Camera3R: Feed-Forward 3D Reconstruction from Heterogeneous Camera Models

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

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

Abstract

Feed-forward 3D reconstruction (3R) from unconstrained multi-view images has recently emerged as a prominent research direction in 3D computer vision. Various foundational reconstruction models have been proposed with different architectures and reconstruction paradigms. Despite being trained on large-scale perspective datasets with substantial computational resources, these models generally struggle to generalize to non-perspective images. This limitation hinders their deployment in applications involving arbitrary, unknown, and potentially heterogeneous camera models. To build on the capabilities of existing reconstruction models with minimal retraining, we propose Camera3R, a low-rank adaptation framework that efficiently extends perspective-pretrained 3R networks to any other camera models. Extensive experiments demonstrate that, with only 0.5% additional parameters and less than a day of training, Camera3R adapts a foundational 3D reconstruction model to non-perspective cameras while fully preserving its original performance on perspective inputs. Code is available at https://github.com/android-xr/camera3r.

1 Introduction

Three-dimensional (3D) visual mapping of an environment is a fundamental component in many spatial applications, ranging from autonomous navigation to extended reality systems. Decision-making within these applications rests on the fidelity of the map, e.g., accuracy of estimated points (and position and orientation of the platform), as well as completeness, or coverage, of the environment as observed by a set of cameras. In practice, the two metrics are not independently optimizable, but are coupled through sensing trade-offs. For a fixed sensing budget, a reduction in field-of-view (FoV) increases the angular resolution, which often results in higher fidelity estimates. On the other hand, an increase in FoV naturally improves environmental coverage, but typically introduces non-trivial lens distortion and measurement noise that degrade per-point accuracy. This trade-off motivates the use of multiple cameras, e.g., combinations of perspective and fisheye or panoramic (360°) cameras, within a mixed-camera system whose cameras provide complementary fidelity and coverage.

While mixed-camera systems offer the potential to balance map fidelity and environmental coverage in 3D reconstruction, existing feed-forward 3D reconstruction models [15, 26, 29] cannot fully exploit the non-perspective or heterogeneous camera inputs. Despite being trained on large-scale datasets comprising tens of millions of images, these models cannot generalize to images captured by wide-FoV or mixed cameras. This failure mode can be attributed to a covariate shift between the training and deployment imaging platforms: The training data used by current feed-forward reconstruction models is predominantly composed of perspective images. In contrast, images captured by different camera models exhibit varying fields of view, resolutions, and distortion profiles, leading to heterogeneous photometric and spatial sampling statistics. Consequently, these models do not

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

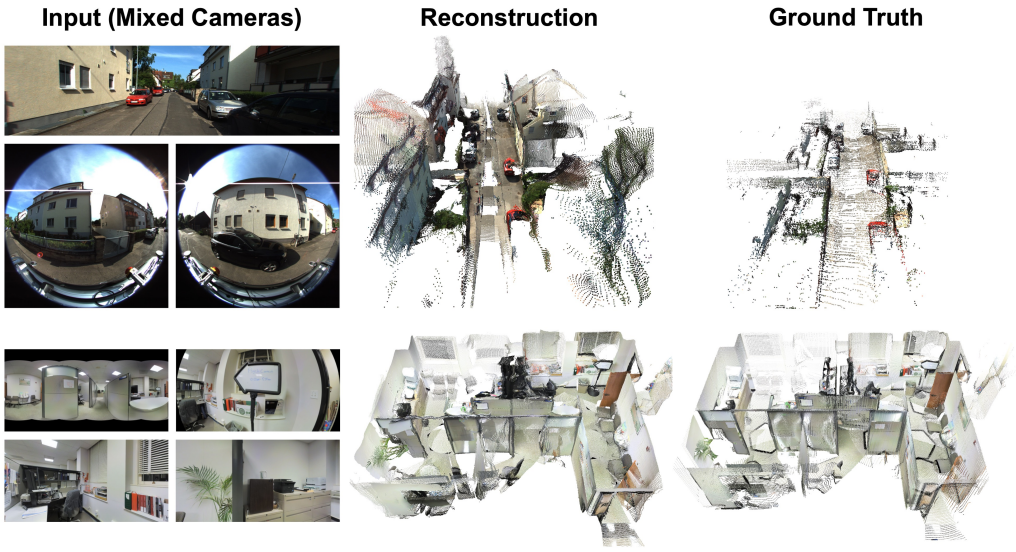


Figure 1: **3D reconstruction from heterogeneous cameras.** Given an image sequence of arbitrary, unknown, and mixed camera models, our method reconstructs the scene in a single forward pass.

directly transfer to mixed-camera streams, limiting their potential to provide high-fidelity maps with broad environmental coverage, despite their strong generalization across diverse 3D scenes.

A natural response to this limitation is to train separate 3D reconstruction models tailored to each camera type or imaging configuration, and subsequently register their outputs and combine them into a single map. However, such an approach would incur substantial operational overhead, and is poorly suited to spatial applications, which are often constrained by compute, memory, and overall system complexity. A key insight is that, despite differences in photometric and spatial sampling statistics due to the image formation process, images of the same 3D scene captured by different camera models should correspond to a single consistent environmental map. In other words, the heterogeneity in camera models may alter the observations, but not the underlying 3D structures being reconstructed. This intuition motivates our central thesis of whether a single feed-forward 3D reconstruction model trained primarily on images captured by perspective cameras can be systematically extended to operate on heterogeneous camera inputs without retraining from scratch.

To this end, we introduce Camera3R, a framework that enables a single feed-forward model to ingest images captured by multiple camera types to produce a consistent map. The crux of our method lies in selective adaptation to inputs during inference time through a mixture of experts. We assume access to a pretrained feed-forward model and that images can originate from a broad categorization of perspective, fisheye, and panoramic cameras. We associate each camera model with an expert, each parameterized by a set of weights. We begin by freezing the pretrained model; as it is pretrained on large-scale datasets of perspective images, we consider it an expert in itself. We surmise that the pretrained model has already learned sufficiently expressive features to model diverse 3D scenes and that the same features can be efficiently modulated, rather than relearned, to extend the pretrained model to accommodate alternative camera types. Accordingly, we introduce low-rank adaptation (LoRA) [11] to the network layers to yield experts for fisheye and panoramic cameras. However, it remains a question as to how much capacity should be allocated to each camera type to model specific distortions and viewing statistics. To address this, we employ a rank-truncation scheme that redistributes capacity by allocating higher ranks to layer-expert pairs based on need, allowing us to leverage the existing knowledge to its full extent while adding minimal overhead in parameters to the model. The result is an efficient model unified across cameras that can be trained with an order of magnitude less data than the base model itself.

In most spatial applications, the origin of the camera data stream is known; therefore, expert selection based on camera identity is sufficient to adaptively run inference on each image. Nonetheless, to accommodate general usage, we propose to learn a camera router which determines the expert to use for modulating latent features. Hence, at test time, we do not assume access to camera metadata

and can handle inputs of any single camera type, as well as a collection of images originating from a mixture of heterogeneous cameras. This design naturally enables backward compatibility, where the generalizability and performance of existing feed-forward 3D reconstruction models are preserved. At the same time, new cameras can be introduced into an existing platform to improve both map fidelity and environmental coverage, and be accommodated without disruption to established workflows. We evaluate our method on multiple 3D reconstruction datasets, spanning indoor and outdoor environments. In single-camera settings, our method achieves a significant improvement over the baseline methods and fully preserves performance on perspective images. We further demonstrate that, in mixed-camera scenarios, our approach can handle a collection of images originating from any of the cameras from a heterogeneous set without prior knowledge of their identity to yield consistent and high-fidelity reconstructions. Our method is applied to three different feed-forward 3D reconstruction models to demonstrate its generalizability.

Our contributions are as follows: (1) We extend existing feed-forward 3D reconstruction models pretrained on perspective imagery to operate on fisheye and panoramic camera inputs through a mixture-of-experts framework with adaptive routing. (2) We introduce an adaptive capacity allocation mechanism that selectively allocates LoRA-based adaptation capacity across experts and layers, significantly reducing the parameter overhead. (3) We achieve state-of-the-art performance in both single-camera and mixed-camera reconstruction settings across multiple indoor and outdoor datasets. (4) To the best of our knowledge, we are the first to enable joint feed-forward 3D reconstruction from heterogeneous perspective, fisheye, and panoramic images in a single forward pass.

2 Related Work

Feed-Forward 3D Reconstruction. Recent progress in 3D reconstruction has been strongly impacted by the introduction of unified, feed-forward foundational models that infer camera poses and scene geometry from unconstrained images in a single forward pass. The trend can be traced back to DUSt3R [27], which directly predicts dense point maps from image pairs. MASt3R [17] further advances the framework by improving reconstruction quality through explicit feature matching. Spann3R [25] subsequently extends this pairwise paradigm to image collections by aggregating information with spatial memory and directly regressing global point maps without optimization-based global alignment. VGGT [26] pioneered a unified, general-purpose framework that predicts camera parameters, depth maps, point maps, and point tracks from multiple images in a single forward pass with an alternating-attention transformer architecture. π^3 [29] introduces a permutation-equivariant formulation that eliminates performance dependence on a fixed reference view and input order. MapAnything [15] incorporates optional geometric inputs for flexible metric reconstruction. These works collectively mark a shift towards unified, feed-forward visual-geometry foundational models for general-purpose 3D reconstruction.

