# 6. Implementation Details

**Implementation Details.** Our models are built on the pre-trained Stable Diffusion V2.1 model [53]. To train camera intrinsic estimation model, we employ the AdamW optimizer with a learning rate of $3e^{-5}$ and train the model for 30,000 iterations with a total batch size of 196 on a cluster of 8 Nvidia A800 GPUs. For metric depth estimation, we use the same optimizer and learning rate with a total batch size of 96, and the training process takes approximately 5 days to converge. For all of our downstream 3D vision tasks, we did not use the ground truth camera image but instead relied on intrinsic parameters predicted by our diffusion model.

## 6.1. Camera intrinsic prediction

We train our model on a diverse range of datasets, ensuring balance by selecting one dataset per batch with equal probability and sampling from it. Most datasets follow the setup of Zhu et al. [96], with additional data incorporated to better leverage the capabilities of stable diffusion. A detailed description of the datasets is provided in Tab. 6. Notably, our training set includes more data compared to He et al. [23]. For a fair comparison, we also report our results using the same training dataset and results is shown in Tab. 8. Regarding the Camera Image, we normalize its values to the range $[-1, 1]$ by dividing by $\pi$, and instead of force-resizing, we pad the Camera Image to a resolution of $768 \times 768$. Unlike previous works [23, 96] that directly resize images to a fixed size, we resize the images while preserving their aspect ratios, padding the remaining areas with zeros. This approach is necessary because the data we used were collected with various aspect ratios even within a single dataset. Following the data augmentation strategy applied in [96], we randomly scale images up to twice their original size and then crop them back to the original resolution, with the camera intrinsics adjusted accordingly.

## 6.2. Metric depth prediction

For metric depth prediction, we do not pad the images. Instead, we resize the maximum dimension of the images to 768 while maintaining their aspect ratios. Additionally, we apply random horizontal flipping and random cropping to enhance dataset diversity even in one dataset. Inspired by [15], we incorporate a "scene distribution decoupler" into our model through text-guided conditioned depth generation. Specifically, we utilize the CLIP tokenizer and encoder to encode the terms "indoor geometry" and "outdoor geometry" for different environments. Based on this setting, we treat the metric depth with different scale factor for indoor and outdoor: $s = \{s_{\text{in}}, s_{\text{out}}\}$, and the depth label become $\boldsymbol{d}_s = \boldsymbol{d}/s_i$ with $s_i \in s$ to fit the output of the training VAE decoder.

Table 6. **Datasets List for camera calibration.** List of the training and testing datasets: number of images, scene type, and method of calibration. SfM: Structure-from-Motion.

| | Dataset | Images | Scene | Intrinsic |
|---|---|---|---|---|
| **Training Set** | NuScenes [7] | 28k | Outdoor | Calibrated |
| | KITTI [11] | 18 k | Outdoor | Calibrated |
| | CityScapes [11] | 23k | Outdoor | Calibrated |
| | NYUv2 [44] | 6k | Indoor | Calibrated |
| | SUN3D [78] | 33k | Indoor | Calibrated |
| | ARKitScenes [3] | 48k | Indoor | Calibrated |
| | Objectron [1] | 33k | Indoor | SfM |
| | MVImgNet [86] | 27k | Indoor | SfM |
| | Hypersim [52] | 54k | Indoor | Synthetic |
| | Virtual KITTI [6] | 20k | Outdoor | Synthetic |
| | Taskonomy [87] | 420k | Indoor | Rendered |
| | TartanAir [74] | 305k | Mix | Synthetic |
| **Testing Set** | Waymo [67] | 800 | Outdoor | Calibrated |
| | RGBD [65] | 160 | Indoor | Pre-defined |
| | ScanNet [13], | 800 | Indoor | Calibrated |
| | MVS [16] | 132 | Outdoor | Pre-defined |
| | Scenes11 [10] | 256 | Mixed | Pre-defined |

Table 7. **Datasets List for Metric Depth estimation.** List of the training and testing datasets for metric depth estimation: number of images, scene type, and method of Acquisition.

