Fisheye3R: Adapting Unified 3D Feed-Forward Foundation Models to Fisheye Lenses

Ruxiao Duan[1,2], Erin Hong[2], Dongxu Zhao[2],
Eric Turner[2], Alex Wong[1], and Yunwen Zhou[2]

[1] Yale University
{ruxiao.duan, alex.wong}@yale.edu
[2] Google
{ruxiao, honge, dongxuz, elturner, verse}@google.com

Abstract. Feed-forward foundation models for multi-view 3-dimensional (3D) reconstruction have been trained on large-scale datasets of perspective images; when tested on wide field-of-view images, e.g., from a fisheye camera, their performance degrades. This degradation arises from changes in spatial arrangements of pixels induced by the non-linear projection model that maps 3D points onto the 2D image plane. While one may surmise that training on fisheye images would resolve this problem, there are far fewer fisheye images with ground truth than perspective images, which limits generalization. To enable inference on imagery exhibiting high radial distortion, we propose *Fisheye3R*, a novel adaptation framework that extends these multi-view 3D reconstruction foundation models to natively accommodate fisheye inputs without performance regression on perspective images. To address the scarcity of fisheye images and ground truth, we introduce flexible learning schemes that support self-supervised adaptation using only unlabeled perspective images and supervised adaptation without any fisheye training data. Extensive experiments across three foundation models, including VGGT, π^3, and MapAnything, demonstrate that our approach consistently improves camera pose, depth, point map, and field-of-view estimation on fisheye images. Code is available at https://github.com/android-xr/fisheye3r.

Keywords: 3D Reconstruction · Fisheye · Adaptation

1 Introduction

Generalist feed-forward foundation models [18, 41, 46] have recently emerged as a dominant paradigm in multi-view 3D reconstruction (3R) from uncalibrated images, unifying traditionally disparate geometric tasks (camera pose, depth, point map, and field-of-view estimation) into a single, cohesive framework. As they are trained on large-scale datasets of abundant perspective (rectilinear) images, these foundation models generalize well across 3D scenes. Yet, when evaluated on images captured by fisheye cameras, which are commonly used in spatial applications from robotics to Augmented/Virtual Reality (AR/VR) [3,22, 34, 52] due to their large scene coverage for situational awareness, these models

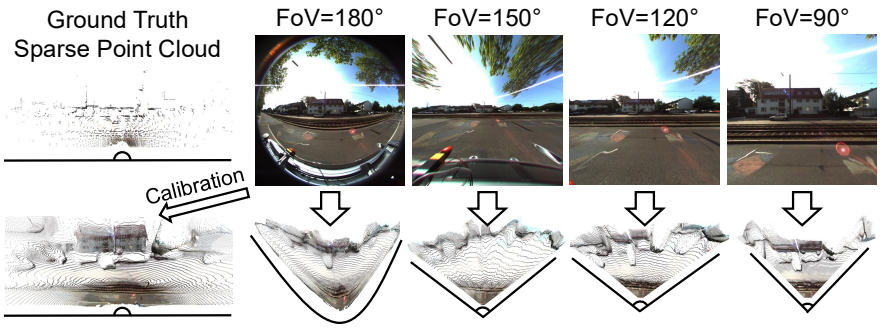

Fig. 1: Foundation models for 3D reconstruction generally fail on fisheye images. **Top**: Ground truth sparse LiDAR points of a fisheye image, with undistorted perspective images of varying FoVs. **Bottom**: Reconstructed point clouds from the corresponding images by [18]; the leftmost one is calibrated by our method. Naively undistorting the fisheye image for 3D reconstruction either sacrifices scene coverage or introduces extreme resampling artifacts and peripheral stretching that deviate from the model's training distribution, as evident from the mismatch of predicted and actual FoVs.

produce degraded outcomes due to curvilinear distortions inherent to fisheye optics (Fig. 1). As many spatial applications are deployed on platforms with wide field-of-view (FoV) cameras or mixed-camera systems, e.g., comprising both perspective and fisheye cameras, these foundation models have been unsuitable for direct integration into these commercial systems.

A conventional approach to handling wide FoV images is to pre-process them through a rectification step, which transforms the fisheye images to a rectilinear pinhole viewport. Yet, this comes with a significant loss of peripheral coverage (due to cropping), which was the precise advantage that motivated the use of wide FoV cameras in the first place. On the other hand, retaining the original fisheye FoV in perspective images results in images with extreme resampling artifacts. Furthermore, the undistorted image would similarly degrade the performance of downstream models due to processing of ultra-wide-angle projections that lie far outside the natural distribution of their training data, which again leads to performance degradation as illustrated in Fig. 1. Additionally, the rectification process incurs latency and relies on accurate calibration, where errors accumulate and propagate through the vision system and further degrade estimates.

An alternative approach is to train a model from scratch or finetune an existing model specifically for fisheye images; however, unlike the abundance of publicly available perspective image datasets, there is a comparatively small corpus of fisheye images, and even fewer fisheye images with ground truth to support 3D tasks. This fisheye data scarcity makes it challenging to train a foundation model that generalizes well across diverse 3D scenes. While finetuning may be an option, this risks model parameters drifting away from a desirable minimum, resulting in a loss of fidelity or generalizability. In both cases, the re-

sulting model becomes camera-specific, requiring separate models to be deployed for mixed-camera systems, which in turn increases operational overhead.

Finally, one may also train a foundation model from scratch on a joint corpus of perspective and fisheye datasets. Not only is this computationally taxing for multi-view foundation models, but it also presents a high imbalance of data between the two camera types, resulting in trade-offs between fidelity and generalizability [30]. We consider whether there exists a middle ground that enables a model to infer with similar fidelity on both perspective and fisheye images to avoid the operational overhead of deploying multiple models, but also without the computational overhead of large-scale training.

Given that existing foundation models already exhibit high generalizability across 3D scenes and that errors stem primarily from curvilinear distortions, we aim to recalibrate the fisheye input, not in the image space as done conventionally, but in the model's latent space, such that its embeddings become conducive to an existing foundation model pre-trained on perspective images. To this end, we propose *Fisheye3R*, a lightweight and efficient adaptation framework that can potentially extend any pre-trained foundation model to fisheye imagery with similar fidelity as perspective images and without loss of generalizability.

Taking advantage of the transformer architectures [7] in foundation 3R models, we adopt a token-based adaptation strategy, where trainable calibration tokens are inserted into transformer blocks to extend the performance of a given pre-trained model to fisheye images. The influence of these tokens is controlled by a masked attention scheme, where their effect is eliminated for perspective images and activated on fisheye images, enabling the model to remain backwards compatible with perspective imagery. Considering the limited availability of ground truth for fisheye images on 3D tasks, we propose three distinct learning schemes to accommodate various degrees of data accessibility: (i) self-supervised learning with unlabeled perspective RGB images only; (ii) supervised learning with annotated perspective data; and (iii) supervised learning with perspective and fisheye data. Through extensive evaluation, we demonstrate that Fisheye3R consistently improves 3D reconstruction performance on fisheye images across all three schemes, providing a versatile solution regardless of data constraints. Our work offers the following advantages:

- Generalizability. We apply our framework generically to adapt three unified multi-view feed-forward foundation 3D reconstruction models [18, 41, 46].
- Efficiency. We show that the models can be adapted through the insertion of some lightweight learnable tokens, requiring minimal training time and negligible computational overhead compared to full-parameter finetuning.
- Backwards compatibility. We demonstrate that the model can fully preserve the original performance on perspective data while gaining the unique capability to handle heterogeneous camera models within a single sequence.
- Data flexibility. Our method supports three different learning schemes tailored to varying data regimes, with which we observe consistent improvements on fisheye data, even with unlabeled perspective images for training.

