Rig3R: Rig-Aware Conditioning for Learned 3D Reconstruction

Samuel Li[*1,2] **Pujith Kachana**[*1,2] **Prajwal Chidananda**[1] **Saurabh Nair**[1]
Yasutaka Furukawa[1] **Matthew Brown**[1]

[1] Wayve Technologies [2] Carnegie Mellon University

Abstract

Estimating agent pose and 3D scene structure from multi-camera rigs is a central task in embodied AI applications such as autonomous driving. Recent learned approaches such as DUSt3R have shown impressive performance in multiview settings. However, these models treat images as unstructured collections, limiting effectiveness in scenarios where frames are captured from synchronized rigs with known or inferable structure. To this end, we introduce **Rig3R**, a generalization of prior multiview reconstruction models that incorporates rig structure when available, and learns to infer it when not. Rig3R conditions on optional rig metadata including camera IDs, timestamp, and rig calibrations to develop a rig-aware latent space that remains robust to missing information. It jointly predicts global pointmaps and two types of raymaps: a pose raymap relative to a global frame, and a rig raymap relative to a rig-centric frame consistent across time. The global pose raymaps allow the model to reason about the agent's ego-motion, while the rig raymaps allow the model to infer rig structure directly from input images when metadata is missing. Rig3R achieves state-of-the-art performance in 3D reconstruction, camera pose estimation, and rig discovery—outperforming both traditional and learned methods by 17–45% mAA across diverse real-world rig datasets, all in a single forward pass without post-processing or iterative refinement.

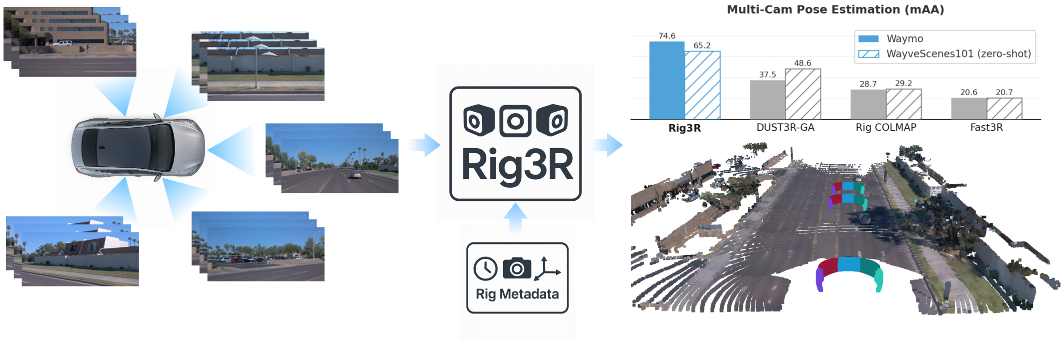


Figure 1: Rig3R is the first learned 3D vision model to leverage rig constraints when available, and the first method to support rig calibration discovery from unordered images when they are not—achieving strong 3D consistency and performance across diverse settings and rig configurations.

[*] Equal contribution, order decided by coin toss. Work done while at Wayve.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Multi-view scene estimation of camera poses and 3D structure from images is a core capability in computer vision and has enabled spatial understanding for embodied agents, robotic systems, and large-scale visual localization [1]. Accurate estimation of structure and motion is essential for tasks such as simultaneous localization and mapping (SLAM)[2, 3], scene relocalization and view synthesis applications [4–7]. Traditional pipelines based on Structure-from-Motion (SfM) and Multi-View Stereo (MVS) reconstruct scenes by optimizing for geometry via matched image features [8–10]. While effective in controlled settings, these methods are brittle in the presence of dynamic objects, visual repetition, or feature poor environments, and often require careful tuning.

Recent learned methods such as DUSt3R [11] have shown impressive capabilities in multiview 3D reconstruction, with many successors [12, 13] extending this to single-pass inference. A limitation, however, is that these works treat images as unstructured collections. This overlooks a key structural prior common in real-world applications: images are often captured using synchronized multi-camera rigs with fixed relative configurations. Rig metadata—such as camera ID, timestamp, and relative poses—can provide valuable cues, especially when field-of-view overlap is limited, as is often the case in rig setups. While classical pipelines can exploit this structure [14–16], feedforward models currently leave it untapped.

We introduce **Rig3R**, a transformer-based model for multiview 3D reconstruction and pose estimation that leverages rig metadata when available and learns to infer rig structure when it is not. Rig3R handles unstructured image sets, calibrated rigs, and everything in between, predicting dense pointmaps and raymaps for each image in a single forward pass. These raymaps spatially encode camera intrinsics and extrinsics, which can be recovered in closed form—even in ambiguous regions such as sky or dynamic pixels. To enable this flexibility, Rig3R combines metadata embeddings with dropout training, and includes a dedicated rig prediction head that infers rig structure directly from image content when metadata is unavailable. This allows Rig3R to produce accurate reconstructions, even in cases with minimal view overlap.

Our key contributions are:

- The first learned method that leverages rig constraints to improve 3D reconstruction and pose estimation, while generalizing to inputs with partial or missing metadata (e.g., camera ID, timestamp, rig poses).
- A novel output representation based on global and rig-relative raymaps, enabling closed-form pose estimation and rig structure discovery from unordered image inputs.
- Extensive experiments across diverse real-world driving datasets show that Rig3R achieves state-of-the-art performance in 3D reconstruction, camera pose estimation, and rig discovery, outperforming both traditional and learned methods, all in a single forward pass.

2 Related Works

Multi-View 3D Reconstruction. Classical pipelines follow a two-stage paradigm: Structure-from-Motion (SfM) for sparse pose and point recovery, followed by Multi-View Stereo (MVS) for densification. Systems like COLMAP [8, 17, 18] rely on feature matching, triangulation, and bundle adjustment, but remain sensitive to occlusion, motion, and low-texture regions [1]. Early learning-based methods improved robustness by introducing learned features and matching [19–23]. Photometric representations such as NeRFs [4, 24, 25] and Gaussian splats [5, 26] reconstruct scenes via view synthesis but typically require accurate camera poses. More recent approaches bypass both explicit feature matching and known poses, predicting 3D structure directly from RGB images [27–30]. DUSt3R [11] pioneered pointmap regression from single image pairs without known poses, with follow-up works addressing multi-frame input [31–33], dynamic scenes [33–35], and downstream tasks [36–38]. MV-DUSt3R [13] introduces multi-frame attention, while Fast3R [12] scales to hundreds of views with global consistency. VGGT [39] jointly predicts depth, pose, and structure using a transformer backbone. Pow3R [40] improves flexibility through lightweight conditioning on inputs such as intrinsics, relative pose, or depth. While these models support efficient scene understanding, they treat input views as unordered. Rig3R builds on this single-pass design, conditioning on rig metadata and enabling structure discovery even in the absence of such priors.

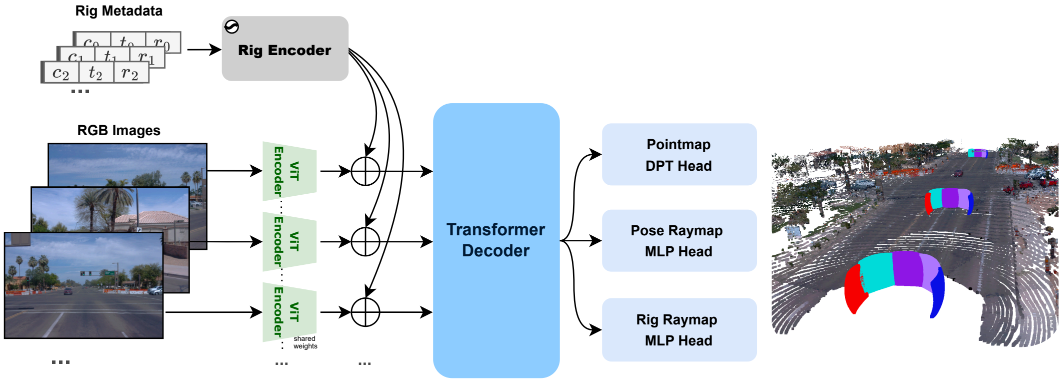


Figure 2: Rig3R jointly predicts pointmaps, global raymaps, and rig-relative raymaps, with dropout conditioning on rig embeddings. Global raymaps are color-coded by the discovered rig structure.