| | Dataset | Images | Scene | Acquisition |
|---|---|---|---|---|
| **Training Set** | Hypersim [52] | 54k | Indoor | Synthetic |
| | Virtual KITTI [6] | 20k | Outdoor | Synthetic |
| | Taskonomy [87] | 40M | Indoor | RGB-D |
| | TartanAir [74] | 305k | Mix | Synthetic |
| | Argoverse2 [76] | 403k | Outdoor | LiDAR |
| | Waymo [68] | 223k | Outdoor | LiDAR |
| | Self-rendered | 10k | Outdoor | Synthetic |
| **Testing Set** | Scannet [13] | 83k | Indoor | RGBD |
| | Diode [70] | 771 | Mix | LiDAR |
| | ETH3D [59] | 454 | Outdoor | RGB-D |
| | IBims-1 [35] | 100 | Indoor | RGB-D |
| | NuScenes [7] | 3k | Outdoor | LiDAR |
| | NYU [44] | 654 | Indoor | RGB-D |
| | VOID [77] | 800 | Indoor | RGB-D |

## 6.3. More implementation details and discussions relating Figures and Tables.

**Fig. 3:** Our Camera Image is image-dependent, unlike other camera representations that are not. For other methods, lines can be plotted directly based on different FoV values. In contrast, we generate the line chart for the Camera Image using the GSV dataset [2], which includes 20 different types of cameras.

**Tab. 9:** We assess the generalization ability across five zero-shot datasets by aligning the predicted depth $\hat{\boldsymbol{d}}$ to the ground-truth depth $\boldsymbol{d}$ with a scale factor $s$ and translation $t$, resulting in the aligned depth map $\boldsymbol{a} = s \times \hat{\boldsymbol{d}} + t$

**Tab. 10:** The pose estimation is compared against pseudo-

Table 8. **Monocular Camera Calibration on Zero-Shot Datasets.** We report the calibration errors for both focal length and optical center. *Small* means we train our model with same dataset with Zhu et al. [96] and He et al. [23].

| Method | Waymo | | RGBD | | ScanNet | | MVS | | Scenes11 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $e_f$ | $e_b$ | $e_f$ | $e_b$ | $e_f$ | $e_b$ | $e_f$ | $e_b$ | $e_f$ | $e_b$ | $e_f$ | $e_b$ |
| Ours-small | 0.138 | 0.033 | 0.051 | 0.012 | 0.084 | 0.023 | 0.080 | 0.010 | 0.071 | 0.014 | 0.085 | 0.017 |
| Ours | 0.115 | 0.036 | 0.041 | 0.010 | 0.089 | 0.024 | 0.087 | 0.008 | 0.061 | 0.010 | 0.078 | 0.017 |

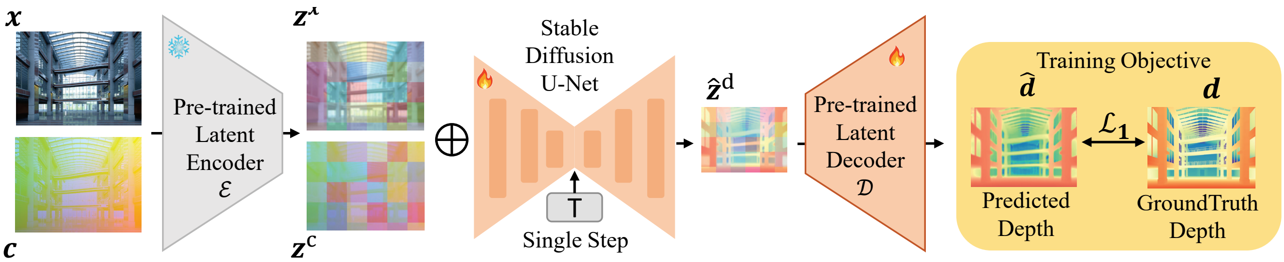


Figure 8. **The overview of metric depth training pipeline.** The encoded image and camera image $\boldsymbol{z}^x$ and $\boldsymbol{z_c}$ are concatenated and sent to pretrained U-Net. Then we employ single-step diffusion at timestamp $T$ to generate depth latent code $\hat{\boldsymbol{z}}_{\boldsymbol{d}}$, which is then decoded into predicted metric depth $\hat{d}$.

ground truth generated using COLMAP [58] from 60 images of a single object, leveraging the ground truth focal length for improved accuracy. For the reconstruction, we select 20 of these images and compare the pose estimation with and without intrinsic cues. Note that SE(3) and scale alignment are applied for the comparison.

**Fig. 7 and Fig. 10:** From a single input image, we first estimate the camera intrinsics and metric depth map, transform them into a 3D point cloud using the pinhole camera model, and calculate the 3D distance between key points.