2 Related Work

Traditional 3D Reconstruction. To recover 3D geometry from single or multiple views, traditional methodologies typically employ Structure-from-Motion (SfM) [1, 9, 13, 28, 32, 35, 36, 47] to estimate sparse point clouds and camera parameters, followed by Multi-View Stereo (MVS) [6, 14, 33, 40, 50, 54] to reconstruct dense scene geometry. These multi-stage pipelines rely on a sequence of independent tasks, including keypoint detection, feature extraction, matching, triangulation, and bundle adjustment, where optimization noise and errors can accumulate at each step.

Feed-Forward 3D Reconstruction. Recently, transformer-based models have emerged to estimate 3D geometry from multiple images. DUSt3R [44] regressed point maps from image pairs, while MASt3R [19] further improved upon this by adding local feature regression to enhance reconstruction accuracy. Subsequent variants further extended their capabilities to dynamic scenes [53], streaming data [43], and parallelized reconstruction [49].

Unified 3D Feed-Forward Models. Unified 3D foundation models serve as general-purpose geometric backbones capable of solving multiple downstream tasks simultaneously. VGGT [41] unifies 3D vision by treating it as a sequence-to-sequence translation problem, regressing camera parameters, depth maps, point maps, and point tracks from uncalibrated images in a single forward pass. π^3 [46] addresses VGGT's limitation regarding reference frame dependency by introducing a permutation-equivariant architecture, enabling robust prediction of local point maps and camera poses regardless of the input order. MapAnything [18] further expands the versatility by leveraging a multi-modal encoder to infer metric geometry from images and optional priors.

Fisheye-Aware 3D Perception. Utilizing uncalibrated fisheye inputs for 3D reconstruction remains a challenging problem due to arbitrary distortions. Traditional pipelines rely on integrating explicit calibration [12, 17, 26, 31] into geometric solvers [3, 25, 55], while recent learning-based attempts often circumvent native distortion by learning from canonical projections [5, 15, 20, 42, 56]. Other data-driven approaches focus on isolated subtasks: [37, 39] target intrinsic calibration, while [23] adapts stereo depth via warping. More recently, [30] proposes a universal model for local camera 3D estimation using spherical basis functions, while [10] employs calibration tokens and [11] proposes distortion extenders to adapt monocular depth estimation models to fisheye cameras.

3 Method

3.1 Problem Setup and Representations

Multi-view feed-forward 3D reconstruction models [18, 41, 46] estimate camera pose and dense geometric attributes (e.g., depth, point maps, ray directions) from a set of uncalibrated RGB images capturing the same 3D scene. This process generally consists of three stages.

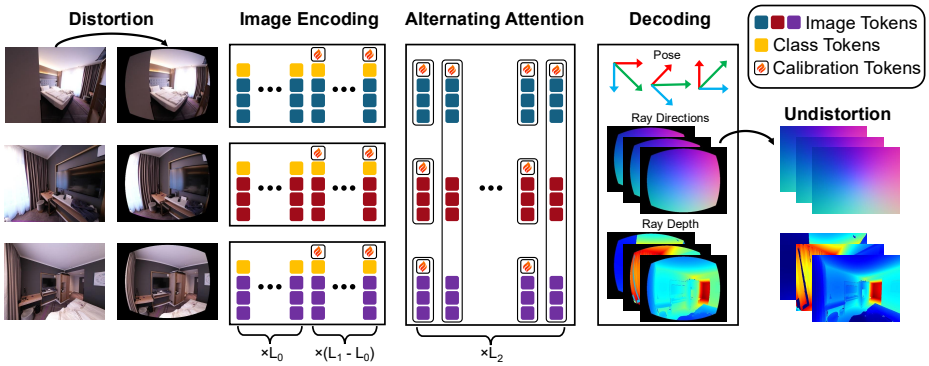

Fig. 2: Fisheye adaptation framework with perspective training data. Our method leverages existing perspective datasets by synthesizing curvilinear distortion in the input sequences. The distorted images are processed by the frozen transformer backbone, where learnable calibration tokens are injected into the encoder layers. The decoder predicts dense geometric attributes, which are undistorted for loss computation. Supervision is derived either from ground truth or through a self-supervised distillation of the model's own predictions on the original perspective sequence.

Image encoding. Consider a sequence of S images $(I_s)_{s=1}^{S}$ from a scene where $I_s \in \mathbb{R}^{H\times W\times 3}$. Each frame is independently passed to some pre-trained image encoder, typically DINOv2 [27], for feature extraction. Specifically, for each $s \in [S] = \{1, \ldots, S\}$, the image I_s is divided into $N = HW/P^2$ patches of size $P \times P$ pixels, which are then projected to patch embeddings $\{\boldsymbol{x}_{s,n}^{(0)}\}_{n=1}^{N}$ where $\boldsymbol{x}_{s,n}^{(0)} \in \mathbb{R}^d$ and concatenated with a class token $\boldsymbol{x}_{s,0}^{(0)} = \boldsymbol{x}_{\text{class}} \in \mathbb{R}^d$ as

$$\boldsymbol{x}_s^{(0)} = \left[\boldsymbol{x}_{s,0}^{(0)}, \boldsymbol{x}_{s,1}^{(0)}, \ldots, \boldsymbol{x}_{s,N}^{(0)}\right] \in \mathbb{R}^{(1+N)\times d}, \tag{1}$$

where $[\cdot]$ denotes concatenation and d is the embedding dimension. For an encoder with L_1 layers, the encoder module $f_{\text{E}}^{(\ell)}$ at a layer $\ell \in [L_1]$ derives

$$\boldsymbol{x}_s^{(\ell)} = f_{\text{E}}^{(\ell)}\left(\boldsymbol{x}_s^{(\ell-1)}\right) \in \mathbb{R}^{(1+N)\times d} \tag{2}$$

through multi-head self-attention and MLP blocks [7]. After the last layer, the N patch tokens are passed to the next stage as $\boldsymbol{z}_s^{(0)} = [\boldsymbol{x}_{s,n}^{(L_1)}]_{n=1}^{N} \in \mathbb{R}^{N\times d}$.

Alternating attention. With a total of L_2 alternating attention (AA) blocks, each block $\ell \in [L_2]$ sequentially applies frame-wise and global self-attention as

$$\boldsymbol{z}_s'^{(\ell)} = f_{\text{F}}^{(\ell)}\left(\boldsymbol{z}_s^{(\ell-1)}\right) \in \mathbb{R}^{N\times d},\ \forall s \in [S], \tag{3}$$

$$\left[\boldsymbol{z}_s^{(\ell)}\right]_{s=1}^{S} = f_{\text{G}}^{(\ell)}\left(\left[\boldsymbol{z}_s'^{(\ell)}\right]_{s=1}^{S}\right) \in \mathbb{R}^{SN\times d}, \tag{4}$$

where the tokens of each frame attend to each other in frame-wise attention layer $f_{\text{F}}^{(\ell)}$ and the tokens across all frames attend to each other in global attention layer

$f_{\mathrm{G}}^{(\ell)}$. The order of frame and global attention can vary depending on the model's architectural design.