Camera Pose Estimation. Traditional pose estimation relies on geometric solvers such as PnP with RANSAC [41, 42] and global optimization via bundle adjustment [9], but performance is brittle under occlusion, dynamic motion, or sparse correspondences. Learned methods such as PoseNet [43] regress 6-DoF poses directly from images [44–46], while unsupervised approaches [47] optimize photometric losses to jointly estimate depth and ego-motion. Systems like DROID-SLAM [48] combine differentiable updates with learned features for increased robustness, and several related method adopt a learned SLAM approach [49–51]. Several 3D reconstruction models, including those discussed above, also infer poses alongside or through 3D structure [11, 52, 39, 12, 30, 13, 53]. Rig3R extends this trend by predicting dense raymaps that encode per-pixel directions and camera centers, enabling closed-form recovery of intrinsics and extrinsics while enforcing multiview consistency.

Rig-Aware Multi-View Geometry. Rig constraints can provide strong geometric cues for multi-view reconstruction, enabling more accurate pose estimation and improved robustness in low-overlap or ambiguous settings. Classical works have leveraged such constraints in various ways. COLMAP [8] incorporates rig structure by modeling the rig as a single moving entity, jointly optimizing global poses while keeping intra-rig calibration fixed through bundle adjustment. Kaess and Dellaert [14] introduced a probabilistic SLAM framework for multi-camera rigs that models cross-camera feature associations under motion. Carrera et al. [16] proposed a SLAM-based method for fully automatic extrinsic calibration of multi-camera systems, even with non-overlapping fields of view. Heng et al. [15] developed an infrastructure-based calibration method using image-based localization and prebuilt maps, requiring no manual intervention. Rig3R takes a markedly different approach from these prior works, providing rig information as optional embeddings on the input, enabling accurate and generalizable 3D reconstruction across both structured and unstructured multi-camera configurations.

3 Rig-aware 3D Reconstruction

We address the task of predicting 3D structure and camera poses from a set of N RGB images $\{I_i\}_{i=1}^N$, where $I_i \in \mathbb{R}^{3\times H\times W}$. Inputs may range from unordered image collections to temporally distributed views captured by multi-camera rigs. Each image may optionally include metadata $M_i = \{c_i, t_i, r_i\}$, where c_i is a camera ID, t_i is a timestamp, and r_i is a rig-relative raymap encoding the camera pose. Each metadata field is optional and may be independently omitted during training and inference.

Given inputs $\{I_i, M_i\}_{i=1}^N$, the model predicts for each image: a pointmap $P_i \in \mathbb{R}^{3\times H\times W}$ representing per-pixel 3D coordinates in the first image's frame; a confidence map $C_i \in \mathbb{R}^{H\times W}$ used to weight the pointmap loss; a pose raymap $R_i^{\text{pose}} \in \mathbb{R}^{H\times W\times 6}$ encoding camera parameters relative to the first image's frame; and a rig raymap $R_i^{\text{rig}} \in \mathbb{R}^{H\times W\times 6}$ encoding camera parameters relative to a rig-centric frame, decoupled from ego-motion. Together, these outputs form Rig3R's predictions:

$$\text{Rig3R} : \{I_i, M_i\}_{i=1}^N \rightarrow \{P_i, C_i, R_i^{\text{pose}}, R_i^{\text{rig}}\}_{i=1}^N$$

The following sections detail our raymap representation, metadata encoding, and model architecture.

3.1 Raymap Representation

Rig3R outputs dense raymaps from both its pose and rig heads. A raymap is a directional field that assigns a unit ray direction to each pixel, with all rays originating from a shared camera center. This representation encodes both camera intrinsics and pose in a unified, geometrically consistent format.

For each pixel (u, v), the viewing ray $\hat{\mathbf{r}}_{uv} \in \mathbb{S}^2$ is computed as $\hat{\mathbf{r}}_{uv} = \mathbf{R} \cdot \mathbf{K}^{-1}[u, v, 1]^\top$, where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic matrix, $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation matrix, and the output is unit-normalized. All rays share a common camera center $\mathbf{c} \in \mathbb{R}^3$.

Raymaps offer key advantages over alternative representations. They provide spatially aligned, per-pixel supervision and serve as a stable signal even in ambiguous regions such as sky or dynamic objects. Unlike pointmaps, which infer pose indirectly through 3D predictions and often fail in such regions, raymaps offer a more direct and consistent representation for pose estimation. They also encode interpretable geometry, enabling closed-form recovery of camera intrinsics and extrinsics from ray directions and pixel distances. In our implementation, we recover focal lengths using angular constraints derived from pixel–ray correspondences, and estimate rotations in closed form using SVD from aligned camera and world rays [54]. See Section B of supplementary material for more details.

3.2 Rig-Aware Metadata

Each image may optionally be associated with a metadata tuple $M_i = \{c_i, t_i, r_i\}$, where c_i is a discrete camera identifier shared by all images from the same physical camera, $t_i \in \mathbb{R}$ is a continuous timestamp normalized in seconds, and $r_i \in \mathbb{R}^{H\times W\times 6}$ is a rig-relative raymap encoding the camera's pose within the rig. This metadata provides geometric and temporal context for multiview reasoning. The combination of camera ID and timestamp forms a structured decomposition of frame identity, offering strong cues for spatiotemporal alignment. All metadata fields are optional and can be independently dropped during training to encourage robustness to missing information.

3.3 Model Architecture

Image Encoder. Rig3R (see Fig. 2) employs a shared ViT-Large encoder [55, 56] to independently patchify and encode each input image using 2D sine-cosine positional encodings. We initialize from DUSt3R [11], though other works [39, 13, 12] indicate that performance is not sensitive to this choice.

Metadata Embedding. Each patch is optionally augmented with rig-aware metadata: (1) frame index N, (2) camera ID c_i, (3) timestamp t_i, and (4) rig raymap patch r_i. The discrete IDs N and c_i are randomly sampled from a larger index range and encoded using 1D sine-cosine embeddings, following [12], enabling generalization to varying numbers of frames and cameras. The timestamp $t_i \in \mathbb{R}$ is normalized in seconds and encoded similarly. The rig raymap patch $r_i \in \mathbb{R}^6$ is linearly projected to the model dimension. All components are concatenated and added to the patch tokens. During training, c_i, t_i, and r_i are randomly dropped out to promote robustness, while the frame index N is always included to uniquely identify each image within the transformer.

Transformer Decoder. Patch tokens from all images are passed to a second ViT-Large transformer, trained from scratch, that performs joint self-attention across the full set. This enables Rig3R to aggregate information across views and time, conditioned on metadata when available. Unlike the shared image encoder, the decoder fuses multiview features in a shared latent space.

