EXTENDING 3D FOUNDATION MODELS TO ANY CENTRAL CAMERA WITH DISTORTION AWARENESS TOKEN

Anonymous authors
Paper under double-blind review

ABSTRACT

3D foundation models, such as Visual Geometry Grounded Transformer (VGGT), have shown superior performance in inferring 3D attributes of a scene from several views. However, they do not generalize to non-perspective cameras, limiting their usage in other camera models, such as fisheye and panoramic cameras. To this end, we propose **VGGT-Any**, which extends VGGT to any type of central camera and to heterogeneous camera systems without retraining the base model. At the core of VGGT-Any is a novel camera representation: it decomposes camera geometry into a canonical space, which follows the usual perspective-camera formulation, and a distortion space, which expresses each camera's specific projection. We further introduce a distortion token, which is injected into the frozen foundation model to reconstruct scenes from various central cameras, or even from heterogeneous camera systems. Moreover, since existing datasets each cover only a limited range of camera types, we construct **LensZoo**, a synthetic indoor dataset in which any central camera model can be synthesized at shared poses, enabling controlled comparisons across camera models. Our evaluations show that VGGT-Any is competitive with camera-specific and fine-tuned baselines on real fisheye and panoramic benchmarks, and exactly retains the base model's pinhole reconstruction ability.

1 INTRODUCTION

Feed-forward 3D foundation models recover camera poses and dense geometry from unposed images (Wang et al., 2024; 2025). As VGGT-Ω shows, their reconstruction accuracy scales with training data and model size (Wang et al., 2026a), so a pretrained model embodies knowledge that is costly to reproduce. These models are increasingly used in SLAM and autonomous driving (Murai et al., 2025; Maggio et al., 2025; Jia et al., 2025). Robots and vehicles in these applications often carry fisheye or panoramic cameras, whose wide field of view captures more of the surroundings in each image and thus provides more information about the scene, strengthening their spatial perception. Such cameras are frequently combined with pinhole cameras in heterogeneous camera systems (Figure 1, left) (Duan et al., 2026; Zhou et al., 2026; Ou et al., 2026). However, these models are trained on 3D data captured predominantly by pinhole cameras (Wang et al., 2025; 2026a), and their camera outputs describe every image with a pinhole camera. Despite their strong 3D perception, they are therefore confined to the pinhole camera model and do not generalize to other camera models, such as the fisheye and panoramic cameras above.

The conventional remedy is to rectify each fisheye or panoramic image into a pinhole view by undistortion (Kannala & Brandt, 2006) (or, for panoramas, reprojection) and cropping. Because no pinhole camera can represent rays at or beyond 90° off its axis, this gives up the wide field of view for which these cameras are chosen (Duan et al., 2026; Jung et al., 2026), and the rectified view, resampled from only part of the original pixels (Figure 1, right), is even inferior to an image taken directly by a pinhole camera. Other methods replace the pinhole camera with a ray-based or spherical representation and train the model on non-pinhole 3D data (Guo et al., 2026; Guruprasad et al., 2026; Jung et al., 2026). Although datasets have been collected with non-pinhole cameras (Ou et al., 2026; Yeshwanth et al., 2023; Liao et al., 2023), they remain far smaller than the data on which VGGT-Ω is trained. Moreover, once the pretrained weights are altered (Guruprasad et al., 2026; Jung

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

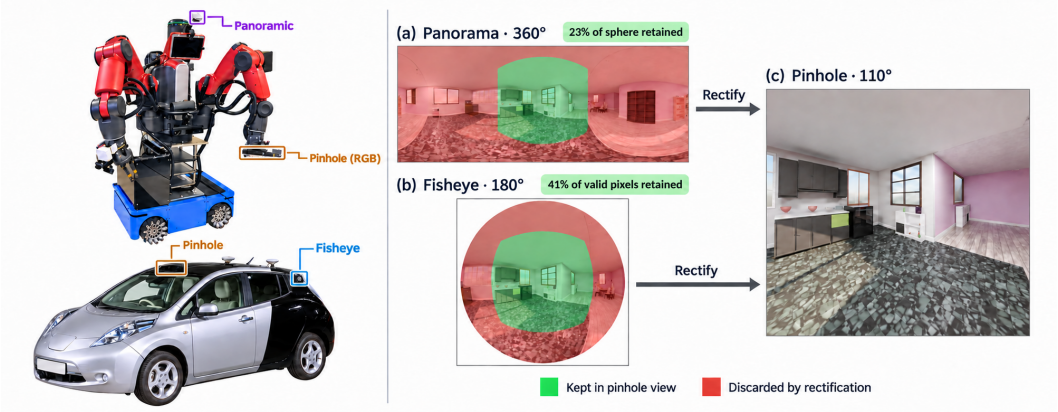

Figure 1: **Heterogeneous cameras and rectification loss.** Left: examples of heterogeneous camera systems. Right: rectifying a 360° panorama (a) or an ideal 180° equidistant fisheye image (b) to a square 110° pinhole view (c) retains 23% of the sphere and 41% of valid fisheye pixels, respectively. Green indicates retained content; red indicates discarded content.

et al., 2026), the model's pinhole reconstruction ability is no longer guaranteed. In contrast, methods that keep the foundation model frozen and insert learned tokens preserve the pretrained weights and, with them, the pinhole reconstruction ability (Gangopadhyay et al., 2025; Duan et al., 2026). These tokens, however, are free parameters trained end-to-end on images rather than derived from camera geometry. What they learn is therefore tied to the training images rather than to a given camera's calibration.

To address these issues, we propose **VGGT-Any**, which extends a frozen VGGT-Ω to any type of central camera and to heterogeneous camera systems, taking each image at its native field of view and leaving the pretrained weights unchanged. Our key insight is that what a frozen foundation model lacks about a camera is only its distortion from the perspective camera model it already assumes. We therefore decompose the geometry of a central camera, one whose rays pass through a single optical center, into a *canonical space*, aligned with the foundation model's native perspective-camera formulation, and a *distortion space*, which captures the camera-specific projection (§3.1). Because a camera's distortion depends on its geometry alone, not on the scene, we compute it from the calibration and encode it as a *distortion token* learned purely from camera geometry; the token therefore cannot overfit to scene content or image appearance (§3.2). Injected into the frozen model through adapters, the token makes the model distortion-aware; assigned per frame, it also lets one input window mix camera models (§3.3).

Since existing non-pinhole datasets each cover only a few camera types, we further construct **LensZoo**, a synthetic indoor dataset built so that any central camera model can be synthesized at shared physical poses, to enable controlled comparisons across camera models (§4). VGGT-Any is competitive with camera-specific and fine-tuned methods on real fisheye and panoramic benchmarks, and a single model reconstructs scenes from every non-pinhole camera model evaluated on LensZoo and from a heterogeneous mixture of camera models (§5). In summary, our contributions are as follows:

- We propose a camera representation that decomposes central-camera geometry into a canonical space, aligned with the foundation model's perspective-camera formulation, and a distortion space.
- We introduce a distortion token learned purely from camera geometry, which therefore cannot overfit to scene content or image appearance.
- We present **VGGT-Any**, which makes a frozen VGGT-Ω distortion-aware for any type of central camera and for heterogeneous camera systems, while exactly retaining its pinhole reconstruction ability.

Table 1: Camera-model coverage of 3D reconstruction methods.

Method	Pinhole	Brown–Conrady	Division	FOV	Equidistant	KB4	Equisolid	UCM	EUCM	Double sphere	ERP	Mirrored ball	Smooth angular
Monocular methods													
UniK3D (Piccinelli et al., 2025)	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗	✓	✗	✗
Multiview methods													
VGGT (Wang et al., 2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
VGGT-Ω (Wang et al., 2026a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
π^3 (Wang et al., 2026b)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
MapAnything (Keetha et al., 2026)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Wid3R (Jung et al., 2026)	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗	✓	✗	✗
PanoVGGT (Guo et al., 2026)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
VGGT-360 (Yuan et al., 2026)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Fisheye3R (Duan et al., 2026)	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗
VGGT-Any	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

For baselines, ✓: reported support, including exact submodels, on shared valid domains; ✗: not established. Camera models are defined, with references, in Appendix C.1.

- We construct **LensZoo**, a synthetic indoor dataset of 1.32M images, built so that any central camera model can be synthesized at shared physical poses, with ground-truth depth, calibration and pose.

2 Related Work

Feed-forward 3D reconstruction. DUSt3R predicts 3D points from image pairs without calibration or poses (Wang et al., 2024), and VGGT estimates cameras and geometry from unposed image sets in one forward pass (Wang et al., 2025); VGGT-Ω scales up its model and data (Wang et al., 2026a), π^3 treats views symmetrically (Wang et al., 2026b), and MapAnything optionally takes the calibration of any central camera as ray directions (Keetha et al., 2026), yet still assumes perspective images and degrades on wide-field-of-view imagery (Jung et al., 2026; Duan et al., 2026).

Omnidirectional camera models. The Taylor model writes the ray of an affinely corrected image point (u, v) at radius ρ as proportional to $(u, v, a_0 + a_2\rho^2 + \cdots + a_N\rho^N)$ (Scaramuzza et al., 2006): a pinhole ray plus an axial increment vanishing to second order at the center; the ray passes 90° where the polynomial changes sign. Our ray decoder (§3.2; equation 27 in Appendix C.6) has this form, with a learned three-vector increment. Unlike the radially symmetric Taylor family, our representation is a model-free map on ray directions, built from any calibrated ray map, covers non-radial cameras such as equirectangular panoramas (§3.1), and is learned as a token conditioning a frozen model (§3.2–§3.3).

Reconstruction beyond perspective cameras. Single-image methods estimate geometry, not poses across views. UniK3D supports different camera projections (Table 1) without calibration (Piccinelli et al., 2025), Depth Any Camera (DAC) remaps calibrated images into equirectangular patches for one metric-depth model (Guo et al., 2025), and DA^2 extends monocular depth to panoramas (Li et al., 2026); VGGT-360 (Yuan et al., 2026) and RPG360 (Jung et al., 2025) process virtual views of one panorama without training, and PaGeR fine-tunes a multiview foundation model on the panorama's cubemap faces (Bozic et al., 2026). Several multiview methods train on non-pinhole 3D data: PanoVGGT for native panoramas (Guo et al., 2026), Wid3R for wide-angle image sets given the camera type (Jung et al., 2026), and CAM3R, which aligns pairwise predictions on wide-angle images into one scene (Guruprasad et al., 2026); neither Wid3R nor CAM3R requires numerical calibration.

Camera conditioning of pretrained models. Lightweight modules can adapt a pretrained model without rectification, leaving its weights intact. Fisheye3R adapts frozen reconstruction models with tokens trainable on unlabeled perspective images alone, supporting uncalibrated fisheye images, heterogeneous perspective/fisheye sequences and transfer across projection models (Duan et al., 2026). Calibration Tokens gives a frozen monocular depth model one token set, shared by all fisheye lenses and dropped for perspective images (Gangopadhyay et al., 2025), and RayTun3R fits positional-

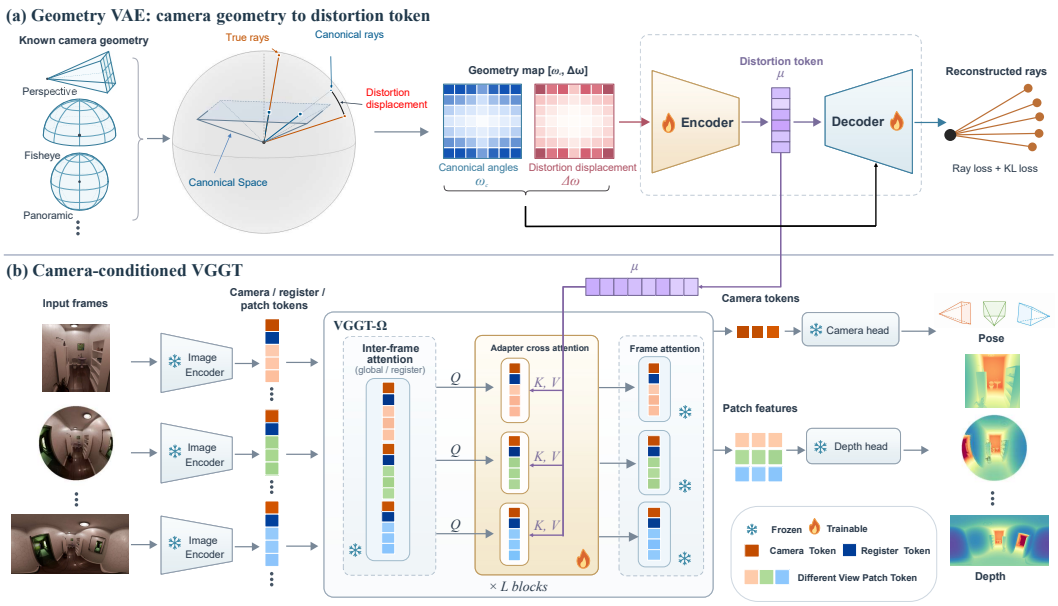

Figure 2: System overview. (a) Geometry-only pretraining: a camera's canonical angles ω_c and its distortion displacement $\Delta\omega^\star$, the coordinates of its distortion map, are encoded into a distortion token (Kingma & Welling, 2014) and decoded back into rays. (b) Inside the frozen VGGT-Ω (Wang et al., 2026a), the token conditions cross-attention adapters; the frozen camera and depth heads output pose, field of view and depth, and points place the resulting range along the known rays (§3.3).

encoding corrections per fisheye sequence at test time (Sinitsyn et al., 2026). EO-VGGT adapts a frozen VGGT to satellite pushbroom cameras (Luo et al., 2026), and X-Lens trains its own compact model (Zhou et al., 2026); both describe each camera by its full calibrated ray map rather than by what the pinhole model lacks. VGGT-Any complements these approaches: it gives a frozen 3D foundation model each camera's distortion map, computed here from known camera geometry, to extend the model across central-camera projections.

3 Method

VGGT-Any extends a frozen VGGT-style 3D foundation model, here VGGT-Ω, to calibrated central cameras without changing its weights (Figure 2). Its design follows one observation: the foundation model describes every image with a perspective camera, whose field of view it predicts together with pose and depth (Appendix B.1); a perspective camera captures any central camera to first order, so the model lacks only the camera's distortion from that perspective description. We therefore represent a camera by what the model already expresses and what it lacks (§3.1), learn a compact token for the latter from camera geometry alone (§3.2), and let this token condition the frozen model (§3.3). Appendices B and C give implementation details and derivations.

3.1 Canonical space and distortion space

In the traditional pinhole camera model, light travels in a straight line through the optical center onto a flat sensor, so the projection is linear in homogeneous coordinates and fully described by a matrix. Fisheye cameras instead use lens groups that refract incoming light: rays at large angles are bent before reaching the sensor, which enables fields of view up to and beyond $180°$. Each pixel still corresponds to a ray through a single optical center, but the perspective projection no longer holds, so the mapping becomes nonlinear and cannot be represented by a matrix alone.

A calibrated central camera thus maps each pixel p of its image domain $\mathcal{U}$ to a unit ray, $\Phi^\star : \mathcal{U} \to \mathbb{S}^2$ (Grossberg & Nayar, 2001). We split this ray map into two parts. The first is the camera's *paraxial*

camera, the pinhole that matches $\Phi^\star$ to first order at the principal pixel p_0, where $\Phi^\star(p_0) = e_z$:

$$A_{\mathrm{c}} = \left.\frac{\partial \Phi^\star_{xy}}{\partial p}\right|_{p=p_0}, \qquad K_{\mathrm{c}} = \begin{bmatrix} A_{\mathrm{c}}^{-1} & p_0 \\ 0^\top & 1 \end{bmatrix}, \tag{1}$$

with $\Phi^\star_{xy}$ the first two ray components. Its rays $\Phi_{\mathrm{c}}(p) = \mathrm{normalize}(K_{\mathrm{c}}^{-1}\tilde{p})$, for the homogeneous pixel $\tilde{p}$, form the *canonical space*, which is aligned with the perspective formulation that the foundation model assumes. The second is the *distortion map* $\Gamma = \Phi^\star \circ \Phi_{\mathrm{c}}^{-1}$, which carries each canonical ray to the camera's actual ray:

$$\Phi^\star = \Gamma \circ \Phi_{\mathrm{c}}, \qquad \Gamma(e_z) = e_z, \qquad d\Gamma(e_z) = I. \tag{2}$$

Because the paraxial camera absorbs all first-order behavior, every distortion map fixes the optical axis and is tangent to the identity there; these maps form the *distortion space* $\mathcal{D}$. The split is unique, and a pinhole camera has zero distortion by design: its distortion map is the identity, the origin of $\mathcal{D}$ (Appendix C.5). Classical lens models make this split within specific parametric families (Brown, 1966; 1971; Kannala & Brandt, 2006); defining it on ray directions and building it from the calibrated rays extends it to every central camera, including panoramas (Appendix C.10). In practice, central cameras are calibrated with a small set of projection models: Table 1 lists the thirteen found in our survey, and token pretraining and adapter training cover each of them, as a family or an exact submodel, except the far-field mirrored ball (Appendix C.1).

3.2 A DISTORTION TOKEN LEARNED FROM CAMERA GEOMETRY

Because the distortion map depends only on camera geometry, not on the scene, we learn its encoding without any image. We read it in angular coordinates, the longitude θ and twice the latitude φ of a ray (Appendix C.3): with $\omega_{\mathrm{c}} = (\theta_{\mathrm{c}}, \varphi_{\mathrm{c}})$ the canonical and $(\theta^\star, \varphi^\star)$ the actual angles at pixel p, the distortion map becomes the displacement

$$\Delta\omega^\star(p) = \big(\,\mathrm{wrap}_\pi(\theta^\star(p) - \theta_{\mathrm{c}}(p)),\ \varphi^\star(p) - \varphi_{\mathrm{c}}(p)\big), \tag{3}$$

which vanishes for a pinhole. An encoder E_η maps $[\omega_{\mathrm{c}}, \Delta\omega^\star]$ to a Gaussian posterior q_η over a single *distortion token* $z \in \mathbb{R}^{d_z}$ (Kingma & Welling, 2014) (Appendix B.3). It is trained jointly with a ray decoder that reconstructs the camera's rays $\hat{r}$ from z and ω_{c}, on procedurally sampled cameras of thirteen families, from the pinhole to fisheye lenses and panoramas (Appendix B.2):

$$\mathcal{L}_{\mathrm{tok}} = \mathbb{E}\big[2\arcsin\big(\tfrac{1}{2}\|\hat{r} - r^\star\|_2\big)\big] + \beta\, D_{\mathrm{KL}}(q_\eta \,\|\, \mathcal{N}(0, I))\,, \tag{4}$$

where the first term is the spherical angle to the true rays $r^\star$; by construction, the decoder produces only maps in the distortion space (Appendices B.4 and C.8). The token therefore retains the camera's geometry (Appendix A.3) but carries no scene content or image appearance to overfit to. After this stage, the decoder is discarded and the encoder frozen: camera geometry is learned once, independently of how the foundation model uses it, and one token per calibration suffices whatever the image resolution.

3.3 CONDITIONING THE FROZEN FOUNDATION MODEL

The token conditions the frozen foundation model through residual cross-attention adapters at sites $\ell \in \mathcal{L}$ that follow its inter-frame attention layers (Appendix B.7). At each site, frame i's token z_i, the posterior mean, becomes n_c conditioning tokens, and every token h_i of the frame attends to them:

$$T_{\ell,i} = F_\ell\, \mathrm{LN}_{\mathrm{shared}}(z_i), \qquad h_i \leftarrow h_i + P_\ell\, \mathrm{CrossAttn}_\ell\big(\mathrm{LN}_\ell(h_i),\, T_{\ell,i}\big). \tag{5}$$

The output projections P_ℓ start at zero, so training begins from the foundation model's features. Only the adapters and the shared normalization are trained, with ℓ_1 losses on pose and range over posed windows from all camera families, including heterogeneous windows, while the foundation model and encoder stay frozen (Appendix B.8); the model's perspective prior is thus kept and only the use of the distortion is learned. The foundation model's pinhole reconstruction ability is unaffected. Since each frame carries its own camera's token, one input window can mix camera models.

Points. With the predicted world-to-camera pose (R_i, t_i), depth D_i and the pinhole intrinsics $\hat{K}_i$ of the predicted field of view, each pixel's range is placed along its known ray:

$$X_i(p) = R_i^\top \big(D_i(p)\, \|\hat{K}_i^{-1}\tilde{p}\|_2\, \Phi^\star_i(p) - t_i\big). \tag{6}$$

Points on rays at and beyond 90°, which no pinhole camera can form, are thus reconstructed directly, without dividing by any ray component (Appendix C.13).

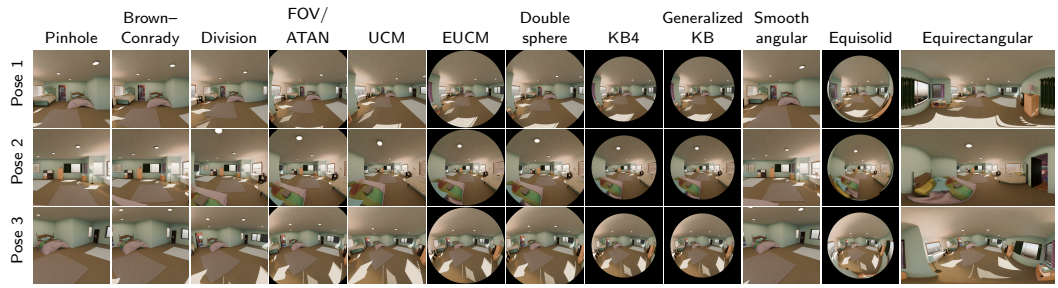

Figure 3: Twelve camera families (columns) at three poses (rows) in one room of LensZoo. Each row shares one physical camera pose; native aspect ratios and valid-domain boundaries are preserved.