Scene representation prediction. The outputs of the AA module $\{\boldsymbol{z}_s^{(L_2)}\}_{s=1}^{S}$ are passed to decoder f_{D} as

$$E_s, D_s = f_{\mathrm{D}}\left(\boldsymbol{z}_s^{(L_2)}\right), \tag{5}$$

where $E_s = [R_s|\boldsymbol{t}_s]$ denotes camera pose for frame s with rotation $R_s \in \mathbb{R}^{3\times3}$ and translation $\boldsymbol{t}_s \in \mathbb{R}^3$, and $D_s \in \mathbb{R}^{H\times W\times C}$ is the combination of dense predictions whose types vary across different architectures: VGGT [41] predicts depth map in the local camera coordinates and point map in the global world coordinates; π^3 [46] predicts a local point map and infers the global point map from the local one and the predicted camera pose; MapAnything [18] predicts local ray directions and ray depths to derive local camera coordinates, from which global point coordinates are inferred with the predicted pose.

3.2 Model Adaptation with Calibration Tokens

While foundation models trained on massive rectilinear datasets excel at processing sequences of perspective images, they fail to generalize to fisheye imagery. When presented with fisheye inputs, the frozen backbone generates features that fall outside the learned data manifold of perspective images, resulting in distorted geometric reconstructions and erroneous camera pose estimates.

To bridge this gap, we align fisheye feature representations with those of perspective images by introducing K learnable calibration tokens into each transformer layer. As illustrated in Fig. 2, these tokens are inserted into all the encoder layers except the initial L_0 layers to modulate the high-level features. The learnable tokens "recalibrate" the values of fisheye features such that they resemble those of perspective images to enable a pre-trained model to infer attributes of the 3D scene with high fidelity. Specifically, we reformulate the transformer encoder, frame-wise, and global attention layers from Eqs. (2) to (4) as

$$\left[\boldsymbol{x}_s^{(\ell)}, \left[\boldsymbol{\phi}_{\mathrm{E},k}^{\prime(\ell)}\right]_{k=1}^{K}\right] = f_{\mathrm{E}}^{(\ell)}\left(\left[\boldsymbol{x}_s^{(\ell-1)}, \left[\boldsymbol{\phi}_{\mathrm{E},k}^{(\ell)}\right]_{k=1}^{K}\right]\right),\ \forall \ell \in (L_0, L_1], \tag{6}$$

$$\left[\boldsymbol{z}_s^{\prime(\ell)}, \left[\boldsymbol{\phi}_{\mathrm{F},k}^{\prime(\ell)}\right]_{k=1}^{K}\right] = f_{\mathrm{F}}^{(\ell)}\left(\left[\boldsymbol{z}_s^{(\ell-1)}, \left[\boldsymbol{\phi}_{\mathrm{F},k}^{(\ell)}\right]_{k=1}^{K}\right]\right),\ \forall \ell \in [L_2], \tag{7}$$

$$\left[\left[\boldsymbol{z}_s^{(\ell)}\right]_{s=1}^{S}, \left[\boldsymbol{\phi}_{\mathrm{G},k}^{\prime(\ell)}\right]_{k=1}^{K}\right] = f_{\mathrm{G}}^{(\ell)}\left(\left[\left[\boldsymbol{z}_s^{\prime(\ell)}\right]_{s=1}^{S}, \left[\boldsymbol{\phi}_{\mathrm{G},k}^{(\ell)}\right]_{k=1}^{K}\right]\right),\ \forall \ell \in [L_2], \tag{8}$$

where $\boldsymbol{\phi}_{\mathrm{M},k}^{(\ell)} \in \mathbb{R}^d$ represents the k-th calibration token at the ℓ-th layer of module $\mathrm{M} \in \{\mathrm{E}, \mathrm{F}, \mathrm{G}\}$. In other words, at each encoder and frame-wise attention layer, K calibration tokens adapt the N image tokens separately. At each global attention layer, K calibration tokens adapt the SN image tokens of the entire sequence jointly. Each layer has its own set of calibration tokens, which are dropped immediately at each layer to localize the latent calibration effect

to each layer. During training, only calibration tokens are learned, while the original backbone is entirely frozen for efficiency and to preserve the model's knowledge on perspective images.

3.3 Learning

We denote the entire feed-forward model by f. With a perspective image sequence $I^{\mathrm{p}} = (I^{\mathrm{p}}_s)_{s=1}^{S}$, we predict

$$\hat{E}^{\mathrm{p}}, \hat{D}^{\mathrm{p}} = f\left(I^{\mathrm{p}}\right), \tag{9}$$

where $\hat{E}^{\mathrm{p}}$ and $\hat{D}^{\mathrm{p}}$ denote the predicted camera poses and dense geometry outputs for the perspective sequence. To adapt the model to fisheye data, we can use a sequence of fisheye images $I^{\mathrm{f}} = (I^{\mathrm{f}}_s)_{s=1}^{S}$ for training, i.e.,

$$\hat{E}^{\mathrm{f}}, \hat{D}^{\mathrm{f}} = f\left(I^{\mathrm{f}}, \boldsymbol{\phi}\right), \tag{10}$$

where $\boldsymbol{\phi}$ is the set of all calibration tokens and $\hat{E}^{\mathrm{f}}, \hat{D}^{\mathrm{f}}$ are outputs corresponding to the fisheye sequence.

The ideal scenario assumes abundant fisheye data is available for training; given this is not the case in practice, we opt for an alternative of generating fisheye images from perspective ones. Particularly, we adopt Kannala-Brandt distortion model [17] for fisheye image synthesis from perspective images: $I^{\mathrm{f}}_s = T(I^{\mathrm{p}}_s)$ for each frame s where T denotes the distortion transformation from perspective to fisheye image. The inverse distortion operation can be represented by T^{-1}, which is used to obtain undistorted geometric predictions from fisheye predictions as $\hat{D}^{\mathrm{p}}_s = T^{-1}(\hat{D}^{\mathrm{f}}_s)$. The camera extrinsic parameters are unrelated to the camera type, thus they do not need to be transformed: $\hat{E}^{\mathrm{p}}_s = \hat{E}^{\mathrm{f}}_s$.

Finally, the calibration tokens can be learned by supervising the outputs. The supervision type depends on the data availability. We propose three supervision schemes according to the types of data available.