Feed-Forward 3D Reconstruction for Non-Perspective Images. Despite their strong performance on standard perspective imagery, these foundational reconstruction models generally fail on non-perspective inputs. Some recent works have developed reconstruction models for non-perspective images. PanoVGGT [9] and Argus [19] achieve feed-forward reconstruction from panoramic images based on VGGT. Wid3R [13] employs spherical harmonics for ray representations to handle perspective, fisheye, and panoramic inputs in reconstruction models, but does not demonstrate reconstructions from mixed-camera sequences. These approaches either specialize in a particular camera model or rely on known camera types during inference. Furthermore, they do not explicitly address mixed-camera input sequences or guarantee backward compatibility on perspective images. CAM3R [10] extends the pairwise reconstruction to camera-agnostic settings by predicting per-pixel rays and radial distances, using a ray-aware global alignment stage to combine pairwise predictions across views. Fisheye3R [4] formulates the task of generalizing reconstruction models to fisheye cameras as an adaptation problem and introduces calibration tokens [8] to extend perspective-pretrained reconstruction networks [15, 26, 29] to fisheye cameras without performance degradation on perspective inputs, but the scope is limited to the fisheye domain. In contrast, our work extends foundational multi-view reconstruction models to handle perspective, fisheye, and panoramic inputs simultaneously in a single forward propagation, accepting sequences with unknown and heterogeneous camera models, while ensuring backward compatibility on perspective data.

Low-Rank Adaptation and Adaptive Rank Allocation. Low-rank adaptation (LoRA) [11] is a parameter-efficient finetuning strategy originally proposed to adapt large language models. Instead

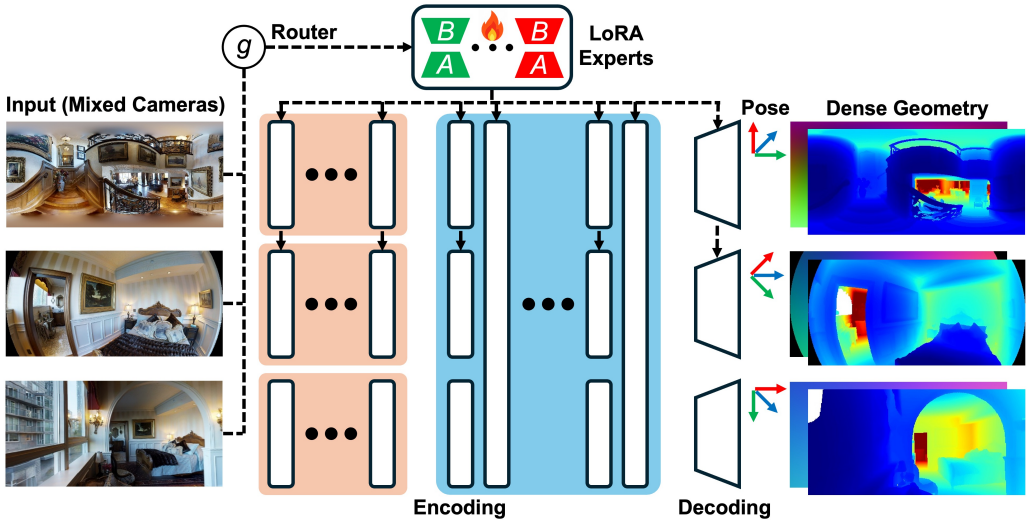


Figure 2: **Camera-agnostic feed-forward reconstruction.** Given an input sequence from arbitrary, unknown, and mixed camera models, we adapt a perspective-pretrained reconstruction network by camera-specific LoRA experts. Each frame is passed to a camera router, which selects the corresponding LoRA expert to adapt the model to the frame's camera geometry, enabling pose and dense geometry (depth, ray directions, etc.) prediction from non-perspective cameras. Perspective frames bypass the adaptation, preserving the original behavior of the pretrained backbone.

of updating the entire set of model parameters, LoRA freezes the backbone and injects lightweight learnable low-rank matrices into selected layers for efficient adaptation, reducing the number of trainable parameters while preserving the original model's knowledge. Recent works further combine LoRA with mixture-of-experts routing [18, 30, 33], where multiple low-rank experts are selected or combined to improve specialization across inputs or tasks. To adaptively allocate resources and achieve lightweight adaptation, adaptive-rank and post-hoc compression approaches such as [16, 24, 34] have been introduced.

3 Method

We study the problem of extending multi-view 3D reconstruction models pretrained on perspective imagery to handle sequences of arbitrary, unknown, and heterogeneous camera types during inference. Three camera models are considered specifically, including perspective, fisheye, and panoramic, due to their relative prevalence in real-world 3D reconstruction applications.

3.1 Feed-Forward Multi-View 3D Reconstruction

Given a set of V unconstrained RGB images $\{I_i\}_{i=1}^{V}$ describing a 3D scene, where the i-th frame $I_i \in [0,1]^{H\times W\times 3}$ for each $i \in [V] = \{1,\dots,V\}$, we denote the camera model of I_i as $c_i \in \mathcal{C} = \{c_{\text{persp}}, c_{\text{fish}}, c_{\text{pano}}\}$. Typical 3D reconstruction methods [15, 26, 29] assume all the images are perspective, i.e., $c_i = c_{\text{persp}}$ for all i, and process an input sequence by three steps in general.

First, an image encoder f_{img}, usually a pretrained frozen DINOv2 [21], extracts features from each frame separately. An image I_i is divided into $N = HW/p^2$ patches of size $p \times p$ pixels, projected to embeddings $\{\boldsymbol{z}_{i,j}^{(0)}\}_{j=1}^{N}$ where $\boldsymbol{z}_{i,j}^{(0)} \in \mathbb{R}^d$ is the j-th patch embedding of the i-th frame and d is the embedding dimension, and then concatenated along with a class token $\boldsymbol{z}_{i,0}^{(0)} = \boldsymbol{\phi}_{\text{cls}} \in \mathbb{R}^d$ as

$$\boldsymbol{z}_i^{(0)} = \left[\boldsymbol{z}_{i,j}^{(0)}\right]_{j=0}^{N} \in \mathbb{R}^{(N+1)\times d}, \quad \forall i \in [V]. \tag{1}$$

For an encoder f_{img} with L transformer blocks, the ℓ-th block processes the frame embeddings as

$$\boldsymbol{z}_i^{(\ell)} = f_{\text{img}}^{(\ell)}\left(\boldsymbol{z}_i^{(\ell-1)}\right) \in \mathbb{R}^{(N+1)\times d}, \quad \forall i \in [V],\ \forall \ell \in [L], \tag{2}$$

by multi-head self-attention and multi-layer perceptrons (MLPs) [3].

Second, the encoded frame features of the entire sequence are passed to a sequence encoder f_{seq}, which typically employs an alternating attention architecture [26], to jointly extract sequence features for 3D reconstruction as

$$[\boldsymbol{z}_i]_{i=1}^{V} = f_{\text{seq}}\left(\left[\boldsymbol{z}_i^{(L)}\right]_{i=1}^{V}\right) \in \mathbb{R}^{V\times(N+1)\times d}, \tag{3}$$

where $\boldsymbol{z}_i \in \mathbb{R}^{(N+1)\times d}$ is the final features of the i-th frame after the entire encoding process.

Finally, the features of each frame are forwarded to a decoder f_{dec} to predict the camera pose and dense scene attributes as

$$(\boldsymbol{q}_i, \boldsymbol{t}_i), \mathcal{G}_i = f_{\text{dec}}(\boldsymbol{z}_i), \quad \forall i \in [V], \tag{4}$$

where $\boldsymbol{q}_i \in \mathbb{R}^4$ and $\boldsymbol{t}_i \in \mathbb{R}^3$ represent the predicted camera-to-world rotation quaternion and translation of frame i, and $\mathcal{G}_i \in \mathbb{R}^{H\times W\times D}$ denotes the pixel-wise dense predictions concatenated to D output channels, possibly including depth maps, point maps, ray directions, or other geometric quantities, depending on the model design.