Prediction Heads. Rig3R uses three multitask heads: one for pointmap prediction and two for raymaps (pose-relative and rig-relative), with shared weights across frames. The pointmap head is a DPT module [57] that predicts a 3D pointmap $P_i \in \mathbb{R}^{3\times H\times W}$ and a confidence map $C_i \in \mathbb{R}^{H\times W}$. Each raymap head consists of two MLPs: one predicts per-pixel ray directions, and the other predicts a global camera center via average pooling over patch tokens. This design avoids dedicated query tokens and ensures all gradients flow through the patch tokens, promoting coherence. The prediction head design follows the nature of the output: pointmaps capture fine geometric detail and benefit from the multi-scale DPT head, while raymaps encode global pose information and are efficiently modeled with lightweight MLPs.

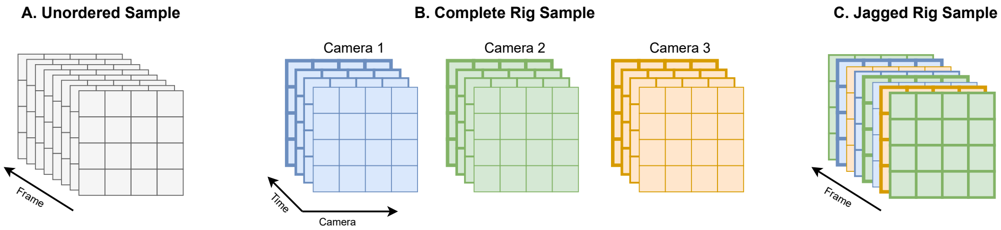


Figure 3: (A) Unordered samples treat all frames as independent, ignoring rig or temporal structure. (B) Complete rig samples use a fixed $C \times T$ grid, capturing all cameras at all timesteps. (C) Jagged rig samples, the approach used in Rig3R, unifies and generalizes strategies A and B. All frames are flattened into a single sequence, maintaining rig-awareness through metadata embeddings while naturally handling missing cameras or irregular timestamps.

These three outputs are tightly coupled: pointmaps are expected to lie along rays defined by the pose raymap, and rig and pose raymaps are related through ego-motion. This multitask formulation acts as a structural prior, improving consistency and generalization across diverse multiview settings.

3.4 Data Sampling

To enable Rig3R to reason about diverse rig configurations, the data sampling process plays a critical role. Unlike prior learned methods that treat multi-view inputs as completely unstructured collections of images as shown in Fig. 3A, Rig3R's design acknowledges that real-world embodied system often use synchronized multi-camera rigs with implicit spatial consistency. When introducing rig-aware conditioning, one sampling approach is to explicitly structure data as a $C \times T$ tensor, where C is the number of cameras and T is the number of temporal steps, allowing the model to learn from complete spatiotemporal grids as shown in Fig. 3B. However, this formulation restricts flexibility, as it assumes fixed rig configurations and uniform frame availability. Instead, Rig3R flattens all frames into a single sequence as demonstrated in Fig. 3C, preserving the unordered format of prior models while embedding each frame with its own rig metadata (camera ID, timestamp, and rig raymap). This design unifies rig information and unstructured sampling within a single representation, making it a truly general sampling choice; this is conceptually equivalent to learning over dense $C \times T$ configurations with stochastic dropout across cameras and timesteps. As a result, Rig3R receives a stronger and more varied learning signal, exposed to all possible sub-rigs and partial observations during training, keeping it flexible to missing views, variable rig geometries, and asynchronous capture setups.

3.5 Training

Training Losses. We train with a multitask loss over pointmaps, pose raymaps, and rig raymaps:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pmap}} + \lambda_{\text{p}} \mathcal{L}_{\text{p\_rmap}} + \lambda_{\text{r}} \mathcal{L}_{\text{r\_rmap}},$$

where λ_{p} and λ_{r} are weighting terms for the pose and rig raymap losses.

The pointmap loss $\mathcal{L}_{\text{pmap}}$, following [11], is a confidence-weighted regression objective [58, 59] with scale-normalized ground truth. For frame v, depth-normalized pointmap error is:

$$\mathcal{L}_{\text{pmap}} = \sum_{i \in \mathcal{D}^v} C_i^v \left\| X_i^v - \frac{1}{\bar{z}} \bar{X}_i^v \right\| - \alpha \log C_i^v,$$

where X_i^v is the predicted 3D point at pixel i, $\bar{X}_i^v$ is the ground truth, C_i^v is the predicted confidence, $\bar{z}$ is the average scene depth used for normalization, and α is the weight of the regularization term.

The raymap loss $\mathcal{L}_{\text{rmap}}$ includes terms for both ray directions and camera centers:

$$\mathcal{L}_{\text{rmap}} = \sum_{h,w} \|\mathbf{r}_{v,h,w} - \bar{\mathbf{r}}_{v,h,w}\| + \beta \left\| \mathbf{c}_v - \frac{1}{\bar{z}} \bar{\mathbf{c}}_v \right\|.$$

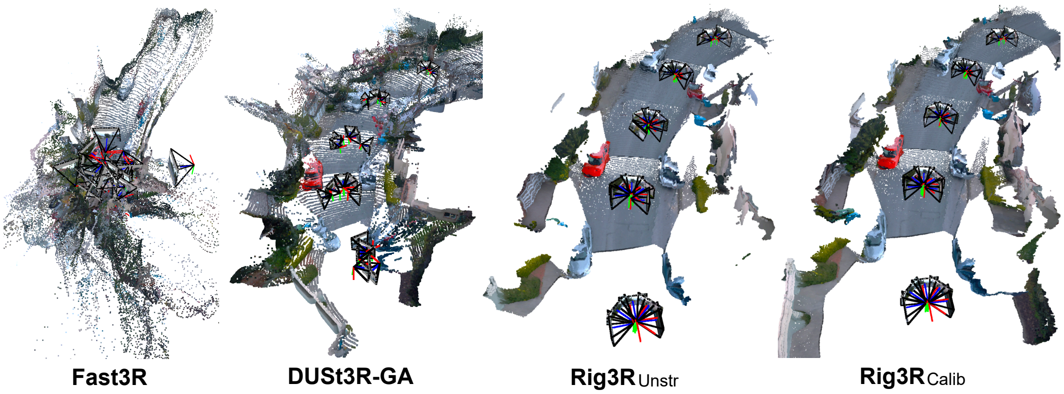


Figure 4: Qualitative results of baselines vs Rig3R with and without rig embeddings. Fast3R fails to find consistent structure and poses. DUSt3R with global alignment gets poses roughly correct, but with inconsistent rig geometry. Rig3R$_{\text{Unstr}}$ and Rig3R$_{\text{Calib}}$ show increasingly refined rig geometry.

Here, $\mathbf{r}_{v,h,w}$ is the predicted unit ray direction at pixel (h, w), and $\mathbf{c}_v$ is the predicted camera center for frame v. The ground-truth ray direction and camera center are denoted by $\bar{\mathbf{r}}_{v,h,w}$ and $\bar{\mathbf{c}}_v$, respectively. The average scene depth $\bar{z}$ is used for scale normalization, and β weights the center loss term. Following [39], we find training is more stable when the model learns scale and direction norms directly—especially for camera centers near the origin.

Training Data. We train Rig3R on a diverse data mix: CO3D-v2 [60], BlendedMVS [61], Map-free [62], ScanNet++ v2 [63], MVImgNet [64], PointOdyssey [65], Virtual KITTI2 [66], TartanAir V2 [67], PandaSet [68], KITTI [69], Argoverse2 [70], nuScenes [71], Waymo [72], and an internal dataset. We process relevant driving datasets following [73]. These cover a broad range of scene types—including indoor, driving, synthetic, and object-centric—with an emphasis on data from multi-camera rigs. For COLMAP datasets, we sample images based on covisibility. For others, we use a random stride within a specified range. In rig-based datasets, we subsample the rig cameras per sequence to increase diversity in rig configurations. Where available, the front-facing camera is always included to ensure overlap and reflect common monocular setups.