Self-supervised learning with perspective images (SSL). If only perspective RGB images I^{p} are available with no ground truth, we generate pseudo-labels using the original model's predictions for self-supervision. Inspired by AugUndo [48], the loss function can therefore be expressed as

$$\mathcal{L}^{\mathrm{p}}_{\mathrm{SSL}} = \mathcal{L}\left(f(I^{\mathrm{p}}); T^{-1} \circ f(T(I^{\mathrm{p}}), \boldsymbol{\phi})\right). \tag{11}$$

Supervised learning with perspective images (SL). If both perspective images I^{p} and their ground truth $(E^{\mathrm{p}}, D^{\mathrm{p}})$ are available, they can be directly used for supervision as

$$\mathcal{L}^{\mathrm{p}}_{\mathrm{SL}} = \mathcal{L}\left((E^{\mathrm{p}}, D^{\mathrm{p}}); T^{-1} \circ f(T(I^{\mathrm{p}}), \boldsymbol{\phi})\right). \tag{12}$$

Supervised learning with fisheye images (SL+). If fisheye images I^{f} with ground truth $(E^{\mathrm{f}}, D^{\mathrm{f}})$ are available, we can train the calibration tokens via

$$\mathcal{L}^{\mathrm{f}}_{\mathrm{SL}} = \mathcal{L}\left((E^{\mathrm{f}}, D^{\mathrm{f}}); f(I^{\mathrm{f}}, \boldsymbol{\phi})\right). \tag{13}$$

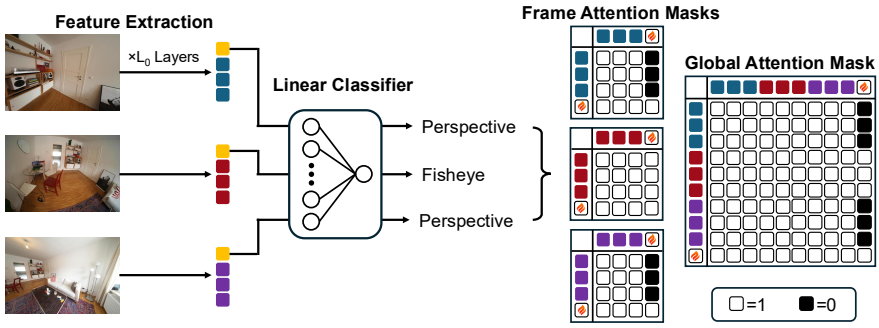

Fig. 3: Masked attention for mixture sequences. To handle a sequence of mixed camera types, we introduce a masked attention mechanism to block the influence of calibration tokens on perspective image tokens. After L_0 layers of feature extraction in image encoder, the class tokens are passed to a linear classifier for camera type prediction. Then, frame-wise and global attention masks are created accordingly and used in all the remaining encoder layers, including the $L_1 - L_0$ ones in the image encoder, and the L_2 frame and L_2 global attention layers in alternating attention.

The loss function $\mathcal{L}$ is the loss used by the original 3D reconstruction method. Undistortion T^{-1} is applied to the dense geometry outputs but not the pose. Supervision is consistently applied at the spatial resolution of the ground truth observation, preventing artifacts that may arise from resampling.

3.4 Controlled Adaptation with Masked Attention

We further extend the model to handle unknown and potentially mixed types of cameras in a sequence during inference. Though calibration tokens can learn to align fisheye features to perspective ones effectively, we observe that they may not be backwards compatible with perspective images once trained, i.e., given perspective images, the calibration tokens are likely to modulate their features, leading to performance degradation on perspective data. This can be explained by the increased task complexity for calibration tokens to not only calibrate fisheye features but also preserve perspective ones simultaneously. Therefore, we propose to control the effect of calibration tokens depending on the camera type of images observed by using a masked attention scheme (Fig. 3). The effect of calibration tokens is eliminated by a binary mask for perspective images, but is activated for fisheye images. This mechanism ensures backwards compatibility with perspective images and enables inference on mixed-camera sequences.

Camera type classification. Due to the significant difference between perspective and fisheye images in terms of distortion characteristics, classifying the two types of images from their feature vectors is feasible. As we have the class tokens readily available from the image encoder as a compact representation of the image feature, we employ them to determine the camera type of an image. Formally, for a frame I_s, we extract its class token $\boldsymbol{x}_{s,0}^{(L_0)}$ from the L_0-th layer and pass it to an image classifier $\psi : \mathbb{R}^d \to (0, 1)$, which can be as simple as a linear

Dataset	Method	Pose (Angular)			Pose (Distance)			Depth Map			Point Map			FoV Map		
		RRA↑	RTA↑	AUC↑	ATE↓	RPEt↓	RPEr↓	Rel↓	RMSE↓	δ_1↑	Acc↓	Comp↓	CD↓	hErr↓	vErr↓	AUC↑
ScanNet++ [52]	VGGT [41]	0.956	0.759	0.381	0.200	0.420	13.877	0.287	0.440	0.618	0.122	0.102	0.112	12.962	10.253	0.305
	w/ **SSL**	**0.999**	0.813	0.533	0.148	0.284	6.097	0.234	0.354	0.744	0.077	0.052	0.065	7.092	5.708	0.605
	w/ **SL**	**0.999**	0.910	0.656	0.097	0.190	4.855	0.230	0.343	0.742	0.074	0.046	0.060	4.439	3.614	0.722
	w/ **SL+**	**0.999**	0.916	0.677	0.088	0.168	4.363	0.222	0.327	0.759	0.073	0.046	0.060	4.602	3.606	0.718
	π^3 [46]	0.989	0.814	0.463	0.164	0.347	10.651	0.282	0.421	0.661	0.095	0.073	0.084	8.288	6.332	0.551
	w/ **SSL**	**0.999**	0.844	0.571	0.114	0.220	6.146	0.239	0.357	0.741	0.070	0.050	0.060	3.926	3.090	0.773
	w/ **SL**	**0.999**	0.872	0.617	0.101	0.194	5.424	0.239	0.350	0.744	0.068	0.048	0.058	3.036	2.561	0.810
	w/ **SL+**	**0.999**	0.926	0.719	0.072	0.139	**3.710**	0.212	0.320	0.771	0.064	**0.044**	0.054	**2.941**	3.018	0.873
	MapAnything [18]	0.993	0.848	0.519	0.163	0.310	8.851	0.274	0.394	0.672	0.091	0.076	0.083	6.631	4.821	0.647
	w/ **SSL**	**0.999**	**0.972**	0.766	0.070	0.143	4.569	0.190	0.283	0.785	**0.057**	0.050	0.054	3.146	2.575	0.875
	w/ **SL**	**0.999**	0.959	0.730	0.080	0.164	5.032	0.195	0.289	0.786	0.059	0.050	0.054	4.011	3.049	0.853
	w/ **SL+**	**0.999**	0.970	**0.774**	**0.066**	**0.134**	4.370	**0.171**	**0.263**	**0.810**	0.058	0.045	**0.051**	3.264	**2.074**	**0.927**
ADT [29]	VGGT [41]	0.964	0.777	0.458	0.134	0.260	10.055	0.189	0.739	0.732	0.237	0.179	0.208	7.382	7.566	0.475
	w/ **SSL**	0.997	0.882	0.662	0.079	0.150	2.761	0.110	0.502	0.885	0.140	0.098	0.119	1.992	1.992	0.983
	w/ **SL**	0.995	0.931	0.746	0.048	0.095	2.242	0.082	0.354	0.930	0.080	0.053	0.066	1.418	1.344	**1.000**
	w/ **SL+**	0.998	0.948	0.778	0.038	0.073	**1.820**	0.076	0.334	0.934	0.072	0.045	0.059	1.665	1.598	**1.000**
	π^3 [46]	0.984	0.893	0.601	0.078	0.157	7.148	0.111	0.410	0.909	0.126	0.106	0.116	4.044	3.961	0.927
	w/ **SSL**	**1.000**	0.953	0.776	0.035	0.069	2.241	0.075	0.332	0.937	0.075	0.054	0.064	1.217	**0.989**	**1.000**
	w/ **SL**	0.996	0.945	0.768	0.041	0.079	3.325	**0.073**	**0.323**	**0.940**	**0.068**	0.047	0.058	**1.161**	1.068	**1.000**
	w/ **SL+**	0.996	**0.960**	**0.819**	**0.034**	**0.066**	2.799	0.087	0.362	0.915	0.074	**0.039**	**0.056**	1.211	1.534	0.997
	MapAnything [18]	0.959	0.793	0.448	0.125	0.264	9.601	0.145	0.411	0.860	0.164	0.130	0.147	8.313	8.141	0.399
	w/ **SSL**	0.996	0.908	0.704	0.060	0.126	3.525	0.090	0.333	0.921	0.079	0.056	0.068	2.032	2.015	**1.000**
	w/ **SL**	0.997	0.903	0.691	0.063	0.134	3.421	0.091	0.337	0.920	0.081	0.054	0.068	1.987	2.001	**1.000**
	w/ **SL+**	**1.000**	0.934	0.753	0.045	0.080	2.111	0.087	0.330	0.925	0.072	0.043	0.058	1.338	1.101	**1.000**
KITTI360 [22]	VGGT [41]	0.829	0.968	0.440	0.768	2.624	20.242	0.270	6.736	0.565	1.029	1.580	1.304	19.430	7.273	0.212
	w/ **SSL**	0.945	0.989	0.592	0.537	1.814	12.882	0.200	5.678	0.724	0.956	1.249	1.102	15.737	5.880	0.313
	w/ **SL**	0.982	0.991	0.649	0.443	1.503	10.290	0.204	4.456	0.694	0.893	1.148	1.021	11.953	4.247	0.435
	w/ **SL+**	**1.000**	**0.993**	0.904	0.117	0.356	2.409	0.111	3.955	0.890	0.644	0.808	0.726	4.790	2.252	0.757
	π^3 [46]	0.955	0.961	0.548	0.428	1.840	11.717	0.153	4.159	0.806	0.772	1.081	0.927	11.731	4.316	0.444
	w/ **SSL**	0.981	0.986	0.649	0.365	1.457	9.938	0.101	3.923	0.897	0.758	0.895	0.827	8.864	3.804	0.531
	w/ **SL**	0.965	**0.993**	0.634	0.404	1.531	11.555	0.105	4.091	0.893	0.749	0.909	0.829	7.816	3.203	0.566
	w/ **SL+**	**1.000**	**0.993**	**0.942**	**0.092**	**0.239**	**1.484**	0.097	3.645	**0.922**	**0.399**	**0.475**	**0.437**	6.316	4.001	0.751
	MapAnything [18]	0.818	0.947	0.428	1.215	3.023	20.568	0.258	5.775	0.607	1.097	1.505	1.301	18.034	6.642	0.259
	w/ **SSL**	0.906	0.989	0.540	0.707	2.152	15.528	0.156	4.294	0.809	0.933	1.098	1.015	15.739	6.025	0.291
	w/ **SL**	0.906	0.982	0.538	0.682	2.141	15.342	0.153	4.458	0.814	0.938	1.094	1.016	15.750	5.998	0.295
	w/ **SL+**	**1.000**	0.992	0.917	0.152	0.350	1.565	**0.091**	**3.282**	**0.922**	0.549	0.601	0.575	**2.963**	**1.938**	**0.805**