3.2 Extension to Non-Perspective Camera Models

Low-rank adaptation to arbitrary cameras. An overview of our approach is provided in Figure 2. For a reconstruction model with K linear layers, suppose the k-th linear layer has a weight matrix $W_k \in \mathbb{R}^{n_k\times m_k}$ where m_k and n_k denote the layer's input and output dimensions. The layer derives

$$\boldsymbol{y}_{i,k} = W_k \boldsymbol{x}_{i,k}, \quad \forall i \in [V],\ \forall k \in [K], \tag{5}$$

where $\boldsymbol{x}_{i,k} \in \mathbb{R}^{m_k}$ is an input vector from frame i sent to layer k (with the frame's token index omitted for simplicity) and $\boldsymbol{y}_{i,k} \in \mathbb{R}^{n_k}$ is the output vector. To extend the reconstruction model to non-perspective images while preserving its performance on perspective inputs, each layer is adapted by camera-specific LoRA experts as

$$\boldsymbol{y}_{i,k} = W_k \boldsymbol{x}_{i,k} + \sum_{c\in\mathcal{C}\setminus\{c_{\text{persp}}\}} \mathbb{1}_{c=c_i} B_k^c A_k^c \boldsymbol{x}_{i,k}, \quad \forall i \in [V],\ \forall k \in [K], \tag{6}$$

where $B_k^c \in \mathbb{R}^{n_k\times r_k^c}$ and $A_k^c \in \mathbb{R}^{r_k^c\times m_k}$ are learnable low-rank matrices with rank $r_k^c \le \min(m_k, n_k)$ that are injected into the layer for adaptation to a non-perspective camera c, and $\mathbb{1}$ is the indicator function to activate the corresponding camera expert for adaptation.

With this formulation, each non-perspective camera LoRA expert specializes in adapting the layer to that specific camera's projection geometry, while perspective frames bypass all these adaptations and are processed by the original weights, thereby preserving the model's prior performance. During training, the original weights W_k are frozen and the LoRA matrices are optimized, with random Gaussian initialization for each A_k^c and zero for each B_k^c's [11]. Although this paper focuses on perspective, fisheye, and panoramic images, this framework naturally extends to other camera models by introducing additional experts.

Input-conditioned adaptation for unknown cameras. In Equation (6), the camera type of each frame c_i must be known to obtain the binary indicator $\mathbb{1}_{c_i=c}$. However, in real-world scenarios, camera metadata is often unavailable at test time. To generalize the framework to handle potentially unknown cameras during inference, we modify each layer as

$$\boldsymbol{y}_{i,k} = \begin{cases} W_k \boldsymbol{x}_{i,k} + \sum_{c\in\mathcal{C}\setminus\{c_{\text{persp}}\}} \mathbb{1}_{c=c_i} B_k^c A_k^c \boldsymbol{x}_{i,k}, & \text{if } c_i \text{ known,} \\ W_k \boldsymbol{x}_{i,k} + \sum_{c\in\mathcal{C}\setminus\{c_{\text{persp}}\}} g_c\left(I_i\right) B_k^c A_k^c \boldsymbol{x}_{i,k}, & \text{if } c_i \text{ unknown,} \end{cases} \quad \forall i \in [V],\ \forall k \in [K], \tag{7}$$

where I_i is the i-th frame. The function $g : \mathbb{R}^{H\times W\times 3} \to \{0,1\}^{|\mathcal{C}|}$ is a LoRA router that predicts a one-hot camera vector for each frame, and $g_c(\cdot) \in \{0,1\}$ denotes the binary prediction of $g(\cdot)$ on camera c. This discrete routing strategy that selects a single LoRA expert is employed instead of the soft mixture over experts in order to let each expert focus solely on adapting to images of that specific camera projection model, to prevent leakage of adaptation to perspective frames, and to facilitate future introduction of new camera experts.

Incremental learning with novel cameras. A prospective advantage of our camera-specific LoRA experts is its natural feasibility for continual learning. In standard full finetuning and single-expert

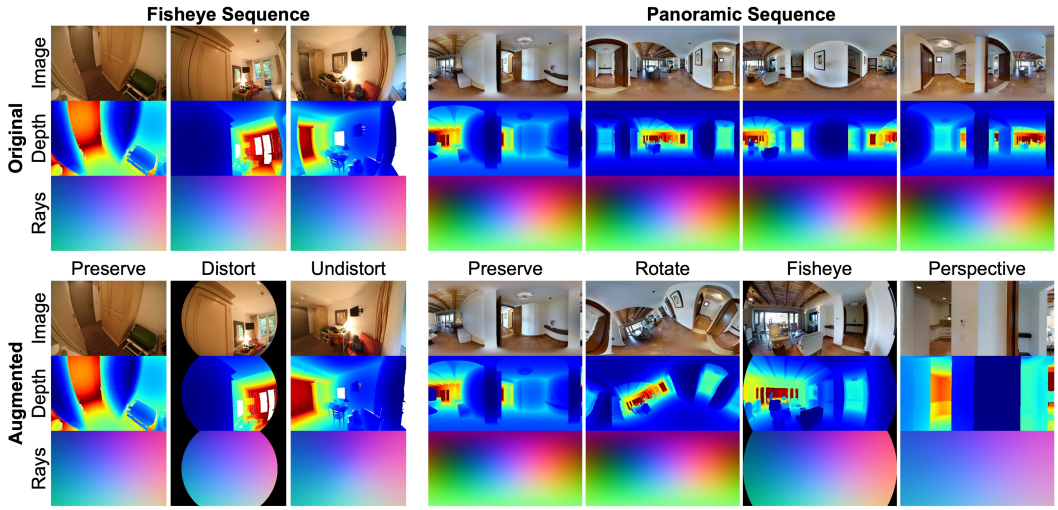


Figure 3: **Cross-camera data augmentation for model adaptation.** Multi-view 3D reconstruction data from fisheye and panoramic sequences are augmented by synthesizing diverse camera projections. For fisheye sequences, we randomly preserve the original views, create additional fisheye distortions, and undistort the image to obtain perspective views. For panoramic sequences, we randomly preserve the original views, apply rotations, and render fisheye or perspective images by sampling from the panoramic sphere. Dense ground truth quantities such as depth and ray maps are transformed together with the image, providing geometrically aligned supervision across various camera models.

adaptation design, incorporating a new camera sensor inevitably risks catastrophic forgetting of earlier knowledge on existing camera models. In contrast, as our framework freezes the underlying foundational backbone and encapsulates camera-specific geometric modulations into isolated LoRA experts, new camera models (e.g., catadioptric or microscopic) can be incorporated incrementally. Specifically, supporting a novel camera model requires only instantiating a new lightweight LoRA expert and extending the camera predictor by one classification category, without full retraining of the perspective backbone or existing adapters.

Post-hoc model compression. Different camera models exhibit different distortion characteristics and may require different levels of adaptation. Moreover, the adaptation effect of LoRA can vary substantially across layers depending on the role of each layer in the specific reconstruction model. Therefore, instead of uniformly assigning the same rank to every layer and camera expert or using expensive hyperparameter search to obtain the sufficient rank for each layer-expert pair, we apply a post-hoc adaptive rank truncation scheme that automatically allocates ranks to more important layer-expert pairs. Specifically, the camera adapters across all layers are initially trained with a moderately high rank, obtaining matrices (A_k^c, B_k^c) for each layer k and non-perspective camera c. Then, singular value decomposition (SVD) is performed on the learned update matrix as

$$B_k^c A_k^c = U_{k,c} \Sigma_{k,c} V_{k,c}^\top, \quad \forall k \in [K],\ \forall c \in \mathcal{C} \setminus \{c_{\text{persp}}\}, \tag{8}$$

where $U_{k,c} \in \mathbb{R}^{n_k \times r_k^c}$, $V_{k,c} \in \mathbb{R}^{m_k \times r_k^c}$, and $\Sigma_{k,c} = \text{Diag}(\boldsymbol{\sigma}_{k,c}) \in \mathbb{R}^{r_k^c \times r_k^c}$ is a diagonal matrix with a diagonal $\boldsymbol{\sigma}_{k,c} \in \mathbb{R}^{r_k^c}$ that contains the singular values in descending order. According to the Eckart-Young-Mirsky theorem [5], truncated SVD yields the optimal low-rank matrix approximation under unitarily invariant norms. Leveraging this theoretical guarantee, we apply a global threshold σ_0 across all layers and camera models to truncate singular values below it as

$$U'_{k,c} \Sigma'_{k,c} V'^\top_{k,c} = \left(U'_{k,c} \Sigma'^{1/2}_{k,c}\right) \left(\Sigma'^{1/2}_{k,c} V'^\top_{k,c}\right) = B'^c_k A'^c_k \tag{9}$$

where $U'_{k,c} \in \mathbb{R}^{n_k \times r'^c_k}$, $V'_{k,c} \in \mathbb{R}^{m_k \times r'^c_k}$, and $\Sigma'_{k,c} = \text{Diag}(\boldsymbol{\sigma}'_{k,c}) \in \mathbb{R}^{r'^c_k \times r'^c_k}$ are the truncated SVD matrices, $\boldsymbol{\sigma}'_{k,c} \in \mathbb{R}^{r'^c_k}$ contains the first r'^c_k singular values of $\boldsymbol{\sigma}_{k,c}$ that reach the threshold σ_0, r'^c_k is the truncated rank, and $B'^c_k \in \mathbb{R}^{n_k \times r'^c_k}$ and $A'^c_k \in \mathbb{R}^{r'^c_k \times m_k}$ are the truncated LoRA matrices. The global threshold σ_0 is chosen such that LoRA experts consume a predetermined parameter budget.

