MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction

Zehao Yu[1] **Songyou Peng**[2,3] **Michael Niemeyer**[1,3] **Torsten Sattler**[4] **Andreas Geiger**[1,3]

[1]University of Tübingen [2]ETH Zurich [3]MPI for Intelligent Systems, Tübingen
[4]Czech Technical University in Prague

https://niujinshuchong.github.io/monosdf

Abstract

In recent years, neural implicit surface reconstruction methods have become popular for multi-view 3D reconstruction. In contrast to traditional multi-view stereo methods, these approaches tend to produce smoother and more complete reconstructions due to the inductive smoothness bias of neural networks. State-of-the-art neural implicit methods allow for high-quality reconstructions of simple scenes from many input views. Yet, their performance drops significantly for larger and more complex scenes and scenes captured from sparse viewpoints. This is caused primarily by the inherent ambiguity in the RGB reconstruction loss that does not provide enough constraints, in particular in less-observed and textureless areas. Motivated by recent advances in the area of monocular geometry prediction, we systematically explore the utility these cues provide for improving neural implicit surface reconstruction. We demonstrate that depth and normal cues, predicted by general-purpose monocular estimators, significantly improve reconstruction quality and optimization time. Further, we analyse and investigate multiple design choices for representing neural implicit surfaces, ranging from monolithic MLP models over single-grid to multi-resolution grid representations. We observe that geometric monocular priors improve performance both for small-scale single-object as well as large-scale multi-object scenes, independent of the choice of representation.

1 Introduction

3D reconstruction from multiple RGB images is a fundamental problem in computer vision with various applications in robotics, graphics, animation, virtual reality, and more. Recently, coordinate-based neural networks have emerged as a powerful tool for representing 3D geometry and appearance. The key idea is to use compact, memory efficient multi-layer perceptrons (MLPs) to parameterize implicit shape representations such as occupancy or signed distance fields. While early works [9, 42, 50] relied on 3D supervision, several recent works [47, 66, 82] use differentiable surface rendering to reconstruct scenes from multi-view images. At the same time, neural radiance fields (NeRFs) [44] achieved impressive novel view synthesis results with volume rendering techniques. [49, 76, 81] combine surface and volume rendering for the task of 3D reconstruction by expressing volume density as a function of the underlying 3D surface, which in turn improves scene geometry.

Current neural implicit-based surface reconstruction approaches achieve impressive reconstruction results for simple scenes with dense viewpoint sampling. Yet, as shown in the first row of Fig. 1, they struggle in the presence of limited input views (DTU with 3 views) or for scenes that contain large textureless regions (walls in ScanNet or Tanks & Temples). A key reason for this behavior is that these model are optimized using a per-pixel RGB reconstruction loss. Using only RGB images as input leads to an underconstrained problem as there exist an infinite number of photo-consistent

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

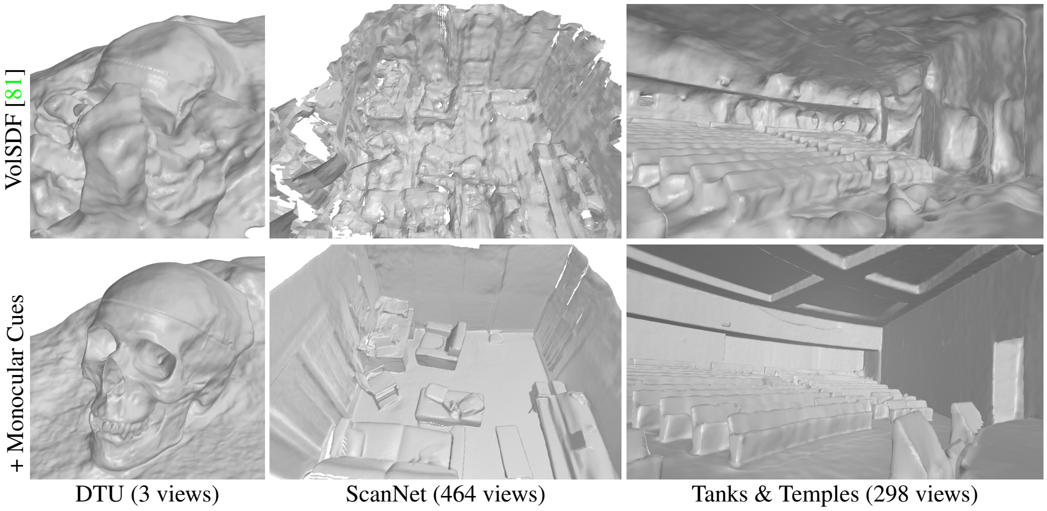


Figure 1: **MonoSDF.** Top: State-of-the-art neural implicit surface reconstruction methods fail in the presence of limited input views or when applied to complex multi-object scenes. Bottom: We demonstrate that incorporating geometric cues from general-purpose monocular predictors enables scaling to larger scenes while yielding more accurate reconstructions and speeding up optimization. An image resolution of 384×384 pixels was used for all results shown above.

explanations [5, 90]. Previous works address this problem by incorporating priors on the structure of the scene into the optimization process, e.g., depth smoothness [46], surface smoothness [49, 91], semantic similarity [30], or Manhattan world assumptions [21]. In this paper, we explore monocular geometric priors as they are readily available and efficient to compute. We show that using such priors significantly improves 3D reconstruction quality in challenging scenarios (see second row of Fig. 1).

