. The encoders are swapped between experiments. The common encoder interface returns image patch tokens T_{img}, their cached positions, and per-layer camera tokens $\mathcal{C}_\ell$ or spatial tokens $\mathcal{S}_\ell$. In the DINOv3-ViT-B configuration, the encoder supplies image tokens with $\mathcal{C}_\ell = \mathcal{S}_\ell = \emptyset$. In the X-Lens configuration, the encoder separates intermediate features from alternating frame and global attention into an appearance stream and a cross-view geometry stream, the latter supplying $\mathcal{S}_\ell$. Local and global spatial features are projected through separate MLPs and concatenated into a 1024-dimensional representation:

$$S' = [\mathrm{MLP}_{\text{local}}(S) \,\|\, \mathrm{MLP}_{\text{global}}(S)] \in \mathbb{R}^{1024}. \tag{13}$$

A ResNetAdapter compresses the features to a $2 \times 2 \times 1024$ visual map, which is flattened and combined with proprioception for policy conditioning.

F.2 SCAN-OBJECT TASK EVALUATION ON ROBOTWIN2

We evaluate the Scan-Object (Easy) task in RoboTwin2 (Chen et al., 2025), comparing DINOv3-Base and DINOv3-Large (Siméoni et al., 2025), VGGT (Wang et al., 2025a), and X-Lens as visual representations within VO-DPP. VO-DPP is an unreleased proprietary extension of VO-DP (Ni et al., 2025) that supports multi-view feed-forward inputs, whereas the original VO-DP uses a single view. Every visual encoder is frozen; only the downstream trainable modules are optimized for 300 epochs with a learning rate of 10^{-5}. Each method is evaluated over 100 episodes. Table 16 reports successful episodes, task success rate (TSR), and a 95% Wilson confidence interval (Wilson, 1927) under a binomial model of episode outcomes. These intervals quantify uncertainty from the finite number of evaluation episodes, not variability across training seeds.

Table 16: Preliminary RoboTwin2 Scan-Object (Easy) evaluation with VO-DPP over 100 episodes per method. TSR denotes task success rate; CI is the 95% Wilson confidence interval.

Visual encoder	Encoder status	Successes / Episodes	TSR (%) ↑	95% CI (%)
DINOv3-Base	Frozen	50 / 100	50	[40.4, 59.6]
DINOv3-Large	Frozen	55 / 100	55	[45.2, 64.4]
VGGT	Frozen	67 / 100	67	[57.3, 75.4]
X-Lens	Frozen	72 / 100	**72**	[62.5, 79.9]

X-Lens achieves the highest observed task success rate among the evaluated visual encoders, providing preliminary evidence that geometric pre-training can benefit this multi-view scanning task. However, the evaluation is limited to one task and does not quantify variability across training seeds. We therefore do not claim a statistically significant advantage over VGGT or general policy superiority across tasks.

G DISCUSSION

X-Lens accepts camera intrinsics and extrinsics as geometric inputs but relies on known calibration; it does not jointly estimate these quantities online. Although its ray-space formulation supports varied camera configurations, the synthetic training data cover a bounded range of intrinsics. Unseen fisheye lens models or extreme fields of view can therefore cause a remaining simulation-to-real gap and reduce accuracy. Broader real-world data and intrinsic diversity are needed to assess and improve this generalization.

Quantitative validation of joint fisheye–pinhole inputs in real-world scenes remains a limitation of this work. Our real-world depth evaluations assess fisheye and pinhole settings separately, while joint heterogeneous evaluation relies on synthetic OmniScene data. KITTI360 (Liao et al., 2022) includes both camera types, but their limited field-of-view overlap restricts the shared visual evidence available for cross-camera attention. We therefore do not include a joint fisheye–pinhole evaluation on KITTI360, as this configuration is less suited to assessing the full benefit of cross-attention through cross-view correspondences. We plan to construct a real-world heterogeneous multi-camera dataset with sufficient overlap, synchronized images, accurate calibration, and reliable metric-depth annotations to evaluate joint inference and quantify the simulation-to-real gap.

Temporal aggregation is another direction for improving geometric consistency. Beyond the preliminary vision–action study, X-Lens could be explored as a geometric prior for world models, SLAM, and robotic loco-manipulation. The current experiments establish depth-estimation performance and one downstream case study, rather than validating all of these applications.