Table 2: Fisheye reconstruction on ScanNet++ (Yeshwanth et al., 2023) and KITTI-360 (Liao et al., 2023).

		Point geometry			Pose metrics		
Dataset	Method	PointErr↓	Acc.↓	Comp.↓	AUC_{30} ↑	RRA_{30} ↑	RTA_{30} ↑
ScanNet++	VGGT-Ω PIN (Wang et al., 2026a)	0.5369	0.1693	0.1359	0.4321	0.9890	0.8058
	MapAnything + rays PIN	0.2637	0.1025	0.0688	0.5826	0.9896	0.9027
	Wid3R (Jung et al., 2026)	**0.0557**	**0.0260**	**0.0218**	**0.9670**	**1.0000**	**0.9993**
	VGGT-Any	0.0627	0.0320	0.0270	0.9306	0.9957	0.9926
KITTI-360	VGGT-Ω PIN	6.0649	1.0145	2.1041	0.4067	0.9325	0.8211
	MapAnything + rays PIN	8.4497	3.8866	1.8808	0.5200	0.8678	0.9304
	Wid3R	8.4577	1.1363	3.2669	0.2042	0.9965	0.4068
	VGGT-Any	**1.6384**	**0.3447**	**0.3951**	**0.9730**	**1.0000**	**0.9995**

PointErr, Acc., Comp.: lower is better; AUC, RRA, RTA at 30°: higher is better. VGGT-Any is close to Wid3R on ScanNet++ and best on KITTI-360. MapAnything (Keetha et al., 2026) + rays is given the exact per-pixel rays, yet its predicted rays stay within about 60° of the optical axis: it is conditioned on calibration, not a calibrated reconstruction; 191 of the 200 ScanNet++ test scenes are among its training scenes. PIN/ERP: pinhole-/ERP-only method. **Blue bold**/orange underline: best/second best per dataset and metric, over all methods.

4 Dataset

We introduce LensZoo, a synthetic indoor dataset for 3D reconstruction across central camera models. Existing non-pinhole 3D datasets each cover only a few camera types; LensZoo instead renders every physical camera pose under any central camera model, directly with Blender Cycles or by ray-map resampling of a co-located equirectangular RGB and range master (Appendix E). Scene geometry and poses stay fixed while image formation changes, which supports controlled comparisons across cameras and input windows that mix camera models.

LensZoo builds on Infinigen indoor environments (Raistrick et al., 2023; 2024): 505 rooms from 101 procedurally generated houses, with poses chosen by active view selection. Each of the 4,399 pose groups is rendered under 301 camera configurations spanning twelve central-camera families (Table 14), giving 1,324,099 RGB images with radial range, calibration and pose (Figure 3). Whereas the geometry-only camera sampling of §3.2 supplies ray fields for learning the distortion token, LensZoo supplies rendered scenes with appearance and geometric targets (§5.2).

5 Experiments

5.1 Experimental setup

Core implementation. We first train a token encoder and ray decoder using known calibration from thirteen procedural camera families. We then freeze these modules and the pretrained VGGT-Ω

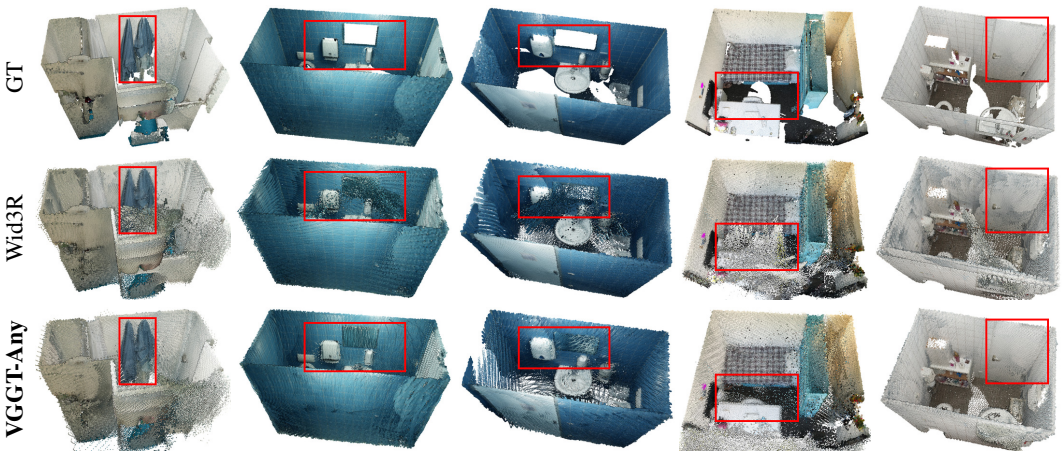

Figure 4: Fisheye reconstruction on five ScanNet++ (Yeshwanth et al., 2023) scenes. Red boxes: VGGT-Any cleaner than Wid3R (Jung et al., 2026).

foundation model, adapting it with twelve cross-attention adapters, their normalization parameters and eight conditioning tokens per adapter. Training on real and synthetic posed windows from all camera families, including heterogeneous windows, uses range and pose supervision; both stages run on 24 AMD MI210 GPUs. Appendix B gives the full architecture, objectives and training details.

Datasets. We evaluate on ScanNet++ (Yeshwanth et al., 2023), KITTI-360 (Liao et al., 2023), Holo360D (Ou et al., 2026), 360Loc (Huang et al., 2024), Replica360 (Straub et al., 2019; Rey-Area et al., 2022), PanoInfinigen, ZuriPano (Bozic et al., 2026) and LensZoo. Adapter training additionally uses TartanAir V2 (Wang et al., 2020), Aria Gen 2 Pilot (Kong et al., 2025), FIORD (Gunes et al., 2025), LaFiDa (Urban & Jutzi, 2017), TUM VI (Schubert et al., 2018), TUM monoVO (Engel et al., 2016), PanoraMIS (Benseddik et al., 2020), Princeton365 (Kayan et al., 2025), MP3D360 (Chang et al., 2017; Rey-Area et al., 2022), WideDepth (Indyk et al., 2026), Helvipad (Zayene et al., 2025) and SynWoodScape (Sekkat et al., 2022). Appendix D.1 details the datasets and their 80/20 scene split, and Appendix D.2 defines the metrics.

5.2 3D RECONSTRUCTION ACROSS CAMERA MODELS

Fisheye cameras. On ScanNet++, VGGT-Any far surpasses the unadapted VGGT-Ω, which describes the fisheye images with its pinhole camera, and nearly matches Wid3R (Table 2). On KITTI-360, VGGT-Any leads every metric, whereas Wid3R, which receives only a camera class, falls behind even VGGT-Ω on most metrics; given each camera's calibration, VGGT-Any stays accurate on both fisheye datasets. In the all-image ScanNet++ reconstructions of Figure 4, VGGT-Any is cleaner than Wid3R in the boxed regions.

Panoramic cameras. On Holo360D, VGGT-Any leads the camera-general methods on every metric and is second only to the camera-specific PanoVGGT, which is built and trained for panoramas alone; it is more accurate than PanoVGGT in Acc. (Table 3). On 360Loc, VGGT-Any estimates the most accurate poses of all methods, camera-specific ones included, while Wid3R reconstructs the most accurate point geometry, followed by PanoVGGT. On both datasets, the adaptation turns VGGT-Ω, which ranks last or second to last on every metric, into a competitive panoramic reconstructor. Figure 5 shows three Holo360D scenes; Appendix A.2 adds KITTI-360, 360Loc, PanoInfinigen, Replica360 and the V2 release of TartanAir.

Heterogeneous cameras. Table 4 covers ten non-pinhole camera models and a heterogeneous mix of six within each input window. Among camera-general methods, VGGT-Any has lower PointErr than Wid3R in every setting and is better on every metric on average, while Wid3R keeps an edge in pose on several single-camera models, notably double sphere, and on FOV leads every metric but

Table 3: Panoramic (ERP) reconstruction on Holo360D (Ou et al., 2026) and 360Loc (Huang et al., 2024).

Dataset	Method	Point geometry			Pose metrics		
		PointErr↓	Acc.↓	Comp.↓	AUC_{30} ↑	RRA_{30} ↑	RTA_{30} ↑
Holo360D	PanoVGGT ERP (Guo et al., 2026)	**1.2192**	0.5354	**0.4117**	**0.9441**	**1.0000**	**1.0000**
	CasaMaestro ERP (Ji et al., 2026)	4.2298	1.1074	1.3786	0.0240	0.1991	0.3788
	VGGT-Ω PIN (Wang et al., 2026a)	7.7980	1.5712	2.4194	0.0118	0.1234	0.3268
	MapAnything + rays PIN	118.2871	132.6354	2.2289	0.0028	0.0952	0.2294
	Wid3R (Jung et al., 2026)	2.4026	0.6244	0.7482	0.5367	0.8506	0.7468
	VGGT-Any	1.3863	**0.5165**	0.4204	0.6930	0.9567	0.9502
360Loc	PanoVGGT ERP	1.0305	0.4175	0.3799	0.0897	0.2283	0.4819
	CasaMaestro ERP	3.4354	1.1844	1.1590	0.1231	0.3394	0.4348
	VGGT-Ω PIN	8.7426	1.7645	2.6805	0.0300	0.2234	0.3345
	MapAnything + rays PIN	160.7525	220.2971	2.3736	0.0177	0.2017	0.3068
	Wid3R	**0.9438**	**0.2583**	**0.2771**	0.4135	0.4819	0.6993
	VGGT-Any	1.3682	0.5578	0.4745	**0.6419**	**0.9384**	**0.9662**

Metrics as in Table 2. PanoVGGT leads Holo360D, where VGGT-Any is second and best in Acc.; VGGT-Any leads pose on 360Loc. MapAnything (Keetha et al., 2026) + rays is given the exact per-pixel rays, yet its predicted rays stay within about 60° of the optical axis: it is conditioned on calibration, not a calibrated reconstruction. PIN/ERP: pinhole-/ERP-only method. **Blue bold**/orange underline: best/second best per dataset and metric, over all methods.

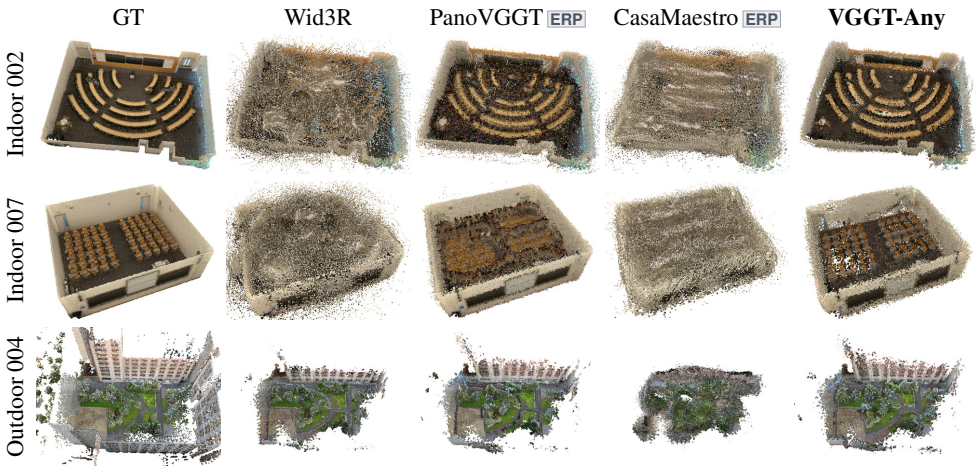

Figure 5: Panoramic (ERP) reconstruction on three Holo360D (Ou et al., 2026) scenes.

PointErr. Mixing camera models costs VGGT-Any less than Wid3R: for both methods, heterogeneous PointErr and AUC rank second-worst among their single-camera settings, but only Wid3R's RRA and RTA fall that low as well, so VGGT-Any leads every metric on heterogeneous input. Against camera-specific methods, VGGT-Any is second only to PanoVGGT on ERP.

5.3 Single-view depth estimation

On single-view radial range (Table 6, Appendix A.1), VGGT-Any, a multiview model given a single panorama, has the lowest Abs Rel among camera-general methods on all three datasets, including monocular models built for arbitrary cameras. It leads both metrics on ZuriPano and, in Abs Rel, is second only to the camera-specific PanoVGGT on Replica360 and PanoInfinigen.

Table 4: Point-cloud geometry and pose on LensZoo.

Camera	Method	Point geometry↓ PointErr	Acc.	Comp.	Pose metrics↑ AUC_{30}	RRA_{30}	RTA_{30}
KB4	Wid3R	0.4036	0.1641	0.0948	0.6727	**0.9670**	0.9299
	VGGT-Any	**0.1002**	**0.0576**	**0.0451**	**0.7771**	0.9335	**0.9571**
ERP	Wid3R	1.0782	0.2711	0.1008	0.0960	0.6290	0.1886
	CasaMaestro	0.3754	0.2335	0.1035	0.3209	0.7452	0.8276
	PanoVGGT	**0.0557**	**0.0374**	**0.0292**	**0.9287**	**0.9670**	**0.9794**
	VGGT-Any	0.1059	0.0689	0.0468	0.7901	0.9478	0.9624
Brown–Conrady	Wid3R	0.2173	0.0866	0.0640	**0.6750**	**0.8795**	0.8908
	VGGT-Any	**0.1081**	**0.0632**	**0.0445**	0.6191	0.8418	**0.9229**
UCM	Wid3R	0.5488	0.2211	0.1048	0.7041	**0.9504**	**0.9731**
	VGGT-Any	**0.0665**	**0.0415**	**0.0336**	**0.7801**	0.9212	0.9507
EUCM	Wid3R	0.9515	0.3284	0.2697	0.7336	0.9670	0.9677
	VGGT-Any	**0.0934**	**0.0566**	**0.0419**	**0.8366**	**1.0000**	**0.9859**
Division	Wid3R	0.4074	0.1782	0.1352	0.8297	0.9368	0.9599
	VGGT-Any	**0.0814**	**0.0484**	**0.0382**	**0.8370**	**0.9771**	**0.9885**
Double sphere	Wid3R	0.7564	0.2783	0.2178	**0.7750**	**0.9514**	**0.9562**
	VGGT-Any	**0.1374**	**0.0859**	**0.0474**	0.5802	0.9289	0.9325
Equisolid	Wid3R	0.8327	0.2849	0.1529	0.3917	**0.9615**	0.7731
	VGGT-Any	**0.1935**	**0.1049**	**0.0665**	**0.6073**	0.8450	**0.9160**
FOV	Wid3R	0.1166	**0.0589**	**0.0442**	**0.7946**	**0.9514**	**0.9616**
	VGGT-Any	**0.1102**	0.0669	0.0453	0.7358	0.9196	0.9481
Smooth angular	Wid3R	0.2040	0.1014	0.0723	0.6557	0.9368	0.9066
	VGGT-Any	**0.0970**	**0.0531**	**0.0432**	**0.8186**	**0.9594**	**0.9643**
Hetero-geneous	Wid3R	0.9574	0.3435	0.1228	0.2511	0.8083	0.5781
	VGGT-Any	**0.1630**	**0.0983**	**0.0488**	**0.5871**	**0.9702**	**0.9720**
Average	Wid3R	0.5885	0.2106	0.1254	0.5981	0.9036	0.8260
	VGGT-Any	**0.1142**	**0.0677**	**0.0456**	**0.7244**	**0.9313**	**0.9546**

Average: mean over the eleven settings. CasaMaestro, PanoVGGT: ERP-only; Wid3R: camera class only (Appendix D.1). **Blue bold**/orange underline: best/second best per camera.

Table 5: Distortion-token ablation on heterogeneous LensZoo cameras.

Dataset	Method	Condition	Projected range Abs Rel_{ov} ↓	δ_1^{ov} ↑	δ_1^{all} ↑	Cov.↑
LensZoo	**VGGT-Any**	Correct	**0.2024**	**0.9143**	**0.7360**	**0.8047**
	VGGT-Any	Bypass	0.3908	0.6605	0.3787	0.5698
	VGGT-Any	Random	0.3851	0.6681	0.3894	0.5794

Correct token, adapters bypassed, or random token (mean of three seeds). **Blue bold**: best.

5.4 Ablation Study

Distortion-token conditioning. On the same checkpoint and inputs, we replace the correct token with a Gaussian random one or bypass the adapters altogether (protocol in Appendix A.4). On heterogeneous input, the correct token outperforms both alternatives by a wide margin on every metric, while a random token performs about as well as bypassing the adapters (Table 5); the correct token is also best or tied-best in every non-pinhole homogeneous setting (Table 9). The adapters' gain thus comes from the camera information in the token, not from their added capacity.

6 Conclusion

We presented VGGT-Any, which extends a frozen, pinhole-trained 3D foundation model to any calibrated central camera. It separates each camera into a canonical space, aligned with the perspective formulation the model already assumes, and a distortion space, and conveys the distortion through a token learned from camera geometry alone, without affecting the foundation model's pinhole reconstruction ability. A single model thus handles fisheye, panoramic and heterogeneous-camera inputs: it leads the camera-general methods on most benchmarks, nearly matches Wid3R on ScanNet++ and is second only to the panorama-specific PanoVGGT on most panoramic benchmarks. VGGT-Any currently requires known calibration; future work will add the ability to predict it from the images.

AI use statement

In this work, we used generative AI tools to assist with translation, to discuss the feasibility of the theoretical framework, and to organize the written presentation of proofs. We have not used generative AI tools for any other task that requires disclosure: the theoretical framework, the mathematical claims and their proofs, the hypotheses, the design of the method and the experiments, the implementation, the dataset construction and processing, and the interpretation of the results are the authors' own work, and qualitative data analysis is not applicable to this work. Additionally, we

used generative AI tools to polish and check experiment code written by the authors, to polish the writing of the paper, to search for related work, and to produce the artwork of Figure 1; all other figures, including Figure 2, were made by the authors. We have reviewed all AI-assisted work, and every AI suggestion to the code and the text was checked by the authors before use. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Simon Baker and Shree K. Nayar. A Theory of Single-Viewpoint Catadioptric Image Formation. *International Journal of Computer Vision*, 35(2):175–196, 1999. doi: 10.1023/A:1008128724364.

Houssem-Eddine Benseddik, Fabio Morbidi, and Guillaume Caron. PanoraMIS: An ultra-wide field of view image dataset for vision-based robot-motion estimation. *The International Journal of Robotics Research*, 39(9):1037–1051, 2020. doi: 10.1177/0278364920915248.

Vukasin Bozic, Isidora Slavkovic, Dominik Narnhofer, Nando Metzger, Denis Rozumny, Konrad Schindler, and Nikolai Kalischek. Unified Panoramic Geometry Estimation via Multi-View Foundation Models. arXiv preprint arXiv:2605.26368, 2026.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Duane C. Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.

Duane C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 667–676, 2017. doi: 10.1109/3DV.2017.00081.

A. E. Conrady. Decentred Lens-Systems. *Monthly Notices of the Royal Astronomical Society*, 79 (5):384–390, 1919. doi: 10.1093/mnras/79.5.384.

Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 189–198. ACM, 1998. doi: 10.1145/280814.280864.

Frédéric Devernay and Olivier Faugeras. Straight Lines Have to Be Straight: Automatic Calibration and Removal of Distortion from Scenes of Structured Environments. *Machine Vision and Applications*, 13(1):14–24, 2001. doi: 10.1007/PL00013269.

Ruxiao Duan, Erin Hong, Dongxu Zhao, Eric Turner, Alex Wong, and Yunwen Zhou. Fisheye3R: Adapting Unified 3D Feed-Forward Foundation Models to Fisheye Lenses. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 17041 of *Lecture Notes in Computer Science*, pp. 1–19, 2026. doi: 10.1007/978-3-032-37429-5_1.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pp. 2366–2374, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/91c56ce4a249fae5419b90cba831e303-Abstract.html.

Jakob Engel, Vladyslav Usenko, and Daniel Cremers. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. arXiv preprint arXiv:1607.02555, 2016.

Andrew W. Fitzgibbon. Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 125–132, 2001. doi: 10.1109/CVPR.2001.990465.

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 (CVPR)*, pp. 26953–26963, 2026.

Suchisrit 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5198–5209, 2025. doi: 10.1109/ICCV51701.2025.00494.

Christopher Geyer and Kostas Daniilidis. A Unifying Theory for Central Panoramic Systems and Practical Implications. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1843 of *Lecture Notes in Computer Science*, pp. 445–461, 2000. doi: 10.1007/3-540-45053-X_29.

Michael D. Grossberg and Shree K. Nayar. A General Imaging Model and a Method for Finding Its Parameters. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pp. 108–115, 2001. doi: 10.1109/ICCV.2001.937611.

Ulas Gunes, Matias Turkulainen, Xuqian Ren, Arno Solin, Juho Kannala, and Esa Rahtu. FIORD: A Fisheye Indoor-Outdoor Dataset with LIDAR Ground Truth for 3D Scene Reconstruction and Benchmarking. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, volume 15726 of *Lecture Notes in Computer Science*, pp. 3–17, 2025. doi: 10.1007/978-3-031-95918-9_1.

Yijing Guo, Mengjun Chao, Luo Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. PanoVGGT: Feed-Forward 3D Reconstruction from Panoramic Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36444–36453, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26996–27006, 2025. doi: 10.1109/CVPR52734.2025.02514.

Namitha Guruprasad, Abhay Yadav, Rama Chellappa, and Cheng Peng. CAM3R: Camera-Agnostic Model for 3D Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 17040 of *Lecture Notes in Computer Science*, pp. 247–264, 2026. doi: 10.1007/978-3-032-37321-2_14.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary Position Embedding for Vision Transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 15068 of *Lecture Notes in Computer Science*, pp. 289–305, 2024. doi: 10.1007/978-3-031-72684-2_17.

Huajian Huang, Changkun Liu, Yipeng Zhu, Hui Cheng, Tristan Braud, and Sai-Kit Yeung. 360Loc: A Dataset and Benchmark for Omnidirectional Visual Localization with Cross-Device Queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22314–22324, 2024. doi: 10.1109/CVPR52733.2024.02106.