3.3 Learning

Data augmentation. As demonstrated in Figure 3, we adapt the models on multi-view 3D fisheye and panoramic datasets, and further augment them through cross-camera synthesis. It is unnecessary to introduce perspective datasets for training which can be synthesized from wide-FoV data. Specifically, from fisheye sequences, we randomly distort the images with varying distortion parameters to synthesize diverse distortions, and we randomly undistort the images to obtain perspective views to enable cross-camera learning [7, 8]. From panoramic sequences, we apply random rotation first and then render with both perspective and fisheye projections from the panoramic view. These transformations significantly enrich the diversity of camera distortions in the training set and provide multiple camera projections of the same underlying scene, facilitating effective learning of camera-dependent features and improving the robustness of the camera predictor.

Model training. The camera classifier is trained beforehand with cross-entropy loss based on the ground-truth camera labels. The individual LoRA experts are firstly trained with fully fisheye and panoramic sequences separately. Global attention LoRA parameters are subsequently finetuned on the mixed-camera sequences. Adaptation is first performed with a rank of 64 for all layers, after which post-hoc compression is applied and the LoRA matrices are compressed to consume only $\epsilon = 0.5\%$ of backbone parameters. For LoRA modules, we use their ground-truth camera labels for routing to train the weight matrices, and the training objective is simply the 3D reconstruction loss used by the chosen backbone model. During inference, the router is replaced by the camera predictor we trained to deal with unknown cameras.

4 Experiments

4.1 Settings

Datasets. We train models on three fisheye datasets (ASE [6], ScanNet++ [32], and KITTI360 [20]) and four panoramic datasets (Matterport3D [2], Gibson [31], TartanAirV2 [28], and PanoCity [9]). The models are evaluated on four camera settings. Perspective: undistorted ADT [22] and ETH3D [23]. Fisheye: ADT [22] and ScanNet++ [32]. Panoramic: Stanford2D3DS [1] and 360Loc [12]. Heterogeneous: KITTI360 [20] with mixed perspective and fisheye images, and Stanford2D3DS [1] and 360Loc [12] with mixed perspective, fisheye, and panoramic images.

Models. We test the effectiveness of adaptation on three foundational backbone models: VGGT [26], MapAnything [15], and π^3 [29]. We use the latest MapAnything v1.1 and π^3-X checkpoints.

Baselines. We compare our performance with previous foundational reconstruction models for non-perspective images, including Wid3R [13], Fisheye3R [4], PanoVGGT [9] (with a low-resolution version and a high-resolution one denoted as PanoVGGT-H), and Argus [19]. We additionally compare with calibration tokens (CT) introduced in [4] as a baseline of adaptation algorithm, applied in the same models and trained with the same data and loss for an apples-to-apples comparison.

Metrics. We evaluate the models from four different tasks: camera pose, depth map, point map, and FoV map estimation, following [4, 15, 26, 27, 29]. For camera pose, we adopt the Relative Rotation Accuracy (RRA), Relative Translation Accuracy (RTA), and Area Under Curve (AUC) of the accuracy-threshold curve at 5° for angular accuracy and Absolute Trajectory Error (ATE), Relative Pose Error for translation (RPEt), and Relative Pose Error for rotation (RPEr) for distance error quantification. For depth map estimation, Absolute Relative Error (Rel), Root Mean Squared Error (RMSE), and δ_1 accuracy are selected as metrics. For point map estimation, Accuracy (Acc), Completeness (Comp), Chamfer Distance (CD), and Normal Consistency (NC) are chosen. For FoV map evaluation, the median error for horizontal (hErr) and vertical (vErr) FoVs in degrees, along with the Recall (Rec) at a 1° threshold for FoV are measured. The arrow $\uparrow$ behind a metric means the higher the better, while $\downarrow$ means the lower the better.

4.2 Results

Camera router training and classification accuracy. The camera model predictor is implemented as a lightweight convolutional neural network. With our training setup, the classifier converges in a few minutes and achieves 100% classification accuracy on our test sets. This observation suggests that the distortion characteristics of the three camera models are sufficiently distinct from their images.

	Dataset/Method	Pose (Angular)			Pose (Distance)			Depth			Point			FoV		
		RRA↑	RTA↑	AUC↑	ATE↓	RPEt↓	RPEr↓	Rel↓	RMSE↓	δ_1↑	Acc↓	Comp↓	NC↑	hErr↓	vErr↓	Rec↑
ADT	Wid3R	0.411	0.403	0.077	0.111	0.192	7.153	0.088	0.468	0.943	0.136	0.115	0.703	5.852	2.267	0.120
	Fisheye3R	0.941	0.486	0.218	0.111	0.166	3.195	0.083	0.355	0.950	0.052	0.049	0.747	1.889	1.907	0.115
	VGGT	0.392	0.273	0.058	0.191	0.274	7.515	0.199	0.787	0.741	0.233	0.148	0.628	5.992	6.068	0.012
	+CT	0.984	0.720	0.452	0.061	0.106	2.592	0.071	0.337	0.952	0.046	0.041	0.741	1.441	0.655	0.458
	+Ours	0.985	0.802	0.502	0.047	0.093	**2.401**	0.069	0.331	0.954	0.039	0.040	0.752	1.330	0.913	0.472
	MapAnything	0.466	0.275	0.057	0.192	0.287	6.928	0.122	0.436	0.909	0.087	0.087	0.718	5.263	5.256	0.019
	+CT	0.982	0.620	0.354	0.080	0.143	2.960	0.085	0.354	0.949	0.046	0.042	0.746	1.308	0.942	0.451
	+Ours	0.986	0.679	0.407	0.065	0.121	2.634	0.083	0.347	0.950	0.041	0.041	0.750	**1.274**	0.762	0.519
	π^3	0.540	0.248	0.061	0.200	0.275	6.216	0.106	0.425	0.925	0.099	0.086	0.739	5.248	5.205	0.015
	+CT	0.971	0.732	0.437	0.054	0.103	2.957	0.082	0.345	0.953	0.042	0.045	0.747	1.498	1.359	0.392
	+Ours	**0.990**	**0.860**	**0.610**	**0.034**	**0.078**	2.473	**0.061**	**0.317**	**0.956**	**0.032**	**0.039**	**0.763**	1.276	**0.531**	**0.600**
ScanNet++	Wid3R	0.995	0.964	0.846	0.036	0.062	1.396	0.035	0.244	0.977	0.025	0.016	0.850	0.587	0.689	0.870
	Fisheye3R	0.997	0.879	0.627	0.054	0.087	1.987	0.047	0.200	0.975	0.034	0.016	0.803	0.917	0.886	0.517
	VGGT	0.190	0.154	0.013	0.334	0.572	11.971	0.128	0.371	0.857	0.129	0.072	0.734	11.714	9.419	0.011
	+CT	0.983	0.775	0.469	0.084	0.131	2.636	0.053	0.218	0.973	0.046	0.018	0.784	0.849	0.823	0.523
	+Ours	0.996	0.955	0.830	0.037	0.058	1.386	0.036	**0.191**	0.978	0.022	0.013	0.839	0.349	0.381	0.927
	MapAnything	0.379	0.305	0.058	0.237	0.397	8.604	0.092	0.273	0.946	0.087	0.046	0.762	5.777	4.629	0.104
	+CT	0.995	0.829	0.559	0.065	0.102	2.265	0.052	0.212	0.973	0.040	0.018	0.799	0.867	0.873	0.516
	+Ours	**0.998**	0.901	0.702	0.048	0.072	1.668	0.043	0.204	0.976	0.026	0.015	0.829	0.512	0.501	0.841
	π^3	0.269	0.198	0.024	0.263	0.455	9.330	0.083	0.260	0.957	0.091	0.042	0.779	8.611	6.720	0.050
	+CT	0.995	0.889	0.664	0.054	0.084	1.865	0.052	0.217	0.975	0.043	0.018	0.791	0.913	0.824	0.533
	+Ours	0.997	**0.968**	**0.870**	**0.030**	**0.048**	**1.282**	**0.034**	**0.191**	**0.979**	**0.019**	**0.012**	**0.851**	**0.254**	**0.324**	**0.974**

Table 1: Quantitative evaluation on fisheye sequences.