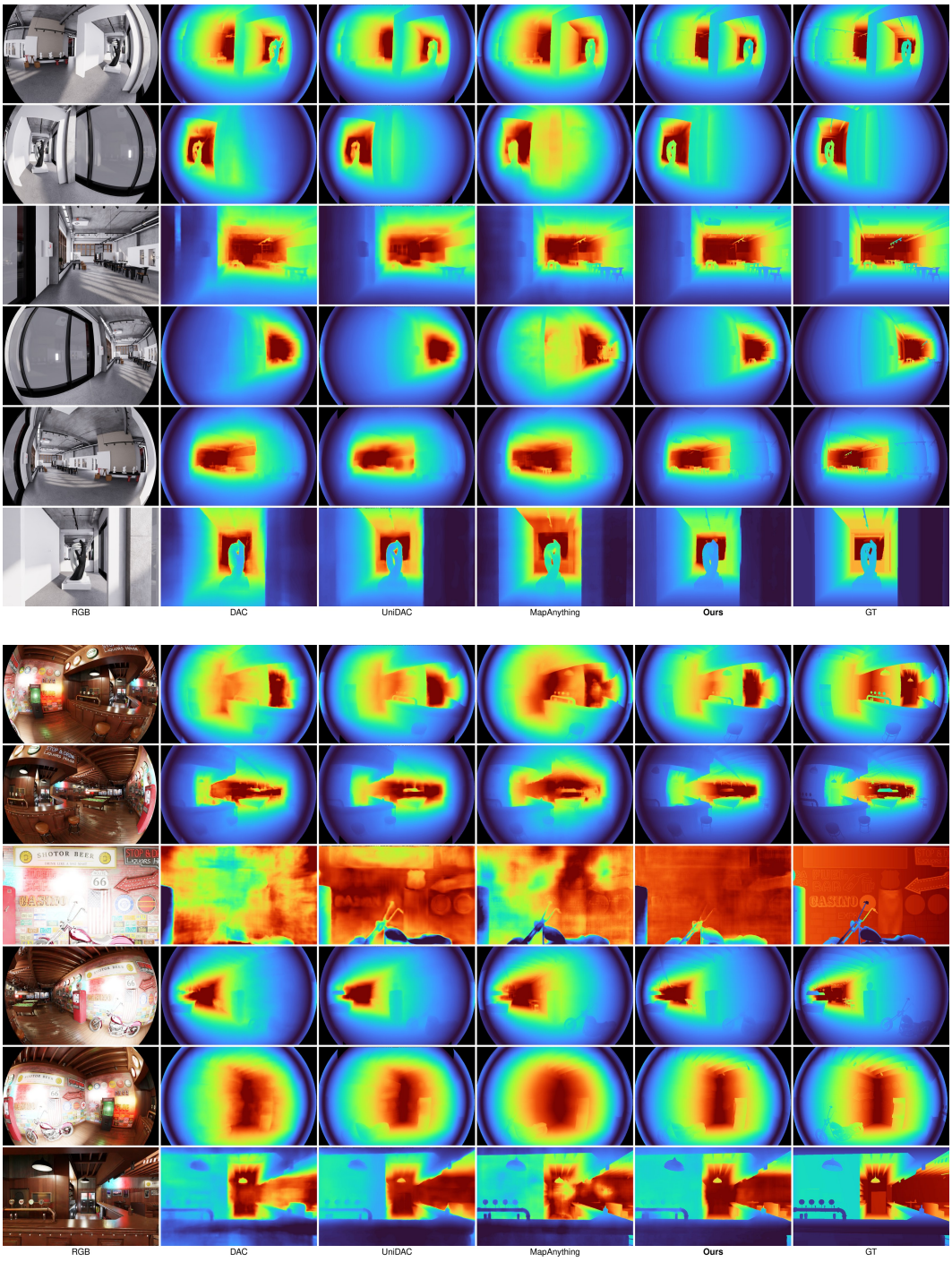


Figure 13: Additional six-view OmniScene-Full comparisons. Each group contains all four fisheye and two pinhole views. The columns are RGB, DAC, UniDAC, MapAnything, X-Lens, and ground truth, respectively. The examples show complementary surround and perspective observations in two different environments.