Ilia Indyk, Ignat Penshin, Ivan Sosin, Maxim Monastyrny, Aleksei Valenkov, and Ilya Makarov. WideDepth: Millimeter-Accurate Benchmark for Fisheye Depth Estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Yuzhou Ji, Xiaotian Yang, and Zhipeng Zhang. CasaMaestro: Multi-view Panoramas for House-Scale 3D Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 17017 of *Lecture Notes in Computer Science*, pp. 205–221, 2026. doi: 10.1007/978-3-032-37252-9_12.

Xiaosong Jia, Yanhao Liu, Yu Hong, Renqiu Xia, Junqi You, Bin Sun, Zhihui Hao, and Junchi Yan. DriveVGGT: Calibration-Constrained Visual Geometry Transformers for Multi-Camera Autonomous Driving. arXiv preprint arXiv:2511.22264, 2025.

Dongki Jung, Jaehoon Choi, Yonghan Lee, and Dinesh Manocha. RPG360: Robust 360 Depth Estimation with Perspective Foundation Models and Graph Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 190851–190871, 2025. doi: 10.52202/085713-5730.

Dongki Jung, Jaehoon Choi, Adil Qureshi, Somi Jeong, Dinesh Manocha, and Suyong Yeon. Wid3R: Wide Field-of-View 3D Reconstruction via Camera Model Conditioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 17018 of *Lecture Notes in Computer Science*, pp. 665–684, 2026. doi: 10.1007/978-3-032-37517-9_37.

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. doi: 10.1109/TPAMI.2006.153.

Karhan Kayan, Stamatis Alexandropoulos, Rishabh Jain, Yiming Zuo, Erich Liang, and Jia Deng. Princeton365: A Diverse Dataset with Accurate Camera Pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7645–7654, 2025. doi: 10.1109/ICCV51701.2025.00717.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal Feed-Forward Metric 3D Reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 499–509, 2026. doi: 10.1109/3DV69130.2026.00054.

Bogdan Khomutenko, Gaetan Garcia, and Philippe Martinet. An Enhanced Unified Camera Model. *IEEE Robotics and Automation Letters*, 1(1):137–144, 2016. doi: 10.1109/LRA.2015.2502921.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6114.

Chen Kong, James Fort, Aria Kang, Jonathan Wittmer, Simon Green, Tianwei Shen, Yipu Zhao, Cheng Peng, Gustavo Solaira, Andrew Berkovich, Nikhil Raina, Vijay Baiyya, Evgeniy Oleinik, Eric Huang, Fan Zhang, Julian Straub, Mark Schwesinger, Luis Pesqueira, Xiaqing Pan, Jakob Julian Engel, Carl Ren, Mingfei Yan, and Richard Newcombe. Aria Gen 2 Pilot Dataset. arXiv preprint arXiv:2510.16134, 2025.

Haodong Li, Wangguandong Zheng, Jing He, Yuhao Liu, Xin Lin, Xin Yang, Ying-Cong Chen, and Chunchao Guo. DA^2: Depth Anything in Any Direction. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=323ximYcsk.

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, 2023. doi: 10.1109/TPAMI.2022.3179507.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Qiyan Luo, Yingdong Pi, Lekang Wen, Jie Yang, Xiaoyu Wang, Haiming Zhang, and Mi Wang. EO-VGGT: Orbital Ray-Conditioned 3D Foundation Models for Satellite Multi-View Reconstruction. arXiv preprint arXiv:2607.00417, 2026.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM Optimized on the SL(4) Manifold. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 143976–144004, 2025. doi: 10.52202/085713-4324.

Christopher Mei and Patrick Rives. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3945–3950, 2007. doi: 10.1109/ROBOT.2007.364084.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16695–16705, 2025. doi: 10.1109/CVPR52734.2025.01556.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Jing Ou, Zidong Cao, Yinrui Ren, Zhuoxiao Li, Jinjing Zhu, Tongyan Hua, Shuai Zhang, Hui Xiong, and Wufan Zhao. Holo360D: A Large-Scale Real-World Dataset with Continuous Trajectories for Advancing Panoramic 3D Reconstruction and Beyond. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria Digital Twin: A New Benchmark Dataset for Egocentric 3D Machine Perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20076–20086, 2023. doi: 10.1109/ICCV51070.2023.01842.

Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal Camera Monocular 3D Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1028–1039, 2025. doi: 10.1109/CVPR52734.2025.00104.

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*, 48(3):2354–2367, 2026. doi: 10.1109/TPAMI.2025.3628473.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite Photorealistic Worlds Using Procedural Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12630–12641, 2023. doi: 10.1109/CVPR52729.2023.01215.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen Indoors: Photorealistic Indoor Scenes using Procedural Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, 2024. doi: 10.1109/CVPR52733.2024.02058.

Manuel Rey-Area, Mingze Yuan, and Christian Richardt. 360MonoDepth: High-Resolution 360° Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3752–3762, 2022. doi: 10.1109/CVPR52688.2022.00374.

Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5695–5701, 2006. doi: 10.1109/IROS.2006.282372.

David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1680–1687, 2018. doi: 10.1109/IROS.2018.8593419.

Ahmed Rida Sekkat, Yohan Dupuis, Varun Ravi Kumar, Hazem Rashed, Senthil Yogamani, Pascal Vasseur, and Paul Honeine. SynWoodScape: Synthetic Surround-View Fisheye Camera Dataset for Autonomous Driving. *IEEE Robotics and Automation Letters*, 7(3):8502–8509, 2022. doi: 10.1109/LRA.2022.3188106.

Daniil Sinitsyn, Nikita Araslanov, and Daniel Cremers. RayTun3R: Online Camera Adaptation in 3D Foundation Models. arXiv preprint arXiv:2607.02711, 2026.

John P. Snyder. Map Projections: A Working Manual. Professional Paper 1395, U.S. Geological Survey, Washington, D.C., 1987.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica Dataset: A Digital Replica of Indoor Spaces. arXiv preprint arXiv:1906.05797, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Javier Tirado-Garín and Javier Civera. AnyCalib: On-Manifold Learning for Model-Agnostic Single-View Camera Calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8044–8055, 2025. doi: 10.1109/ICCV51701.2025.00754.

Steffen Urban and Boris Jutzi. LaFiDa—A Laserscanner Multi-Fisheye Camera Dataset. *Journal of Imaging*, 3(1):5, 2017. doi: 10.3390/jimaging3010005.

Vladyslav Usenko, Nikolaus Demmel, and Daniel Cremers. The Double Sphere Camera Model. In *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 552–560, 2018. doi: 10.1109/3DV.2018.00069.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL https://jmlr.org/papers/v9/vandermaaten08a.html.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025. doi: 10.1109/CVPR52734.2025.00499.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21486–21499, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024. doi: 10.1109/CVPR52733.2024.01956.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A Dataset to Push the Limits of Visual SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916, 2020. doi: 10.1109/IROS45743.2020.9341801.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://openreview.net/forum?id=DTQIjngDta.

Jianxiong Xiao, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2695–2702, 2012. doi: 10.1109/CVPR.2012.6247991.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22, 2023. doi: 10.1109/ICCV51070.2023.00008.

Senthil Yogamani, Ciaran Hughes, Jonathan Horgan, Ganesh Sistu, Sumanth Chennupati, Michal Uricar, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed, Sanjaya Nayak, Saquib Mansoor, Padraig Varley, Xavier Perrotton, Derek O'Dea, and Patrick Perez. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9307–9317, 2019. doi: 10.1109/ICCV.2019.00940.

Jiayi Yuan, Haobo Jiang, De Wen Soh, and Na Zhao. VGGT-360: Geometry-Consistent Zero-Shot Panoramic Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19874–19883, 2026.

Mehdi Zayene, Jannik Endres, Albias Havolli, Charles Corbière, Salim Cherkaoui, Alexandre Kontouli, and Alexandre Alahi. Helvipad: A Real-World Dataset for Omnidirectional Stereo Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26975–26984, 2025. doi: 10.1109/CVPR52734.2025.02512.

Heng Zhou, Shuhong Liu, Yonghao He, Bohao Zhang, Fa Fu, Chenhui Hou, Xianbao Hou, Lijun Han, Cong Yang, and Wei Sui. X-Lens: Real-Time Metric Depth Estimation with Heterogeneous Cameras. arXiv preprint arXiv:2607.12993, 2026.

The appendix is organized as follows; the last column lists the main-text sections that refer to each part.

Section	Contents	Main text
A	Single-view depth, qualitative results, and distortion-token results (ray accuracy, conditioning, visualization)	§3.2, §§5.2–5.4
B	Foundation-model camera parameterization; token pretraining (procedural pool, encoder, decoder, optimization); adapters; training configuration and symbols	§§3.2, 3.3 and 5.1
C	Camera models (Table 11), canonical and distortion spaces, derivations and Proposition 1	§2, §§3.1–3.3
D	Dataset roles and splits (Table 12) and evaluation metrics	§§5.1 and 5.2
E	LensZoo construction	§4

A Additional results

A.1 Single-view depth results

This table accompanies §5.3; the LensZoo table is Table 4 in §5.2.

Table 6: Single-view panoramic radial-range (depth) estimation.

Dataset	Method	Abs Rel↓	δ_1 ↑
Replica360 (Straub et al., 2019; Rey-Area et al., 2022)	UniDepthV2 PIN MONO (Piccinelli et al., 2026)	0.1326	0.8124
	PanoVGGT ERP (Guo et al., 2026)	**0.0283**	**0.9955**
	VGGT-360 ERP (Yuan et al., 2026)	0.0934	0.9450
	Wid3R (Jung et al., 2026)	0.4804	0.4053
	DAC MONO (Guo et al., 2025)	0.1109	0.9159
	UniK3D MONO (Piccinelli et al., 2025)	0.0780	0.9569
	VGGT-Any	0.0670	0.9556
PanoInfinigen (Bozic et al., 2026)	UniDepthV2 PIN MONO	0.2575	0.6038
	PanoVGGT ERP	**0.0771**	**0.9194**
	VGGT-360 ERP	0.2140	0.7594
	Wid3R	1.2413	0.2017
	DAC MONO	0.3210	0.4679
	UniK3D MONO	0.1483	0.8228
	VGGT-Any	0.1369	0.8525
ZuriPano (Bozic et al., 2026)	UniDepthV2 PIN MONO	0.1981	0.7163
	PanoVGGT ERP	0.1426	0.7613
	VGGT-360 ERP	0.2386	0.6179
	Wid3R	0.1428	0.7884
	DAC MONO	0.2135	0.7371
	UniK3D MONO	0.1434	0.8306
	VGGT-Any	**0.1224**	**0.8561**

Single-image (SV) radial range on three panorama datasets; VGGT-Any has the lowest Abs Rel among general-camera methods on all three. PIN/ERP: pinhole-/ERP-only method; MONO: monocular; untagged: general-camera multiview. **Blue bold**/orange underline: best/second best per dataset and metric, over all methods.

A.2 Additional qualitative results

Figures 6–10 add qualitative reconstructions on KITTI-360, 360Loc, PanoInfinigen, Replica360 and the V2 release of TartanAir.

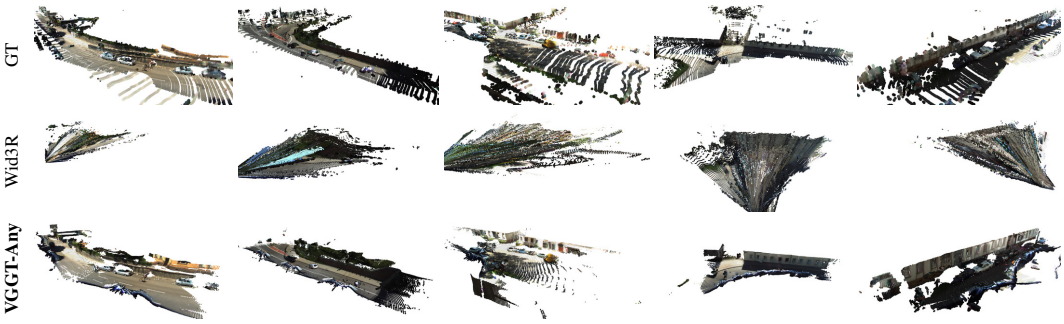

Figure 6: Qualitative reconstructions on five KITTI-360 (Liao et al., 2023) windows (columns): GT, Wid3R (Jung et al., 2026) and **VGGT-Any** (rows). The GT is from LiDAR and covers only a sparse band over the road.

A.3 Ray accuracy of the geometry-only stage

We measure how well the decoded rays reproduce the target rays for the token encoder used throughout the paper, the last checkpoint (step 100,000) of the final geometry-only training stage (Appendix B.9). The ray decoder is used only in pretraining (Appendix B.4), so this measures how completely the token, together with the paraxial camera, encodes a camera's ray field. It does not

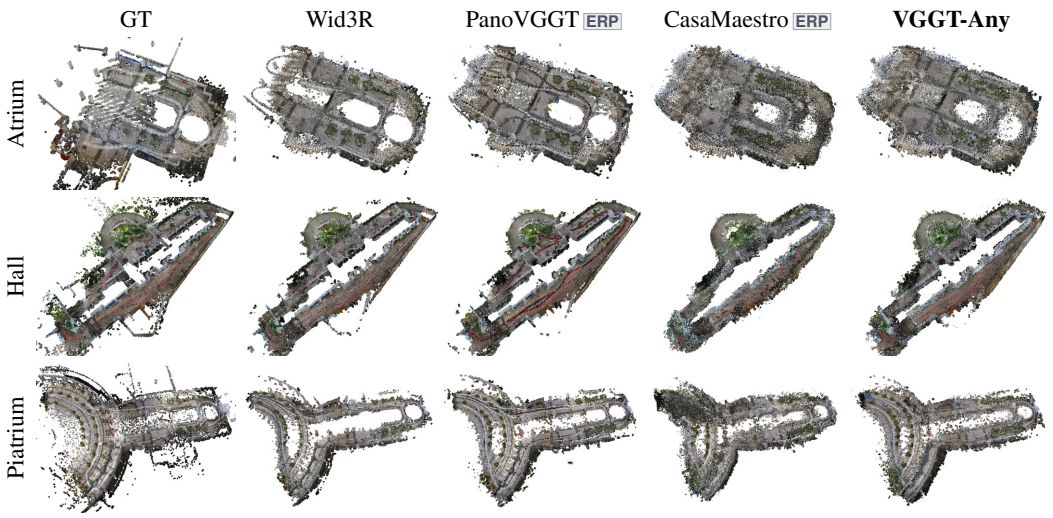

Figure 7: Qualitative reconstructions on three 360Loc (Huang et al., 2024) locations (rows): GT, Wid3R (Jung et al., 2026), PanoVGGT ERP (Guo et al., 2026), CasaMaestro ERP (Ji et al., 2026) and **VGGT-Any**. ERP: tied to equirectangular input.

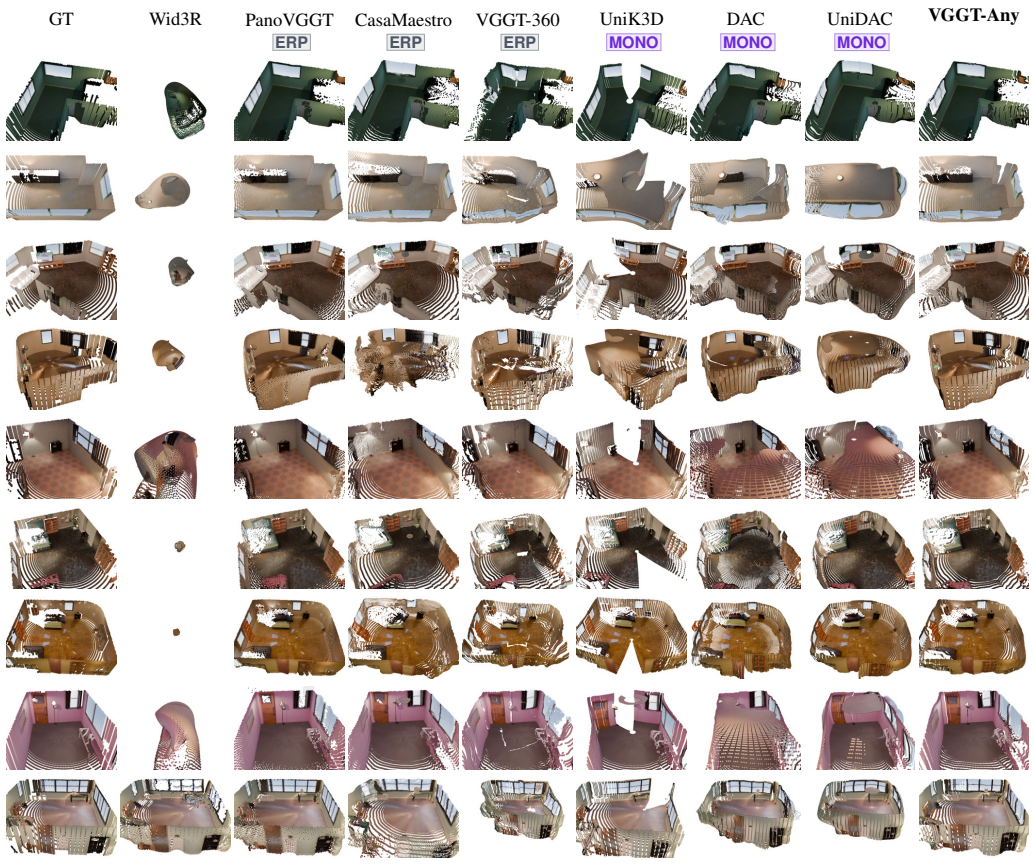

Figure 8: Qualitative single-panorama reconstructions on nine PanoInfinigen (Bozic et al., 2026) panoramas (rows): GT, Wid3R (Jung et al., 2026), PanoVGGT ERP (Guo et al., 2026), CasaMaestro ERP (Ji et al., 2026), VGGT-360 ERP (Yuan et al., 2026), UniK3D MONO (Piccinelli et al., 2025), DAC MONO (Guo et al., 2025), UniDAC MONO (Ganesan et al., 2026) and **VGGT-Any**. ERP: tied to equirectangular input; MONO: monocular.

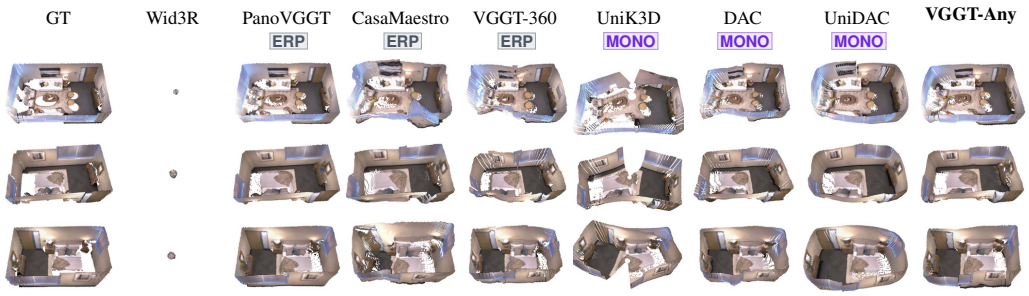

Figure 9: Qualitative single-panorama reconstructions on three Replica360 (Straub et al., 2019; Rey-Area et al., 2022) panoramas (rows); columns and markers as in Figure 8.

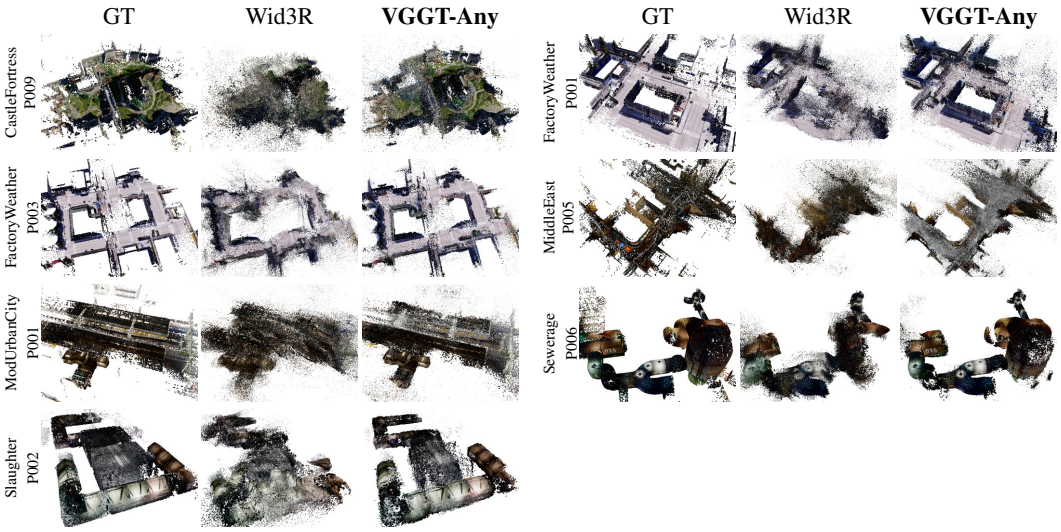

Figure 10: Qualitative reconstructions on seven scenes from the V2 release (`https://tartanair.org`) of TartanAir (Wang et al., 2020), two per row: GT, Wid3R (Jung et al., 2026) and **VGGT-Any**.

measure reconstruction, in which no ray is decoded and points are placed along the calibrated rays (Appendix B.6).

Protocol. For each of the thirteen families we draw 96 cameras from the training distribution of Appendix B.2, with seeds disjoint from training ($999983 + 1000k$ for family index k), 1,248 cameras in total. These are fresh draws from the thirteen families; the equirectangular family has no free parameter, so its 96 draws are the single training camera. Each camera is encoded at 512×512 and decoded from its posterior mean $z = \mu_\eta$ in float32. The error at a pixel is the geodesic angle $2 \arcsin(\frac{1}{2}\|\hat{r} - r^\star\|_2)$ between the decoded and the target ray, over the validity mask used in training; percentiles are exact over all 309,744,040 valid pixels. Table 7 reports the error per family. Table 8 evaluates the same cameras and mask with the camera's own token replaced by the zero token $z = 0$, the prior mean, or by the token of the next camera drawn from the same family, and adds the paraxial camera alone, $r = \mathrm{normalize}(K_c^{-1}\tilde{p})$, as a reference without a token.