Embedding Dropout. To encourage metadata-aware reasoning and improve robustness, we randomly drop each metadata field (camera ID, timestamp, rig pose) with 50% probability during training. This structured masking teaches the model to leverage metadata when available, and to infer missing context from image content and cross-view relationships when it is not—enabling generalization to diverse input configurations at inference time.

Training Details. Rig3R is trained on 24-frame samples with a batch size of 128, using 128 H100 GPUs for 250k steps over 5 days. Images are resized to 512×512 with padding. We apply data augmentations including random per-frame color jitter, Gaussian blur, and centered aspect-ratio crops to simulate variation in focal length and image shape. During training, input sequences are randomly shuffled to vary the reference frame and promote generalization. We use the AdamW optimizer with a learning rate of 0.0001 and cosine annealing. While training uses a fixed 24-frame input for consistency and comparability across baselines, all frame IDs other than the first frame are randomly sampled from a shared ID pool from 1 to 100, enabling the model to generalize to sequences much longer than 24 frames.

4 Experiments

Evaluation Data. We evaluate Rig3R on the Waymo Open [72] validation set and WayveScenes101 [74], both featuring 5-camera rigs and approximately 200 timesteps per scene at 10 FPS under diverse real-world driving conditions. Waymo provides LiDAR-based ground-truth poses and 3D points, while WayveScenes101 uses COLMAP reconstructions. For each scene, we extract two 24-frame samples, each using the full 5-camera rig spaced approximately 2 seconds apart.

Method	Waymo					WayveScenes101 *(unseen)*					Time
	@15° ↑		@5° ↑		@30° ↑	@15° ↑		@5° ↑		@30° ↑	
	RRA	RTA	RRA	RTA	mAA	RRA	RTA	RRA	RTA	mAA	
COLMAP	31.1	24.4	23.0	22.5	22.7	34.1	26.0	28.1	23.0	24.4	>2m
MV-DUSt3R	44.3	23.8	18.9	8.4	15.8	40.0	27.0	13.6	9.5	13.1	10.5s
DUSt3R-GA	56.0	57.9	18.2	37.2	37.5	89.1	61.8	33.7	47.4	48.6	>2m
Fast3R	46.8	31.2	19.2	13.3	20.6	61.1	29.1	23.8	12.3	20.7	3.9s
Rig3R$_{\text{Unstr}}$	96.6	83.9	66.0	71.6	74.6	49.2	52.4	20.5	36.0	25.7	5.7s
Rig COLMAP	38.6	31.1	28.4	28.6	28.7	43.0	31.8	33.9	27.1	29.2	>2m
Rig3R$_{\text{Calib}}$	**99.4**	**91.6**	**67.4**	**77.4**	**82.1**	**95.8**	**75.8**	**77.7**	**60.0**	**65.2**	5.7s

Table 1: Multi-view pose estimation results, reporting RRA, RTA, and mAA at various precision levels. Intrinsics are withheld. **Cyan** and orange indicate the best and second-best results, respectively.

Method	Waymo			WayveScenes101 *(unseen)*		
	Acc. ↓	Comp. ↓	Chamfer ↓	Acc. ↓	Comp. ↓	Chamfer ↓
MV-DUSt3R	1.7	24.0	12.9	6.7	38.0	19.3
DUSt3R-GA	1.9	15.2	8.6	1.4	7.8	4.6
Fast3R	1.9	5.9	3.9	0.7	5.1	2.9
Rig3R$_{\text{Unstr}}$	0.2	1.4	0.8	0.4	8.2	4.3
Rig3R$_{\text{Calib}}$	**0.1**	**0.2**	**0.2**	**0.3**	**4.1**	**2.2**

Table 2: Multi-view pointmap estimation results. We report accuracy and completeness, and their average as the Chamfer distance. Intrinsics are withheld for all methods.

Baselines. We compare Rig3R to both learned and classical baselines for multi-view 3D reconstruction and pose estimation. MV-DUSt3R [13] and Fast3R [12] are architecturally similar to Rig3R, using transformer-based, feedforward multi-view inference. DUSt3R-GA [11] predicts stereo pointmaps refined via global optimization. As a classical baseline, we evaluate COLMAP [8] in both unstructured and rig-aware modes. While learned models operate per sample, we allow COLMAP to process full scenes for stronger global context. The COLMAP ground truth in WayveScenes101 is generated from full, dense sequences, whereas our evaluations use strided sequences with larger baseline shifts and sparser keypoint overlap. To ensure fairness and consistency, learned models operate on 24-frame chunks as in training, while COLMAP is permitted to process the entire strided sequence jointly, leveraging its inherent ability to optimize globally across frames. We evaluate two variants of Rig3R: Rig3R$_{\text{Unstr}}$, which receives no metadata and treats the sequence as unstructured, and Rig3R$_{\text{Calib}}$, which is given full rig metadata (camera ID, timestamp, and rig raymaps). Note that Rig3R$_{\text{Calib}}$ and Rig COLMAP are the only methods that leverage rig constraints, and both receive the same rig calibration information.

4.1 Camera Pose Estimation

We evaluate pose estimation using relative rotation and translation accuracy (RRA, RTA) at 15° and 5° thresholds, along with mean average accuracy (mAA) over thresholds up to 30° [75–77]. These metrics capture both coarse correctness and fine-grained precision. Full results are shown in Table 1.

On Waymo, Rig3R$_{\text{Calib}}$ achieves the best performance (82.1 mAA), and maintains high precision even at 5° thresholds. Rig3R$_{\text{Unstr}}$ ranks second overall, despite lacking rig metadata. Learned baselines exhibit sharp drops at 5°. COLMAP improves with rig constraints, and shows smaller differences between thresholds, consistent with classical optimization's binary convergence behavior. We also report wall-clock time per method to estimate poses. Qualitative results in Fig. 4 show Rig3R's improved spatial consistency—especially with embeddings—even under large spatial displacement.

We also evaluate on WayveScenes101, an unseen dataset with a novel rig configuration constructed from scenes where COLMAP reconstruction succeeded—potentially favoring classical methods. Despite this, Rig3R$_{\text{Calib}}$ achieves the best performance across all metrics, followed by DUSt3R-

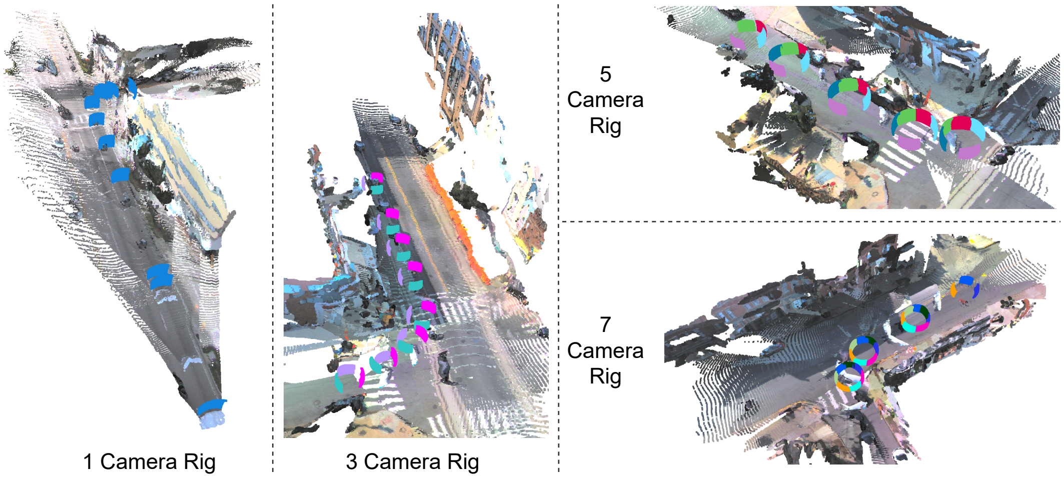


Figure 5: Qualitative results from Rig3R on 24 frames across diverse rig configurations, showing pointmaps and global pose raymaps color-coded by camera ID based on clustered rig raymap outputs.