**Fig.** 12 & **Fig.** 13: We take 20 to 25 images with five different focal lengths (same image focal lengths as shown in Fig. 7) and perform the reconstruction based on these images. Surrouding are cropped for better visulization. Our method complements sparse-view reconstruction methods like Dust3r [73] by providing intrinsic information, rather than serving as a direct comparison. Dust3r [73] delivers less accurate intrinsic estimation because it focuses on sparse-view reconstruction by generating point clouds for image pairs and performing global alignment to jointly optimize intrinsic calibrations and poses. This process is less robust and often converges to a local minimum. In contrast, our method is specifically designed to recover camera intrinsics. The results demonstrate that Dust3r achieves more accurate reconstruction when equipped with our estimated intrinsics.

**Procrustes alignment:** When pointcloud $X$ is given, the relative pose can be obtained by Procrustes alignment [41]:

$$R^*, t^* = \operatorname*{arg\,min}_{\sigma,R,t} \sum_i \left\| \sigma(RX_i^{1,1} + t) - X_i^{1,2} \right\|^2 ,$$

where $X^{1,2}$ represents the pointmap of image 1 in the coordinate frame of image 2. Then, a global alignment of the pointmaps is performed to further refine the pose and obtain the final aligned pointcloud reconstruction

## 7. More experimental Results

### 7.1. Metric Depth

We show more qualitative metric depth prediction in Fig. 9.

### 7.2. Relative Depth

The quantitative comparison for relative depth is shown in Tab. 9

### 7.3. Metrologie

We show more Metrologie results in Fig. 10 compared with Metric3D [85].

We also present the metrologie results for UniDepth [46] in Fig. 11. While it shows some limitations in focal estimation, this leads to slightly less accurate visualizations.

### 7.4. 3D reconstruction

We show more qualitative 3D reconstruction results in Fig. 12 and 13.

### 7.5. Mesh Reconstruction

By using our predict metric depth, we can deduce corresponding normal map, and mesh can be reconstructed via the depth and normal map using BiNI algorithm [8]. We present the reconstruction result of Pisa tower in Fig. 6, and we show the reconstructed mesh in Fig. 14. Noting that we crop all background for better visualization.

### 7.6. Single view 3D reconstuction

In this section, we present single-view 3D reconstruction of different camera focal length results using our estimated

