

---

# Rig3R: Rig-Aware Conditioning for Learned 3D Reconstruction

---

Anonymous Author(s)

Affiliation

Address

email

# 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.

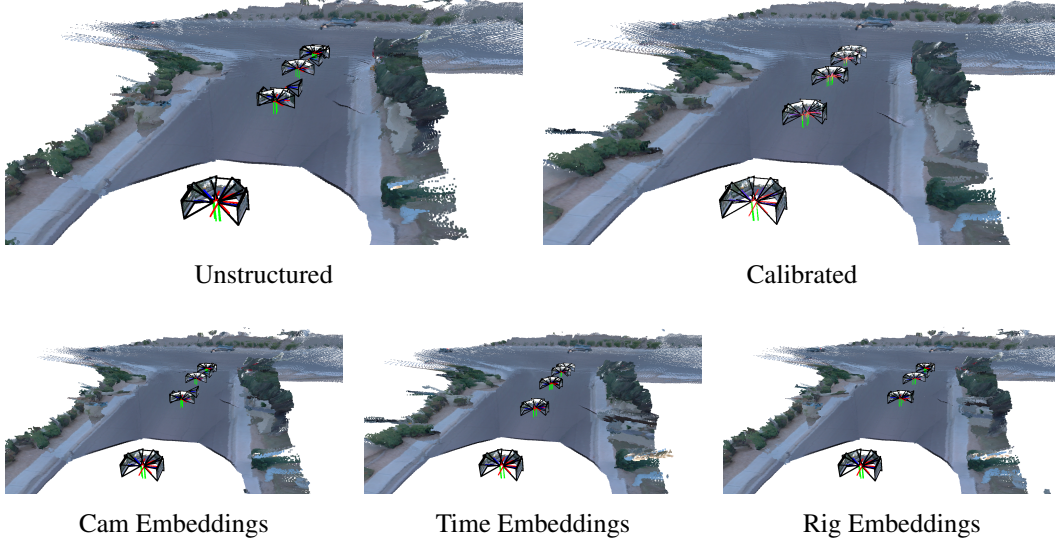


Figure 1: Visualizations of the qualitative effects of rig metadata embeddings on the Waymo validation set. We observe that with added embeddings, the quality of the estimated poses noticeably improves, and the fine details of reconstructed scene are also better captured.

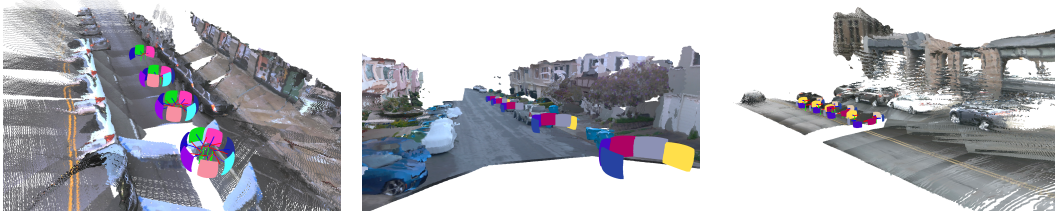


Figure 2: Rig scene reconstructions on Argoverse [1], Waymo [2], and nuScenes [3] validation sets.

## A Additional Visualization

Figure 1 presents qualitative reconstruction results to illustrate the embedding ablations in 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 2 presents additional visualization of Rig3r outputs on diverse rig scenes from Argoverse [1], Waymo [2], and nuScenes [3] 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.

## 18 B Estimating Camera Parameters from Raymaps

19 We describe the method used for estimating camera intrinsics and extrinsics from raymaps, as  
20 introduced in the Raymap Representation section of the main paper.

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

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

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

26 Given two pixels, the angle  $\theta$  between their predicted ray directions must be consistent with the angle  
27 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  
28 this in terms of camera coordinates gives

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

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

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

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

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

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

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

43 We solve for the optimal rotation  $\mathbf{R}$  that minimizes angular error across all correspondences using  
44 cross-covariance alignment and singular value decomposition (SVD), following [4].

## 45 C Raymaps vs. Pointmaps for Pose Estimation

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

52 To test this hypothesis, we evaluate three Rig3R variants: PnP RANSAC [5] on the predicted  
53 pointmaps with confidence thresholding, the same with sky masking, and pose estimation using

Method	@15° ↑		@5° ↑		@30° ↑
	RRA	RTA	RRA	RTA	mAA
Rig3R <sub>Calib</sub> (pointmap)	25.2	26.3	1.3	4.3	7.7
Rig3R <sub>Calib</sub> (pointmap + sky mask)	69.2	46.8	59.7	25.4	34.0
Rig3R <sub>Calib</sub> (raymap)	<b>99.4</b>	<b>91.6</b>	<b>67.4</b>	<b>77.4</b>	<b>82.1</b>

Table 1: Comparison of Pointmaps vs Raymaps for pose estimation on Waymo.

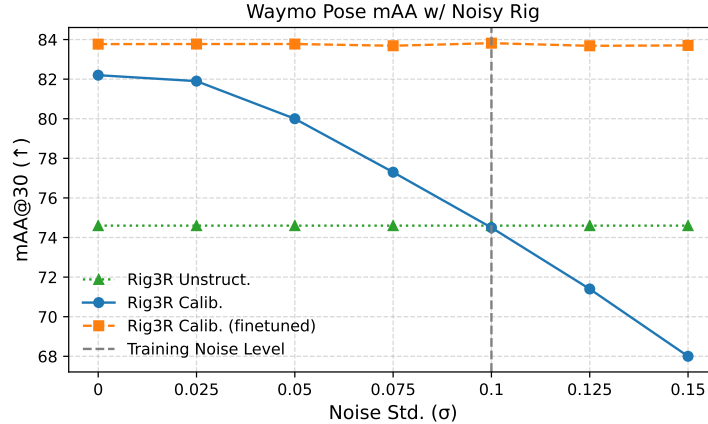


Figure 3: Robustness to rig metadata noise before and after finetuning on noisy embeddings. We plot mAA@30 as Gaussian noise is added to rig pose embeddings at inference time. Finetuning with noisy metadata improves performance across all noise levels.

closed-form solutions on the global raymap. As shown in Table 1, 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<sub>Calib</sub> and Rig3R<sub>Unstr</sub> 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. 3, 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.

## 79 References

- 80 [1] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khan-  
81 delwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argo-  
82 verse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint*  
83 *arXiv:2301.00493*, 2023.
- 84 [2] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul  
85 Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for  
86 autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on*  
87 *Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020.
- 88 [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush  
89 Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for  
90 autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.
- 91 [4] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-d  
92 point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.
- 93 [5] George Terzakis and Manolis Lourakis. *A Consistently Fast and Globally Optimal Solution*  
94 *to the Perspective-n-Point Problem*, pages 478–494. 11 2020. ISBN 978-3-030-58451-1. doi:  
95 10.1007/978-3-030-58452-8\_28.