GA, which benefits from global optimization. Rig3R$_{\text{Unstr}}$ remains competitive, outperforming all feedforward baselines and performing comparably to Rig COLMAP. Rig3R$_{\text{Unstr}}$ remains comparable to other baselines, displaying Rig3R's flexibile performance even when rig metadata is missing. Crucially, the large improvement in performance from Rig3R$_{\text{Unstr}}$ to Rig3R$_{\text{Calib}}$ indicates the value of leveraging rig metadata, suggesting it can enable stronger generalization.

These results confirm that Rig3R$_{\text{Calib}}$ achieves the strongest accuracy and precision across datasets, while Rig3R$_{\text{Unstr}}$ remains robust without rig metadata. Rig-aware embeddings provide a powerful mechanism for generalization and fine-grained pose estimation in diverse multi-view settings.

4.2 Pointmap Estimation

We evaluate 3D reconstruction quality using pointmap accuracy (Acc.), completeness (Comp.), and Chamfer distance (average of the two). Metrics are computed over sparse pointclouds by masking both predictions and ground-truth to valid regions (see Table 2).

On Waymo, Rig3R$_{\text{Calib}}$ achieves the lowest error across all metrics. Rig3R$_{\text{Unstr}}$ follows closely, significantly outperforming all other baselines and confirming the strength of Rig3R's pointmap predictions even without metadata. Fig. 4 highlights Rig3R's improvements over baselines in 3D reconstruction quality. In particular, with rig-aware embeddings, the model confidently reconstructs side-view cameras with minimal overlap, where geometric cues alone are often insufficient.

On WayveScenes101, Rig3R$_{\text{Calib}}$ again leads, demonstrating robust generalization. Interestingly, Fast3R achieves the second-best Chamfer distance, slightly outperforming Rig3R$_{\text{Unstr}}$ despite lower pose accuracy. This highlights a key advantage of Rig3R: it estimates pose directly from raymaps rather than pointmaps, enabling more consistent multi-view reasoning. See Section C of the supplementary material for further discussion on pose inference from raymaps versus pointmaps.

These results show that Rig3R learns robust scene structure with strong spatial precision and completeness, and that rig metadata further improves reconstruction quality and generalization.

4.3 Generalization Across Rig Configurations

We assess Rig3R's ability to generalize across rig configurations on the Argoverse2 [70] validation set. We subsample 1, 3, 5, and 7 camera rigs, increasing strides to maintain scene coverage (Fig. 5).

Rig Calibration Discovery. Figure 6a shows results for Rig3R$_{\text{Unstr}}$, which discovers rig calibrations directly from unordered images—without any rig metadata or assumptions about camera configuration. We evaluate two rig-specific metrics: rig ID accuracy and rig-relative pose mAA. Rig ID accuracy reflects how well frames from the same camera are grouped together; we compute it by first clustering the rig raymap outputs, and then evaluating frame assignment accuracy via the Hungarian algorithm.

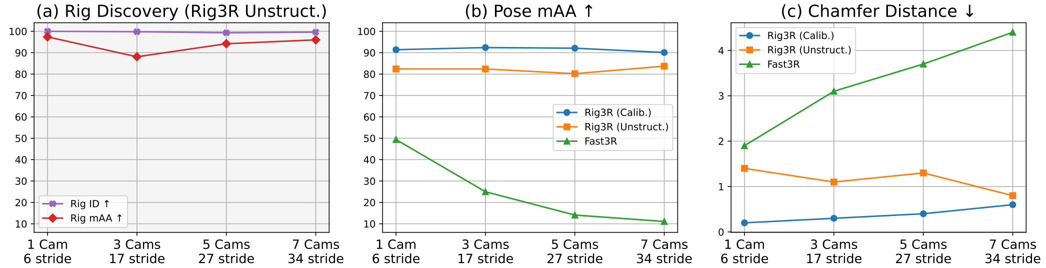


Figure 6: Rig3R generalizes across rig configurations. (a) Rig-relative pose mAA and Rig ID clustering accuracy with Rig3R$_{\text{Unstr}}$. (b) Global pose mAA and (c) Chamfer distance for Rig3R$_{\text{Calib}}$, Rig3R$_{\text{Unstr}}$, and Fast3R. Both Rig3R variants achieve consistent high performance across all settings.

Rig mAA then measures the quality of these predicted clusters, evaluating how accurate the relative orientations and positions of the discovered cameras are in a rig-centeric frame.

Rig3R$_{\text{Unstr}}$ achieves strong rig mAA and Rig ID accuracy across all configurations. Performance remains strong even as spatial layout becomes more complex, confirming Rig3R's ability to discover diverse rig structures without supervision. To the best of our knowledge, this is the first attempt, learned or classical, to address rig discovery with unordered images and no timestamps. Notably, Rig3R also handles monocular and unordered inputs by predicting identity rig raymaps and producing a single cluster at inference—correctly signaling the absence of a rig. Fig. 5 visualizes various predicted rig configurations, with camera clusters color-coded across time.

Flexible Rig Performance. We evaluate performance across rig sizes using pose mAA and Chamfer distance (Fig. 6b–c) for Rig3R$_{\text{Calib}}$, Rig3R$_{\text{Unstr}}$, and Fast3R. Rig3R$_{\text{Calib}}$ performs best overall, maintaining high pose and reconstruction quality across all settings. Rig3R$_{\text{Unstr}}$ also performs strongly, despite receiving no metadata. In contrast, Fast3R degrades as rig size and stride increase—likely due to reduced image overlap, which challenges methods that rely solely on visual correspondence without rig context. These results demonstrate that Rig3R remains robust to spatial variation and generalizes well across diverse rig configurations, as can be seen in Fig. 5.

4.4 Ablation Studies

Metadata Embeddings. In Table 3, we ablate the contribution of each metadata field—camera ID, timestamp, and rig pose—on pose estimation across Waymo and WayveScenes101, which represent previously seen and unseen rig configurations, respectively.

On Waymo, we observe that camera ID and rig pose embeddings provide only modest gains over the unstructured baseline, likely because the rig structure is easily recognized in this familiar setting. In this context, timestamp proves particularly valuable, as it provides dynamic cues for localizing the rig over time. Since the model already reasons about the rig structure implicitly, temporal information becomes the most informative remaining signal.

On WayveScenes101, a previously unseen dataset with a novel rig configuration, we find that camera ID and timestamp embeddings offer only limited gains over the unstructured baseline. This is likely because neither field alone disambiguates the underlying rig layout: camera ID does not indicate motion over time, and timestamp alone does not reveal which camera captured each frame. Classical methods like COLMAP similarly require both to accurately infer rig structure and optimize poses. In contrast, rig pose embeddings provides a direct spatial signal about the novel rig configuration—crucial for reasoning under domain shift. With this input, Rig3R can recognize and adapt to unseen rig geometries, resulting in a substantial boost in generalization and performance (mAA improves from 25.7 to 56.4). These results highlight the unique role of rig pose metadata in enabling generalization to unseen capture setups.