Results. Pooled over all pixels, the error has mean $0.084°$, median $0.071°$, 99.9th percentile $0.642°$ and 99.99th percentile $1.109°$ (Table 7). The mean and the 99.9th percentile are below $1°$ for all thirteen families. The 99.99th percentile exceeds $1°$ for five families (KB4, EUCM, FOV, generalized KB and smooth angular), reaching at most $1.537°$ for FOV, and 1.74×10^{-4} of all valid pixels have an error above $1°$. The decoder uses the token (Table 8): with another camera's token from the same family, the mean error over families rises from $0.084°$ to $2.64°$, and with the zero token to

Table 7: Angular error of the decoded rays (degrees) for the token encoder used in the paper, on 96 fresh cameras per family (Appendix A.3).

Family	Mean	Median	p99	p99.9	p99.99
Pinhole	0.058	0.054	0.164	0.200	0.339
Brown–Conrady	0.083	0.073	0.254	0.344	0.456
Near-orthographic	0.079	0.076	0.183	0.333	0.548
Division	0.071	0.064	0.200	0.258	0.338
KB4	0.082	0.073	0.240	0.423	1.227
Generalized KB	0.102	0.091	0.295	0.606	1.401
Equisolid	0.076	0.065	0.234	0.459	0.787
UCM	0.073	0.061	0.260	0.437	0.742
EUCM	0.065	0.061	0.175	0.365	1.003
Double sphere	0.063	0.061	0.141	0.199	0.428
FOV	0.113	0.093	0.391	0.853	1.537
ERP[†]	0.134	0.130	0.392	0.691	0.928
Smooth angular	0.096	0.075	0.555	0.981	1.245
Mean over 13 families	0.084	0.075	0.268	0.473	0.844
Pooled over all pixels	0.084	0.071	0.284	0.642	1.109

px: xth percentile of the per-pixel error. [†]One fixed camera: the 96 ERP draws are identical. Pooled: 1,248 cameras, 309,744,040 valid pixels.

Table 8: Token substitution: angular error of the decoded rays (degrees, mean / p99.99) on the cameras and mask of Table 7, with the camera's own token replaced as described in Appendix A.3.

Family	Own token	Paraxial camera alone	Zero token ($z = 0$)	Other camera's token
Pinhole	0.058 / 0.339	0.000 / 0.000	7.73 / 70.44	0.239 / 3.11
Brown–Conrady	0.083 / 0.456	0.548 / 8.04	8.57 / 81.34	0.960 / 15.40
Near-orthographic	0.079 / 0.548	0.000 / 0.004	1.70 / 9.21	0.091 / 0.568
Division	0.071 / 0.338	1.315 / 20.64	8.82 / 83.15	1.084 / 12.52
KB4	0.082 / 1.227	9.654 / 36.90	5.95 / 75.22	4.450 / 82.01
Generalized KB	0.102 / 1.401	7.968 / 34.46	6.01 / 83.40	9.622 / 83.48
Equisolid	0.076 / 0.787	19.650 / 76.04	2.31 / 56.57	1.916 / 37.45
UCM	0.073 / 0.742	6.320 / 55.40	11.07 / 96.84	2.854 / 83.04
EUCM	0.065 / 1.003	7.710 / 58.11	7.68 / 76.40	1.596 / 66.52
Double sphere	0.063 / 0.428	4.935 / 49.42	7.62 / 72.66	0.694 / 24.07
FOV	0.113 / 1.537	6.597 / 64.24	7.70 / 81.31	10.105 / 96.36
ERP[†]	0.134 / 0.928	19.415 / 49.43	0.134 / 0.929	n/a[‡]
Smooth angular	0.096 / 1.245	0.491 / 14.05	8.59 / 75.41	0.639 / 10.76
Mean over 13 families	0.084 / 0.844	6.508 / 35.90	6.454 / 66.38	2.645 / 39.71[‡]

Paraxial camera alone: $r = \mathrm{normalize}(K_c^{-1}\tilde{p})$, with no token or decoder. Other camera's token: that of the next of the 96 draws of the same family. [†]One fixed camera. [‡]The 96 ERP draws are identical, so another draw's token is the camera's own; the column mean counts ERP at its own-token value (0.134 / 0.928).

6.45°, about 31 and 77 times the own-token error. The angle map at which the decoder is queried is computed from K_c alone, so camera-specific information beyond K_c can reach the decoder only through the token. The zero token is not a neutral input: the equirectangular camera's posterior mean lies near it (norm 0.18, against average norms of 1.4 to 4.1 for the other families), and the zero token decodes that camera as accurately as its own token. For the pinhole and near-orthographic families, whose paraxial camera is exact or nearly exact, the decoded rays carry a floor of about 0.06–0.08° (0.058° and 0.079°). This floor does not reach reconstruction: no ray is decoded there (Appendix B.6).

Table 9: Distortion-token conditioning across LensZoo cameras.

Method	Camera	Condition	Projected range Abs Rel_{ov} ↓	δ_1^{ov} ↑	δ_1^{all} ↑	Cov.↑
VGGT-Any	KB4	Correct	**0.2415**	**0.9428**	**0.8289**	**0.8791**
VGGT-Any		Bypass	0.5781	0.4752	0.1593	0.3295
VGGT-Any		Random	0.5536 ±0.0056	0.5049 ±0.0030	0.1837 ±0.0014	0.3587 ±0.0007
VGGT-Any	ERP	Correct	**0.0845**	**0.9483**	**0.9396**	**0.9908**
VGGT-Any		Bypass	0.3012	0.5356	0.5307	**0.9908**
VGGT-Any		Random	0.2924 ±0.0069	0.5424 ±0.0070	0.5374 ±0.0070	**0.9908** ±0.0000
VGGT-Any	Brown–Conrady	Correct	**0.2981**	**0.9470**	**0.8699**	**0.9173**
VGGT-Any		Bypass	0.3892	0.9115	0.6480	0.7114
VGGT-Any		Random	0.3845 ±0.0028	0.9114 ±0.0016	0.6586 ±0.0014	0.7234 ±0.0006
VGGT-Any	UCM	Correct	**0.2481**	**0.8969**	**0.6078**	**0.6781**
VGGT-Any		Bypass	0.4460	0.6546	0.2294	0.3447
VGGT-Any		Random	0.4353 ±0.0029	0.6650 ±0.0009	0.2402 ±0.0026	0.3551 ±0.0040
VGGT-Any	EUCM	Correct	**0.2548**	**0.8809**	**0.6822**	**0.7745**
VGGT-Any		Bypass	0.5177	0.5358	0.1927	0.3544
VGGT-Any		Random	0.4891 ±0.0024	0.5601 ±0.0013	0.2158 ±0.0012	0.3808 ±0.0014
VGGT-Any	Heterogeneous	Correct	**0.2024**	**0.9143**	**0.7360**	**0.8047**
VGGT-Any		Bypass	0.3908	0.6605	0.3787	0.5698
VGGT-Any		Random	0.3851 ±0.0132	0.6681 ±0.0031	0.3894 ±0.0028	0.5794 ±0.0022

Correct, Bypass, Random: correct-token conditioning, adapter bypass, Gaussian random conditioning. Runs are averaged over eight rooms; Random is mean ± sample SD over three seeds, i.e. assignment variability, not a confidence interval. "ov": occupied GT-valid overlap; "all": holes count as failures over all valid GT pixels; Cov.: occupied valid-GT coverage. **Blue bold**/orange underline: best/second best by unrounded mean per camera and metric. ERP coverage ties across conditions.

A.4 Distortion-token conditioning: protocol and full results

This experiment intervenes in the adapter conditioning at inference using the same first-epoch checkpoint, token encoder and pretrained foundation-model assets. Every condition receives identical ordered RGB inputs and target GT/support data for eight rooms, with five non-pinhole homogeneous camera settings and the heterogeneous condition. Bypass inference disables all adapter sites; no zero latent is consumed by an adapter. For random conditioning, each original image receives a 1024-dimensional vector with independent standard-normal entries. NumPy PCG64 draws are keyed by the random seed and original-image hash, so repeated instances of one image receive the same latent across camera settings. We use seeds 42, 43 and 44. These are inference interventions on one trained model; no condition is retrained.

Saved XYZ predictions are projected into a shared 512×512 reference raster (with 512×256 ERP content), retaining the nearest positive radial range without filling holes or enlarging splats. All conditions use the same GT-validity rule: native validity, sky/domain masks and finite GT range of at least $1\,\text{mm}$. Each room/camera case fits one pooled median(GT/projected range) scale on valid overlap. Overlap errors consequently have condition-dependent support; all-valid accuracy and coverage expose missing predictions. Brown–Conrady projection uses the inverse of the renderer's five-step ray field for its fixed lens.

A.5 DISTORTION-TOKEN VISUALIZATION PROTOCOL

Figure 11 visualizes posterior-mean distortion tokens for all thirteen procedural camera families. Several family distributions overlap in this shared projection, while individual families occupy different subsets of the map. Family identity enters encoder initialization (Appendix B.3); this is a qualitative view of a family-conditioned representation, without establishing label-free family discovery or reconstruction accuracy. Figure 12 gives a separate panel for each camera family.

Figures 11 and 12 use the token encoder behind every reported result, the last checkpoint (step 100,000) of the final geometry-only training stage (Appendix B.9). We draw 1,000 procedural camera parameter samples from each of the thirteen families in the procedural pool of Appendix B.2, for 13,000 draws in total. The sampler seed is $999983 + 1000k$, where k is the canonical family index. Extraction uses the configuration saved with this checkpoint: the equisolid-family parameter prior, 512×512 center crop/pad and zero mask erosion. The encoder runs in evaluation mode with float32 arithmetic and TF32 disabled. Each vector is the deterministic 1024-dimensional posterior mean $z = \mu_\eta$; the ray decoder is not evaluated. True camera-family identifiers select the encoder's family-specific initialization, as described in Appendix B.3.

Exact duplicate vectors are fitted once and then assigned identical projected coordinates. There are 12,001 unique tokens: one from all 1,000 equirectangular draws and 1,000 from each of the other twelve families, with no cross-family duplicates. We jointly center these unique vectors and apply unwhitened PCA with 50 components, without feature standardization. One Euclidean t-SNE (van der Maaten & Hinton, 2008) fit then uses perplexity 30, PCA initialization, random seed 42, early exaggeration 12, automatic learning rate (resolved to approximately 250.02) and a maximum of 1,500 iterations. Family labels enter neither PCA nor t-SNE fitting. Every camera panel reuses this single embedding and its axis limits, highlighting that family's tokens against the remaining tokens in gray. "Near-orthographic" denotes the near-orthographic long-lens Brown–Conrady family of Appendix B.2.

This balanced visualization sample describes distinct encoded camera geometries; it neither reproduces the training hard-example selection nor estimates camera prevalence. Duplicate collapse changes the weighting of the original draws, particularly for equirectangular cameras. The map therefore supports a qualitative inspection of family overlap, without assigning quantitative meaning to inter-cluster spacing, cluster area or point density. Family conditioning during encoding also precludes interpreting apparent separation as label-free discovery.

B CONVENTIONS AND IMPLEMENTATION DETAILS

B.1 CAMERA PARAMETERIZATION OF VGGT-Ω

VGGT-Ω (Wang et al., 2026a), like VGGT (Wang et al., 2025), maps unposed images of a static scene to per-frame cameras and dense geometry in a single forward pass. Its camera head predicts each frame's pose and field of view, which define a pinhole camera, and its depth head predicts depth, unprojected through that pinhole. The model thus assumes a perspective camera for every image and cannot represent rays at or beyond 90° from the optical axis, which no pinhole camera captures (Appendix C.2).

The nine numbers of the per-view code (Wang et al., 2026a) are stored as $[\,t,\, q,\, \alpha_y,\, \alpha_x\,]$: a translation, a unit quaternion giving $R(q) \in SO(3)$, then the vertical and the horizontal field-of-view angle in that order. The extrinsics map world coordinates to camera coordinates, so the camera center is $-R^\top t$ and not t. The intrinsic half of the code is the angle pair. For an image of size (W, H) and $0 < \alpha_x, \alpha_y < \pi$, the angles define the intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & W/2 \\ 0 & f_y & H/2 \\ 0 & 0 & 1 \end{bmatrix}, \qquad f_x = \frac{W/2}{\tan(\alpha_x/2)}, \qquad f_y = \frac{H/2}{\tan(\alpha_y/2)}, \tag{7}$$

a centered, zero-skew pinhole, and the foundation model forms its points only by unprojecting its depth D_i through it. That is the map used in the point chain (Appendix C.13) to build $\hat{K}$ from the field of view emitted by the frozen camera head, whereas the paraxial camera K_c is constructed from the known calibration (Appendix C.4). Parameterizing the intrinsics as angles rather than as

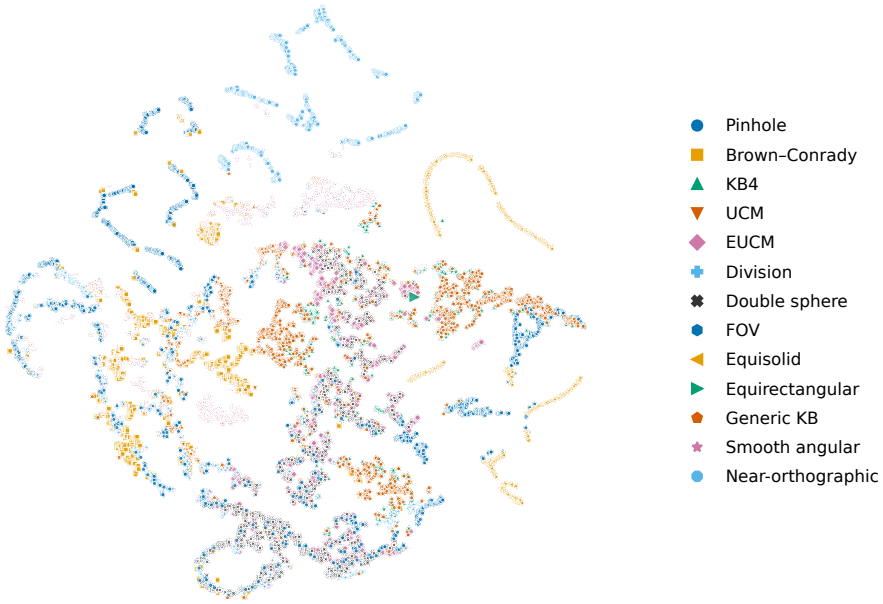

Figure 11: Shared t-SNE (van der Maaten & Hinton, 2008) view of distortion-token posterior means from the token encoder used throughout the paper (Appendix B.9). Colors and markers identify the thirteen camera families, whose projected distributions overlap. The 1,000 equirectangular draws share one unique token, shown with an enlarged marker.

pixel focal lengths makes them resolution-independent under a consistent resize, though not under a crop. Every intrinsic degree of freedom VGGT-Ω's camera model possesses, and every distortion degree of freedom it lacks, resides in this angle pair.

B.2 Procedural pool and validity mask

The procedural pool of §3.2 has thirteen families: the pinhole itself; the Brown–Conrady radial-tangential model (Brown, 1966; 1971; Conrady, 1919) and its near-orthographic long-lens limit; the division model (Fitzgibbon, 2001); the radially symmetric part of the Kannala–Brandt model (Kannala & Brandt, 2006) in its four-coefficient normalization (KB4), a generalized polynomial variant of it (generalized KB), and the equisolid angle fisheye, which Kannala and Brandt list as projection (iv); the unified (Geyer & Daniilidis, 2000; Mei & Rives, 2007), enhanced unified (Khomutenko et al., 2016) and double-sphere (Usenko et al., 2018) models; the FOV model (Devernay & Faugeras, 2001); an ideal central equirectangular projection; and a smooth angular perturbation of a radial base belonging to no closed-form family.

The pool is procedural and has no fixed membership: each training camera is generated on the fly by Algorithm 1 from camera parameters alone, with no image content. The data workers draw from separate random streams derived from the run seed, and the evaluation cameras of Appendix A.3 are drawn with seeds disjoint from these streams. A candidate camera's family is drawn uniformly from all thirteen, with no family held out, and its parameters are drawn uniformly from the ranges of Table 10. The cameras that enter the gradient are selected from these candidates by the rule of Algorithm 2, so their family mix need not be uniform. Every family is generated with short side $S = 512$, a long side set by an aspect ratio $\alpha_{\text{asp}} \sim \mathcal{U}(0.6, 1.8)$ and the principal pixel at the image center, and is then center-cropped to $S \times S$; the crop is therefore the same centered window for every aspect ratio. The equirectangular family instead keeps the full sphere: its $2S \times S$ raster is resized to $S \times S/2$ and padded to $S \times S$. Having no free parameter, this family is a single fixed camera. Target

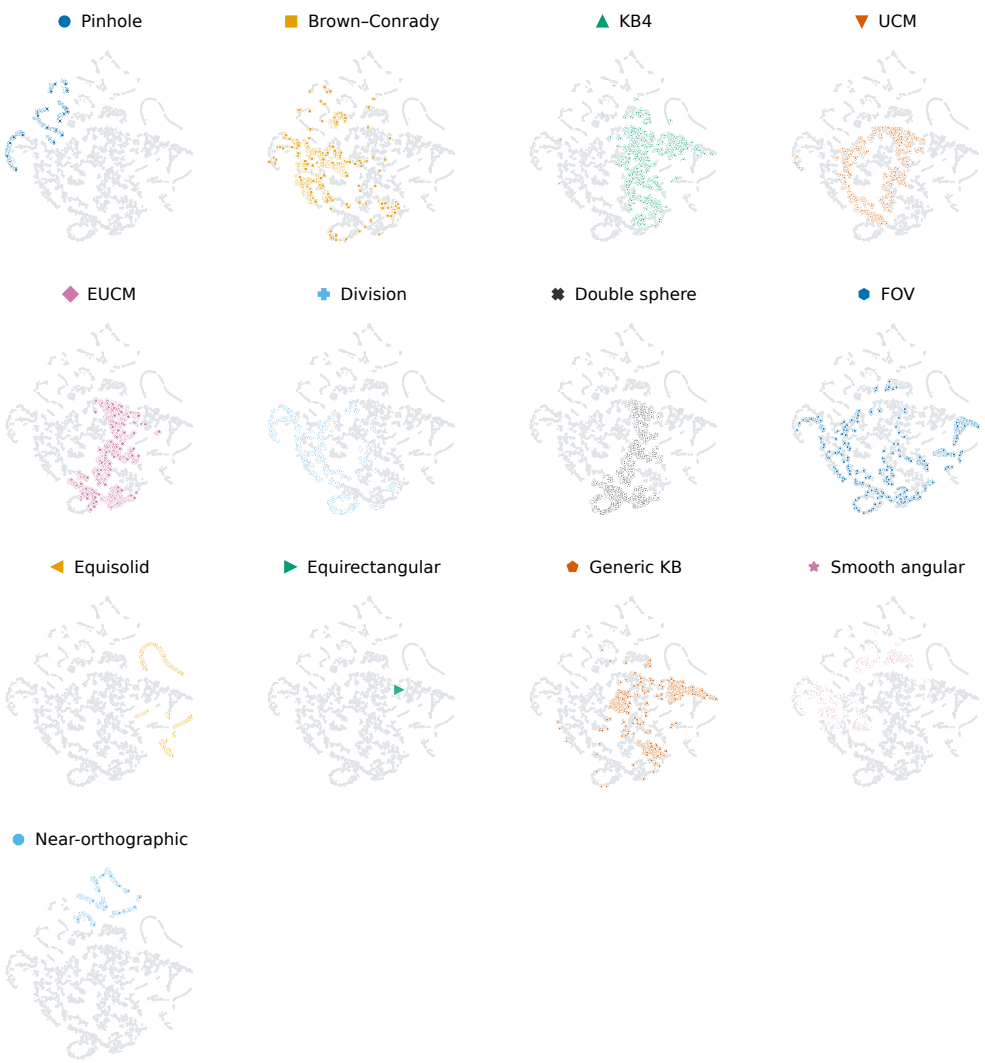

Figure 12: Distortion-token t-SNE (van der Maaten & Hinton, 2008) for each of the thirteen camera families. Each panel highlights one family in the same joint projection and axis limits used in Figure 11; gray points show the other families. The token encoder used throughout the paper encodes 1,000 draws per family. Exact duplicates share coordinates, so all equirectangular draws appear at one point, shown with an enlarged marker; the other families each contribute 1,000 unique tokens.

rays, paraxial intrinsics, canonical angles and distortion displacements are computed from camera geometry alone.

For each sampled camera, the validity mask intersects the family's per-pixel rule of Table 10 with chart validity, which excludes the $\pm y$ poles and the back-meridian seam of Appendix C.3 at a tolerance of 10^{-3} on the ray components. A maximum polar angle bounds validity for the equisolid family only; the other families are limited by their own domains. Folds are handled per family: Brown–Conrady rejects folded draws at draw time (Algorithm 1, step 3) and has no per-pixel monotonicity mask, whereas the two Kannala–Brandt types are masked per pixel to the monotone branch of the radial law on which the Newton inverse converges. The Brown–Conrady base of the smooth angular perturbation is neither fold-tested nor masked for folds, and in about 3.3% of its draws the crop corner lies beyond the base's first radial fold. The encoder input is not masked; the mask enters only the objective and the selection score.

Algorithm 1 Generating one training camera, on the fly in a data worker.