	Dataset/Method	Pose (Angular)			Pose (Distance)			Depth			Point		
		RRA↑	RTA↑	AUC↑	ATE↓	RPEt↓	RPEr↓	Rel↓	RMSE↓	δ_1↑	Acc↓	Comp↓	NC↑
Stanford2D3DS	Wid3R	0.751	0.720	0.517	1.258	1.654	21.361	0.063	0.621	0.969	0.216	0.227	0.747
	Argus	0.839	0.795	0.635	0.918	0.856	12.049	0.067	0.505	0.966	0.196	0.299	0.773
	VGGT	0.082	0.052	0.004	3.844	4.623	81.735	0.470	1.612	0.381	0.535	2.182	0.529
	+CT	0.457	0.396	0.120	1.497	1.874	19.382	0.090	0.494	0.954	0.278	0.265	0.693
	+Ours	0.958	0.909	0.781	0.206	0.200	1.985	0.050	0.382	0.981	0.072	0.064	0.862
	MapAnything	0.085	0.040	0.002	4.398	4.242	81.574	0.681	4.916	0.250	0.466	3.418	0.507
	+CT	0.764	0.702	0.340	0.682	0.743	6.915	0.101	0.535	0.938	0.170	0.115	0.717
	+Ours	0.976	0.921	0.730	0.148	0.205	2.429	0.055	0.425	0.978	0.069	**0.045**	0.842
	π^3	0.076	0.037	0.001	4.365	4.166	87.080	0.457	1.621	0.385	0.559	2.460	0.516
	+CT	0.793	0.764	0.432	0.492	0.600	6.925	0.067	0.424	0.971	0.136	0.097	0.780
	+Ours	**0.978**	**0.945**	**0.857**	**0.105**	**0.144**	**1.364**	**0.043**	**0.377**	**0.982**	**0.048**	0.053	**0.888**
360Loc	PanoVGGT	0.453	0.385	0.090	1.117	1.279	7.649	0.297	3.634	0.700	0.651	0.349	0.689
	PanoVGGT-H	0.997	0.996	0.933	0.087	0.118	0.459	0.259	3.484	0.739	0.418	0.332	0.765
	Argus	0.900	0.866	0.548	0.890	1.140	3.642	0.290	4.982	0.711	0.842	0.763	0.677
	VGGT	0.261	0.089	0.010	4.735	6.193	38.487	0.532	5.313	0.290	2.165	5.092	0.560
	+CT	0.878	0.694	0.321	0.574	0.852	2.517	0.327	4.021	0.720	0.682	0.332	0.684
	+Ours	**1.000**	0.998	0.939	0.086	0.123	0.413	**0.234**	**3.396**	**0.811**	0.348	0.258	0.768
	MapAnything	0.224	0.048	0.006	5.632	7.345	48.683	0.688	13.669	0.239	2.202	6.454	0.550
	+CT	0.945	0.902	0.531	0.302	0.446	1.818	0.377	4.049	0.691	0.654	0.312	0.644
	+Ours	0.998	0.992	0.860	0.126	0.170	0.759	0.297	3.931	0.739	0.503	0.309	0.734
	π^3	0.283	0.053	0.006	5.122	6.734	38.235	0.480	4.734	0.370	2.102	5.327	0.551
	+CT	0.980	0.982	0.694	0.185	0.286	1.371	0.333	3.634	0.742	0.472	**0.240**	0.711
	+Ours	**1.000**	**0.999**	**0.969**	**0.068**	**0.099**	**0.354**	0.272	3.752	0.790	**0.338**	0.265	**0.781**

Table 2: Quantitative evaluation on panoramic sequences.

Quantitative results. We compare with different methods on fisheye and panoramic sequences in Tables 1 and 2 and mixed-camera sequences in Table 3. Results across all these datasets and evaluation metrics demonstrate that our approach significantly beats prior ones by a large margin, with the strongest performance on π^3 backbone. As an adaptation method, our method can be applied to all the three backbones, extending all of them to non-perspective images, with a much better effect than calibration tokens. Table 4 compares several design choices for handling inputs with heterogeneous camera models. The notation for different LoRA expert assignments is defined in the table caption. Compared with the original model, both full finetuning and retraining degrade performance on perspective inputs, as the original backbone weights are no longer preserved. Among the LoRA variants, assigning an adaptation expert to perspective images, as in 3PFP and 1PFP, also leads to worse perspective performance, likely because learning a new perspective expert from a relatively small dataset causes forgetting of the knowledge acquired by the backbone during large-

Dataset	Method	Pose (Angular)			Pose (Distance)			Depth			Point		
		RRA↑	RTA↑	AUC↑	ATE↓	RPEt↓	RPEr↓	Rel↓	RMSE↓	δ_1↑	Acc↓	Comp↓	NC↑
KITTI360	Wid3R	0.255	0.039	0.007	11.218	17.582	114.694	0.486	8.498	0.306	0.692	5.937	0.683
	π^3	0.200	0.082	0.010	6.034	11.387	34.416	0.227	6.173	0.626	1.068	1.881	0.690
	+Ours	**1.000**	**0.881**	**0.753**	**0.249**	**0.428**	**0.909**	**0.084**	**3.970**	**0.933**	**0.326**	**0.327**	**0.795**
S2D3DS	Wid3R	0.260	0.191	0.035	2.912	3.374	31.397	0.364	1.416	0.309	0.455	0.446	0.623
	π^3	0.138	0.103	0.015	3.814	3.865	55.577	0.424	1.449	0.408	0.519	0.929	0.601
	+Ours	**0.833**	**0.767**	**0.525**	**1.022**	**1.202**	**11.347**	**0.075**	**0.469**	**0.954**	**0.170**	**0.160**	**0.787**
360Loc	Wid3R	0.500	0.243	0.097	4.452	5.457	9.092	0.414	7.390	0.285	1.110	0.750	0.627
	π^3	0.283	0.105	0.020	6.122	6.813	27.750	0.578	7.108	0.301	1.586	2.471	0.575
	+Ours	**0.995**	**0.960**	**0.806**	**0.220**	**0.266**	**1.098**	**0.286**	**4.713**	**0.815**	**0.469**	**0.372**	**0.720**

Table 3: Quantitative evaluation on mixed-camera sequences.

Method	Perspective				Fisheye				Panoramic				Mixed			
	ADT		ETH3D		ADT		ScanNet++		S2D3DS		360Loc		S2D3DS		360Loc	
	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑	CD↓	NC↑
Original	**0.036**	**0.790**	**0.052**	**0.907**	0.089	0.741	0.071	0.788	1.598	0.526	3.598	0.568	0.724	0.601	2.028	0.575
Finetune	0.042	0.754	0.076	0.855	0.040	0.751	0.020	0.820	0.448	0.535	1.662	0.496	0.347	0.650	1.251	0.569
Retrain	0.748	0.553	1.458	0.552	0.240	0.635	0.578	0.550	1.520	0.492	4.953	0.521	1.620	0.480	5.100	0.500
LoRA (3PFP)	0.037	0.783	0.063	0.874	0.036	0.755	**0.017**	0.835	**0.050**	0.863	0.543	0.734	0.215	0.771	0.446	0.706
LoRA (1PFP)	0.040	0.757	0.085	0.847	0.037	0.755	0.018	0.827	0.064	0.843	0.372	0.729	0.204	0.762	0.425	0.699
LoRA (1FP)	**0.036**	**0.790**	**0.052**	**0.907**	**0.035**	0.752	0.018	0.834	0.063	0.837	**0.344**	0.739	0.178	0.770	**0.415**	0.707
LoRA (2FP)	**0.036**	**0.790**	**0.052**	**0.907**	**0.035**	**0.756**	**0.017**	**0.842**	0.055	**0.864**	0.373	**0.756**	**0.172**	0.771	0.423	**0.712**
LoRA (2FP) Compressed	**0.036**	**0.790**	**0.052**	**0.907**	0.036	0.750	0.019	0.818	0.064	0.845	0.393	0.740	0.179	**0.772**	0.435	0.701

Table 4: Quantitative point cloud evaluation with π^3 backbone. **3PFP**: 3 experts for perspective, fisheye, and panoramic images, respectively. **1PFP**: 1 expert for all images. **1FP**: 1 expert for only fisheye and panoramic images. **2FP**: 2 experts for fisheye and panoramic images, respectively.

scale pretraining. In contrast, removing LoRA expert for perspective images, as in 1FP and 2FP, perfectly preserves the original perspective performance by routing perspective inputs through the frozen backbone. Compared to 1FP, 2FP generally achieves superior performance in fully fisheye and panoramic images, indicating that separating adaptation of different cameras to different experts is a better design. After rank truncation, the adapter is approximately compressed by 22 times and the performance drops only slightly, presenting an efficiency-performance trade-off.

Qualitative results. Qualitative figures of how our method adapts models to fisheye and panoramic sequences are presented in Figures 5 to 8. Figure 4 illustrates two scenarios of applying our approach to π^3 for mixed cameras, with the predicted point clouds before and after adaptation. In the KITTI360 example, two fisheye cameras are mounted back-to-back and connected through a front-facing perspective camera. Without support for mixed-camera inputs, a fisheye specialist model would struggle in this setting due to the lack of covisibility between the two fisheye views. In contrast, the perspective view provides an intermediate observation that links the two fisheye views, enabling a coherent reconstruction of the entire scene. In the 360Loc example, our model is also able to accurately reconstruct the scene given three completely distinct camera models in the inputs.

Limitations. Our method assumes that each input image belongs to a predefined set of camera categories, currently including perspective, fisheye, and panoramic. Thus when an image from a fourth camera model is encountered, the classifier has to incrementally learn the new camera type before correctly routing the images to a new expert. Additionally, our training employs cross-camera synthesis from fisheye and panoramic datasets. Although these augmentations provide diverse projection geometries, synthetic distortion may not fully capture real lens- and sensor-specific effects.