Across datasets, we find that timestamp embeddings primarily improve translation accuracy, as they provide strong cues about velocity and motion magnitude, while rig pose embeddings more strongly influence rotation accuracy, reinforcing consistent relative orientations between cameras within each timestamp. Providing all metadata yields the best performance across both datasets, confirming that

Cam	Time	Rig	Waymo				WayveScenes101 *(unseen)*			
			@15° ↑		@30° ↑	↓	@15° ↑		@30° ↑	↓
			RRA	RTA	mAA	Chamfer	RRA	RTA	mAA	Chamfer
			96.6	83.9	74.6	0.8	49.2	52.4	25.7	4.3
✓			97.0	84.3	75.1	1.1	48.0	55.4	26.8	4.5
	✓		97.6	**92.7**	81.8	0.3	36.2	60.2	23.9	4.6
		✓	98.0	84.2	76.0	1.2	**96.5**	66.5	56.4	2.5
✓	✓	✓	**99.4**	91.6	**82.1**	**0.2**	95.8	**75.8**	**65.2**	**2.2**

Table 3: Ablation of input metadata (cam ID, time, rig pose) on pose and pointmap estimation results.

Variant	Unstructured					Calibrated				
	@15° ↑		@5° ↑		@30° ↑	@15° ↑		@5° ↑		@30° ↑
	RRA	RTA	RRA	RTA	mAA	RRA	RTA	RRA	RTA	mAA
$\mathcal{L}_{\text{pose}}$	89.2	54.1	59.6	46.7	45.8	**98.6**	89.0	63.6	79.7	78.9
$\mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{rig}}$	90.4	56.6	60.2	49.3	48.5	98.5	**91.7**	64.8	**84.1**	**81.9**
$\mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{pmap}}$	**91.3**	**62.1**	**61.7**	**52.5**	**53.5**	98.3	90.1	**66.6**	79.8	79.8

Table 4: Model ablation of pointmap ($\mathcal{L}_{\text{pmap}}$) and rig ($\mathcal{L}_{\text{rig}}$) heads on the pose raymap head ($\mathcal{L}_{\text{pose}}$).

spatial calibration, temporal cues, and view identity are complementary. These results show that Rig3R generalizes well from partial metadata and fully benefits from rig calibration when available.

Multi-task Learning. We evaluate the impact of each auxiliary head by training three Rig3R variants: with only the pose raymap head ($\mathcal{L}_{\text{pose}}$), with pose + rig raymap ($\mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{rig}}$), and with pose + pointmap ($\mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{pmap}}$). As this ablation requires training separate models from scratch, we perform it at reduced scale (batch size 32, 5 datasets) for computational efficiency; results are not directly comparable to full-scale evaluations. As shown in Table 4, both auxiliary heads individually improve performance in the unstructured setting. The pointmap head provides the largest gain—raising mAA from 45.8 to 53.5—reflecting the value of 3D grounding. The rig head also improves results over using only the pose raymap head, suggesting that even without metadata, it helps maintain a coherent spatial layout. In the calibrated setting, all metrics are higher across the board, indicating some degree of performance saturation. Still, the rig head yields the highest mAA, likely by reinforcing the structure provided by the rig metadata across timesteps, followed by the pointmap head also helps in this setting. Despite the reduced scale of this setup, we observe a consistent pattern: both heads improve performance, and the rig head is particularly important when rig constraints are available. Importantly, the rig head also enables Rig3R's novel ability to perform rig discovery. For these reasons, we include both auxiliary heads in the full-scale Rig3R model.

5 Conclusion

We present Rig3R, a transformer-based model for multiview 3D reconstruction and pose estimation that introduces rig-aware conditioning and rig discovery. Rig3R is the first method to leverage rig metadata in a learned setting and the first to perform rig structure discovery from completely unconstrained image inputs. It jointly predicts pointmaps, global raymaps, and rig-relative raymaps in a single forward pass, achieving strong performance through its spatially-grounded representations.

Limitations and Future Work. Rig3R's main performance limitation is data diversity and quality, particularly regarding variety in rig configurations across existing datasets. One promising direction is to incorporate augmentations that simulate diverse rigs across a continuous configuration space. Future work may also explore balancing structured, rig-based temporal sampling with unordered, general sampling to improve generalization and adaptability across capture settings. While raymaps implicitly downweight dynamic content, explicitly modeling motion could further improve robustness in highly dynamic scenes. Overall, we see rig-aware embeddings as a powerful and generalizable cue, readily applicable to existing and future transformer-based models for multiview reasoning.

References

[1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[2] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103.

[3] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. 07 2014. doi: 10.15607/RSS.2014.X.007.

[4] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL `https://arxiv.org/abs/2003.08934`.

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. URL `https://arxiv.org/abs/2308.04079`.

[6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4), 2022. ISSN 1557-7368. doi: 10.1145/3528223.3530127. URL `http://dx.doi.org/10.1145/3528223.3530127`.

[7] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[8] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[11] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

[13] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*, 2024.

[14] Michael Kaess and Frank Dellaert. Probabilistic structure matching for visual slam with a multi-camera rig. *Computer Vision and Image Understanding*, 114(2):286–296, 2010.

[15] Lionel Heng, Paul Furgale, and Marc Pollefeys. Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. *Journal of Field Robotics*, 32(5):775–802, 2015.

[16] Gerardo Carrera, Adrien Angeli, and Andrew J. Davison. Slam-based automatic extrinsic calibration of a multi-camera rig. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2652–2659. IEEE, 2011.

[17] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited, 2024. URL `https://arxiv.org/abs/2407.20219`.

[18] Jiahao Li, Haochen Wang, Muhammad Zubair Irshad, Igor Vasiljevic, Matthew R. Walter, Vitor Campagnolo Guizilini, and Greg Shakhnarovich. Fastmap: Revisiting dense and scalable structure from motion, 2025. URL https://arxiv.org/abs/2505.04612.

[19] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. URL https://arxiv.org/abs/1712.07629.

[20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020. URL https://arxiv.org/abs/1911.11763.

[21] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the feature: Learning robust camera localization from pixels to pose, 2021. URL https://arxiv.org/abs/2103.09213.

[22] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021.

[23] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *arXiv preprint arXiv:2305.15404*, 2023.

[24] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections, 2021. URL https://arxiv.org/abs/2008.02268.

[25] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. Das3r: Dynamics-aware gaussian splatting for static scene reconstruction, 2024. URL https://arxiv.org/abs/2412.19584.

[26] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. *MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images*, page 370–386. Springer Nature Switzerland, October 2024. ISBN 9783031726644. doi: 10.1007/978-3-031-72664-4_21. URL http://dx.doi.org/10.1007/978-3-031-72664-4_21.

[27] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024.

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin ElNouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[29] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[30] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[31] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state, 2025. URL https://arxiv.org/abs/2501.12387.

[32] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory, 2024. URL https://arxiv.org/abs/2408.16061.

[33] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion, 2025. URL https://arxiv.org/abs/2410.03825.

[34] Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos, 2025. URL https://arxiv.org/abs/2412.09621.

[35] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training, 2025. URL https://arxiv.org/abs/2503.24391.

[36] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos, 2025. URL https://arxiv.org/abs/2412.09401.

[37] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. URL https://arxiv.org/abs/2406.09756.

[38] Riku Murai, Eric Dexheimer, and Andrew J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors, 2024. URL https://arxiv.org/abs/2412.12392.

[39] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[40] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors, 2025. URL https://arxiv.org/abs/2503.17316.

[41] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[42] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[43] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization, 2016. URL https://arxiv.org/abs/1505.07427.

[44] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion, 2024. URL https://arxiv.org/abs/2402.14817.

[45] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization, 2025. URL https://arxiv.org/abs/2412.08376.