1: Draw a family uniformly from the thirteen.
2: Draw the aspect ratio $\alpha_{\text{asp}} \sim \mathcal{U}(0.6, 1.8)$ ($\alpha_{\text{asp}} = 2$ for ERP) and the family parameters, one uniform draw each from the ranges of Table 10.
3: **if** the family is Brown–Conrady **then** redraw (k_1, k_2), at most 64 times, while the distorted corner radius $\sqrt{1 + \max(\alpha_{\text{asp}}, 1/\alpha_{\text{asp}})^2}\,/\,(2\bar{q})$ is at least the peak distorted radius $r_d(r_u^{\text{fold}})$ at the first radial fold r_u^{fold}. The corner is that of the uncropped raster, so the test is conservative.
4: Set the raster: short side $S = 512$, long side $16\,\text{round}\big(S\max(\alpha_{\text{asp}}, 1/\alpha_{\text{asp}})/16\big)$ ($2S \times S$ for ERP); principal pixel p_0 at the image center; $f_x = f_y = \bar{q}S$, zero skew.
5: Evaluate the target rays $r^\star(p) = \Phi^\star(p)$ in float64 and the family's per-pixel validity $\text{val}_{\text{fam}}(p)$ at every pixel.
6: $A_{\text{c}} \leftarrow D_p\Phi^\star_{xy}(p_0)$ by central differences, and K_{c} by equation 20 (Appendix C.4).
7: $\omega_{\text{c}}(p) \leftarrow \mathcal{A}\big(\Phi_{\text{c}}(p)\big)$, with $\Phi_{\text{c}}(p) = \text{normalize}(K_{\text{c}}^{-1}\tilde{p})$.
8: $\omega^\star(p) \leftarrow \mathcal{A}\big(r^\star(p)\big)$, and $\Delta\omega^\star(p)$ by equation 23.
9: $\text{val}(p) \leftarrow \text{val}_{\text{fam}}(p)$, set to false where $\omega^\star(p)$ is not finite or $r^\star(p)$ lies at the $\pm y$ poles or on the back seam.
10: Center-crop ω_{c}, $\Delta\omega^\star$, $r^\star$ and val to $S \times S$ (ERP: resize to $S \times S/2$ and pad to $S \times S$).
11: **return** the encoder input $[\omega_{\text{c}}; \Delta\omega^\star] \in \mathbb{R}^{4\times S\times S}$ (float32, unmasked), the family identifier, $r^\star$ and val.

B.3 Token encoder

The ray maps we target are low-dimensional, since each is determined by a few parameters. We therefore compress each ray map into one token instead of feeding it densely to the foundation model; because the token is trained to reproduce the ray map, it compresses the camera's geometry rather than discarding it. The token is computed once per calibration, shared by every frame with that calibration, and adds a fixed number of conditioning tokens regardless of image resolution. The encoder maps a four-channel $S \times S$ geometry map to a token posterior (Kingma & Welling, 2014), $E_\eta : \mathbb{R}^{4\times S\times S} \to \mathbb{R}^{d_z} \times \mathbb{R}^{d_z}_{>0}$. Its read-out defines

$$q_\eta(z \mid \text{camera}) = \mathcal{N}\big(\mu_\eta, \text{diag}(\sigma_\eta^2)\big), \qquad z = \mu_\eta + \sigma_\eta \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_{d_z}). \tag{8}$$

The input stacks the canonical angles $\omega_{\text{c},i} = \mathcal{A} \circ \Phi_{\text{c},i}$ with the distortion displacement $\Delta\omega_i^\star$, with longitude differences wrapped as in Appendix C.5. Cameras that differ are encoded separately. Training samples this posterior; adaptation and inference use its mean. The sampling rule does not supply scene appearance, because all four input channels are computed from camera geometry.

A plain vision transformer in the DINOv2 (Oquab et al., 2024) style, trained only on the four-channel geometry map and never initialized from RGB-pretrained weights, patchifies the 512×512 input at stride 16 into a 32×32 grid of 1024-dimensional tokens and applies twelve pre-norm blocks of sixteen heads with axial two-dimensional rotary position embeddings (Su et al., 2024; Heo et al., 2024) on patch queries and keys (205 M parameters). One dedicated `[distortion]` token is prefixed to the sequence. Each block first attends over the whole sequence, then lets that token read the patch features $H^{(\ell)}$ again through a dedicated cross-attention,

$$t_\delta^{(\ell+1)} = t_\delta^{(\ell)} + \text{CrossAttn}\big(Q = t_\delta^{(\ell)},\ K = V = H^{(\ell)}\big), \tag{9}$$

with the projections omitted for clarity. A final layer normalization and a linear head map the token to the posterior parameters of equation 8. Its initial value is a per-family prototype, so family identity enters initialization rather than a classification objective. The prototypes are rows of an embedding table with one row per family plus a neutral row, selected by family identifier and detached from the gradient. The rows are refreshed offline by an exponential moving average toward the mean token read-out over a balanced per-family buffer and stay fixed during training; the rows of the generalized KB and smooth angular families and the neutral row were never refreshed and keep their random initialization. There is no classification loss or arg-max selection of an output family.

B.4 Ray decoder

Using the per-query shorthand from Appendix C.6, the decoder's core is

$$g_\psi : \Omega \times \mathbb{R}^{d_z} \to \mathbb{R}^3, \qquad \Delta(\omega; z) = m(\omega) g_\psi(\omega; z). \tag{10}$$

Table 10: Per-family priors of the procedural pool; all ranges are uniform, with focal length $f = \bar{q}S$ and $S = 512$. The last column is measured rather than prescribed: the largest polar angle of a valid ray in the $S \times S$ crop, median / maximum over the 96 evaluation cameras of Appendix A.3; ERP covers the full sphere.

Family	$\bar{q}$	Shape and distortion parameters	Per-pixel validity	Max. polar angle (°)
Pinhole	$[0.70, 2.00]$	none	all	28.7 / 44.3
Brown–Conrady	$[0.60, 1.60]$	$k_1 \in [-0.40, 0.25]$, $k_2 \in [-0.05, 0.21]$, $k_3 \in [-0.01, 0.01]$, $p_1, p_2 \in [-0.002, 0.002]$	finite unit ray	32.0 / 50.6
Near-orthographic	$[4, 9]$	Brown–Conrady with $k_1 \in [-0.02, 0.02]$, $k_2 = k_3 = p_1 = p_2 = 0$	finite unit ray	6.5 / 10.0
Division	$[0.50, 1.20]$	$\lambda \in [-0.40, 0]$	$r_d^2 < -1/\lambda$	43.9 / 76.6
KB4	$[0.25, 0.60]$	$k_1 \in [-0.05, 0.06]$, $k_2 \in [-0.02, 0.02]$, $k_3 \in [-0.01, 0.01]$, $k_4 \in [-0.005, 0.005]$	monotone branch; Newton residual $<10^{-6}$	83.2 / 90.0
Generalized KB	$[0.25, 0.60]$	$r_d = \theta\big(1 + \sum_{i=1}^{4} c_i \theta^{2i}\big)$, $c_i \in [-0.05, 0.05]$	$r_d \le \pi/2$; monotone branch; Newton residual $<10^{-6}$	70.0 / 90.0
Equisolid	$[0.20, 0.45]$	$\theta_{\max} \in [90°, 140°]$	$\theta \le \theta_{\max}$	109.4 / 139.2
UCM	$[0.30, 0.80]$	$\xi \in [0, 2.10]$	$r_d^2 < 1/(\xi^2 - 1)$ if $\xi > 1$	69.9 / 130.7
EUCM	$[0.30, 0.80]$	$\alpha_{\mathrm{EU}} \in [0.50, 0.70]$, $\beta_{\mathrm{EU}} \in [1.0, 1.2]$	$r_d^2 \le 1/\big(\beta_{\mathrm{EU}}(2\alpha_{\mathrm{EU}} - 1)\big)$	74.5 / 124.3
Double sphere	$[0.30, 0.80]$	$\xi \in [-0.30, -0.10]$, $\alpha_{\mathrm{DS}} \in [0.54, 0.62]$	$r_d^2 < 1/(2\alpha_{\mathrm{DS}} - 1)$; real inverse	59.0 / 117.8
FOV	$[0.40, 1.00]$	$\omega_{\mathrm{FOV}} \in [0.40, 2.10]$ rad	$r_d\, \omega_{\mathrm{FOV}} < \pi/2$	57.9 / 90.0
ERP	–	none; full $2S \times S$ raster resized to $S \times S/2$ and padded to $S \times S$	all	full sphere
Smooth angular	$[0.60, 1.60]$	Brown–Conrady base with $k_1 \in [-0.20, 0.10]$, $k_2 = k_3 = p_1 = p_2 = 0$; warp of equation 40 with $\sin(\nu\theta + \zeta)$ and $\sin(\nu\varphi + \zeta)$ in place of $\sin 2\theta$ and $\sin 2\varphi$; $A \in [0, 0.05]$, $\nu \in \{1, 2, 3\}$, $\zeta \in [0, 2\pi)$	base valid; $\det J_{\mathrm{warp}} > 0$	34.9 / 68.2

$\bar{q}$: focal length normalized by the short side S, as in Appendix E. r_d and θ: normalized image radius $\|\bar{p}\|_2$ and incidence angle in the forms of Table 11; in the smooth-angular row, θ and φ are the chart angles of equation 18. Target rays use closed-form inverses except for Brown–Conrady, its near-orthographic limit and the smooth-angular base, which use OpenCV's five-iteration fixed-point inverse (Bradski, 2000), and the two Kannala–Brandt types, which use a Newton inverse that clamps r_d at $\pi/2$ and returns front-hemisphere rays.

The actual network evaluates the whole angle map together, so this notation does not imply an independent coordinate MLP.

The decoder is used only in geometry-only pretraining. It patchifies the canonical angle map $\omega_{\mathrm{c}}(\cdot\,; K_{\mathrm{c}})$ with a stride-16 convolution into 1024 tokens of width 128 carrying the same axial rotary embedding, and projects the sampled distortion token to a single conditioning token. Six pre-norm residual blocks each run one fused attention, with the patch tokens as queries and the patch tokens together with that one distortion token as keys and values, and with zero-initialized output projections. The last block's tokens are upsampled, smoothed by a 3×3 convolution, concatenated at full resolution with a learned embedding of ω_{c} and mapped by a 1×1 convolution to the three channels of g_ψ, linearly and without squashing (1.54 M parameters). The envelope of equation 26 multiplies that output, so the anchoring conditions of Appendix C.6 hold for every token.

Algorithm 2 One optimizer step with camera resampling, run on each GPU with its own threshold τ (unset before the first step); $b = 4$ cameras per GPU, a budget of $n_{\text{cand}} = 500$ candidates, $\kappa = 0.02$.

1: $\mathcal{Q} \leftarrow$ empty list.
2: **repeat**
3: Draw b cameras with Algorithm 1 and append them to $\mathcal{Q}$.
4: Score each without gradient, the model in training mode (z sampled): $\chi =$ mean of the $\max\big(1, \text{round}(\kappa\, n_{\text{valid}})\big)$ largest per-pixel errors $2\arcsin(\frac{1}{2}\|\hat{r} - r^\star\|_2)$ over valid pixels, in degrees.
5: **until** (τ is set and at least b cameras in $\mathcal{Q}$ have $\chi > \tau$) or $|\mathcal{Q}| \geq n_{\text{cand}}$
6: **if** τ is unset **then** $\tau \leftarrow \max_{\mathcal{Q}} \chi$; keep the b highest-scoring cameras.
7: **else if** at least b cameras have $\chi > \tau$ **then** keep the first b of them in draw order.
8: **else** $\tau \leftarrow 0.7\,\tau$; keep the b highest-scoring cameras.
9: Forward the kept cameras again with gradient and $z = \mu_\eta + \sigma_\eta \odot \epsilon$. The loss is the mean over cameras of the mean error of their worst fraction κ of valid pixels, plus β, warmed up linearly over the first T_{w} steps, times the mean over cameras of $D_{\text{KL}}\big(q_\eta \,\|\, \mathcal{N}(0, I)\big)$.
10: All-reduce the gradients over all GPUs, clip the gradient norm at 1.0, take an AdamW (Loshchilov & Hutter, 2019) step and advance the learning-rate schedule.

B.5 Token encoder and ray decoder optimization

The encoder and ray decoder train jointly using target rays computed from camera parameters. The decoder receives a sampled token and the canonical angle map ω_{c}. Its objective is

$$\mathcal{L}_{\text{ray}} = \mathbb{E}\big[2\arcsin\big(\tfrac{1}{2}\|\hat{r} - r^\star\|_2\big)\big]\,, \qquad \mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{ray}} + \beta D_{\text{KL}}(q_\eta \,\|\, \mathcal{N}(0, I))\,. \tag{11}$$

The ray term is the spherical angle between unit directions, avoiding the nonuniform metric of raw angular coordinates. Training on geometric outputs rather than raw camera coefficients also follows the calibration approach of AnyCalib (Tirado-Garín & Civera, 2025). The loss acts on the enveloped reconstruction rather than dividing the target by the envelope. The thirteen-family pool and validity masks are specified in Appendix B.2; all thirteen families are drawn as training candidates, and none is held out. This geometry-only training scope covers the token encoder and ray decoder, while adapter training uses real and synthetic posed windows.

The expectation in equation 11 uses the worst fraction κ of valid pixels within each camera, and the objective averages this per-camera term over the cameras of a step. The cameras of each step are selected by resampling (Algorithm 2), which keeps cameras above a per-GPU difficulty threshold, lowered whenever too few candidates exceed it, rather than always the hardest ones. The Gaussian regularizer weight β is warmed up linearly over T_{w} steps. An auxiliary loss on $\Delta\omega^\star$ is disabled in the reported configuration; the active ray objective never divides a target by the envelope. The anchoring is structural, so no first-order penalty is needed. Appendix A.3 reports the accuracy of the decoded rays and shows that the decoder uses the continuous token, although it also receives the angular map.

B.6 Reconstruction and two-stage training

The adapted foundation model predicts a pose (R_i, t_i), a z-planar depth D_i and a field of view, from which we construct $\hat{K}_i$ using equation 7; in an all-pinhole window these are exactly the foundation model's predictions. The known calibration supplies the encoder input (the canonical angles of K_{c} and the distortion displacement) and the rays $\Phi_i^\star$ along which points are placed. The predicted $\hat{K}_i$ serves only to convert depth to range, for the reason given in Appendix C.13; no ray is decoded at reconstruction time.

Range-preserving point reconstruction. For $q_{K,i}(p) = \hat{K}_i^{-1}\tilde{p}$, with $\tilde{p}$ the homogeneous pixel coordinate, we retain the magnitude of the foundation model's unprojected point and place it along the known ray:

$$\text{range}_i(p) = D_i(p)\|q_{K,i}(p)\|_2, \qquad X_i(p) = R_i^\top\big(\text{range}_i(p)\,\Phi_i^\star(p) - t_i\big)\,, \tag{12}$$

which is equation 36 with the range written out. The construction divides by no component of the ray and therefore also accommodates backward directions. It takes the direction from the calibration

and the magnitude from the prediction, without establishing metric scale. For a pinhole camera the point differs from the foundation model's unprojected point only in using the known ray instead of the predicted one, and equals it when $\hat{K}_i$ equals the calibration (Appendix C.13).

Training stages and scope. First, procedural camera geometry trains the encoder and decoder with equation 11; the decoder is not used afterwards. Second, we freeze the encoder and train only the adapters and shared normalization on real and synthetic posed image windows, with ℓ_1 losses on the pose code and on range; supervision is normalized per window (Appendix B.8). These windows span all thirteen camera families and include heterogeneous-camera windows; pinhole frames occur only within heterogeneous windows. At inference, the adapters are conditioned on the posterior mean, and every frame's range is placed along its known rays (Figure 2). Recovering camera parameters from pixels is outside this method; its reconstruction applies to images accompanied by known central-camera geometry.

B.7 Inference composition and adapter details

Let $\mathcal{F}_A$ denote the foundation model with its adapters, let ξ_i contain the predicted extrinsics and field of view, and let $\mathcal{C}$ be the set of frames conditioned by a distortion token. For known per-frame camera parameters, inference composes

$$\begin{aligned}(\mu_{\eta,i}, \sigma_{\eta,i}) &= E_\eta\big([\omega_{\mathrm{c},i},\ \Delta\omega_i^\star]\big), \qquad z_i = \mu_{\eta,i}, \qquad i \in \mathcal{C},\\ (\xi_i,\ D_i) &= \mathcal{F}_A\big(\{I_j\}_{j=1}^N,\ \{z_j\}_{j\in\mathcal{C}}\big),\\ X_i(p) &= R_i^\top\Big(D_i(p)\,\big\|\hat{K}_i^{-1}\tilde{p}\big\|_2\,\Phi_i^\star(p) - t_i\Big).\end{aligned} \tag{13}$$

On an all-pinhole window $\mathcal{F}_A$ reduces to the foundation model. The known calibration defines the encoder input and the rays $\Phi_i^\star$, whereas the predicted $\hat{K}_i$ defines only the depth-to-range conversion. The encoder is frozen, and the mean μ_i is detached before the adapter path. The shared-camera case sets $\mu_i \equiv \mu$ and reuses one encoding across the window.

In the foundation model, an aggregator alternating frame-wise and inter-frame attention feeds the task heads; the camera head reads each frame's camera token and the depth head its patch tokens. Each adapter acts on that frame's complete feature sequence after the selected inter-frame block. It is a residual cross-attention: at its site $\ell \in \mathcal{L}$, the frame's distortion token z_i passes through one shared LayerNorm and a site-specific linear map F_ℓ, whose output is reshaped into n_c conditioning tokens, and the frame's full token sequence h_i (camera, register and patch tokens) then attends to them:

$$T_{\ell,i} = F_\ell\,\mathrm{LN}_{\mathrm{shared}}(z_i) \in \mathbb{R}^{n_c\times d_h}, \qquad h_i \leftarrow h_i + P_\ell\,\mathrm{CrossAttn}_\ell\big(\mathrm{LN}_\ell(h_i),\,T_{\ell,i}\big), \tag{14}$$

where the first argument supplies the queries, the second the keys and values, and d_h is the adapter width. Queries come from the foundation model's tokens, so every token receives its own update from the frame's camera. The adapter's site-specific conditioning projection and its attention and output projections are trainable, along with its feature normalization; the distortion-token normalization is shared across sites. There are $L_A = |\mathcal{L}|$ sites in the N_{layer}-layer aggregator, each producing n_c conditioning tokens from one geometry latent. The token count is configurable; the reported configuration uses the default $n_c = 8$. The twelve sites follow inter-frame layers $\mathcal{L} = \{0, 2, \ldots, 22\}$ of the 24-layer aggregator. Each adapter uses eight attention heads of width 128; together with the shared normalization, the trainable modules contain 151,175,168 parameters. The number L_A and the total trainable parameter count N_A therefore do not grow with the number of frames or cameras. Zero output initialization preserves the foundation model's features, pose and depth at initialization.

B.8 Adapter training

The adapter-training pool provides approximately $N_{\mathrm{win}} = 490{,}000$ posed windows from the $N_{\mathrm{col}} = 18$ collections of Table 12. It spans all $N_{\mathrm{fam}} = 13$ camera families and includes heterogeneous-camera windows; pinhole frames occur only within heterogeneous windows. Dataset roles are detailed in Appendix D.1. The objective combines a masked ℓ_1 term on the foundation model's nine-dimensional pose code, weighted five times, and an ℓ_1 term on the range of equation 12. Points lie on the known rays, so a camera-frame point term would reduce to the range term,

$\|\text{range}\,\Phi^\star - \text{range}^\star\Phi^\star\|_2 = |\text{range} - \text{range}^\star|$, and none is used; no ray is decoded. Ground truth is normalized per window by recentering on the first frame and dividing by a scale statistic, so the supervision is scale-invariant and metric scale is not learned.

B.9 Training configuration

The token encoder and ray decoder are trained jointly. The token encoder used throughout the paper is the last checkpoint (step 100,000) of the final geometry-only training stage, configured as follows. It runs 100,000 optimizer steps on three nodes of eight GPUs with four cameras per GPU, that is 96 cameras per step without gradient accumulation. It uses AdamW (Loshchilov & Hutter, 2019) with learning rate 2×10^{-4}, betas $(0.9, 0.999)$ and weight decay 0.05; a linear warm-up over the first 1000 steps followed by cosine decay (Loshchilov & Hutter, 2017) to 2×10^{-6} at step 100,000; bfloat16 autocast with float32 parameters; and gradient-norm clipping at 1.0. The Gaussian regularizer reaches weight $\beta = 10^{-4}$ after a linear warm-up of 2000 steps; the hard-pixel term uses the most difficult 2% of valid pixels per camera, and the cameras are selected by Algorithm 2 (Appendix B.5). Candidate cameras from all thirteen procedural families are generated on the fly by eight data workers per GPU (Algorithm 1), and the random seed is 42. The encoder has $205\,\text{M}$ parameters and the decoder $1.54\,\text{M}$.

The adapter stage trains the twelve adapters and their normalization parameters with AdamW under a one-cycle schedule with 5% warm-up: a peak learning rate of 2×10^{-5}, then a warm restart at 5×10^{-6} for three epochs of $4{,}164$ optimizer steps each, with gradient accumulation 2, at most 24 images per GPU at 512 pixels and 2–24 views per window. Both stages run on 24 AMD MI210 GPUs. The token encoder, the VGGT-Ω aggregator and its camera and depth heads remain frozen; posterior-mean latents are detached before conditioning the adapters. The evaluated model is the first-epoch checkpoint of the warm restart. Trainable and frozen components are specified in Appendix B.6, and the adapter objective is given in Appendix B.8.

B.10 Symbols and their values

The Method and the implementation details above use the following named symbols. Their values in the configuration described here are collected below; the experimental setup is where they are exercised.