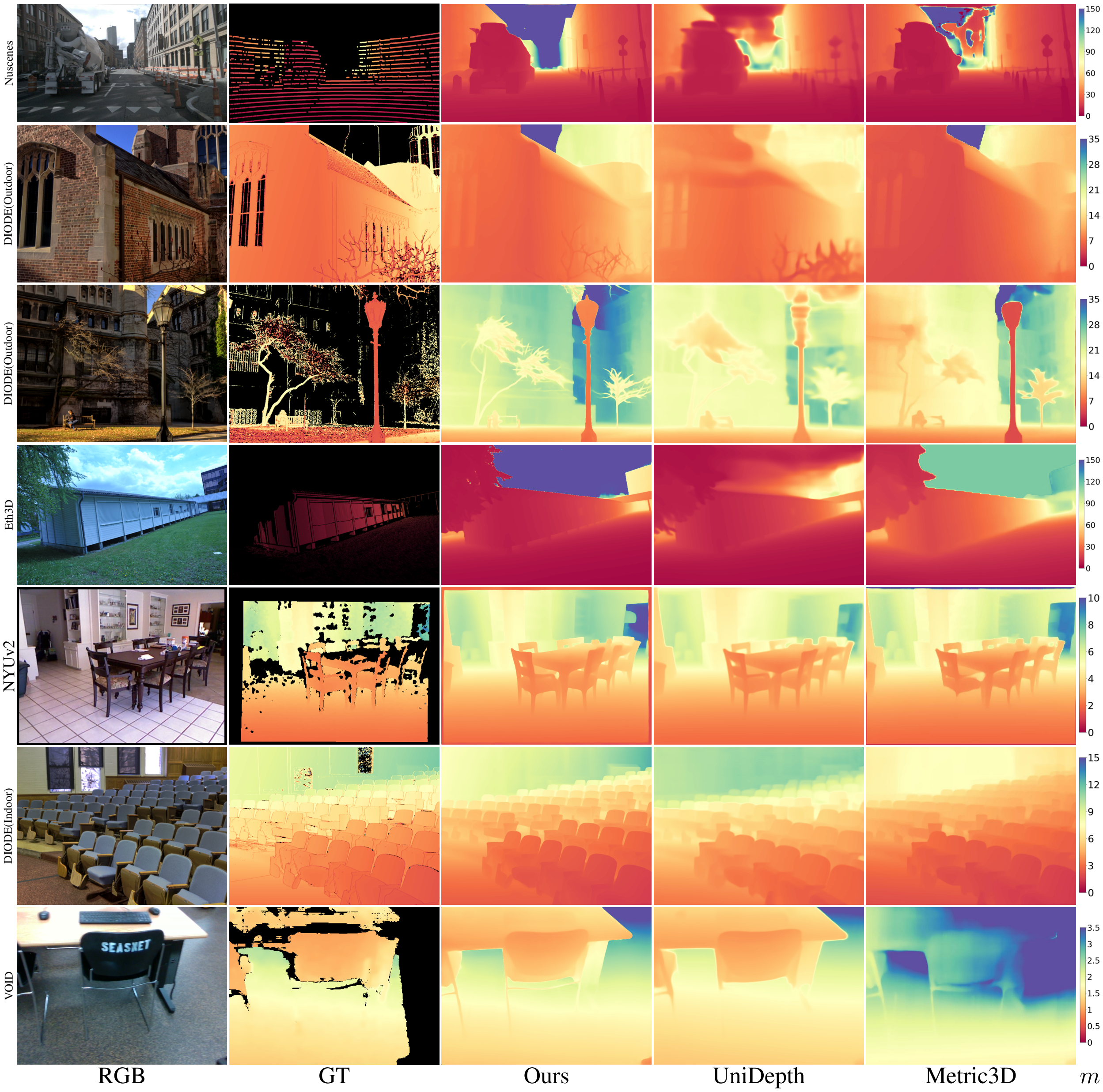


Figure 9. **Zero-Shot Metric Depth Estimation Results.** We present the predicted metric depth in both outdoor and indoor scenes. Our method provides more detailed results and recovers accurate metric depths.

camera intrinsics and metric depth map. By applying the pinhole camera model, we transform the estimated intrinsics and depth map into a 3D point cloud. We demonstrate the robustness of our intrinsic estimation and depth prediction through in-the-wild single-view 3D reconstructions. Qualitative results can be found in Fig 15.

### 7.7. The Importance of Principal Point Evaluation and the Assessment of Both Vertical and Horizontal Focal Lengths

In our work, we evaluate the focal length as well as the principle points. Some previous works [30, 72] focuses solely on focal length. We prove the indispensability to evaluate the principal points. We have a significant amount of data where the principal point does not lie at the image center in certain datasets, and our model effectively learns the position of the principal points rather than ignoring them. To validate this, we conduct an ablation study comparing

Table 9. **Quantitative Comparison on 5 Zero-shot Affine-invariant Depth Benchmarks.** Despite targeting metric depth, we achieve performance comparable to SoTA affine-invariant depth methods.

| Method | NYUv2 | | KITTI | | ETH3D | | ScanNet | | DIODE-Full | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AbsRel ↓ | $\delta1$ ↑ | AbsRel ↓ | $\delta1$ ↑ | AbsRel ↓ | $\delta1$ ↑ | AbsRel ↓ | $\delta1$ ↑ | AbsRel ↓ | $\delta1$ ↑ |
| DiverseDepth [82] | 11.7 | 87.5 | 19.0 | 70.4 | 22.8 | 69.4 | 10.9 | 88.2 | 37.6 | 63.1 |
| MiDaS [51] | 11.1 | 88.5 | 23.6 | 63.0 | 18.4 | 75.2 | 12.1 | 84.6 | 33.2 | 71.5 |
| LeReS [83] | 9.0 | 91.6 | 14.9 | 78.4 | 17.1 | 77.7 | 9.1 | 91.7 | 27.1 | 76.6 |
| Omnidata v2 [31] | 7.4 | 94.5 | 14.9 | 83.5 | 16.6 | 77.8 | 7.5 | 93.6 | 33.9 | 74.2 |
| HDN [90] | 6.9 | 94.8 | 11.5 | 86.7 | 12.1 | 83.3 | 8.0 | 93.9 | 24.6 | 78.0 |
| DPT [50] | 9.8 | 90.3 | 10.0 | 90.1 | 7.8 | 94.6 | 8.2 | 93.4 | **18.2** | 75.8 |
| Metric3D [85] | 5.8 | 96.3 | **5.8** | **97.0** | 6.6 | 96.0 | 7.4 | 94.1 | 22.4 | 78.5 |
| DepthAnything [80] | **4.3** | **98.1** | 7.6 | 94.7 | 12.7 | 88.2 | **4.2** | **98.0** | 27.7 | 75.9 |
| Marigold [33] | 5.5 | 96.4 | 9.9 | 91.6 | 6.5 | 96.0 | 6.4 | 95.1 | 30.8 | 77.3 |
| GeoWizard [15] | 5.2 | 96.6 | 9.7 | 92.1 | **6.4** | **96.1** | 6.1 | 95.3 | 29.7 | 79.2 |
| Ours | 4.8 | 97.1 | 8.5 | 93.5 | 7.1 | 95.3 | 5.7 | 96.5 | 25.6 | **79.4** |