[46] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2d images in 3d: Metric relative pose from metric correspondences, 2024. URL https://arxiv.org/abs/2404.06337.

[47] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video, 2017. URL https://arxiv.org/abs/1704.07813.

[48] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras, 2022. URL https://arxiv.org/abs/2108.10869.

[49] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[50] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[51] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[52] Vincent Leroy, Yohann Cabon, and Jérôme Révaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024.

[53] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024.

[54] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL `https://arxiv.org/abs/2010.11929`.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

[57] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021. URL `https://arxiv.org/abs/2103.13413`.

[58] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

[59] David Novotny, Diane Larlus, and Andrea Vedaldi. Capturing the geometry of object categories from video supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[60] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

[61] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[62] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022.

[63] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[64] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[65] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking, 2023. URL `https://arxiv.org/abs/2307.15055`.

[66] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. URL `https://arxiv.org/abs/2001.10773`.

[67] 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. 2020.

[68] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021.

[69] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012.

[70] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

[71] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[72] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020.

[73] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024.

[74] Jannik Zürn, Paul Gladkov, Sofía Dudas, Fergal Cotter, Sofi Toteva, Jamie Shotton, Vasiliki Simaiaki, and Nikhil Mohan. Wayvescenes101: A dataset and benchmark for novel view synthesis in autonomous driving, 2024. URL `https://arxiv.org/abs/2407.08280`.

[75] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment, 2024. URL `https://arxiv.org/abs/2306.15667`.

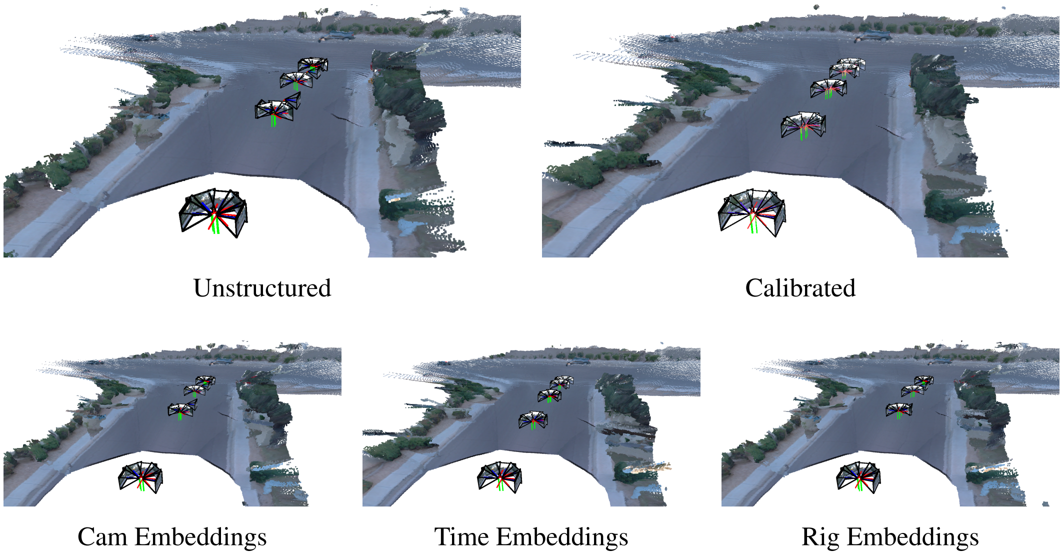
[76] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision (ECCV)*, pages 592–611. Springer, 2022.

[77] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

[78] George Terzakis and Manolis Lourakis. *A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem*, pages 478–494. 11 2020. ISBN 978-3-030-58451-1. doi: 10.1007/978-3-030-58452-8_28.

Appendix

The supplementary document provides A) additional visualization of our reconstructions; B) details of the pose estimation algorithm from raymaps; C) discussion on pose inference from raymaps versus pointmaps; and D) an additional experiment evaluating the robustness to calibration errors by injecting Gaussian noise.



Unstructured

Calibrated

Cam Embeddings

Time Embeddings

Rig Embeddings

Figure 7: Visualizations of the qualitative effects of rig metadata embeddings on the Waymo validation set. We observe that with added embeddings, the quality of the estimated poses noticeably improves, and the fine details of reconstructed scene are also better captured.

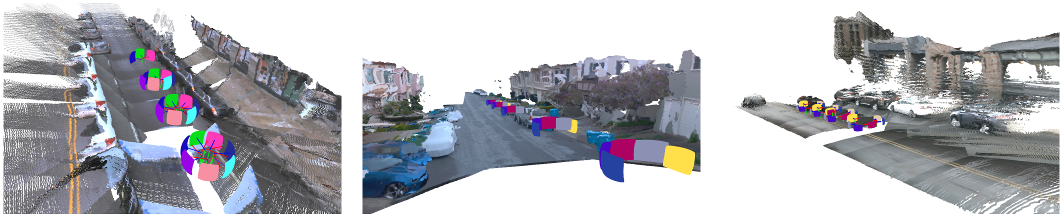


Figure 8: Rig scene reconstructions on Argoverse [70], Waymo [72], and nuScenes [71] validation sets.

A Additional Visualization

Figure 7 presents qualitative reconstruction results to illustrate the embedding ablations in Section 4.4 of the main paper. While the unstructured model produces visually reasonable reconstructions, some frames are misaligned in position and orientation relative to the rig. Introducing time embeddings helps correct positional drift, while rig pose embeddings improve orientation alignment. The fully calibrated model achieves the highest reconstruction quality and most accurate pose estimates, demonstrating the compounding benefits of both embedding types.

Figure 8 presents additional visualization of Rig3r outputs on diverse rig scenes from Argoverse [70], Waymo [72], and nuScenes [71] validation sets. The visualizations include confidence-thresholded pointmaps and global raymaps for rig scenes (color-coded by discovered rig structure), and highlight Rig3R's consistency under diverse conditions. No post-processing is applied to the 3D points or poses, aside from confidence thresholding and sky masking for visualization.

B Estimating Camera Parameters from Raymaps

We describe the method used for estimating camera intrinsics and extrinsics from raymaps, as introduced in the Section 3.1 of the main paper.

Intrinsics At each pixel (u, v), the ray direction from the raymap is interpreted as a unit vector $\hat{\mathbf{r}}$ in a global coordinate frame. In the pinhole camera model, the corresponding normalized ray in the camera frame is given by

$$\hat{\mathbf{r}}_{\text{cam}} = \frac{1}{\|\cdot\|} \begin{bmatrix} u/f_x \\ v/f_y \\ 1 \end{bmatrix}. \tag{1}$$

Here, f_x and f_y are focal lengths, and (u, v) denotes image coordinates relative to a known principal point, which we fix to the image center—a standard simplification in SfM and multiview geometry.

Given two pixels, the angle θ between their predicted ray directions must be consistent with the angle computed from the camera model and intrinsics, i.e., $\cos\theta = \hat{\mathbf{r}}^T\hat{\mathbf{r}}' = \hat{\mathbf{r}}_{\text{cam}}^T\hat{\mathbf{r}}'_{\text{cam}}$. Squaring and writing this in terms of camera coordinates gives

$$\cos^2\theta = \frac{(\tilde{\mathbf{u}}^\top \omega \tilde{\mathbf{u}}')^2}{(\tilde{\mathbf{u}}^\top \omega \tilde{\mathbf{u}})(\tilde{\mathbf{u}}'^\top \omega \tilde{\mathbf{u}}')}, \text{ where } \omega = \operatorname{diag}(1/f_x^2,\ 1/f_y^2,\ 1), \tag{2}$$

$\mathbf{r}, \mathbf{r}'$ are a pair of world rays from the raymap and $\tilde{\mathbf{u}}, \tilde{\mathbf{u}}'$ the corresponding (homogeneous) image coordinates. This equation constrains the focal lengths and can be solved analytically (simultaneous polynomials) or numerically using multiple pixel pairs. The intrinsic matrix is then formed using the assumed camera center and recovered focals.