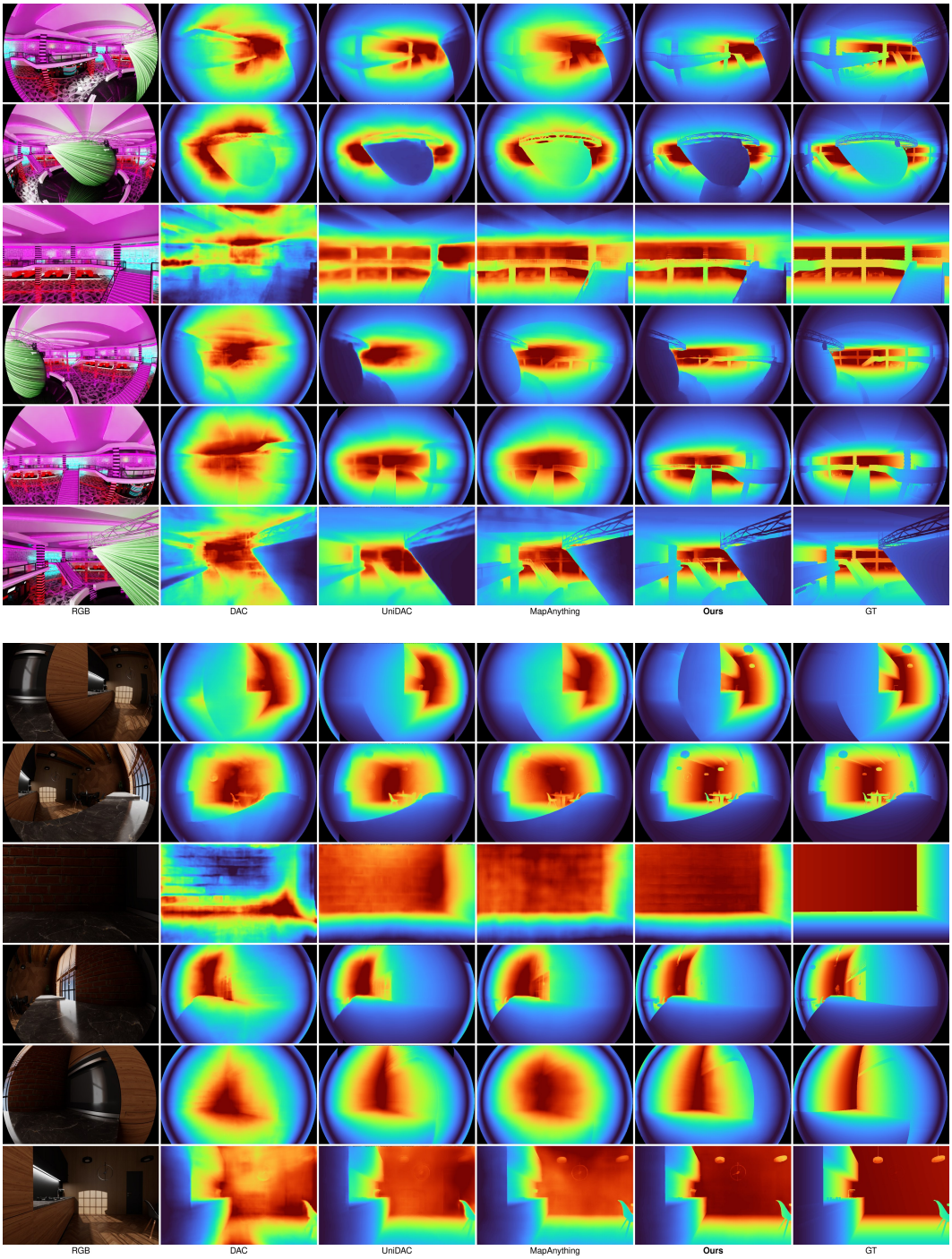


Figure 14: Further six-view OmniScene-Full comparisons under different scene appearance and geometry. Each group uses the same four-fisheye/two-pinhole rig and column ordering as Figure 13, allowing direct inspection of depth structure across camera types.