Table 1: Quantitative results of fisheye adaptation on three models, three datasets, and three learning schemes. Compared to the baselines, the cell colors denote an improvement of 10%-30%, 30%-50%, and 50%+. The best numbers are marked in bold.

classifier with sigmoid activation. Consequently, a binary predicted camera type can be obtained for each frame s as

$$M_s = \mathbb{I}\left(\psi\left(\boldsymbol{x}_{s,0}^{(L_0)}\right) > 0.5\right) \in \{0, 1\}, \tag{14}$$

where $\mathbb{I}$ denotes the indicator function and M_s is 1 for fisheye and 0 for perspective.

Attention masking. With such binary camera type predictions, the frame-wise attention mask $M_{\mathrm{F},s}$ for each frame s and the global attention mask M_{G} for the sequence can be respectively constructed as

$$M_{\mathrm{F},s} = \begin{pmatrix} \mathbf{1}_{N\times N} & M_s \cdot \mathbf{1}_{N\times K} \\ \mathbf{1}_{K\times N} & \mathbf{1}_{K\times K} \end{pmatrix}, \qquad M_{\mathrm{G}} = \begin{pmatrix} \mathbf{1}_{SN\times SN} & \begin{pmatrix} M_1 \cdot \mathbf{1}_{N\times K} \\ M_2 \cdot \mathbf{1}_{N\times K} \\ \vdots \\ M_S \cdot \mathbf{1}_{N\times K} \end{pmatrix} \\ \mathbf{1}_{K\times SN} & \mathbf{1}_{K\times K} \end{pmatrix}, \tag{15}$$

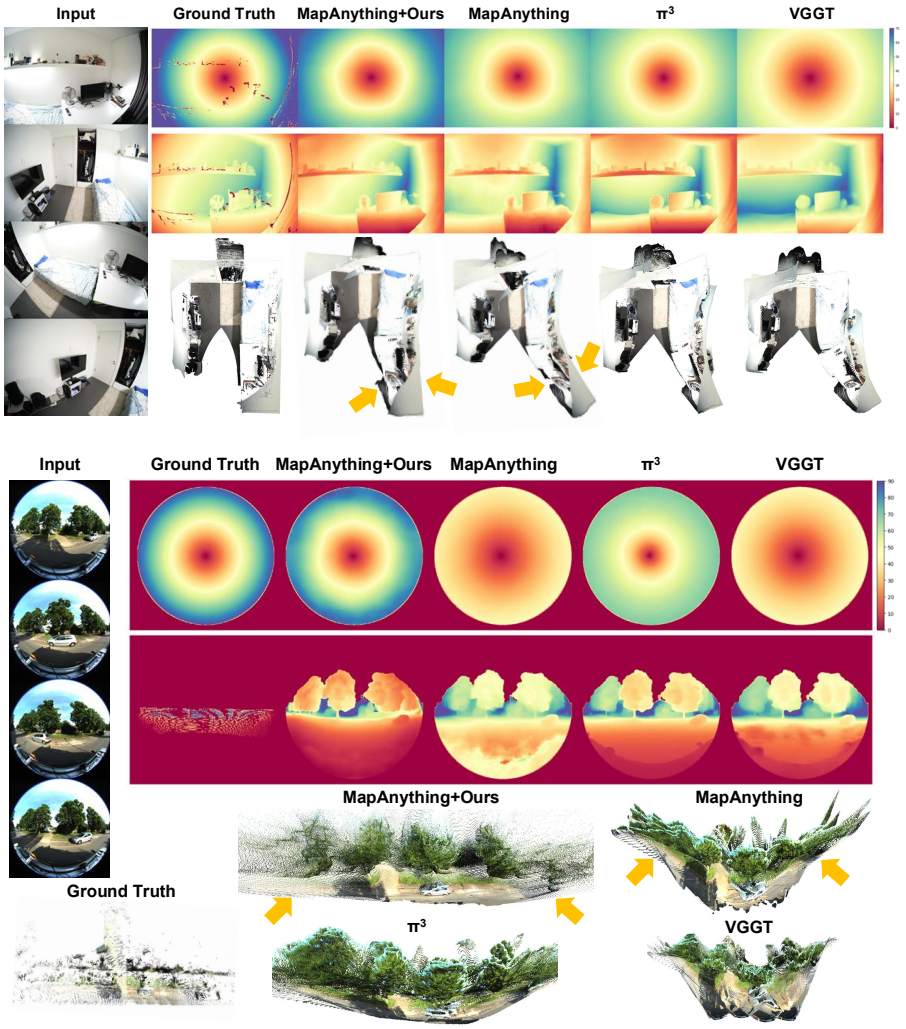

Fig. 4: Qualitative comparison of FoV, depth, and point map estimation on an indoor (top) and outdoor (bottom) fisheye sequence. We apply calibration tokens to MapAnything and consistently produce more geometrically consistent reconstructions compared to the baseline models. The yellow arrows highlight the regions where our calibration tokens effectively rectify significant geometric distortions.