5 Conclusion

We present a parameter-efficient low-rank adaptation framework to extend 3D feed-forward reconstruction models pretrained on large-scale perspective datasets to non-perspective, potentially unknown, and mixed-camera sequences. By freezing the original reconstruction backbone and introducing lightweight learnable camera-specific LoRA experts for fisheye and panoramic images that are selected by a camera router, we separate the adaptation effect on different camera models, enabling the network to handle hybrid inputs while preserving backward compatibility with perspective data. This design allows existing large-scale reconstruction models to be reused with minimal additional parameters and training cost, rather than requiring model-specific retraining or separate networks for

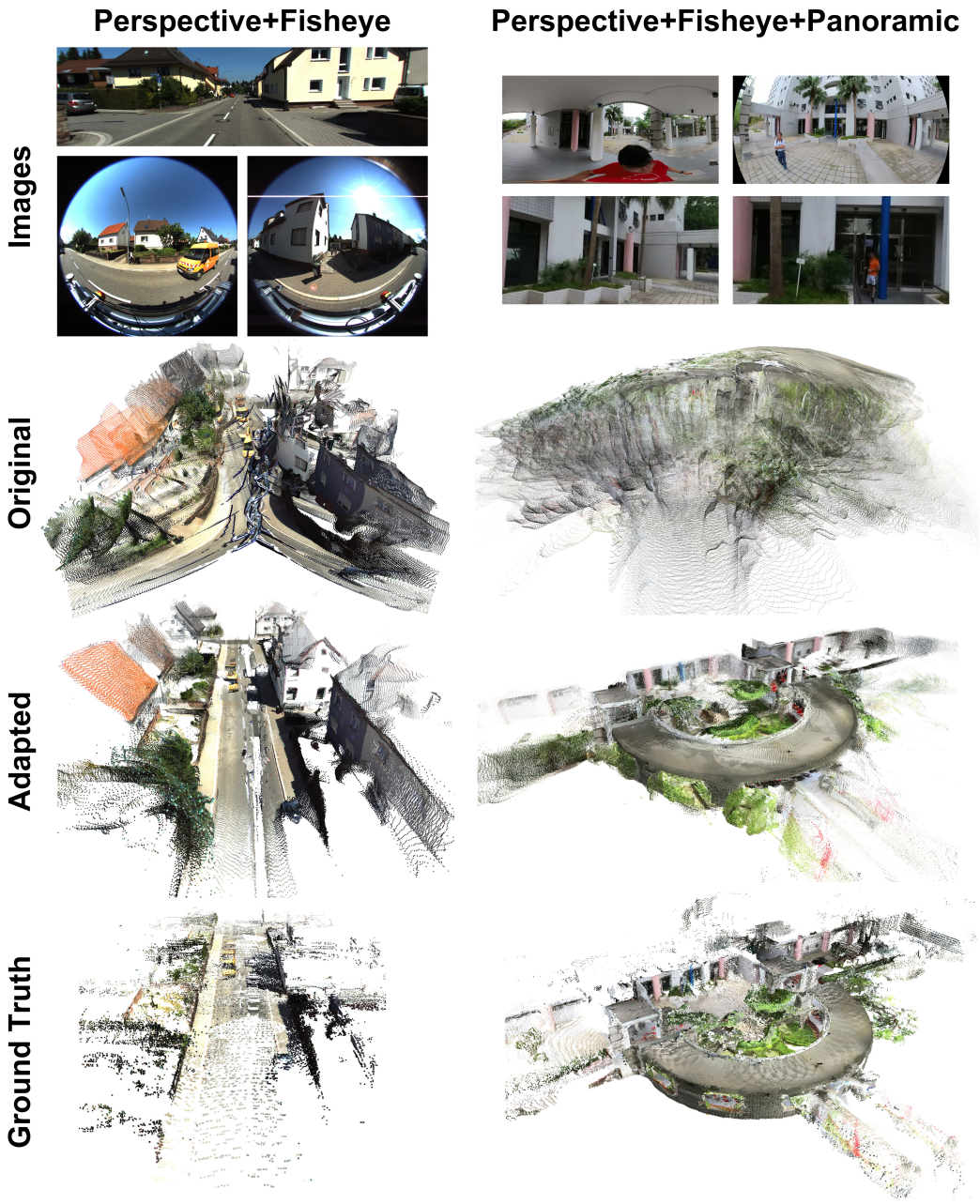


Figure 4: Qualitative reconstructions on mixed-camera sequences.

each camera configuration. We demonstrate that the framework supports mixed-camera sequences, where images from different projection models jointly contribute to a coherent reconstruction. Across three foundational backbones, our method achieves state-of-the-art performance on 3D scene reconstruction with non-perspective images, while maintaining the original model behavior on perspective inputs. We hope this work provides a practical step toward deploying feed-forward 3D reconstruction models in real-world systems with heterogeneous and evolving camera setups.

Acknowledgments

This work is supported by Google Gift Funding and NSF-2112562 Athena AI Institute. Alex Wong was supported by the grant SFI-AN-NC-SCN-00007276 from the Simons Foundation International as part of the SCENE collaboration. The authors have no competing interests to disclose.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[2] 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. *arXiv preprint arXiv:1709.06158*, 2017.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] 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 *European Conference on Computer Vision*, pages 1–19. Springer, 2026.

[5] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1 (3):211–218, 1936.

[6] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

[7] Suchisrit Gangopadhyay and Alex Wong. From perspective to fisheye depth estimation and open-vocabulary segmentation. In *European Conference on Computer Vision*. Springer, 2026.

[8] 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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5198–5209. IEEE, 2025.

[9] 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. *arXiv preprint arXiv:2603.17571*, 2026.

[10] Namitha Guruprasad, Abhay Yadav, Rama Chellappa, and Cheng Peng. Cam3r: Camera-agnostic model for 3d reconstruction. In *European Conference on Computer Vision*, pages 247–264. Springer, 2026.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[12] 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*, pages 22314–22324, 2024.

[13] 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 *European Conference on Computer Vision*, pages 665–684. Springer, 2026.

[14] 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.

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

[16] Vishnuprasadh Kumaravelu, Sunil Gupta, and PK Srijith. Post-optimization adaptive rank allocation for lora. *arXiv preprint arXiv:2604.27796*, 2026.

[17] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[18] Wenbing Li, Zikai Song, Hang Zhou, Yunyao Zhang, Junqing Yu, and Wei Yang. Lora-mixer: Coordinate modular lora experts through serial attention routing. *arXiv preprint arXiv:2507.00029*, 2025.

[19] Xi Li, Linyuan Li, Yan Wu, Tong Rao, Kai Zhang, Xinchen Hui, and Cihui Pan. Argus: Metric panoramic 3d reconstruction for indoor scenes. In *European Conference on Computer Vision*, pages 628–647. Springer, 2026.

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

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[22] 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*, pages 20133–20143, 2023.

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

[24] Ivan Vulić, Adam Grycner, Quentin de Laroussilhe, and Jonas Pfeiffer. Lora-squeeze: Simple and effective post-tuning and in-tuning compression of lora modules. *arXiv preprint arXiv:2602.10993*, 2026.

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

[26] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[27] 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*, pages 20697–20709, 2024.

[28] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[29] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pages 10481–10497, 2026.

[30] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

[31] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018.

[32] 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*, pages 12–22, 2023.

[33] Dacao Zhang, Kun Zhang, Shimao Chu, Le Wu, Xin Li, and Si Wei. More: A mixture of low-rank experts for adaptive multi-task learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1311–1324, 2025.

[34] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

A Training and Evaluation Details

Split	Dataset	Camera	Domain	Environment
Train	ScanNet++ [32]	Fisheye	Real	Indoor
	ASE [6]	Fisheye	Synthetic	Indoor
	KITTI360 [20]	Fisheye&Perspective	Real	Outdoor
	Matterport3D [2]	Panoramic	Real	Indoor
	Gibson [31]	Panoramic	Real	Indoor
	TartanAirV2 [28]	Panoramic	Synthetic	Indoor&Outdoor
	PanoCity [9]	Panoramic	Synthetic	Outdoor
Test	ScanNet++ [32]	Fisheye	Real	Indoor
	ADT [22]	Fisheye	Real&Synthetic	Indoor
	KITTI360 [20]	Fisheye&Perspective	Real	Outdoor
	360Loc [12]	Panoramic	Real	Indoor&Outdoor
	Stanford2D3DS [1]	Panoramic	Real	Indoor

Table 5: Non-perspective datasets used in training and evaluation.

Datasets and train-test split. We list the dataset information in Table 5. We train and evaluate the models on perspective, fisheye, and panoramic cameras, spanning real and synthetic, indoor and outdoor environments. For the two datasets that appear in both training and testing, ScanNet++ follows the split in [15], while KITTI360 follows the split in [4], both of which are enforced at a scene level, thus no sequence from the test set is revealed to the model during training.

Hyperparameters. For adaptation, we initially train with a uniform rank of 64 and then truncate LoRA to $\epsilon = 0.5\%$ of the original backbone's parameters.

Hardware. Training and testing are performed on 8 NVIDIA RTX 6000 Blackwell (48 GB) GPUs.

Training time. The entire training process takes approximately 20 hours for each model.

Data processing. During training, each GPU is given a maximum of 16 images, with each image resized to have a maximum edge of 518 pixels. With a sequence length randomly sampled as $V \in [2, 16]$, the number of sequences per GPU is $\lfloor 16/V \rfloor$, leading to an effective batch size of $8\lfloor 16/V \rfloor$ sequences per iteration. Evaluation is conducted on a fixed sequence length of $V = 16$.