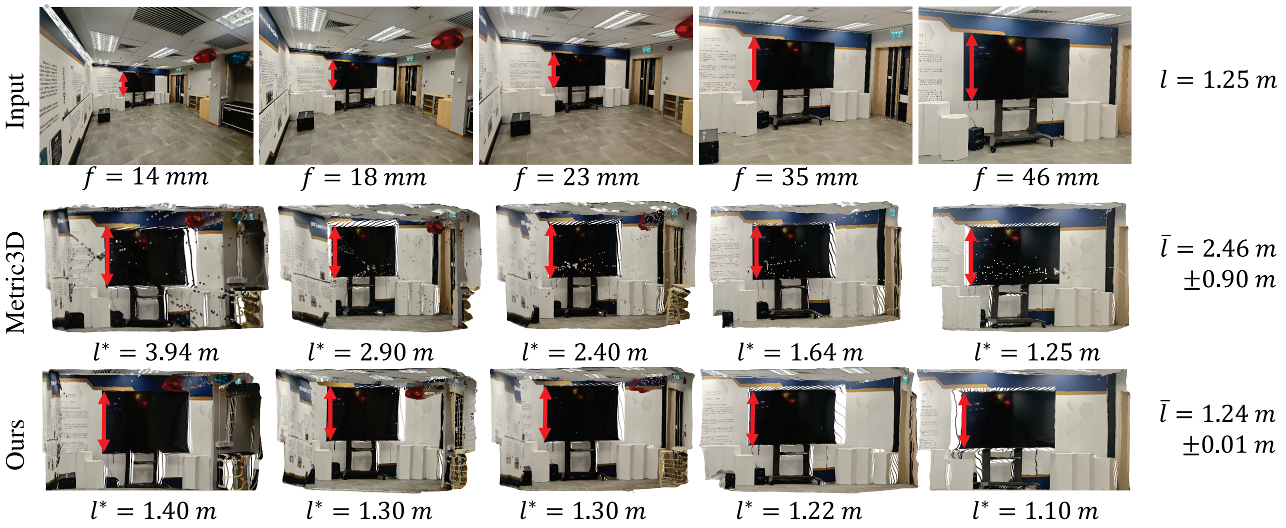


Figure 10. **Metrology of in-the-wild scenes.** Our method accurately recovers real-world metrics while demonstrating robustness to variations in focal length.

Table 10. **Pose error.** We compare the pose error with and without intrinsic cue.

| | $t_{rel}\ (m)$ | $r_{rel}\ (^\circ)$ |
|---|---|---|
| *w/o.* cue | 1.17 | 5.02 |
| *w.* cue | 0.63 | 2.30 |

Table 11. **Principal points error** We compare the error of principle point estimation when assuming principal point lies at the image center with the error of our estimated principal point.

| | **NuScenes** | **KITTI** | **CityScapes** | **NYUv2** |
|---|---|---|---|---|
| $e_b$ | 0.051 | 0.021 | 0.055 | 0.050 |
| $\hat{e}_b$ | 0.007 | 0.014 | 0.011 | 0.009 |

the error when assuming the principal point lies at the image center ($e_b$) with the error of our estimated principal point ($\hat{e}_b$). We show the results on Tab. 11.

Furthermore, not all datasets have $f_x = f_y$ (e.g., CityScapes dataset [11] with $f_x = 2268.36$ and $f_y = 2225.54$). And our method is inherently capable of solving for both $f_x$ and $f_y$ and we take this into account to ensure more robust estimation and support future broader applications and datasets such as Diode [70].