Estimating geometric cues such as depth and normals from a single image has been an active research area for decades. The seminal work by Eigen et al. [18, 19] showed that learned models based on deep convolutional neural networks (CNNs) significantly improved over early work in this area [24–27, 59–61]. Recent work [17, 55, 56], in particular Omnidata [17], has made significant headway in terms of prediction quality and generalization to new scenes using very large datasets for training. These strong results on individual images, and the fact that monocular geometric cues can be computed efficiently, naturally lead to the question whether such models are able to provide the additional constraints required by implicit neural surface reconstruction approaches to handle more challenging settings.

This paper describes a framework, called MonoSDF, for integrating monocular geometric priors into neural implicit surface reconstruction methods: given multi-view images, we infer depth and surface normals for each image, and use them as additional supervision signals during optimization together with the RGB image reconstruction loss. We observe that these priors lead to significant gains in reconstruction quality, especially in textureless and less-observed areas as shown in Fig. 1. This is due to the fact that the photometric consistency cues used by surface reconstruction methods and the recognition cues used by monocular networks are complementary: while photometric consistency fails in texturless regions such as walls, surface normals can be predicted reliably in these areas due to the structured 3D scene layout. Conversely, photoconsistency cues allow for establishing globally accurate 3D geometry in textured regions, while normal and (relative) depth cues only provide local geometric information.

Apart from incorporating monocular geometric cues, we provide a systematic study and analysis of state-of-the-art design choices for coordinate-based neural representations in the context of implicit surface reconstruction. More specifically, we investigate the following architectures: a single, large MLP [49, 76, 81, 82], a dense SDF grid [31], a single feature grid [28, 38, 53, 54] and multi-resolution feature grids [10, 22, 45, 69, 92]. We observe that MLPs act globally and exhibit an inductive smoothness bias while being computationally expensive to optimize and evaluate. In contrast, grid-based representations benefit from locality during optimization and evaluation, hence they are computationally more efficient. However, reconstructions are noisier for sparse views or less-observed areas. Including monocular geometric priors improves neural implicit reconstruction results across different settings with faster convergence times and independent of the underlying representation.

In summary, we make the following contributions:

- We introduce *MonoSDF*, a novel framework which exploits monocular geometric cues to improve multi-view 3D reconstruction quality, efficiency, and scalability for neural implicit surface models.
- We provide a systematic comparison and detailed analysis of design choices of neural implicit surface representations, including vanilla MLP and grid-based approaches.
- We conduct extensive experiments on multiple challenging datasets, ranging from object-level reconstruction on the DTU dataset [1], over room-level reconstruction on Replica [67] and ScanNet [13], to large-scale indoor scene reconstruction on Tanks and Temples [34].

2 Related Work

Architectures for Neural Implicit Scene Representations. Neural implicit scene representations or neural fields [78] have recently gained popularity for representing 3D geometry due to their expressiveness and low memory footprint. Seminal works [9, 42, 50] use a single MLP as the scene representation and show impressive object-level reconstruction quality, but they do not scale to more complicated or large-scale scenes due to the limited model capacity. Follow-up works [10, 22, 41, 45, 54, 69, 92] combine an MLP decoder with one or multi-level voxel grids of low-dimensional features. Such hybrid representations are able to better represent fine geometric details and can be evaluated fast. However, they lead to a larger memory footprint with increasing scene size. In this paper we provide a systematic comparison of four architectural design choices for *implicit surface reconstruction*.

3D Reconstruction from Multi-view Images. Reconstructing the underlying 3D geometry from multi-view images is a long-standing goal of computer vision. Classic multi-view stereo (MVS) methods [2, 6–8, 35, 35, 62, 64, 65] consider either feature matching for depth estimation [6, 62] or represent shapes with voxels [2, 7, 8, 35, 51, 64, 72, 73]. Learning-based MVS methods usually replace some parts of the classic MVS pipeline, e.g., feature matching [23, 36, 40, 74, 88], depth fusion [16, 57], or inferring depth from multi-view images [29, 79, 80, 85]. In contrast to the explicit scene representations used by classic MVS algorithms, recent neural approaches [39, 48, 82] represent surfaces via a single MLP with continuous outputs. Learned purely from posed 2D images, they show appealing reconstruction results and do not suffer from discretization. However, accurate object masks are required. Inspired by the density-based volume rendering in NeRF [44], which demonstrated impressive view synthesis without object masks, several works [49, 76, 81] use volume rendering for neural implicit surface reconstruction without masks. However, these methods lead to poor results in large-scale scenes with textureless regions. In this work, we show that incorporating monocular priors allows these approaches to obtain significantly more detailed reconstructions and to scale to larger and more challenging scenes.