Symbol	Meaning	Value	Introduced
S	token-encoder input side	512	§B.3
d_z	distortion-token dimension	1024	§B.3
σ_{env}	envelope scale, in chart radians	0.02	§C.6
κ	hard-pixel fraction per camera	0.02	§B.5
β	weight on the Gaussian regularizer	10^{-4}	§B.5
T_{w}	linear warm-up length for β, in steps	2000	§B.5
N_{layer}	aggregator depth of the instantiated model	24	§B.7
$\mathcal{L}$	injected layer set	$\{0, 2, \ldots, 22\}$	§B.7
L_A	number of adapters, $\lvert\mathcal{L}\rvert$	12	§B.7
n_c	conditioning tokens per adapter	8	§B.7
N_A	trainable parameters, adapters and shared norm	151 M	§B.7
N_{win}	posed windows in the adapter-training pool	$\approx 4.9 \times 10^5$	§B.8
N_{col}	collections in the adapter-training pool	18	§B.8
N_{fam}	camera families represented among them	13	§B.8

The instantiated model carries about 1 B parameters in total. Encoder and decoder sizes, the optimizer settings and the schedule are in Appendices B.3, B.4 and B.9.

C Camera representation: definitions and derivations

This appendix defines canonical space and distortion space, gives their angular coordinates, and develops the supporting arguments for the three-vector ray decoder used in geometry-only pretraining. Camera coordinates are $+x$ right, $+y$ down and $+z$ forward; $p = (u, v)^\top$ is a pixel and $\tilde{p}$ its

homogeneous lift. A central camera is specified by

$$\Phi^\star : \mathcal{U} \subset \mathbb{R}^2 \to \mathbb{S}^2, \qquad p \mapsto \Phi^\star(p). \tag{15}$$

A noncentral camera would additionally require a per-pixel ray origin. Throughout, the paraxial camera is the pinhole K_c of equation 20, with rays $\Phi_\mathrm{c}(p) = \mathrm{normalize}(K_\mathrm{c}^{-1}\tilde{p})$, canonical space $\Phi_\mathrm{c}(\mathcal{U})$ and canonical angles $\omega_\mathrm{c} = \mathcal{A} \circ \Phi_\mathrm{c}$, and the camera's distortion map is $\Gamma = \Phi^\star \circ \Phi_\mathrm{c}^{-1}$. The paraxial camera shares the target's optical center and coordinate frame: it is an intrinsic projection model, not another view or a relative pose.

C.1 Camera models of Table 1

Table 11 defines the camera models that head the columns of Table 1, in the same order. Each is modeled as a central camera in the sense of equation 15, the mirrored ball only in a far-field limit. A camera-frame point $X = (x, y, z)^\top$, with $d = \|X\|_2$, incidence angle $\theta = \arccos(z/d)$ from the optical axis and azimuth $\gamma = \mathrm{atan2}(y, x)$, maps to a normalized image point $\bar{p} = (\bar{u}, \bar{v})^\top$ and to the pixel $p = \mathrm{diag}(f_x, f_y)\,\bar{p} + (c_x, c_y)^\top$. Radially symmetric laws set $\bar{p} = r_d\,(\cos\gamma, \sin\gamma)^\top$ with image radius $r_d(\theta)$, and the pinhole point $\bar{p}_u = (x_u, y_u)^\top = (x/z, y/z)^\top$ has radius $r_u = \tan\theta$. Laws are stated in this projection direction, except that the division model gives r_u from r_d and the smooth angular model warps the ray map. Here θ is the incidence angle of Appendix C.2, not the chart longitude of Appendix C.3.

UCM is a special case of both EUCM and double sphere. With $\alpha_\mathrm{U} = \xi/(1+\xi)$ and the focal length rescaled by $1 - \alpha_\mathrm{U}$, it takes the form $\bar{p} = (x, y)^\top / \big(\alpha_\mathrm{U} d + (1 - \alpha_\mathrm{U}) z\big)$, which is EUCM at $\beta_\mathrm{EU} = 1$ and double sphere with its own offset ξ set to zero (Usenko et al., 2018). The mirrored ball is central only in the far-field limit, with the scene distant relative to the ball: a physical spherical mirror has no single viewpoint (Baker & Nayar, 1999). The columns are not the thirteen families of token pretraining (Appendix B.2): equidistant and mirrored ball are not separate families there, and the pool's near-orthographic Brown–Conrady and generalized KB families have no column. The LensZoo parameter settings are given in Table 14.

C.2 Pinhole coverage and a possible training trade-off

A pinhole places a ray at angle θ from the optical axis at image radius $r(\theta)$. For image width W and horizontal field of view α,

$$r(\theta) = f \tan\theta, \qquad f(\alpha) = \frac{W/2}{\tan(\alpha/2)}, \qquad \lim_{\theta \to \pi/2^-} r(\theta) = \infty, \qquad \lim_{\alpha \to \pi^-} f(\alpha) = 0. \tag{16}$$

At a fixed positive focal length, the radius diverges as a ray approaches 90° from the optical axis. No finite sensor therefore contains that direction under the pinhole law. Beyond 90°, the tangent changes sign and folds the ray onto the opposite side of the image instead of representing it as a distinct backward direction. A pinhole covers only an open hemisphere; an ideal central panorama can cover the full sphere. Hence no field of view predicted by the camera head can cover the rays at or beyond 90° in a fisheye or panoramic image.

The effect of retraining is a separate optimization question. Write w for the weights, $\mathcal{L}_\mathrm{p}$ for risk on perspective imagery and $\mathcal{L}_\mathrm{w}$ for risk on wide-field imagery. A simplified comparison is

$$w_\mathrm{p}^\star = \arg\min_w \; \mathcal{L}_\mathrm{p}(w), \qquad w_\lambda^\star = \arg\min_w \; \big[(1-\lambda)\,\mathcal{L}_\mathrm{p}(w) + \lambda\,\mathcal{L}_\mathrm{w}(w)\big], \qquad \lambda \in (0,1). \tag{17}$$

By definition of the perspective minimum, $\mathcal{L}_\mathrm{p}(w_\lambda^\star) \geq \mathcal{L}_\mathrm{p}(w_\mathrm{p}^\star)$. This inequality is non-strict: it does not prove that heterogeneous-camera training reduces perspective accuracy, and different minimizers of the two individual risks do not by themselves imply $w_\lambda^\star \neq w_\mathrm{p}^\star$. It describes the possibility that optimizing a mixed objective changes performance on the original regime. It also does not assume that pretrained weights are a global minimizer of the perspective risk. The motivation for freezing is to retain those weights while learning additional camera-conditioned adapters; preserving weights alone does not guarantee unchanged predictions after adaptation. An all-pinhole window yields exactly the foundation model's pose, field of view and depth.

Table 11: Camera models of Table 1 in column order, in the notation of Appendix C.1. All also carry f_x, f_y, c_x, c_y, fixed by the raster for ERP and by the ball's image for the mirrored ball.

Model	Projection	Parameters	Name and typical capture
Pinhole	$r_d = r_u = \tan\theta, \theta < \pi/2$	–	Rectilinear lenses; the foundation model's camera of equation 7
Brown–Conrady	$\bar{p} = (1 + k_1 r_u^2 + k_2 r_u^4 + k_3 r_u^6)\,\bar{p}_u$ plus decentering, in the OpenCV (Bradski, 2000) convention: $2p_1 x_u y_u + p_2(r_u^2 + 2x_u^2)$ in $\bar{u}$ and $p_1(r_u^2 + 2y_u^2) + 2p_2 x_u y_u$ in $\bar{v}$	k_1, k_2, k_3; p_1, p_2	Radial–tangential model; barrel or pincushion distortion, decentered elements (Brown, 1966; 1971; Conrady, 1919)
Division	$r_u = r_d/(1 + \lambda r_d^2)$	λ	Wide-angle lenses with barrel distortion (Fitzgibbon, 2001)
FOV	$r_d = \omega_{\mathrm{FOV}}^{-1} \arctan\big(2r_u \tan(\omega_{\mathrm{FOV}}/2)\big)$	ω_{FOV}	Field-of-view model, first order; wide-angle and fisheye lenses (Devernay & Faugeras, 2001)
Equidistant	$r_d = \theta$, i.e. KB4 with $k_1 = \cdots = k_4 = 0$	–	f-θ fisheye lenses (Kannala & Brandt, 2006)
KB4	$r_d = \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9$	$k_1, \ldots, k_4$	Four-coefficient radial Kannala–Brandt model; fisheye lenses (Kannala & Brandt, 2006)
Equisolid	$r_d = 2\sin(\theta/2)$	–	Equisolid-angle (equal-area) fisheye lenses (Kannala & Brandt, 2006)
UCM	$\bar{p} = (x, y)^\top/(z + \xi d)$: the unit-sphere point X/d projected from $(0, 0, -\xi)^\top$	ξ	Unified camera model; central catadioptric cameras, fisheye lenses (Geyer & Daniilidis, 2000; Mei & Rives, 2007)
EUCM	$\bar{p} = (x, y)^\top/\big(\alpha_{\mathrm{EU}} d_\beta + (1 - \alpha_{\mathrm{EU}})z\big)$, $d_\beta = \sqrt{\beta_{\mathrm{EU}}(x^2 + y^2) + z^2}$: an ellipsoid replaces the sphere	$\alpha_{\mathrm{EU}}, \beta_{\mathrm{EU}}$	Enhanced unified camera model; fisheye and catadioptric cameras (Khomutenko et al., 2016)
Double sphere	$\bar{p} = (x, y)^\top/\big(\alpha_{\mathrm{DS}} d_2 + (1 - \alpha_{\mathrm{DS}})z_\xi\big)$, $z_\xi = \xi d + z$, $d_2 = \|(x, y, z_\xi)\|_2$: two unit spheres ξ apart	$\xi, \alpha_{\mathrm{DS}}$	Fisheye lenses, also beyond 180° (Usenko et al., 2018)
ERP	Image axes linear in longitude and latitude (the chart angles of equation 18); a $2H \times H$ raster spans $360^\circ \times 180^\circ$	–	Equirectangular projection; full-sphere panoramas (Snyder, 1987)
Mirrored ball	$r_d = 2\sin(\theta/2)$ with $f = W/4$ on $\theta \in [0, \pi]$, θ from the ball-to-camera axis: the equisolid law over the full sphere	–	Mirrored sphere of image diameter W viewed from afar, as a light probe (Debevec, 1998)
Smooth angular	The chart angles ω_{b} of a radial base camera receive a smooth, tapered displacement $\delta\omega(\omega_{\mathrm{b}})$, as in equation 40	base; A	Our procedural family, in none of the other twelve; models no particular lens

C.3 Spherical angular coordinates

Define $\omega = (\theta, \varphi)^\top \in \Omega = (-\pi, \pi] \times [-\pi, \pi]$ by

$$\mathcal{A}(r) = \begin{bmatrix} \operatorname{atan2}(r_x, r_z) \\ 2\operatorname{atan2}\big(r_y, \sqrt{r_x^2 + r_z^2}\big) \end{bmatrix}, \qquad \mathcal{R}(\theta, \varphi) = \begin{bmatrix} \cos(\varphi/2)\sin\theta \\ \sin(\varphi/2) \\ \cos(\varphi/2)\cos\theta \end{bmatrix}, \tag{18}$$

with $\|\mathcal{R}(\omega)\|_2 = 1$ identically, so $\mathcal{R}$ inverts $\mathcal{A}$ on representatives. Here θ is the longitude about $+y$ and φ is twice the latitude, the factor of two carrying the $\pm 90^\circ$ latitude range onto $[-\pi, \pi]$. The longitude carries a branch cut and is undefined at the $\pm y$ poles, where a convention may be assigned for evaluation. The pair covers the sphere but does not constitute a single globally smooth chart,

which is why Proposition 1 is stated off the singular set and why the training mask of Appendix B.2 excludes the poles and the back-meridian seam. In VGGT-Any, the chart supplies only input coordinates: to the encoder, which reads a distortion map as its distortion displacement in these coordinates (Appendix C.5), and to the queries of the ray decoder, which predicts three-dimensional directions (Appendix C.6).

C.4 First-order matching of the paraxial camera

Definition 1 (First-order matched pinhole). *Let $\Phi^\star : \mathcal{U} \to \mathbb{S}^2$ be the target ray map and p_0 its principal pixel, with coordinates chosen so that $\Phi^\star(p_0) = e_z$. A pinhole K_c with an invertible transverse intrinsic block and principal point $(c_x, c_y) = p_0$ is* first-order matched *at p_0 when*

$$\Phi_\mathrm{c}(p_0) = \Phi^\star(p_0), \qquad D_p\Phi_\mathrm{c}(p_0) = D_p\Phi^\star(p_0). \tag{19}$$

For the centered zero-skew model of equation 7,

$$D_p\Phi_\mathrm{c}(p_0) = \begin{bmatrix} 1/f_x & 0 \\ 0 & 1/f_y \\ 0 & 0 \end{bmatrix},$$

so matching by two positive focal lengths requires the target Jacobian to take that form in the chosen coordinates. The condition is restrictive rather than formal: an arbitrary central camera may have cross-axis derivatives or an off-center principal pixel, and matching it then requires further intrinsic parameters or an explicit change of pixel coordinates and camera frame. The paraxial camera is therefore not read from the camera head: equation 20 constructs it from the calibration, with a full transverse block and principal point p_0, and it satisfies equation 19 whenever A_c is invertible (construction below). The foundation model's intrinsics $\hat{K}_i$ enter only the depth-to-range conversion of Appendix C.13. Because every decoded distortion map is tangent to the identity at the optical axis (Appendix C.8), it could not absorb a first-order mismatch at p_0; constructing K_c from the calibration leaves none beyond numerical differentiation error.

Construction from the calibrated ray map. Let p_0 be the camera's principal pixel, with $\Phi^\star(p_0) = e_z$, and let $\Phi^\star_{xy}$ denote the first two components of $\Phi^\star$. The paraxial camera describes the camera to first order at p_0 in perspective form:

$$A_\mathrm{c} = \left.\frac{\partial \Phi^\star_{xy}}{\partial p}\right|_{p=p_0}, \qquad K_\mathrm{c} = \begin{bmatrix} A_\mathrm{c}^{-1} & p_0 \\ 0^\top & 1 \end{bmatrix}. \tag{20}$$

The implemented construction of equation 20 evaluates the target ray map at its principal pixel p_0. It assumes $\Phi^\star(p_0) = e_z$; it does not recenter the map or rotate its camera frame. The constructor computes $J = D_p\Phi^\star(p_0)$ in float64 using central differences with step $h = 10^{-6}$ pixels and takes its first two rows as A_c. In particular, for the image-coordinate basis vectors e_j, $j \in \{1, 2\}$,

$$A_\mathrm{c}[:, j] \approx \frac{\Phi^\star_{xy}(p_0 + he_j) - \Phi^\star_{xy}(p_0 - he_j)}{2h}. \tag{21}$$

The determinant tolerance for inversion is 10^{-12}. The matrix K_c is recomputed from the calibrated ray map; a separate supplied calibration matrix is not substituted for this construction. Its full 2×2 block can contain cross-axis terms, unlike the two-focal-length camera head in equation 7.

At a unit ray $\Phi^\star(p_0) = e_z$, the third (optical-axis) row of its derivative is zero. Thus the pinhole map $\Phi_\mathrm{c} = \mathrm{normalize}(K_\mathrm{c}^{-1}\tilde{p})$ has derivative $\begin{bmatrix} A_\mathrm{c} \\ 0_{1\times 2} \end{bmatrix}$ at p_0 and matches the target to first order, up to numerical differentiation error. Its canonical angles $\omega_\mathrm{c} = \mathcal{A} \circ \Phi_\mathrm{c}$ form the encoder's first input map. The second is the distortion displacement $\Delta\omega^\star$ of equation 23, the pixel-indexed coordinates of the distortion map Γ, computed from the known target and not from the learned three-vector increment. For a pinhole target the construction returns the target's own intrinsic matrix, so Γ is the identity and $\Delta\omega^\star$ vanishes, both up to the differentiation error.

C.5 Canonical space, distortion space and angular coordinates

A finite pinhole's pixel-to-direction map is injective, so the paraxial camera maps $\mathcal{U}$ bijectively onto canonical space $\Phi_c(\mathcal{U})$. This set lies in the open front hemisphere and, for an interior principal pixel, contains a neighborhood of the optical axis $e_z = \Phi_c(p_0)$.

Definition 2 (Distortion space)**.** *A* distortion map *is a map* $\Gamma : \Phi_c(\mathcal{U}) \to \mathbb{S}^2$*, defined on the canonical space of a camera, that fixes the optical axis and is tangent to the identity there,*

$$\Gamma(e_z) = e_z, \qquad d\Gamma(e_z) = I \text{ on } T_{e_z}\mathbb{S}^2, \tag{22}$$

where Γ *is differentiable at* e_z *and* $T_{e_z}\mathbb{S}^2$ *is the tangent* (xy-)*plane there. The distortion maps form the* distortion space $\mathcal{D}$*. Each element carries its domain, and the origin of* $\mathcal{D}$ *is the identity on that domain. Two distortion maps are compared on the intersection of their domains, which contains a neighborhood of the optical axis.*

Lemma 1 (Factorization)**.** *Here a pinhole is* $\Phi(p) = \mathrm{normalize}(K^{-1}\tilde{p})$ *with* $K = \left[\begin{smallmatrix} M & c \\ 0^\top & 1 \end{smallmatrix}\right]$*,* M *invertible, principal point* c*. Let* $\Phi^\star$ *be differentiable at its principal pixel* p_0*, an interior point of* $\mathcal{U}$*, with* $\Phi^\star(p_0) = e_z$ *and* A_c *of equation 20 invertible. Then* $\Gamma = \Phi^\star \circ \Phi_c^{-1}$ *lies in* $\mathcal{D}$ *and* $\Phi^\star = \Gamma \circ \Phi_c$*. The pair* (K_c, Γ) *is unique: if* $\Phi^\star = \Gamma' \circ \Phi'$ *for a pinhole* Φ' *with principal point* p_0 *and a map* Γ' *satisfying equation 22, then* $\Phi' = \Phi_c$ *and* $\Gamma' = \Gamma$*; and* $\Gamma = \mathrm{id}$ *if and only if* $\Phi^\star$ *is a pinhole.*

Proof. By equation 20, $K_c^{-1}\tilde{p} = \big((A_c(p-p_0))^\top, 1\big)^\top$, so $\Phi_c(p_0) = e_z$, Φ_c^{-1} is smooth on canonical space, and $D_p\Phi_c(p_0)$ equals $D_p\Phi^\star(p_0) = \begin{bmatrix} A_c \\ 0_{1\times 2} \end{bmatrix}$ (Appendix C.4). Hence $\Gamma(e_z) = \Phi^\star(p_0) = e_z$ and, by the chain rule, $d\Gamma(e_z) = D_p\Phi^\star(p_0)\,[D_p\Phi_c(p_0)]^{-1} = I$ on $T_{e_z}\mathbb{S}^2$, the inverse being taken from $T_{e_z}\mathbb{S}^2$ to the pixel plane; thus $\Gamma \in \mathcal{D}$, and $\Phi^\star = \Gamma \circ \Phi_c$ holds by definition. For uniqueness, write $\Phi'(p) = \mathrm{normalize}\big(((B(p-p_0))^\top, 1)^\top\big)$ with B invertible. Differentiating $\Phi^\star = \Gamma' \circ \Phi'$ at p_0 and using $d\Gamma'(e_z) = I$ gives $D_p\Phi^\star(p_0) = D_p\Phi'(p_0)$, whose first two rows read $A_c = B$; hence $\Phi' = \Phi_c$ and $\Gamma' = \Phi^\star \circ \Phi_c^{-1} = \Gamma$. If $\Phi^\star(p) = \mathrm{normalize}(K^{-1}\tilde{p})$ is a pinhole, $\Phi^\star(p_0) = e_z$ makes p_0 its principal point, and equation 20 returns $K_c = K$, so Γ is the identity. Conversely, $\Gamma = \mathrm{id}$ gives $\Phi^\star = \Phi_c$, a pinhole. □

Classical lens models. The factorization of Lemma 1 is the paraxial decomposition behind classical lens models. Brown–Conrady applies, on the normalized image plane defined by its intrinsic matrix, a distortion that is the identity to first order at the principal point (Brown, 1966; 1971); that plane holds no ray at or beyond $90°$. The radially symmetric Kannala–Brandt model, in the normalization of Table 11, has the same first-order structure but reaches past $90°$ through a radial law in the incidence angle referenced to the equidistant projection (Kannala & Brandt, 2006). We build the paraxial camera from the calibrated ray map rather than from model parameters and apply the distortion map to ray directions, so one model-free representation, in which a pinhole camera has zero distortion, covers both, as well as non-radial cameras such as the equirectangular projection.

Scale and shape. The paraxial camera carries a camera's scale and principal point, and the distortion map the shape of its projection. Suppose $\Phi^\star(p) = \Psi\big(S^{-1}(p-p_0)\big)$ for a normalized ray map Ψ with $\Psi(0) = e_z$ and invertible $D\Psi_{xy}(0)$, and an invertible focal matrix S; the models of Table 11 have this form with $S = \mathrm{diag}(f_x, f_y)$. Then equation 20 gives $\Phi_c(p) = \Pi\big(S^{-1}(p-p_0)\big)$, with $\Pi(q) = \mathrm{normalize}\big(((D\Psi_{xy}(0)\,q)^\top, 1)^\top\big)$ the paraxial camera of Ψ itself, so $\Gamma = \Psi \circ \Pi^{-1}$ does not depend on S or p_0: the distortion maps of such cameras agree on the intersection of their domains. For a radially symmetric law with image radius $r_d(\theta)$, Γ keeps the azimuth and sends the canonical polar angle ϑ_c to $r_d^{-1}\big(r_d'(0)\tan\vartheta_c\big)$: the equidistant law sends ϑ_c to $\tan\vartheta_c$ at every focal length. This is a property of Γ; we do not claim that the learned token is invariant to focal length.