where N is the number of image tokens per frame, K is the number of calibration tokens per layer, and $\mathbf{1}$ represents a matrix of ones. The (i, j)-th entry of each mask controls whether the j-th input token affects the i-th output token. Then, $M_{\mathrm{F},s}$ are passed to the remaining $L_1 - L_0$ image encoder layers and L_2 frame-wise layers in the AA module, while M_{G} is passed to the global attention layers in the AA module. Altogether, these masks nullify the effect of calibration tokens on perspective frames.

4 Experiments

4.1 Settings

Datasets. We train the models with 3 learning schemes as mentioned in Sec. 3.3. For **SSL** and **SL**, we use 6 perspective datasets for training: ScanNet++ (perspective) [52], MegaDepth [21], BlendedMVS [51], TartanAir [45], MVS-Synth [16], and ParallelDomain-4D [38], spanning both real and synthetic, indoor and outdoor scenes. For **SL+**, we additionally add two fisheye datasets for training: ASE [8] and KITTI360 [22]. For testing, we evaluate the performance on 3 fisheye datasets: ScanNet++ (fisheye) [52], ADT [29], and KITTI360 [22].
Metrics. We evaluate the models on 4 different tasks and 15 quantitative metrics following [18, 37, 46]. For camera pose estimation, we use Relative Rotation Accuracy (RRA), Relative Translation Accuracy (RTA), and Area Under Curve (AUC) of the accuracy-threshold curve at 30° as angular accuracy metrics and Absolute Trajectory Error (ATE), Relative Pose Error for translation (RPEt), and Relative Pose Error for rotation (RPEr) as distance error metrics. For depth map estimation, we use Absolute Relative Error (Rel), Root Mean Squared Error (RMSE), and δ_1 accuracy. For point map estimation, we use Accuracy (Acc), Completeness (Comp), and Chamfer Distance (CD). For FoV map estimation, we use median error for horizontal (hErr) and vertical (vErr) FoVs in degrees, as well as the Area Under Curve (AUC) up to a 10° threshold for FoV.
Models. We apply our method on three foundation reconstruction models, VGGT [41], π^3 [46], and MapAnything [18], based on their public checkpoints.

4.2 Results

Quantitative improvements on fisheye datasets. Table 1 summarizes the quantitative results on three datasets of fisheye sequences. Fisheye3R yields consistent and substantial performance gains over the baseline models, demonstrating that calibration tokens can effectively bridge the geometric domain gap between perspective and fisheye imagery. Remarkably, even self-supervised learning using only unlabeled perspective RGB images boosts performance on fisheye data by a large margin, achieving a 50%+ improvement in 26/135 tests across all models and metrics. This highlights our framework's potential to scale with larger corpora of unlabeled data for further enhancement. The introduction of annotated perspective data further strengthens these results, with 37/135 metrics showing improvements exceeding 50%. Performance ultimately peaks when incorporating annotated fisheye data, where a 50%+ improvement is achieved in 77/135 cases. This shows that while Fisheye3R is highly effective in data-constrained regimes, it scales efficiently as more supervision becomes available.
Qualitative comparisons. We qualitatively compare our method with previous unified feed-forward foundation models in Fig. 4. Our method achieves significantly more accurate 3D scene reconstruction from the sequence of fisheye images thanks to the calibration tokens. While traditional foundation models

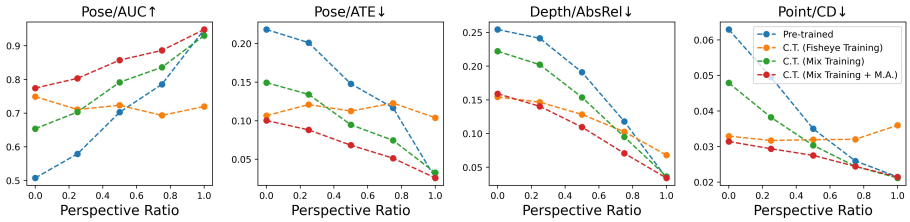

Fig. 5: Performance analysis on hybrid sequences of ScanNet++ with varying perspective image ratios. **Pre-trained**: the vanilla MapAnything backbone without adaptation. **C.T. (Fisheye Training)**: calibration tokens trained on fully fisheye sequences. **C.T. (Mix Training)**: tokens trained on mixture sequences of both camera types. **C.T. (Mix Training + M.A.)**: masked attention introduced during mixed training.

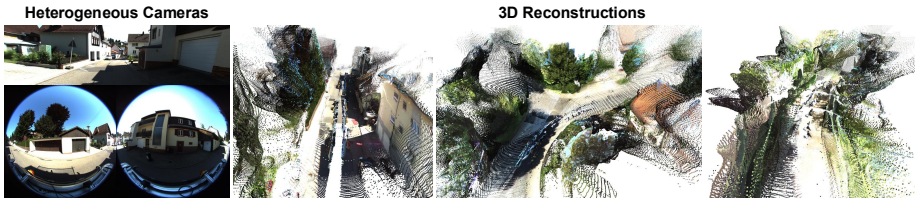

Fig. 6: Reconstructions over a hybrid camera system in driving scenes [22]. The left and right fisheye cameras capture the two sides of the scene, while the front-facing perspective camera bridges the two views. Without a model capable of handling heterogeneous cameras, reconstructions from the two fisheye views would remain disconnected.

are incapable of accurately predicting 3D scene geometry from distorted fisheye lenses, our method produces reconstructions with higher fidelity and better alignment with ground truth.

Performance on hybrid inputs. We analyze the calibration performance across hybrid sequences containing both perspective and fisheye frames in Fig. 5 and Fig. 6. By varying the ratio of perspective images, Fig. 5 shows that while calibration tokens excel at optimizing pure fisheye sequences, they suffer from parameter drift caused by specialized training on fisheye images, leading to performance degradation on perspective images. Conversely, while training tokens on mixed-camera sequences preserves perspective accuracy, it weakens adaptation to fisheye data. By contrast, our proposed masked attention mechanism not only enables backwards compatibility by retaining the model's native performance on perspective data, but also unleashes the representational power of calibration tokens for fisheye adaptation, achieving the most robust performance across all perspective ratios. The reconstructions in Fig. 6 illustrate a practical deployment scenario of our mixed-camera model for autonomous driving, where a vehicle is equipped with heterogeneous cameras. Leveraging the forward-facing perspective camera to connect the left and right fisheye views, our method reconstructs a single, geometrically consistent panoramic scene from hybrid sensors.