As a practical simplification, we can estimate f_x and f_y analytically by sampling pixel pairs along the image axes. For example, selecting the optical center and a second pixels at $(\Delta u, 0)$, we obtain:

$$f_x = \frac{|\Delta u|}{\tan\theta}, \text{ and similarly for } f_y.$$

This works well in practice due to the high consistency of Rig3R's predicted raymaps, which provide stable and geometrically faithful directions across pixel locations and views—enabling accurate and efficient focal length estimation.

Extrinsics. Once intrinsics are estimated, we compute the ray direction $\hat{\mathbf{r}}_{\text{cam}}^{(i)}$ for each pixel (u, v) using Equation 1. The global raymap predicts the corresponding unit ray directions $\hat{\mathbf{r}}^{(i)}$ in a shared global reference frame. Since both sets of rays are defined at the same pixel locations, we obtain a dense correspondence between camera-frame and global-frame rays. The relationship between them is a rigid transformation consisting of a single rotation $\mathbf{R}$, such that

$$\hat{\mathbf{r}}^{(i)} = \mathbf{R}\hat{\mathbf{r}}_{\text{cam}}^{(i)}.$$

We solve for the optimal rotation $\mathbf{R}$ that minimizes angular error across all correspondences using cross-covariance alignment and singular value decomposition (SVD), following [54].

C Raymaps vs. Pointmaps for Pose Estimation

This section provides further analysis and experimental results comparing raymaps and pointmaps as output representations for pose estimation. While raymaps encode per-pixel ray directions, and can be derived directly from camera intrinsics and extrinsics, pointmaps require predicting full 3D coordinates via per-pixel depth estimation. This makes pointmap-based inference strictly harder and more error-prone—especially in low-texture, reflective, dynamic, or sky regions where depth is ill-posed. Thus we expect pose estimation from raymaps to be more stable than for pointmaps.

To test this hypothesis, we evaluate three Rig3R variants: PnP RANSAC [78] on the predicted pointmaps with confidence thresholding, the same with sky masking, and pose estimation using

Method	@15° ↑		@5° ↑		@30° ↑
	RRA	RTA	RRA	RTA	mAA
Rig3R$_{\text{Calib}}$ (pointmap)	25.2	26.3	1.3	4.3	7.7
Rig3R$_{\text{Calib}}$ (pointmap + sky mask)	69.2	46.8	59.7	25.4	34.0
Rig3R$_{\text{Calib}}$ (raymap)	**99.4**	**91.6**	**67.4**	**77.4**	**82.1**

Table 5: Comparison of Pointmaps vs Raymaps for pose estimation on Waymo.

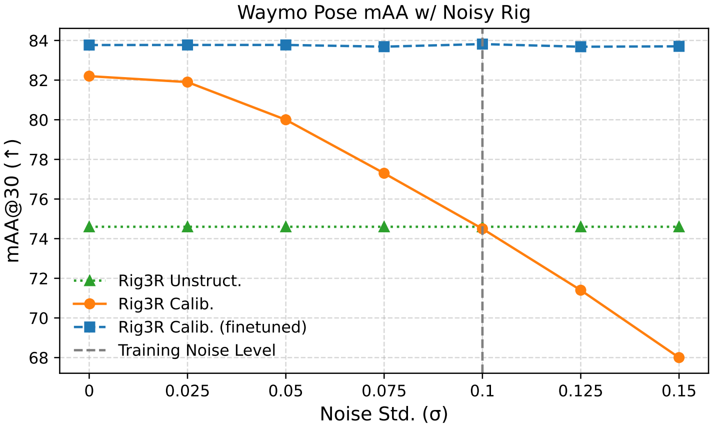


Figure 9: Robustness to rig metadata noise before and after finetuning on noisy embeddings. We plot mAA@30 as Gaussian noise is added to rig pose embeddings at inference time. Finetuning with noisy metadata improves performance across all noise levels.

closed-form solutions on the global raymap. As shown in Table 5, the raymap-based method consistently outperforms both pointmap variants. Even after masking sky pixels—where depth is undefined—pointmap-based estimates remain less accurate and more variable, indicating that relying on intermediate 3D points is suboptimal for pose inference.

D Sensitivity to Noisy Calibration Embeddings

We present results of an additional experiment to evaluate Rig3R's robustness to rig calibration error, which commonly arises in real-world systems due to hardware tolerances, sensor drift, or coarse offline estimation. We additionally test whether training on noisy inputs improves performance when calibration metadata is degraded at inference time.

We simulate noise by independently perturbing rig extrinsics—translation and rotation (roll, pitch, yaw)—with zero-mean Gaussian noise. During training, we use a fixed standard deviation of 0.1. Rig positions are normalized so their average distance from the reference camera is 1, making this noise roughly equivalent to 10 cm deviation for a 1-meter rig. Preliminary tests showed similar performance between Rig3R$_{\text{Calib}}$ and Rig3R$_{\text{Unstr}}$ at this noise level, guiding our choice. At inference time, we evaluate robustness across a range of increasingly severe noise levels.

We compare three models: (1) Rig3R without rig embeddings (Unstr.), (2) the original Rig3R trained on clean metadata, and (3) Rig3R finetuned on noisy metadata. As shown in Fig. 9, the clean model performs well under mild noise but degrades steadily as noise increases. In contrast, the finetuned model maintains high performance across all noise levels. The unstructured variant remains flat across noise levels, as it does not use rig embeddings and serves as a lower-bound control. We observe that the base model still stays above the unstructured model even as noise increases, and even has higher performance than the base calibrated model. These results show that training on noisy rig metadata enables Rig3R to remain robust to calibration errors at inference time. This enhances robustness to degraded inputs and makes the approach more practical for real-world deployment, where calibration is often approximate but still informative.

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 abstract and introduction clearly state the paper's contributions, which are supported by Sections 3 and 4.

   Guidelines:

   - The answer NA 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 NA 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: Section 5 discusses limitations related to data diversity and generalization to novel rig configurations, and outlines directions for future improvement.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The paper focuses on empirical methods; it builds on established theoretical components (e.g., ray geometry) without introducing new theorems or proofs.

Guidelines:

- The answer NA 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: In Section 3, the paper provides full model details, training setup, and dataset sources and processing methods, enabling reproduction of the main experimental results.

   Guidelines:

   - The answer NA 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: While we do not provide open access to code at submission time, the method is clearly described and most datasets are publicly available for reproducibility; we will look into releasing code and a model trained only on public datasets in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3.5 details training hyperparameters, optimizer settings, data preprocessing, and augmentation; evaluation splits are described in Section 4.

   Guidelines:

   - The answer NA 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: We do not report error bars due to the high computational cost of rerunning large-scale experiments; however, we examine extensive ablations and diverse dataset evaluations.

   Guidelines:

   - The answer NA 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: Section 3.5 outlines the compute setup, including GPU type, training duration, and batch size.

   Guidelines:

   - The answer NA 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 adheres to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: This work focuses on foundational research in 3D vision and does not directly involve applications or consequences with immediate societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce models or datasets with high risk for misuse; it focuses on 3D vision tasks trained on existing, publicly available datasets and one internal dataset.

    Guidelines:

    - The answer NA 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: We use public datasets and evaluate against third-party models, all of which are properly credited and used in accordance with their licenses.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new datasets, models or other assets at submission time.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs in any important, original, or non-standard way.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.