Angular coordinates. Write $\omega_c(p) = \mathcal{A}(\Phi_c(p))$ and $\omega^\star(p) = \mathcal{A}(\Phi^\star(p))$, with $B = \omega_c(\mathcal{U})$ canonical space in angular coordinates, indexed by pixel. Since $\mathcal{R}(\omega_c(p)) = \Phi_c(p)$, the distortion map carries $\omega_c(p)$ to $\omega^\star(p)$, and its coordinates are its distortion displacement: for $\mathrm{wrap}_\pi(a)$ the representative of a modulo 2π in $(-\pi, \pi]$,

$$\Delta\omega^\star\big(\omega_c(p)\big) = \begin{bmatrix} \mathrm{wrap}_\pi\big(\theta^\star(p) - \theta_c(p)\big) \\ \varphi^\star(p) - \varphi_c(p) \end{bmatrix}, \qquad \Gamma\big(\mathcal{R}(\omega)\big) = \mathcal{R}\big(\omega + \Delta\omega^\star(\omega)\big) \text{ on } B. \tag{23}$$

Only the longitude difference is wrapped. The displacement describes lens distortion and changes of projection alike, and $\Delta\omega^\star \equiv 0$ exactly when Γ is the identity, the origin of $\mathcal{D}$. It need not be small, and its norm generally differs from the spherical angle between the rays (Appendix C.9). A 180° equidistant fisheye whose image circle spans the image width, for instance, has a paraxial field of view of $2\arctan(\pi/2) \approx 115^\circ$, and its distortion map carries the canonical direction 57.5° off-axis on the image circle to the true ray at 90°. A 2:1 equirectangular panorama of width W has $A_\mathrm{c} = (2\pi/W)I_2$, so its paraxial camera has focal length $W/(2\pi)$ and a horizontal field of view of $2\arctan\pi \approx 144.7^\circ$. At the panorama's side edge, the canonical longitude is $\arctan\pi \approx 72.3^\circ$ while the true longitude is 180°: a displacement of 107.7°, larger than the canonical angle itself.

A learned angular field $\Delta\omega_\psi : B \times \mathcal{Z} \to \mathbb{R}^2$ defines in the same way a learned distortion map $\mathcal{R}(\omega) \mapsto \mathcal{R}\big(\Pi_\Omega[\omega + \Delta\omega_\psi(\omega; z)]\big)$ and, composed with Φ_c, the ray map

$$\widehat{\Phi}(p; K_\mathrm{c}, z) = \mathcal{R}\Big(\Pi_\Omega\big[\omega_\mathrm{c}(p) + \Delta\omega_\psi(\omega_\mathrm{c}(p); z)\big]\Big),\ \Pi_\Omega(\theta, \varphi) = \big(\mathrm{wrap}_\pi(\theta), \mathrm{clip}(\varphi, -\pi, \pi)\big). \quad (24)$$

Wrapping changes the coordinate representative without changing the ray. Backward directions need not cross the branch cut; clipping bounds latitude but introduces a zero-gradient region outside its interval. These output operations belong to the angular formulation and are not used by the three-vector decoder of equation 25, which never inverts the chart.

For the paraxial camera, $\omega_0 = \omega_\mathrm{c}(p_0) = 0$. The anchoring conditions (C1) $\Delta\omega_\psi(\omega_0; z) = 0$ and (C2) $D_\omega \Delta\omega_\psi(\omega_0; z) = 0_{2\times 2}$ are equation 22 written in angular coordinates (Appendix C.7), so they place the learned map in $\mathcal{D}$. They are imposed by $\Delta\omega_\psi(\omega; z) = m(\omega) g^\Omega_\psi(\omega; z)$, with the envelope of equation 26 and a two-channel core g^Ω_ψ. This theoretical core is distinct from the three-channel core g_ψ used by the ray decoder, and $\Delta\omega_\psi$ from the decoder's three-vector increment Δ. No distortion coefficient is required to define Γ or its displacement once the ray map is supplied; camera-family information still enters the implemented encoder through prototype initialization.

The four input channels of the encoder in equation 13 are the canonical angles ω_c, which fix the canonical space, and the distortion displacement $\Delta\omega^\star$ of equation 23, the coordinates of the distortion map. A pinhole camera is its own paraxial camera, so its distortion map is the identity and $\Delta\omega^\star$ vanishes identically. Angular wrapping remains part of the construction of the distortion displacement even though the pretraining decoder outputs a normalized three-vector.

C.6 Anchored ray decoding

The ray decoder is the token's training device in geometry-only pretraining. It is queried at the canonical angles $\omega = \omega_\mathrm{c}(p; K_\mathrm{c})$ of the known paraxial camera, is conditioned on a sampled distortion token z and predicts a three-dimensional increment, gated by an envelope that vanishes quadratically at the optical axis, $\omega_0 = \omega_\mathrm{c}(p_0) = 0$:

$$\Delta(\omega; z) = m(\omega)\, g_\psi(\omega; z) \in \mathbb{R}^3, \qquad \hat{r} = \mathrm{normalize}\big(\mathcal{R}(\omega) + \Delta(\omega; z)\big), \quad (25)$$

$$m(\omega) = 1 - \exp\left(-\frac{\|\omega - \omega_0\|_2^2}{\sigma^2_\mathrm{env}}\right). \quad (26)$$

Here $\mathcal{R}$ is the inverse angular map and $\mathrm{normalize}$ normalizes to unit length. Since $\mathcal{R}(\omega_\mathrm{c}(p)) = \Phi_\mathrm{c}(p)$, the decoded ray at pixel p is

$$\hat{r}(p) = \mathrm{normalize}\Big(\Phi_\mathrm{c}(p) + m\big(\omega_\mathrm{c}(p)\big)\, g_\psi\big(\omega_\mathrm{c}(p); z\big)\Big), \quad (27)$$

of which equation 25 is the form in angular coordinates. As a map of canonical directions, the decoder is the learned distortion map $\widehat{\Gamma}_z\big(\mathcal{R}(\omega)\big) = \mathrm{normalize}\big(\mathcal{R}(\omega) + \Delta(\omega; z)\big)$, so that $\hat{r} = \widehat{\Gamma}_z \circ \Phi_\mathrm{c}$. The pointwise notation denotes the output at one query; the implemented decoder processes the entire angle map jointly through patch attention. For a locally bounded core, the envelope makes the increment and its first derivative zero at ω_0, so $\widehat{\Gamma}_z$ lies in $\mathcal{D}$ for every token (Appendix C.8). Agreement with the target at p_0 therefore rests on K_c, which is first-order matched by construction (Appendix C.4). Away from the anchor, an unconstrained 3D increment can reach any unit direction, including backward rays, without angular clipping (Appendices C.8 and C.12). Formal matching and coverage statements are given in Appendices C.4 and C.10. After pretraining the decoder is not used: adapter training and reconstruction take the rays $\Phi^\star$ analytically from the calibration.

C.7 First-order matching and the distortion map at the optical axis

Write $F = \mathcal{A} \circ \Phi^\star$ and $a = \mathcal{A} \circ \Phi_{\mathrm{c}}$ for the two ray maps read in angular coordinates. Fix a neighborhood of p_0 on which a is a diffeomorphism onto its image, on which the angular coordinates are nonsingular and on which a consistent unwrapped branch of the longitude is chosen. On that neighborhood $a^{-1} = \Phi_{\mathrm{c}}^{-1} \circ \mathcal{R}$, so $F \circ a^{-1} = \mathcal{A} \circ \Gamma \circ \mathcal{R}$ is the distortion map read in angular coordinates, and the distortion displacement of equation 23 is its difference from the identity,

$$\Delta\omega^\star(\omega) = F\big(a^{-1}(\omega)\big) - \omega = \mathcal{A}\Big(\Gamma\big(\mathcal{R}(\omega)\big)\Big) - \omega. \tag{28}$$

Differentiating at $\omega_0 = a(p_0)$ by the chain rule gives

$$D\Delta\omega^\star(\omega_0) = DF(p_0)\,\big[Da(p_0)\big]^{-1} - I_2. \tag{29}$$

Hence $\Delta\omega^\star(\omega_0) = 0$ is equivalent to $F(p_0) = a(p_0)$, and $D\Delta\omega^\star(\omega_0) = 0$ is equivalent to $DF(p_0) = Da(p_0)$. Since $\mathcal{A}$ is a local diffeomorphism away from its singular set, these two statements are the angular-coordinate form of the first-order matching conditions of equation 19, and equally of $\Gamma(e_z) = e_z$ and $d\Gamma(e_z) = D_p\Phi^\star(p_0)\,[D_p\Phi_{\mathrm{c}}(p_0)]^{-1} = I$ on $T_{e_z}\mathbb{S}^2$, that is, of $\Gamma \in \mathcal{D}$ (Lemma 1). Given that a first-order matched pinhole exists, requiring membership in $\mathcal{D}$, which conditions (C1) and (C2) of Appendix C.5 express in coordinates, therefore imposes no *further* restriction on which central cameras the representation can describe; it fixes which pinhole serves as the paraxial camera. The two-focal-length, zero-skew head of Appendix C.4 cannot match a target whose Jacobian is not of the stated diagonal form. For a general nonsingular transverse Jacobian A_{c}, the full-block construction in Appendix C.4 instead supplies this pinhole, the paraxial camera.

Two caveats attach to equation 28. It is a local statement on an unwrapped branch, and the globally defined distortion displacement of equation 23 may be discontinuous where the branch changes. Separately, writing $\Delta\omega^\star = m\,g_\psi^\Omega$ for a *continuous* core g_ψ^Ω requires the quotient $\Delta\omega^\star/m$ to extend continuously through ω_0; vanishing to first order does not by itself supply that, since m vanishes to exactly second order. This is why the training objective of Appendix B.5 never forms the quotient.

C.8 The envelope keeps every decoded map in the distortion space

Let $u = \|\omega - \omega_0\|_2^2$. Then $m = 1 - \exp(-u/\sigma_{\mathrm{env}}^2) = u/\sigma_{\mathrm{env}}^2 + O(u^2)$ as $u \to 0$, so

$$m(\omega) = \frac{\|\omega - \omega_0\|_2^2}{\sigma_{\mathrm{env}}^2} + O\big(\|\omega - \omega_0\|_2^4\big) = O\big(\|\omega - \omega_0\|_2^2\big). \tag{30}$$

Suppose the core is bounded on some neighborhood of the optical axis ω_0, that is $\|g_\psi^\Omega(\omega; z)\|_2 \le C$ for $\|\omega - \omega_0\|_2 < \tau$ and some finite C. Then on that neighborhood $\|\Delta\omega_\psi(\omega; z)\|_2 \le C\,m(\omega) = O(\|\omega - \omega_0\|_2^2)$. Evaluating at ω_0, where $m(\omega_0) = 0$, gives $\Delta\omega_\psi(\omega_0; z) = 0$, which is (C1). The difference quotient then satisfies

$$\frac{\|\Delta\omega_\psi(\omega; z) - \Delta\omega_\psi(\omega_0; z)\|_2}{\|\omega - \omega_0\|_2} = O\big(\|\omega - \omega_0\|_2\big) \;\longrightarrow\; 0, \tag{31}$$

so $\Delta\omega_\psi$ is differentiable at ω_0 with $D_\omega\Delta\omega_\psi(\omega_0; z) = 0$, which is (C2). The argument uses only local boundedness of the core; it requires neither differentiability of g_ψ^Ω nor any property of its output activation, and it holds for every token z and every choice of weights. The squared distance in the exponent is what makes the argument work: an envelope built on $\|\omega - \omega_0\|_2$ vanishes only to first order and is not differentiable at ω_0, so (C2) would either fail or be undefined there.

The same argument applies componentwise to the three-vector increment $\Delta = m g_\psi$, so $\Delta(\omega_0; z) = 0$ and $D_\omega\Delta(\omega_0; z) = 0$. The decoded map $\widehat{\Gamma}_z\big(\mathcal{R}(\omega)\big) = \mathrm{normalize}\,\big(\mathcal{R}(\omega) + \Delta(\omega; z)\big)$ of Appendix C.6 therefore fixes the optical axis, $\widehat{\Gamma}_z(e_z) = \mathrm{normalize}(e_z) = e_z$, and its differential there is that of $\mathrm{normalize}$ at the unit vector e_z, the orthogonal projection onto $T_{e_z}\mathbb{S}^2$, applied to tangent vectors of the sphere at e_z, which it leaves unchanged: $d\widehat{\Gamma}_z(e_z) = I$. Every decoded map thus lies in $\mathcal{D}$, for every token z and every choice of weights. The construction separates local scale, which K_{c} carries, from higher-order directional variation, which Γ carries (Appendix C.5); it does not establish focal-length invariance of the token.

C.9 SPHERICAL IDENTITIES

Differentiating $\mathcal{R}(\omega)^\top \mathcal{R}(\omega) = 1$ in ω gives $2\,\mathcal{R}^\top J_\mathcal{R} = 0$ with $J_\mathcal{R} = D_\omega \mathcal{R}$, so the columns of the Jacobian are tangent to the sphere at $\mathcal{R}(\omega)$. From equation 18,

$$\frac{\partial \mathcal{R}}{\partial \theta} = \begin{bmatrix} \cos(\varphi/2)\cos\theta \\ 0 \\ -\cos(\varphi/2)\sin\theta \end{bmatrix}, \qquad \frac{\partial \mathcal{R}}{\partial \varphi} = \tfrac{1}{2}\begin{bmatrix} -\sin(\varphi/2)\sin\theta \\ \cos(\varphi/2) \\ -\sin(\varphi/2)\cos\theta \end{bmatrix}, \tag{32}$$

whose squared norms are $\cos^2(\varphi/2)$ and $\frac{1}{4}\big(\sin^2(\varphi/2) + \cos^2(\varphi/2)\big) = \frac{1}{4}$, and whose inner product is $-\frac{1}{2}\cos(\varphi/2)\sin(\varphi/2)\cos\theta\sin\theta + \frac{1}{2}\cos(\varphi/2)\sin(\varphi/2)\sin\theta\cos\theta = 0$. Hence

$$\mathcal{R}^\top J_\mathcal{R} = 0, \qquad G(\omega) = J_\mathcal{R}^\top J_\mathcal{R} = \operatorname{diag}\big(\cos^2(\varphi/2),\ \tfrac{1}{4}\big). \tag{33}$$

The longitude direction contracts as $|\varphi| \to \pi$, so a fixed coordinate error there corresponds to a vanishing directional error; this is why the objective of Appendix B.5 is measured on the sphere rather than in the chart.

The Jacobian maps infinitesimal coordinate changes to tangent vectors, but the finite displacement between two *unit* rays is not one. For unit rays r and r', expanding $\|r' - r\|_2^2 = 2 - 2\,r^\top r'$ gives $r^\top r' = 1 - \frac{1}{2}\|r' - r\|_2^2$ and therefore $r^\top(r' - r) = -\frac{1}{2}\|r' - r\|_2^2$, a radial component that is negative unless $r' = r$. The same expansion gives the chord form of equation 11, since $\|r' - r\|_2 = 2\sin(d/2)$ for the geodesic angle $d \in [0, \pi]$, so $d = 2\arcsin\big(\frac{1}{2}\|r' - r\|_2\big)$.

C.10 COVERAGE AND APPROXIMATION

If $\Delta\omega_\psi \equiv 0$ then $\widehat{\Phi} = \Phi_\mathrm{c}$: the learned map is the identity, the origin of $\mathcal{D}$, the canonical angle already lying inside the box so that Π_Ω acts as the identity. This is an algebraic property of the representation, not a guarantee that a learned predictor outputs zero on every perspective image. Pinhole performance does not rest on this property: on all-pinhole input the foundation model's pose, field of view and depth are unchanged. In this chart form the reconstructed ray has unit norm for every decoder output, since $\mathcal{R}$ is unit by construction, and the coordinates carry a nonuniform metric that makes a raw angular-coordinate error differ from a directional error, markedly so near the poles (Appendix C.9).

Proposition 1 (Coverage and approximation)**.** *Substituting the unrestricted distortion displacement $\Delta\omega^\star$ of equation 23 into equation 24 returns $\Phi^\star(p)$ at every pixel whose angular representatives are defined, with a convention assigned at the poles, so the representation admits full-sphere central ray maps; and on a compact valid pixel domain under pixel-area measure, bounded displacements can be approximated in L^2 to arbitrary accuracy by envelope-weighted neural fields whose core approximates continuous coordinate functions, excluding arbitrarily small neighborhoods of ω_0.*

The first half is a functional identity and the second an integrated one, both argued below with their caveats; neither half establishes exact representation by a fixed continuous network, compression into a fixed token dimension, or generalization to camera families outside the training distribution.

Coverage. Fix a pixel p at which $\omega_\mathrm{c}(p) = \mathcal{A}(\Phi_\mathrm{c}(p))$ and $\omega^\star(p) = \mathcal{A}(\Phi^\star(p))$ are both defined, a pole convention having been assigned where needed. Substituting the unrestricted distortion displacement of equation 23 into equation 24 gives, componentwise,

$$\begin{aligned} \Pi_\Omega\big[\omega_\mathrm{c}(p) + \Delta\omega^\star(\omega_\mathrm{c}(p))\big] &= \Big(\operatorname{wrap}_\pi\big(\theta_\mathrm{c} + \operatorname{wrap}_\pi(\theta^\star - \theta_\mathrm{c})\big),\ \operatorname{clip}(\varphi^\star, -\pi, \pi)\Big) \\ &= \big(\operatorname{wrap}_\pi(\theta^\star),\ \varphi^\star\big), \end{aligned} \tag{34}$$

using that wrap_π is additive modulo 2π and idempotent, and that $\varphi^\star \in [-\pi, \pi]$ already, so the clip is inactive. Since $\mathcal{R}$ inverts $\mathcal{A}$ on representatives, $\widehat{\Phi}(p) = \mathcal{R}(\omega^\star(p)) = \Phi^\star(p)$. Nothing in the argument restricts the target's field of view, so full-sphere central ray maps, the ideal central equirectangular projection included, are admitted: the distortion map carries canonical directions, all in the front hemisphere, to rays at and beyond $90°$. When a displaced longitude leaves $(-\pi, \pi]$ behind the camera, the outer wrap in equation 34 returns it to the equivalent representative. The statement is a functional identity about the representation. It says nothing about a fixed continuous network: $\Delta\omega^\star$ may be discontinuous at a branch cut of the chart or of the wrapped longitude difference even where $\Phi^\star$ is smooth, and no claim is made that it is realizable by a particular core or compressible into a fixed token dimension.

Approximation. Let $\mathcal{U}_c \subset \mathcal{U}$ be a compact valid pixel domain with finite pixel-area measure, and exclude a neighborhood of radius $\varepsilon > 0$ around the optical axis ω_0 together with neighborhoods of the chart's singular set. On the remaining compact set, the envelope is bounded below by $m_{\min} > 0$. For a bounded measurable target, the quotient $\Delta\omega^\star/m$ is therefore square-integrable under the push-forward of pixel-area measure by ω_c. Continuous functions are dense in this L^2 space. First choose a continuous function within any prescribed L^2 error of that quotient, then approximate it uniformly by the neural core, as the proposition assumes. Since $0 \le m \le 1$, multiplication by m cannot increase the L^2 error. This proves arbitrary integrated accuracy on the retained domain without assuming that the wrapped target is continuous. Bounded targets can also be approximated globally in L^2 by choosing continuous approximants that vanish on a smaller neighborhood of the excluded measure-zero sets: division by m is then harmless on their support, and the discarded contribution tends to zero with the neighborhood's pixel area. This argument assumes those excluded sets have zero pixel-area measure. It supplies an integrated approximation, not pointwise or derivative agreement at the anchor, and establishes neither fixed-token compression nor transfer to unseen camera families.

C.11 An alternative to the paraxial camera

Matching the camera to first order at p_0 constrains the base law only to first order, so other laws with the same local behavior could serve as the base law in place of the paraxial camera. An equidistant law with focal length f assigns the polar angle ρ/f to the image radius ρ about p_0, where a pinhole with the same focal length assigns $\arctan(\rho/f)$; the two agree in value and first derivative at p_0 and differ at third order in ρ. Unlike the pinhole's, the equidistant law's rays can enter the back hemisphere. VGGT-Any does not use this law. Backward rays need no such law, since the distortion map (Proposition 1) and the decoded increment (Appendix C.12) reach them from the paraxial camera's canonical space, and only the paraxial camera makes the pinhole camera, which the foundation model assumes, the origin of distortion space.

C.12 Reachability of the three-vector increment

Let ω be a query with $m(\omega) > 0$, let $r_c = \mathcal{R}(\omega)$ be the canonical ray there and let $r^\star \in \mathbb{S}^2$ be any unit target. Choose any $c > 0$ and set $\Delta = c\,r^\star - r_c$. Then

$$\mathcal{R}(\omega) + \Delta = c\,r^\star, \qquad \text{normalize}\left(c\,r^\star\right) = r^\star, \tag{35}$$

since $c > 0$ and $\|r^\star\|_2 = 1$. The core value $g_\psi = \Delta/m(\omega)$ realizes it. The construction places no condition on $r^\star$: the antipode $r^\star = -r_c$ gives $\Delta = -(c+1)r_c$, which normalizes to $-r_c = r^\star$ as required. Every unit direction, including every direction behind the camera, is therefore reachable from any query at which the envelope is nonzero, and no longitude wrap is involved. At $\omega = \omega_0$ the envelope vanishes and $\hat{r} = r_c$ is forced, which is the anchoring condition rather than a failure of reachability. The single inadmissible output is $\Delta = -r_c$, at which the sum vanishes and the direction is undefined; the implementation clamps the norm away from zero there.

Because c is free, the increment is not constrained to the tangent plane at r_c, and for a generic choice of c it is not tangent. It is tangent exactly when $r_c^\top \Delta = c\,r_c^\top r^\star - 1 = 0$, that is when $c = 1/\langle r_c, r^\star\rangle$, which is positive and therefore admissible for every target in the open hemisphere about r_c. The freedom in c therefore contains the tangent parameterization rather than excluding it; what Appendix C.9 rules out is only that the displacement between two *unit* vectors be tangent.