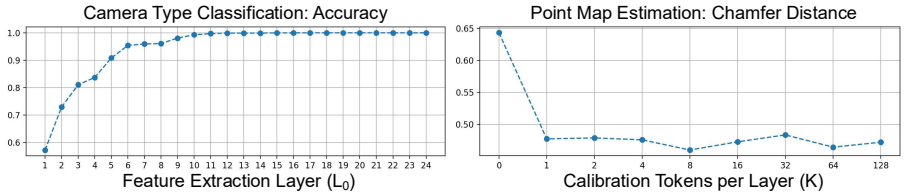

Fig. 7: Ablation studies of hyperparameters. **Feature extraction depth** (left): Classification accuracy of camera types as a function of the number of frozen initial layers (L_0). **Token density** (right): Impact of the number of calibration tokens per layer (K) on reconstruction accuracy, measured by Chamfer Distance. Experiments are conducted on ScanNet++ with MapAnything backbone.

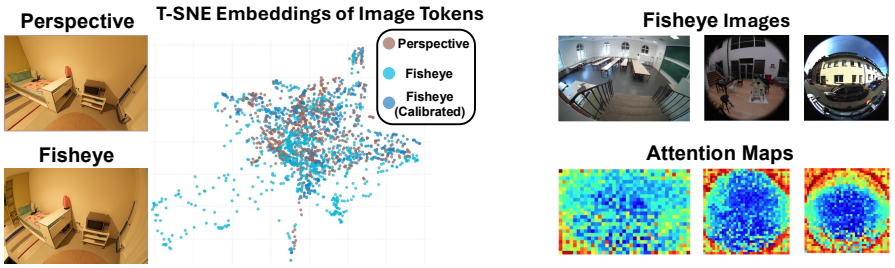

Fig. 8: Visualizing the impact of calibration tokens on the feature space with MapAnything. **Feature alignment** (left): Image tokens from the last alternating attention layer of a perspective and fisheye image pair, with fisheye tokens derived from the model before and after calibration. **Attention maps** (right): Three fisheye test images and their average attention weights from image to calibration tokens in the image encoder.

Ablation experiments. We present two ablation studies in Fig. 7. First, we analyze the feature extraction depth (L_0) required for robust camera type classification. While deeper layers provide the class token with higher-level semantic information, increasing L_0 conversely reduces the number of subsequent layers ($L_1 - L_0$) available for feature alignment via calibration tokens. As shown in Fig. 7 (left), classification accuracy saturates at approximately $L_0 = 12$ layers. We identify this as the optimal bottleneck, as it reserves the remaining 12 encoder layers for calibration, thereby maximizing the potential for geometric adaptation. Second, we evaluate the number of calibration tokens (K) per layer in Fig. 7 (right), which reveals that even a single token ($K = 1$) per layer brings the majority of the performance gain, demonstrating the efficiency of the proposed token-based alignment. While increasing the token count yields further marginal improvements, the reconstruction quality peaks approximately at $K = 8$. Therefore, we adopt $L_0 = 12$ and $K = 8$ as the default configuration for the experiments.

Feature alignment. We employ t-SNE [24] to visualize the patch-level embeddings of a perspective and fisheye image pair in Fig. 8 (left). As shown in

Model	KB (OOD)	Fisheye624	MEI	Equidistant	Stereographic	Equiangular	Orthographic
π^3 CD↓	0.230	0.228	0.264	0.219	0.211	0.241	0.375
+Ours	0.116	0.117	0.095	0.107	0.088	0.116	0.182
Improvement	49.7%	48.5%	64.1%	51.1%	58.1%	52.0%	51.5%

Table 2: Reconstruction CD of π^3 on Stanford2D3DS with various projection models. Although our method employs Kannala-Brandt (KB) for synthesis, it generalizes well to KB with out-of-distribution (OOD) parameters and other projection models.

the embedding space, the original fisheye features (light blue) exhibit a distinct distributional shift and deviate significantly from the perspective manifold (brown). This geometric domain gap explains the performance degradation observed when applying the pre-trained foundation model directly to fisheye data. By introducing calibration tokens, the adapted fisheye embeddings (dark blue) are effectively realigned to the perspective distribution. This demonstrates that the tokens are able to bridge the geometric gap by projecting fisheye patches into a shared representation space consistent with the model's original training distribution.

Attention map visualization. To further understand the spatial influence of our framework, we visualize the attention weights that image tokens pay to the calibration tokens in Fig. 8 (right). These attention maps serve as an indicator of the corrective influence exerted by the calibration tokens on specific image patches. It can be observed that patches in the peripheral regions that suffer from the most significant radial distortion pay more attention to calibration tokens (red). This suggests that the calibration tokens selectively target regions where the features are most divergent from the perspective distribution, prioritizing the correction of extreme geometric warping at fisheye image boundaries to restore global feature consistency.

Generalization to other projection models. To evaluate the projection model domain gap in a controlled setting and to eliminate dataset-dependent confounding effects, we render fisheye images from Stanford2D3DS dataset [2] with varying projection models and measure the improvement of reconstruction quality of our approach in Tab. 2. The results indicate that although only Kannala-Brandt model is employed for fisheye synthesis, the learned calibration tokens generalize well to other projection models.

Generalization to panoramic images. While this work primarily focuses on extending foundation models to fisheye imagery, calibration tokens have the potential to generalize to images from other camera models as well, such as panoramic images. Figure 9 presents multi-view reconstructions from 360° images, where calibration tokens are trained on Matterport3D [4] and tested on Stanford2D3DS [2]. The result implies that calibration tokens can be generalized to camera models beyond fisheye.

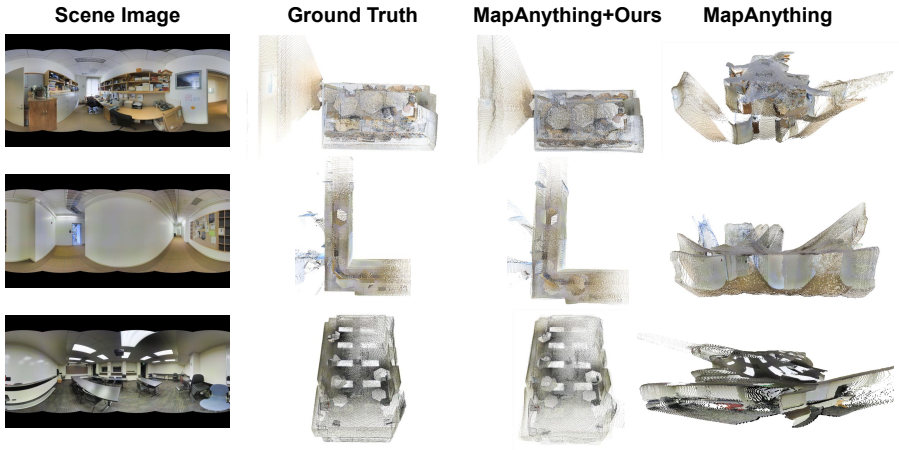

Fig. 9: Generalization of calibration tokens to panoramic images.