Photometric augmentations. Random color jittering, gray-scale conversion, and Gaussian blur are independently applied to each image.

Distortion sampling. For fisheye distortion synthesis, we follow [4] and adopt the Kannala-Brandt camera model [14] with randomly sampled distortion coefficients (k_1, k_2, k_3, k_4), where $k_1, k_2, k_3 \sim U(-0.5, 0.5)$ and $k_4 \sim U(-0.05, 0.05)$. This augmentation generates diverse fisheye distortion patterns from existing training images. For images that are originally fisheye, we retain the original distortion with probability 0.5 and replace it with a newly sampled distortion with probability 0.5. For panoramic image augmentation, we always apply a random yaw rotation sampled from $(-180°, 180°)$. Additionally, with a probability of 0.5, we independently apply random pitch and roll rotations, each sampled uniformly from $(-15°, 15°)$.

Camera model sampling. For camera model sampling during training, if a sequence is originally fisheye, it has a probability of 0.5 to remain entirely fisheye and a probability of 0.5 to be converted into mixed camera types, where each frame is independently sampled as either perspective or fisheye with equal probability. If a sequence is originally panoramic, it has a probability of 0.5 to remain entirely panoramic and a probability of 0.5 to be converted into mixed camera types, where each frame is independently sampled from perspective, fisheye, and panoramic cameras with uniform probabilities. During evaluation, KITTI360 uses the dataset's original perspective and fisheye images with a sampling ratio of $1:1$. For panoramic datasets, we additionally generate perspective and fisheye views for testing, and sample the three camera types uniformly with a ratio of $1:1:1$.

Frame sampling scheme. A pairwise covisibility matrix is precomputed for each scene and only views with at least 25% pixels visible from another sampled view are sampled for both training and evaluation, following [15].

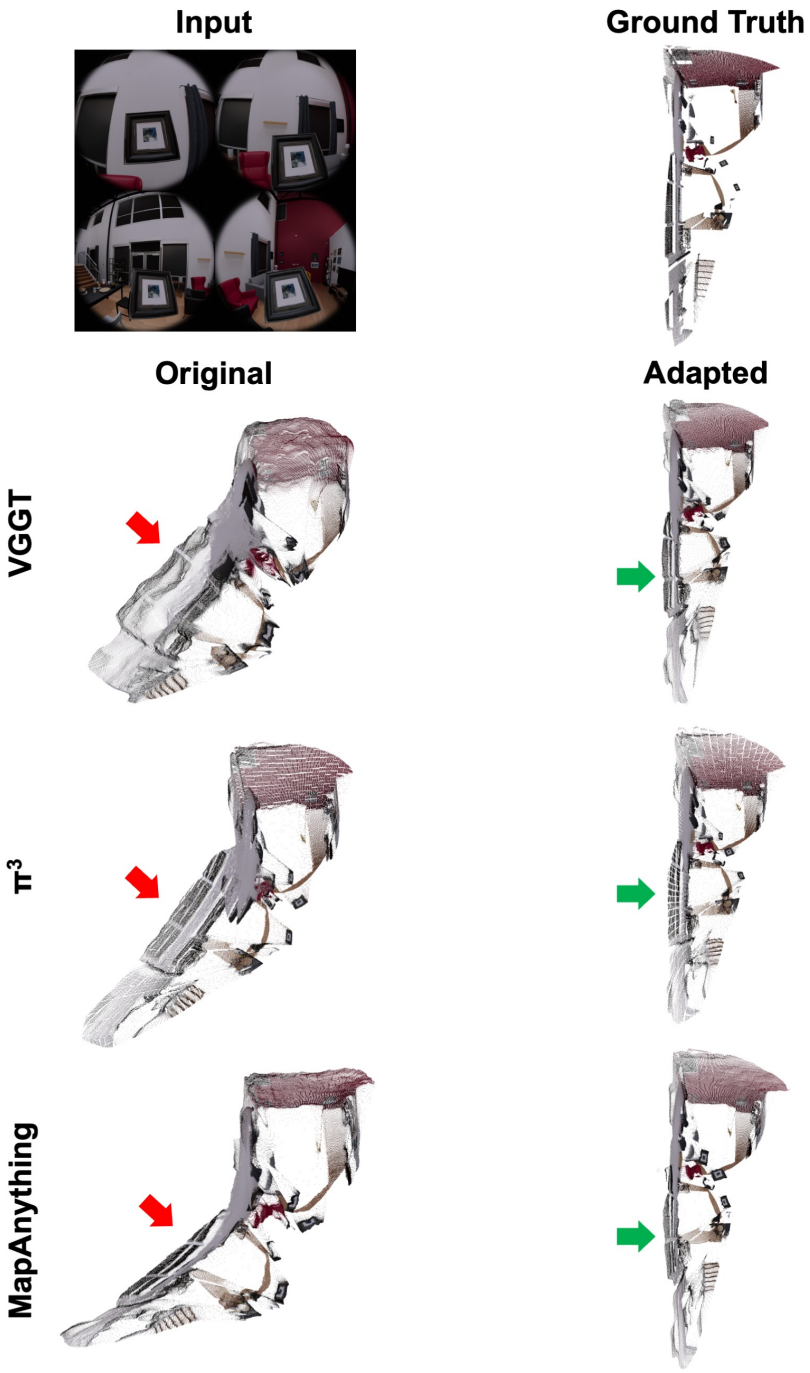


Figure 5: Qualitative reconstructions on ADT.

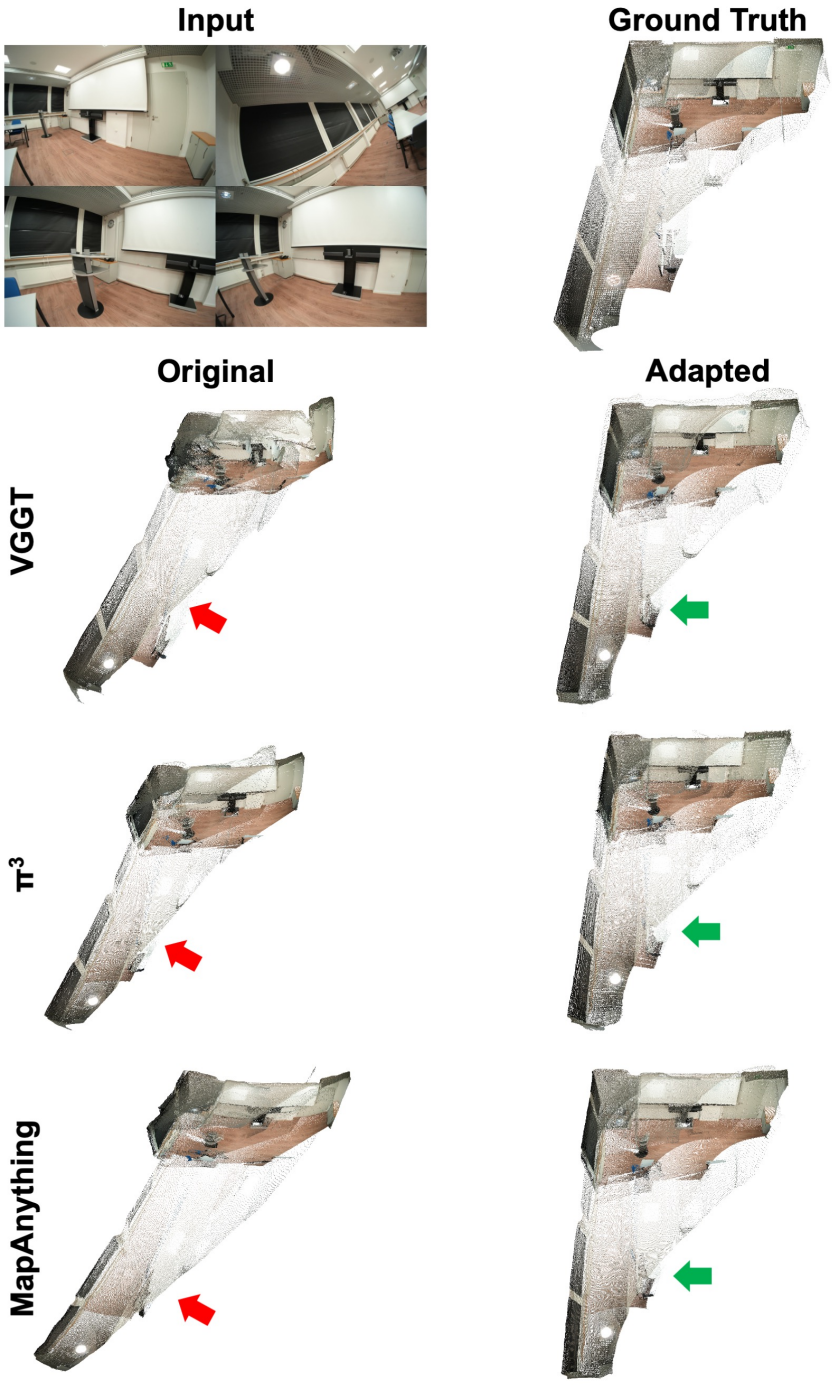


Figure 6: Qualitative reconstructions on ScanNet++.