C.13 The pinhole identity of the point chain

For frame i, the foundation model predicts a world-to-camera pose (R_i, t_i), a depth map D_i and a field of view, which gives its pinhole intrinsics $\hat{K}_i$ by equation 7. Each pixel receives the range $\text{range}_i(p) = D_i(p)\,\|\hat{K}_i^{-1}\tilde{p}\|_2$, with $\tilde{p} = (p^\top, 1)^\top$, which is placed along the known ray,

$$X_i(p) = R_i^\top\left(\text{range}_i(p)\,\Phi_i^\star(p) - t_i\right), \tag{36}$$

so points on sideways and backward rays are formed without dividing by any ray component.

The point chain converts depth to range with the foundation model's intrinsics $\hat{K}_i$ rather than the known K_c because D_i is produced together with $\hat{K}_i$: for a pinhole camera, $\text{range}_i(p)$ is exactly the

distance of the foundation model's unprojected point. For other cameras, D_i is a positive per-pixel quantity that the adapters train so that $\mathrm{range}_i(p)$ equals the true range, and range is positive in every direction.

Consider an all-pinhole window and a frame with intrinsic matrix K. The depth D_i, the pose (R_i, t_i) and the intrinsics $\hat{K}_i$ are the foundation model's own outputs, and the known rays are $\Phi^\star(p) = \mathrm{normalize}(K^{-1}\tilde{p})$. If the predicted intrinsics equal the calibration, $\hat{K}_i = K$, the range of equation 12 placed along the known ray gives

$$\mathrm{range}_i(p)\,\Phi^\star(p) = D_i(p)\,\|K^{-1}\tilde{p}\|_2\,\frac{K^{-1}\tilde{p}}{\|K^{-1}\tilde{p}\|_2} = D_i(p)\,K^{-1}\tilde{p}, \tag{37}$$

which is exactly the foundation model's own unprojection of its predicted z-planar depth; with the pose unchanged, equation 36 returns the foundation model's point cloud. If $\hat{K}_i \neq K$, range and pose are still the foundation model's and only the direction differs: the point lies on the known ray $\mathrm{normalize}(K^{-1}\tilde{p})$ rather than on the predicted ray $\mathrm{normalize}(\hat{K}_i^{-1}\tilde{p})$. This is the only difference between the two clouds on all-pinhole input.

D Datasets and evaluation protocol

D.1 Experimental datasets and split protocol

The datasets and their roles are summarized in Table 12. The procedural camera-family sampler used for token pretraining supplies calibration and ray supervision without RGB scenes. It is separate from both the posed-window pool used for adapter training and LensZoo, used in the heterogeneous-camera comparison.

Training and testing allocation. For each dataset, we allocate 80% of the scenes for training and 20% for testing. Holo360D is evaluated on its test split. The single-view table uses one image per evaluated scene.

LensZoo evaluation cohort. The heterogeneous-camera comparison uses eight rooms from eight distinct houses excluded from the warm-start training split. PanoInfinigen is an independently prepared panoramic-depth collection and is not interchangeable with LensZoo. The rendering procedure, camera families, shared-pose construction and GT validity are described in Appendix E, and the point and pose metrics in Appendix D.2. Wid3R receives only a camera-class name for each image, with placeholder parameters and no calibration. Its six classes are the pinhole (`Pinhole`), Brown–Conrady with thin-prism terms (`OPENCV`), Kannala–Brandt with six radial coefficients and tangential and thin-prism terms (`Fisheye624`), unified camera model with radial–tangential distortion (`MEI`), EUCM (`EUCM`) and ERP (`Spherical`) models; each of these six camera families receives its own class. The five families without a Wid3R class take a fixed one: Brown–Conrady for division and smooth angular perturbation, EUCM for double sphere, and Kannala–Brandt for equisolid and FOV. On heterogeneous input, each frame receives its own camera's class.

D.2 Evaluation metrics

Radial magnitude and scale alignment. For our method, the predicted $\hat{K}$ converts depth to range by equation 12; the ray along which a point is then placed does not change this range. For each method, let ρ_{pred} denote its predicted radial magnitude. For each original evaluation group with valid-pixel set $\mathcal{V}$, we fit $s = \mathrm{median}_{p\in\mathcal{V}}[\rho(p)/\rho_{\mathrm{pred}}(p)]$ and set $\hat{\rho} = s\,\rho_{\mathrm{pred}}$, where ρ is GT range. Abs Rel (Eigen et al., 2014) averages $|\hat{\rho} - \rho|/\rho$, and δ_1 is the fraction satisfying $\max(\hat{\rho}/\rho, \rho/\hat{\rho}) < 1.25$. Single-view refers to inference from one image. This GT-based alignment assesses relative range accuracy without establishing metric-scale recovery.

Point placement from range and ray. PointErr tests the combined camera-frame prediction of range and ray through paired Euclidean error, without trajectory fitting. Acc. and Comp. (Wang et al., 2025) measure mean nearest-neighbor distances from prediction to GT and from GT to prediction. They use clouds fused at GT poses and the native sample of at most 50,000 points per

Table 12: Datasets, camera inputs and roles.

Dataset	Camera inputs	Role
Reported evaluations		
ScanNet++ (Yeshwanth et al., 2023)	Calibrated fisheye	Multiview point geometry and relative pose
KITTI-360 (Liao et al., 2023)	Calibrated fisheye	Multiview point geometry and relative pose
Holo360D (Ou et al., 2026)	Indoor/outdoor ERP	Panoramic multiview point geometry and relative pose
360Loc (Huang et al., 2024)	ERP panoramas	Panoramic multiview point geometry and relative pose
Replica360 (Straub et al., 2019; Rey-Area et al., 2022)	Single ERP image	Single-view panoramic depth
PanoInfinigen (Bozic et al., 2026)	Single ERP image	Single-view panoramic depth; distinct from LensZoo
ZuriPano (Bozic et al., 2026)	Single ERP image	Single-view panoramic depth
LensZoo	Twelve central-camera families	Point-cloud geometry and pose; projected-depth conditioning ablation; built from Infinigen scenes (Raistrick et al., 2023; 2024), Appendix E
Additional posed-window pool for adapter training		
V2 release of TartanAir (Wang et al., 2020)	Synthetic fisheye	Adapter training
Aria Gen 2 Pilot (Kong et al., 2025)	Egocentric RGB fisheye	Adapter training
FIORD (Gunes et al., 2025)	Fisheye	Adapter training
LaFiDa (Urban & Jutzi, 2017)	Multi-fisheye rig	Adapter training
TUM VI (Schubert et al., 2018)	Grayscale stereo fisheye	Adapter training
TUM monoVO (Engel et al., 2016)	Wide-angle fisheye, FOV model	Adapter training
PanoraMIS (Benseddik et al., 2020)	Catadioptric, dual fisheye	Adapter training
Princeton365 (Kayan et al., 2025)	Wide-angle, Brown–Conrady	Adapter training
MP3D360 (Chang et al., 2017; Rey-Area et al., 2022)	ERP panoramas	Adapter training
WideDepth (Indyk et al., 2026)	Fisheye, 120–195°	Adapter training
Helvipad (Zayene et al., 2025)	Top–bottom ERP stereo	Adapter training
SynWoodScape (Sekkat et al., 2022)	Synthetic surround-view fisheye	Adapter training
Obtained but not used in the reported experiments		
SUN360 (Xiao et al., 2012)	ERP panoramas	Not used in the reported experiments
WoodScape (Yogamani et al., 2019)	Surround-view fisheye	Not used in the reported experiments
Aria Digital Twin (Pan et al., 2023)	Egocentric RGB fisheye	Not used in the reported experiments

Role gives each dataset's main role. Adapter training uses 18 collections: the twelve pool datasets above and our 80% training partitions of ScanNet++, KITTI-360, Holo360D, 360Loc, Replica360 and ZuriPano. PanoInfinigen has no training scenes. The warm-start adapter run also drew 2% of its training mix from 37 rooms in eight other houses of the Infinigen corpus underlying LensZoo, excluded from the LensZoo evaluation cohort. TartanAir: V2 release (`https://tartanair.org`).

group. Geometry scores use the same median-scale alignment and GT coordinate units. Thus these measurements assess point placement while bypassing predicted-extrinsic errors; they are not a world-reconstruction score using the predicted poses.

Inter-view geometry from the camera head. We assess predicted poses separately from the ray-based point construction. AUC_{30} (area under the pose-accuracy curve up to 30°) (Wang et al., 2025)

Table 13: Scene-construction and viewpoint parameters: random ranges and fixed controls. Lengths are in meters; furnishing ratios and sun intensity (a model multiplier, not watts) are dimensionless. Acquisition height is drawn once per candidate floor-grid position.

Parameter	Configured draw or fixed value
Wall thickness	$\mathcal{U}(0.20, 0.30)\,\mathrm{m}$
Wall height	$\mathcal{U}(2.80, 3.20)\,\mathrm{m}$
Furniture floor-filling target	$\mathcal{U}(0.60, 0.90)$
Shelf-interior occupancy target	$\mathcal{U}(0.50, 1.00)$
Storage-top occupancy target	$\mathcal{U}(0.50, 1.00)$
Other support-surface occupancy target	$\mathcal{U}(0.20, 1.00)$
Sun elevation	$\mathrm{BG}(40^\circ, 25^\circ; [6^\circ, 70^\circ])$
Sun azimuth	$\mathcal{U}(0^\circ, 360^\circ)$
Sun intensity	$\mathcal{U}(0.80, 1.00)$
Sky strength	0.25 (fixed)
Acquisition camera height	$\mathcal{U}(1.40, 1.70)\,\mathrm{m}$ above floor

is the normalized discrete cumulative-histogram area through 30° for the larger relative-rotation or translation-direction angular error. RRA_{30} and RTA_{30} (relative rotation and translation accuracy) are fractions with the corresponding error strictly below 30°. They and δ_1 lie in $[0, 1]$. The reconstruction tables report these three pose metrics alongside PointErr, Acc. and Comp.; the single-view depth table reports Abs Rel and δ_1.

E Dataset construction details

This appendix describes the construction of LensZoo, introduced in §4. The pipeline separates procedural scene generation, room-wise view acquisition and camera-specific image formation. Reusing scene geometry and physical poses across projections is the central design choice; different camera models can still observe different surfaces and have different valid image domains. Figure 3 illustrates this shared-pose construction.

Procedural indoor environments. We use Infinigen's indoor composition pipeline to generate furnished multi-room homes. A scene seed initializes procedural stages, with room and furniture constraint graphs determining the layout. Staged placement introduces large, medium and small objects, followed by construction of doors, windows, walls, floors and ceilings. Furnishing density, optional objects, material assignments and lighting vary during scene generation. These are scene-level choices; the view-acquisition stage preserves the generated lighting. The resulting Blender scene retains its seed identifier. Room-floor objects provide room identities and bounds for acquisition, so observations from different rooms remain associated with their source house.

Scene randomization ranges. Table 13 lists selected high-level construction parameters. Here $\mathcal{U}(a, b)$ is a continuous uniform draw, and $\mathrm{BG}(\mu, \sigma; [a, b])$ draws from $\mathcal{N}(\mu, \sigma^2)$ until the value lies in $[a, b]$, with the final proposal clipped to the interval after twenty unsuccessful attempts. The furnishing ratios are constraint-solver targets, not measured occupied fractions of the completed room. Material categories depend on room type, and procedural assets have additional internal parameters. These are configured generation ranges, rather than empirical extrema recovered from every rendered scene. The lighting entries describe source-scene generation; acquisition introduces no per-view light randomization.

Candidate viewpoints. The active-view planner operates independently in each room, maintaining its own voxel and coverage state. Candidate floor positions lie on a $0.5\,\mathrm{m}$ grid inset $0.5\,\mathrm{m}$ from the walls, with up to 300 positions retained per room by sampling without replacement. Heights follow Table 13. The yaw support is $\{0^\circ, 45^\circ, \ldots, 315^\circ\}$ and the pitch support is $\{-15^\circ, 0^\circ, 15^\circ\}$; these are deterministic orientation grids, not continuous random angle draws. Fine scoring selects the best orientation for each evaluated position before stochastic position selection. Occupied-voxel rejection can remove positions inside scene geometry. Very small rooms use a centroid fallback

$1.5\,\mathrm{m}$ above the floor when the regular grid yields no suitable positions. This room-wise construction supplies viewpoint candidates before final camera-specific rendering; it does not optimize a separate trajectory for each camera family.

Coverage-aware active selection. The next-view score combines four terms: visible unobserved voxels, spatial diversity, angular diversity and the change in entropy of the occupied-surface view-count histogram. Visibility estimation uses a coarse occupied-voxel depth buffer to account for occlusion. The four terms are standardized across candidates and weighted by 1, 1.5, 1.5 and 1.5, respectively. Candidate selection uses a softmax distribution with inverse temperature 3 and a joint spatial and angular proximity test: a candidate is rejected when it is both within $0.3\,\mathrm{m}$ and within 30° of an acquired pose. Thus the thresholds do not enforce independent minimum translation and rotation changes. If a small room has no remaining candidate, the implementation can halve both thresholds once before terminating.

Coverage is measured against a sampled reference surface obtained from four cardinal-facing depth views at the room centroid, $1.5\,\mathrm{m}$ above the floor. The configured 90% threshold therefore concerns this reference, rather than the entire room mesh. Selection also has a view cap, coverage-plateau termination and a relative-score stopping rule. In the recorded full-corpus configuration, the relative-score rule becomes eligible at iteration twelve; coverage or candidate exhaustion can terminate acquisition earlier. Interpolation between selected poses is disabled. Each room stores the selected poses, reference intrinsics, view count, coverage and termination metadata. The camera-rendering stage consumes this cache directly.

Camera generation at shared poses. LensZoo combines three directly rendered camera families with nine analytic projections. Blender Cycles directly renders pinhole, equisolid fisheye and equirectangular images. The other families are Brown–Conrady, KB4, unified and enhanced unified cameras, division, double sphere, field-of-view, generalized KB and smooth angular perturbation. These are produced from a co-located 4096×2048 equirectangular RGB and range master for each selected pose. For each target pixel, the corresponding camera model provides a ray; rotating that ray into the master frame locates the sample in the spherical image. The implementation bilinearly samples RGB and range, with analytic target images defaulting to 512×512. Thus the twelve LensZoo families combine direct rendering and ray-map resampling, rather than requiring a separate Cycles camera implementation for every family. The exact cached pose is reused throughout, while each camera entry records its own projection parameters and raster dimensions. LensZoo's Brown–Conrady images are warped, and their ground-truth rays computed, with OpenCV's (Bradski, 2000) default five-iteration undistortion (`cv::undistortPoints`); for the fixed lens ($k_1 = -0.183$) this map agrees with the Brown–Conrady polynomial to within 0.5 px on 91.5% of pixels, and the 3.2% corner pixels beyond the polynomial's fold image directions up to 79° off-axis; excluding them changes Table 4's Brown–Conrady point errors by less than 0.01 and no ranking.

Camera parameter coverage. Scene generation and viewpoint acquisition include random draws, but the camera parameters for LensZoo form a fixed bank of 301 configurations across the twelve families. The same bank is reused across physical poses; camera parameters are not redrawn for each image. This is a count of camera settings, not a count of rendered images. It is distinct from the randomized camera sampling used for geometry-only token pretraining.

Table 14 specifies the bank's design intervals. Let $\bar{q} = f/\min(W,H)$ denote normalized focal length. Except for equirectangular images, the bank uses 512×512 rasters, $f_x = f_y = 512\bar{q}$ pixels and $(c_x, c_y) = (255.5, 255.5)$ pixels, without principal-point, aspect-ratio or skew jitter. The full-sphere equirectangular setting uses 1024×512 pixels. For an interval $[a,b]$ and nominal spacing h, the grid is

$$n = \max\left(1, \left\lfloor \frac{b-a}{h} + 10^{-9} \right\rfloor + 1\right), \qquad \mathcal{G}(a,b,h) = \mathrm{linspace}(a,b,n). \tag{38}$$

For $n \geq 2$, both endpoints are included, so the actual spacing can differ from h; for $n = 1$, only a is retained. The nominal focal spacing is $h_{\bar{q}} = 0.15$. For each primary distortion value, the lower focal endpoint is raised according to camera-domain validity before its focal grid is formed. Consequently, the table gives design support rather than a Cartesian product in which every listed focal value is rendered for every distortion value.

Table 14: LensZoo camera-parameter coverage: intervals of a fixed, deterministic bank, not uniform random draws. Coefficients are dimensionless unless angular or defined by the angular-warp equations below.

Family	$\bar{q}$ design interval	Primary parameter interval	Nominal spacing
Pinhole	$[0.70, 2.00]$	No distortion	–
Brown–Conrady	$[0.60, 1.60]$	$k_1 \in [-0.40, 0.25]$	0.10
KB4	$[0.25, 0.60]$	$k_1 \in [-0.05, 0.06]$	0.02
UCM	$[0.30, 0.80]$	$\xi \in [0, 2.10]$	0.30
EUCM	$[0.30, 0.80]$	$\alpha_{\mathrm{EU}} \in [0.50, 0.70]$	0.03
Division	$[0.50, 1.20]$	$\lambda \in [-0.40, 0]$	0.06
Double sphere	$[0.30, 0.80]$	$\xi \in [-0.30, -0.10]$	0.03
FOV/ATAN	$[0.40, 1.00]$	$\omega_{\mathrm{FOV}} \in [0.40, 2.10]\,\mathrm{rad}$	$0.25\,\mathrm{rad}$
Equisolid	$[0.20, 0.45]$	$\theta_{\max} \in [90, 140]^\circ$	15°
Equirectangular	–	$360^\circ \times 180^\circ$ full sphere	–
Generalized KB	$[0.25, 0.60]$	$c_{\mathrm{KB}} \in [0, 0.03]$	0.00375
Smooth angular	$[0.60, 1.60]$	$A \in [0, 0.05]$	0.01

Coefficient dependencies. Secondary coefficients follow fixed paths rather than independent draws. For each family's primary coordinate $d \in [d_{\min}, d_{\max}]$, define $t = (d - d_{\min})/(d_{\max} - d_{\min})$. Brown–Conrady uses

$$k_2 = \begin{cases} 0, & k_1 \le 0, \\ -0.05 + 0.26t, & k_1 > 0, \end{cases} \qquad k_3 = p_1 = p_2 = 0. \tag{39}$$

KB4 sets $k_2 = k_3 = k_4 = 0$. EUCM couples $\beta_{\mathrm{EU}} = 1 + 0.2t$ to α_{EU}, while double sphere couples $\alpha_{\mathrm{DS}} = 0.62 - 0.08t$ to ξ. Division follows the convention of Table 11. The FOV parameter ω_{FOV} is a model coefficient in radians, not the image's field of view. Equisolid sets the native renderer's full-angle parameter to $2\theta_{\max}$. Generalized KB uses nine equally spaced values of c_{KB} and the coefficient vector $(c_{\mathrm{KB}}, -c_{\mathrm{KB}}/2, 0, 0)$.

The smooth-angular family starts from an undistorted perspective ray map and applies a fixed, tapered warp in the chart of equation 18, whose coordinates are longitude θ and twice latitude φ, both in radians. It uses

$$\delta\theta = A\tau \sin(2\theta), \qquad \delta\varphi = \frac{A}{2}\tau \sin(2\varphi), \qquad \tau = (\theta^2 + \varphi^2) \exp\left[-\frac{\theta^2 + \varphi^2}{2}\right]. \tag{40}$$

The frequency is fixed at 2 and the phase at zero. Here A is a coefficient of this chart warp, not a maximum angular displacement. Ray validity additionally requires a positive warp-Jacobian determinant.

Validity-conditioned focal grids. The focal lower bound is selected using a 128×128 ray-field probe requiring at least 58% valid pixels, with additional radial-monotonicity and 150° full-field constraints for Brown–Conrady and smooth-angular models. If $\bar{q}_{\mathrm{usable}}$ denotes the resulting lower limit, the focal sweep begins at $\min(\max(\bar{q}_{\min}, \bar{q}_{\mathrm{usable}}), \bar{q}_{\max} - 0.001)$ with nominal upper endpoint $\bar{q}_{\max}$, using equation 38. Final camera validation requires at least 50% valid rays at the output resolution and retains the applicable monotonicity checks. These are camera-domain checks, separate from scene-dependent range validity. Fields of view are measured from the constructed ray fields, rather than independently sampled, and do not imply the same angular coverage after rendering masks.

Geometric targets and coordinate conventions. LensZoo stores RGB images and Euclidean radial range in meters, including the conversion needed for native outputs that initially encode axial depth. For a valid pixel p with unit ray $\Phi(p)$ and range $\rho(p)$, the camera-frame point is $X_c(p) = \rho(p)\Phi(p)$. A stored camera-to-world pose $(R^{\mathrm{c2w}}, t^{\mathrm{c2w}})$ then gives $X_w(p) = R^{\mathrm{c2w}} X_c(p) + t^{\mathrm{c2w}}$. Camera metadata specifies the family, model parameters, image dimensions, RGB/range paths, coordinate convention and the 4×4 pose matrix. Blender and OpenCV pose representations are recorded explicitly. These records allow geometric targets to be interpreted using the camera that formed each image.

Validity and curation. Analytic resampling rejects rays outside the camera domain or available master content, applies a forward-ray cutoff and uses a depth-discontinuity guard near occlusion boundaries. Invalid range is stored as zero. RGB validity is handled separately, so conservative depth rejection need not create a black boundary in the color image. The forward-ray filter restricts the retained domain of analytic outputs; their valid pixels should not be assumed to cover every direction admitted by the abstract camera model. Resampling can also introduce range artifacts, so these checks do not make every retained pixel exact ground truth. The additional range-validity rule used for the distortion-token ablation is specified in Appendix A.4.

Manifest-driven preparation excludes scenes marked unsuccessful, removes duplicate canonical perspective entries and checks agreement between cached poses and declared view counts. The base rendering stage additionally checks actual image and transform files, because a successful scene-level status alone does not establish complete output. Scene, room, pose and camera identifiers remain distinct throughout preparation. The LensZoo evaluation cohort (Appendix D.1) and the camera models evaluated in Table 4 specify the scope of the reported comparisons; they do not constitute an exhaustive sweep of the supported camera parameterizations.