5 Discussion

Projective geometry describes the mapping between 3D points and 2D image pixels. Different camera lenses induce different projection functions, resulting in different spatial arrangements of pixels in the images formed. As existing models are trained on perspective images, they take on such biases implicitly. When tested on images with curvilinear distortion, even for 3D scenes previously observed, local operations such as patch embedding operate on a different set of visual patterns that induce a covariate shift in the latent features, leading to out-of-distribution behavior and performance degradation. As the change occurs in the value of the features distributed over the spatial domain, our work aims to recalibrate them with learnable tokens to align with the original training distribution of perspective features. While this can be viewed as a form of camera domain adaptation, our masked attention mechanism enables a single network to support different camera models by selectively controlling the activation of adaptation based on camera types.

Limitations. The proposed method is limited to transformer architectures due to its dependency on token-based adaptation. The fidelity of self-supervised adaptation also depends on the original model's performance. While this is partially mitigated by the use of ground truth in the supervised setting, one may be limited for tasks that do not have annotations readily available.

Acknowledgments

This work is supported by Google Gift Funding and NSF-2112562 Athena AI Institute. Additionally, we thank Federico Tombari for his valuable guidance and constructive feedback during the course of this project.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10), 105–112 (2011) 4
2. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017) 14
3. Caruso, D., Engel, J., Cremers, D.: Large-scale direct slam for omnidirectional cameras. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 141–148. IEEE (2015) 1, 4
4. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017) 14
5. Deng, J., Wang, Y., Meng, H., Hou, Z., Chang, Y., Chen, G.: Omnistereo: Real-time omnidireactional depth estimation with multiview fisheye cameras. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1003–1012 (2025) 4
6. Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: Transmvsnet: Global context-aware multi-view stereo network with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8585–8594 (2022) 4
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 3, 5
8. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., et al.: Project aria: A new tool for egocentric multi-modal ai research. arXiv preprint arXiv:2308.13561 (2023) 11
9. Frahm, J.M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.H., Dunn, E., Clipp, B., Lazebnik, S., et al.: Building rome on a cloudless day. In: European conference on computer vision. pp. 368–381. Springer (2010) 4
10. Gangopadhyay, S., Kim, J.H., Chen, X., Rim, P., Park, H., Wong, A.: Extending foundational monocular depth estimators to fisheye cameras with calibration tokens. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5198–5209 (2025) 4
11. Gangopadhyay, S., Wong, A.: From perspective to fisheye depth estimation and open-vocabulary segmentation. In: European Conference on Computer Vision. Springer (2026) 4
12. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: European conference on computer vision. pp. 445–461. Springer (2000) 4
13. Gherardi, R., Farenzena, M., Fusiello, A.: Improving the efficiency of hierarchical structure-and-motion. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 1594–1600. IEEE (2010) 4
14. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2495–2504 (2020) 4
15. Guo, Y., Garg, S., Miangoleh, S.M.H., Huang, X., Ren, L.: Depth any camera: Zero-shot metric depth estimation from any camera. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26996–27006 (2025) 4

16. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2821–2830 (2018) 11
17. Kannala, J., Brandt, S.S.: 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) 4, 7
18. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025) 1, 2, 3, 4, 6, 9, 11
19. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024) 4
20. Li, M., Hu, X., Dai, J., Li, Y., Du, S.: Omnidirectional stereo depth estimation based on spherical deep network. Image and Vision Computing **114**, 104264 (2021) 4
21. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2041–2050 (2018) 11
22. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3292–3310 (2022) 1, 9, 11, 12
23. Lichy, D., Su, H., Badki, A., Kautz, J., Gallo, O.: Fova-depth: Field-of-view agnostic depth estimation for cross-dataset generalization. In: 2024 International Conference on 3D Vision (3DV). pp. 1–10. IEEE (2024) 4
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 13
25. Matsuki, H., Von Stumberg, L., Usenko, V., Stückler, J., Cremers, D.: Omnidirectional dso: Direct sparse odometry with fisheye cameras. IEEE Robotics and Automation Letters **3**(4), 3693–3700 (2018) 4
26. Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. pp. 3945–3950. IEEE (2007) 4
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 5
28. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: European Conference on Computer Vision. pp. 58–77. Springer (2024) 4
29. Pan, X., Charron, N., Yang, Y., Peters, S., Whelan, T., Kong, C., Parkhi, O., Newcombe, R., Ren, Y.C.: Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20133–20143 (2023) 9, 11
30. Piccinelli, L., Sakaridis, C., Segu, M., Yang, Y.H., Li, S., Abbeloos, W., Van Gool, L.: Unik3d: Universal camera monocular 3d estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1028–1039 (2025) 3, 4
31. Scaramuzza, D., Martinelli, A., Siegwart, R.: A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: Fourth IEEE International Conference on Computer Vision Systems (ICVS'06). pp. 45–45. IEEE (2006) 4

32. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016) 4
33. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European conference on computer vision. pp. 501–518. Springer (2016) 4
34. Schops, T., Sattler, T., Pollefeys, M.: Bad slam: Bundle adjusted direct rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 134–144 (2019) 1
35. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM siggraph 2006 papers, pp. 835–846 (2006) 4
36. Sweeney, C., Sattler, T., Hollerer, T., Turk, M., Pollefeys, M.: Optimizing the viewing graph for structure-from-motion. In: Proceedings of the IEEE international conference on computer vision. pp. 801–809 (2015) 4
37. Tirado-Garín, J., Civera, J.: Anycalib: On-manifold learning for model-agnostic single-view camera calibration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8044–8055 (2025) 4, 11
38. Van Hoorick, B., Wu, R., Ozguroglu, E., Sargent, K., Liu, R., Tokmakov, P., Dave, A., Zheng, C., Vondrick, C.: Generative camera dolly: Extreme monocular dynamic novel view synthesis. In: European Conference on Computer Vision. pp. 313–331. Springer (2024) 11
39. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: Geocalib: Learning single-image calibration with geometric optimization. In: European Conference on Computer Vision. pp. 1–20. Springer (2024) 4
40. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: Learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14194–14203 (2021) 4
41. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025) 1, 3, 4, 6, 9, 11
42. Wang, N.H., Solarte, B., Tsai, Y.H., Chiu, W.C., Sun, M.: 360sd-net: 360 stereo depth estimation with learnable cost volume. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 582–588. IEEE (2020) 4
43. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025) 4
44. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024) 4
45. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. in 2020 ieee. In: RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4909–4916 11
46. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025) 1, 3, 4, 6, 9, 11
47. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision-3DV 2013. pp. 127–134. IEEE (2013) 4
48. Wu, Y., Liu, T.Y., Park, H., Soatto, S., Lao, D., Wong, A.: Augundo: Scaling up augmentations for monocular depth completion and estimation. In: European Conference on Computer Vision. pp. 274–293. Springer (2024) 7

49. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21924–21935 (2025) 4
50. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018) 4
51. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1790–1799 (2020) 11
52. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023) 1, 9, 11
53. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825 (2024) 4
54. Zhao, D., Lichy, D., Perrin, P.N., Frahm, J.M., Sengupta, S.: Mvpsnet: Fast generalizable multi-view photometric stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12525–12536 (2023) 4
55. Zhao, G., Liu, Y., Qi, W., Ma, F., Liu, M., Ma, J.: Fisheyedepth: A real scale self-supervised depth estimation model for fisheye camera. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 3780–3787. IEEE (2025) 4
56. Zhao, H., Chen, X., Li, Y., Wang, Q., Lu, H., Gao, F.: Fastvidar: Real-time omnidirectional depth estimation via alternative hierarchical attention. arXiv preprint arXiv:2509.23733 (2025) 4