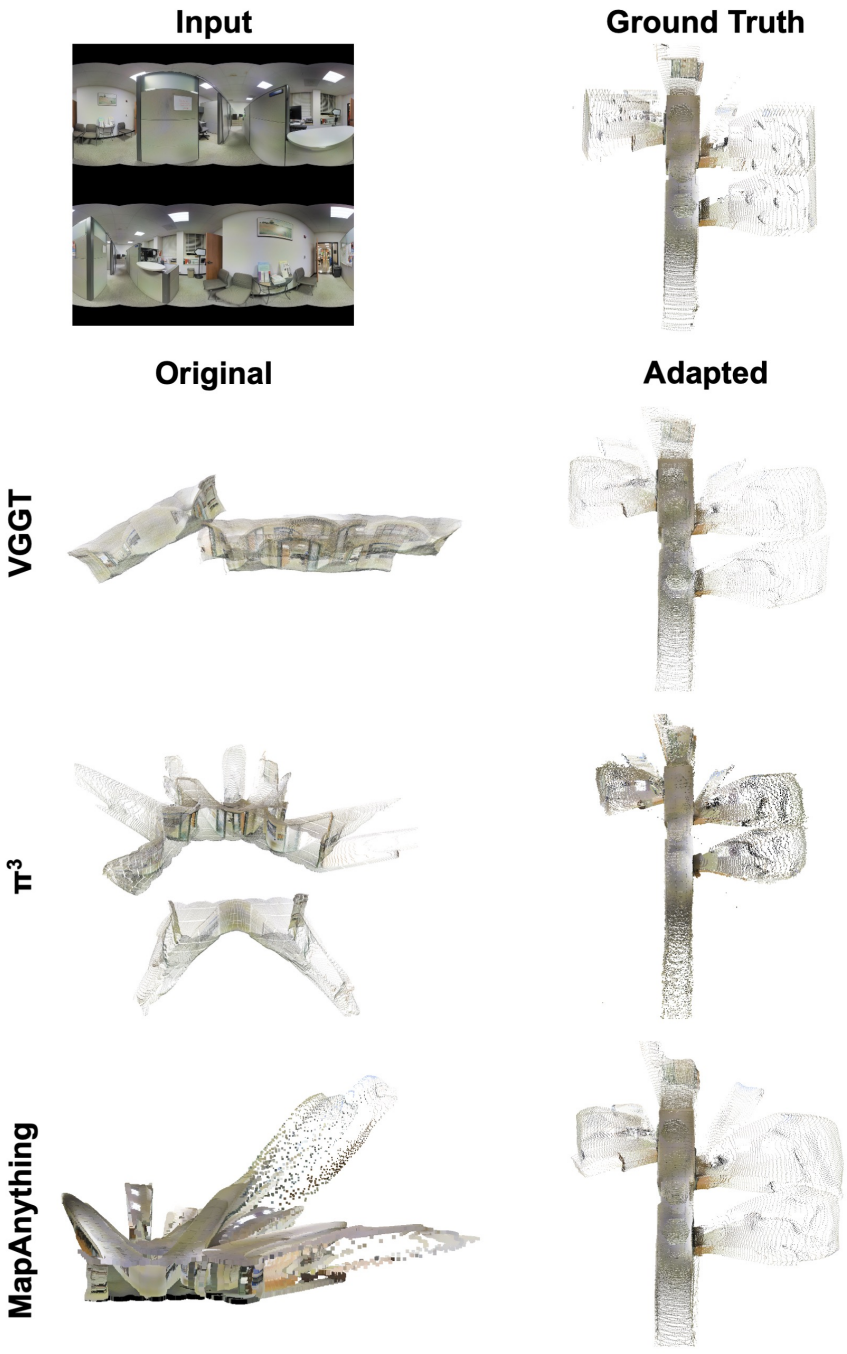


Figure 7: Qualitative reconstructions on Stanford2D3DS.

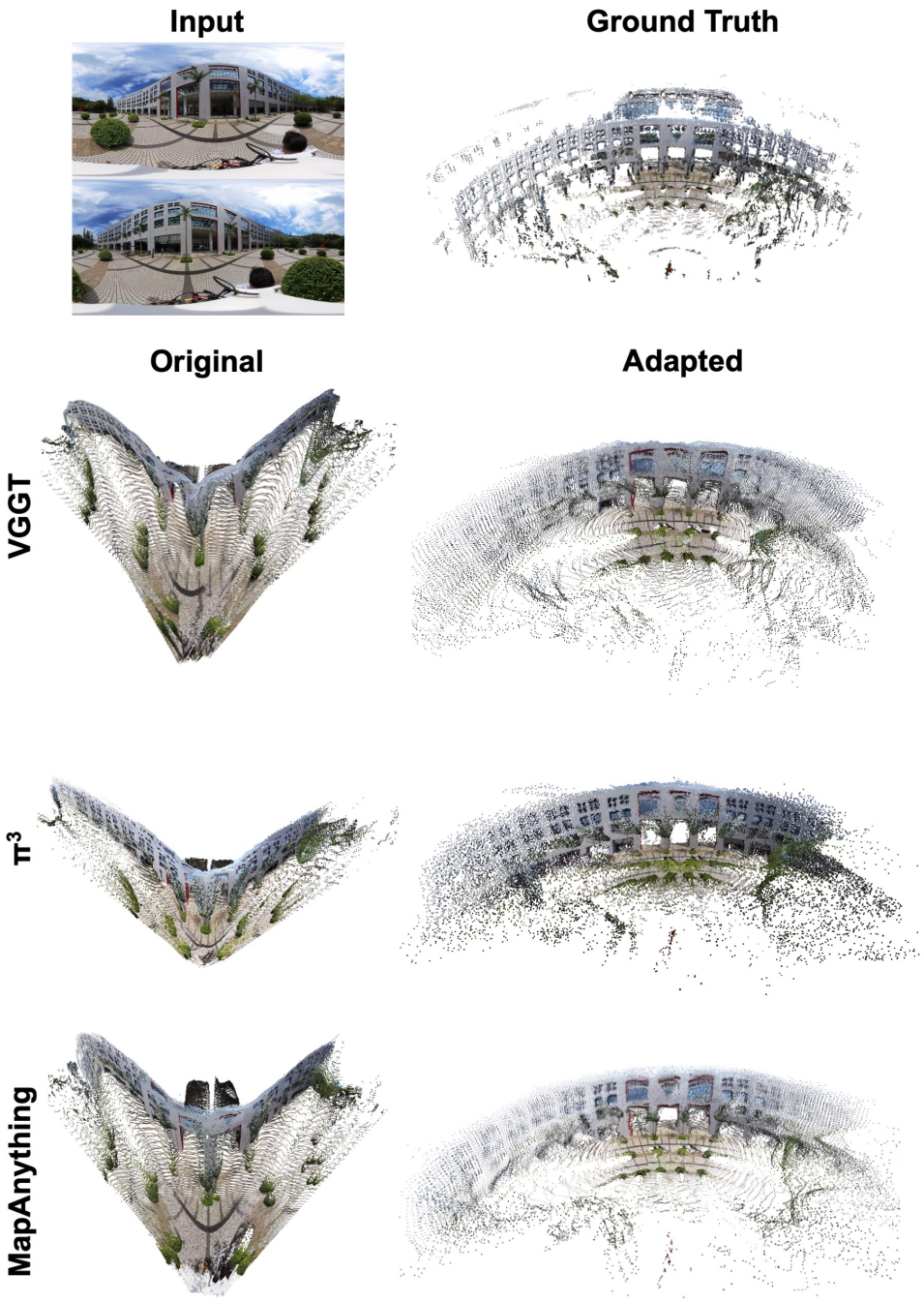


Figure 8: Qualitative reconstructions on 360Loc.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect the paper's contributions and scope to extend perspective-pretrained feed-forward multi-view 3D reconstruction foundational models to non-pinhole cameras.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has discussed the limitations of our approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper has no theoretical results that require mathematical assumptions or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide implementation details such as datasets, model hyperparameters, training settings, evaluation protocols, etc., for reproduction of the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The datasets used in the paper are all public open-sourced data which can be found online. We plan to release the entire codebase upon paper acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We have specified the training and test details in the paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not provided since training multi-view feed-forward 3D reconstruction models is computationally expensive.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported the GPU information, memory, and time of our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work studies multi-view 3D reconstruction from unconstrained images with unknown camera models, which has potential positive impacts in applications such as robotics, augmented/virtual reality, autonomous driving, etc. However, the improved reconstruction capabilities may also introduce negative societal impacts, e.g., the reconstruction models could be used for surveillance, privacy invasion, or unauthorized reconstruction of private environments.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release models or datasets with high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The existing assets used in the paper have been properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We have not yet released the code or checkpoints, but detailed documentation will be provided once they are published.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: **[TODO]**

    Guidelines: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used in the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.