### 7.8. The Importance of camera image in metric depth estimation.

The camera image (intrinsic information) is essential for robust and accurate metric depth estimation. We present the $\delta_1$

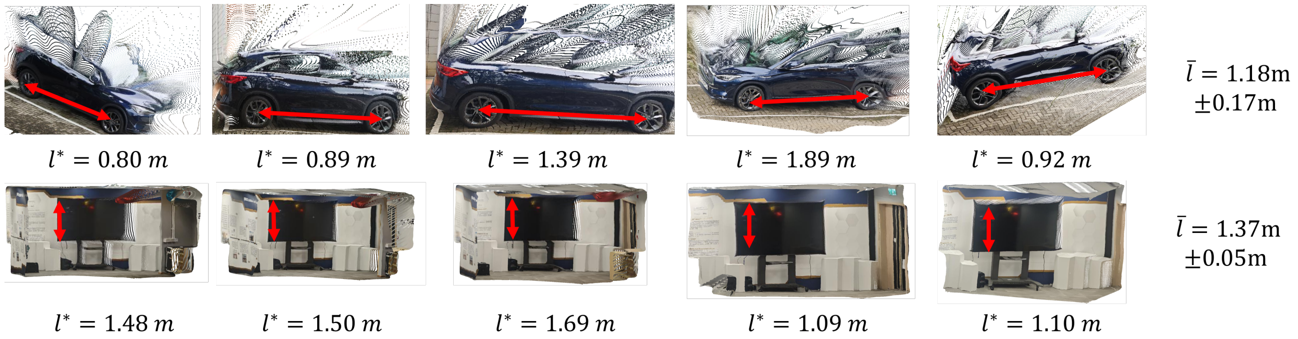


Figure 11. **Metrology of in-the-wild scenes for UniDepth.**

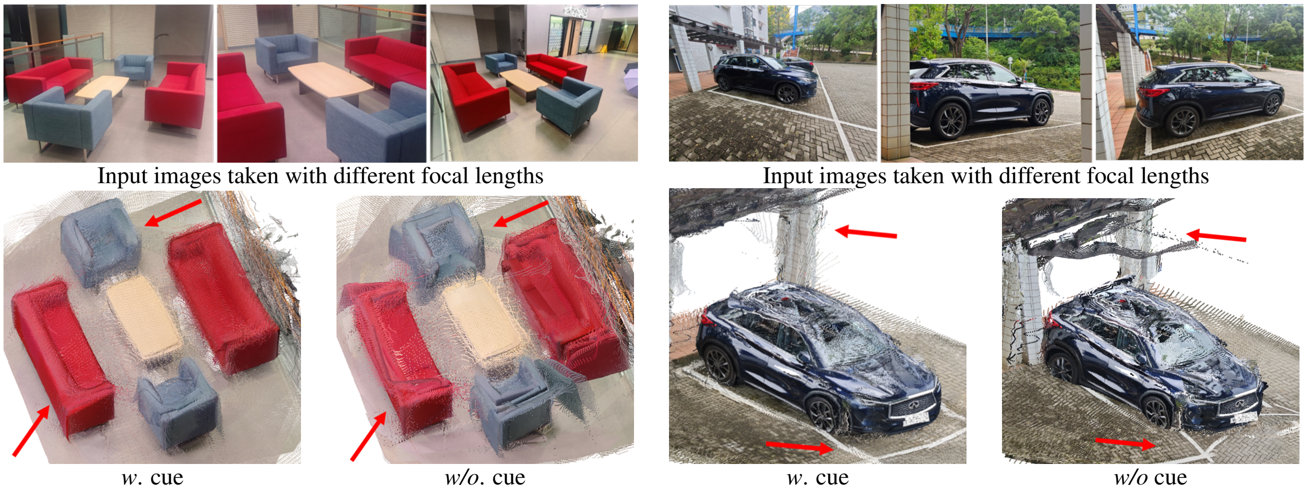

Input images taken with different focal lengths
Input images taken with different focal lengths
*w.* cue
*w/o.* cue
*w.* cue
*w/o* cue

Figure 12. **Sparse View 3D Reconstruction with Intrinsic Cues.** We captured images with various focal lengths and present the reconstruction results. With intrinsic cues, our method achieves more accurate and better-aligned reconstructions.

results on three additional datasets in Tab.12, complementing the findings in Tab.5.

Table 12. Ablation study on the effectiveness of camera images for metric depth estimation.

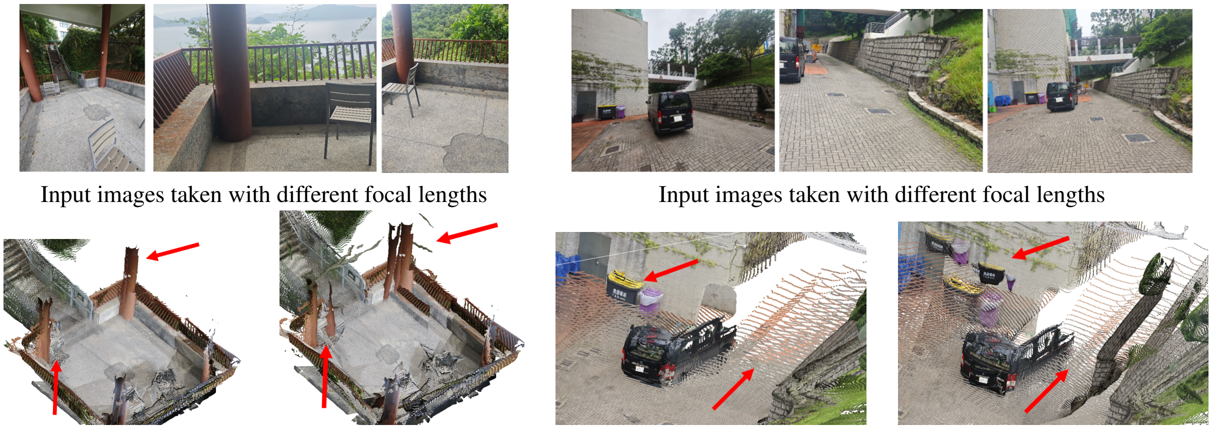
| | ibims | Diode indoor | Diode outdoor |
|---|---|---|---|
| *w.* cam img | 88.7 | 50.1 | 41.0 |
| *w.o* cam img | 82.6 | 35.0 | 25.2 |

As shown, the absence of the camera image leads to a significant performance drop.

### 7.9. Test-time ensembling

To reduce the stochasticity of the process, we aggregate five predicted camera images by taking their mean. This significantly minimizes the randomness of the diffusion model, as evidenced by the small standard deviation in Tab. 13.

Without the aggregation, the standard deviation is sometimes not negligible, as presented in Tab. 14.



Input images taken with different focal lengths

Input images taken with different focal lengths

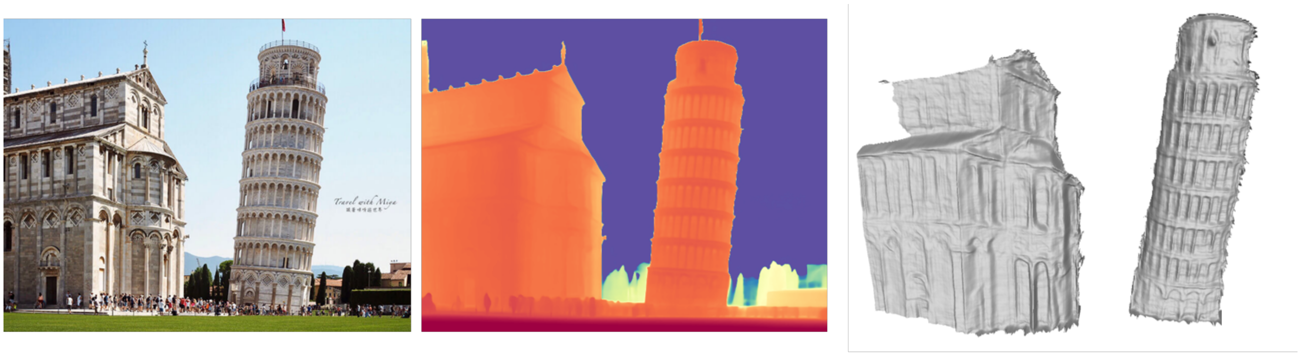
*w.* cue

*w/o.* cue

*w.* cue

*w/o* cue

Figure 13. **Sparse view 3D reconstruction with intrinsic cue.** We captured images at different focal lengths and present the reconstruction results. With intrinsic cues, the reconstruction is more accurate and better aligned.



a). RGB

b). Predicted Metric Depth

c). Reconstructed mesh

Figure 14. **The reconstructed mesh using our predicted intrinsics and metric depth.**

Table 13. **Standard Deviation of estimated intrinsics after ensembling.**

| | **Waymo** | **RGBD** | **ScanNet** | **MVS** | **Scenes11** | **Average** |
|---|---|---|---|---|---|---|
| $e_f$ | $0.115 \pm 0.008$ | $0.041 \pm 0.002$ | $0.089 \pm 0.002$ | $0.087 \pm 0.006$ | $0.061 \pm 0.006$ | $0.078 \pm 0.006$ |
| $e_b$ | $0.036 \pm 0.001$ | $0.010 \pm 0.000$ | $0.024 \pm 0.000$ | $0.008 \pm 0.000$ | $0.010 \pm 0.001$ | $0.017 \pm 0.001$ |

Table 14. **Standard Deviation of estimated intrinsics without ensembling.**

| | **Waymo** | **RGBD** | **ScanNet** | **MVS** | **Scenes11** | **Average** |
|---|---|---|---|---|---|---|
| $e_f$ | $0.115 \pm 0.035$ | $0.041 \pm 0.010$ | $0.089 \pm 0.024$ | $0.087 \pm 0.008$ | $0.061 \pm 0.009$ | $0.078 \pm 0.017$ |
| $e_b$ | $0.036 \pm 0.012$ | $0.010 \pm 0.001$ | $0.024 \pm 0.001$ | $0.008 \pm 0.001$ | $0.010 \pm 0.001$ | $0.017 \pm 0.001$ |

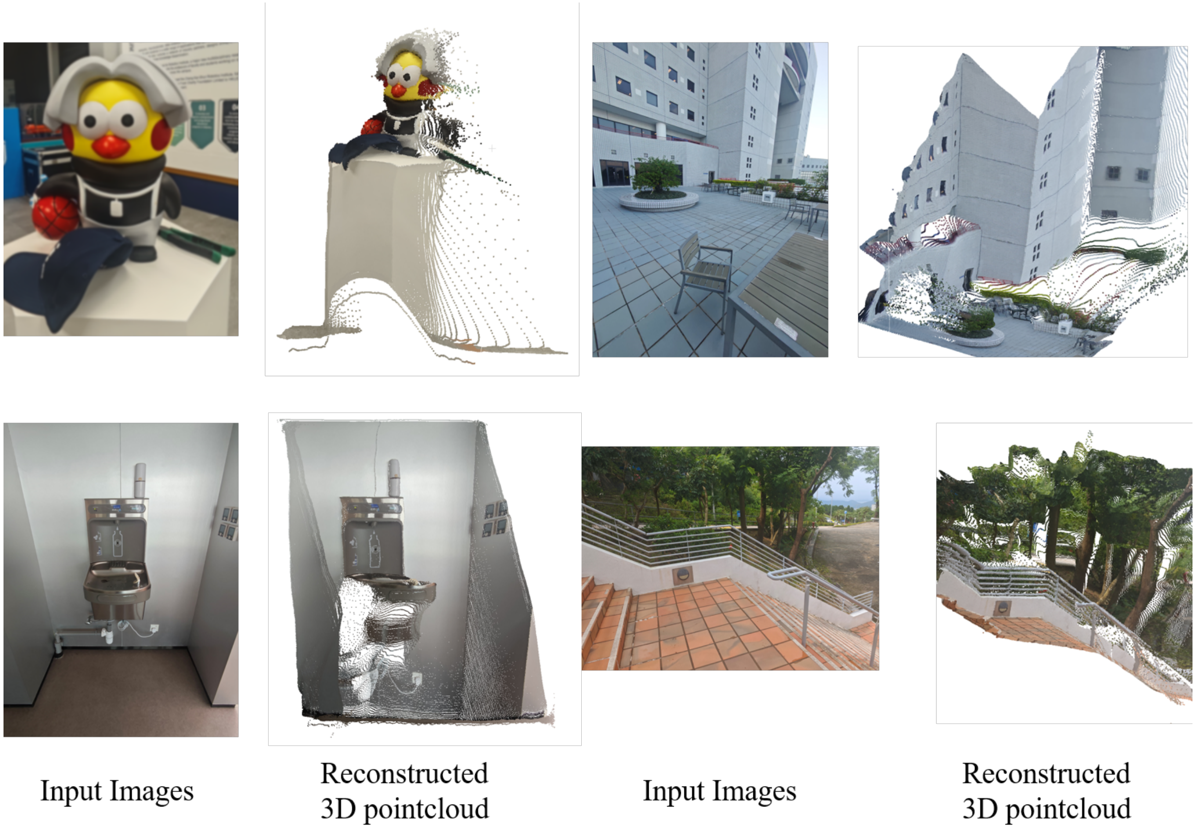


Figure 15. **The reconstructed pointcloud from images with different camera focal length using our predicted instrinc and metric depth.**