Incorporating Priors into Neural Scene Representations. Several researchers proposed to incorporate priors such as depth smoothness [46], semantic similarity [30], or sparse MVS point clouds [58] for the task of *novel view synthesis* from sparse inputs. In contrast, in this work, our focus is on implicit 3D surface reconstruction. Concurrently, Manhattan-SDF [21] uses dense MVS depth maps from COLMAP [63] as supervision and adopts Manhattan world priors [11] to handle low-textured planar regions corresponding to walls, floors, etc. Our approach is based on the observation that data-driven monocular depth and normal predictions [17] provide high-quality priors for the full scene. Incorporating these priors into the optimization of neural implicit surfaces not only removes the Manhattan world assumption [11] but also results in improved reconstruction quality and a simpler pipeline.[1] Compared to NeuRIS [75], a concurrent work that proposes to use normal priors for indoor scene reconstruction, we integrate monocular depth cues and further demonstrate the effectiveness of monocular cues on various neural scene representations, ranging from MLP to multi-resolution feature grids.

[1] Manhattan-SDF [21] requires semantic segmentation to determine where to enforce the assumption.

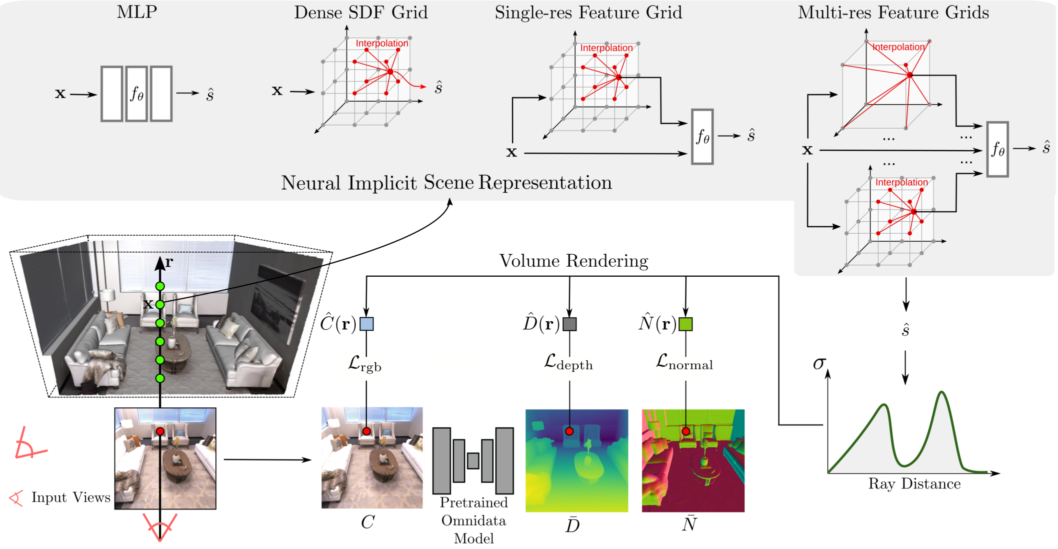


Figure 2: **Overview.** In this work we use monocular geometric cues predicted by a general-purpose pretrained network to guide the optimization of neural implicit surface models. More specifically, for a batch of rays, we volume render predicted RGB colors, depth, and normals, and optimize wrt. the input RGB images and monocular geometric cues. Further, we investigate different design choices for neural implicit architectures and provide an in-depth analysis. For clarity, we only show the SDF and not the color prediction branch above.

3 Method

Our goal is to recover the underlying scene geometry from multiple posed images while utilizing monocular geometric cues to guide the optimization process. To this end, we first review neural implicit scene representations and various design choices in Section 3.1 and discuss how to perform volume rendering of these representations in Section 3.2. Next, we introduce the monocular geometric cues we investigate in our study in Section 3.3 and discuss loss functions and the overall optimization process in Section 3.4. An overview of our framework is provided in Fig. 2.

3.1 Implicit Scene Representations

We represent scene geometry as a signed distance function (SDF). A signed distance function is a continuous function f that, for a given 3D point, returns the point's distance to the closest surface:

$$f : \mathbb{R}^3 \to \mathbb{R} \qquad \mathbf{x} \mapsto s = \text{SDF}(\mathbf{x}) \ . \tag{1}$$

Here, $\mathbf{x}$ is the 3D point and s denotes the corresponding SDF value. In this work, we parameterize the SDF function with learnable parameters θ and investigate several different design choices for representing the function: explicit as a dense grid of learnable SDF values, implicit as a single MLP, or hybrid using an MLP in combination with single- or multi-resolution feature grids.

Dense SDF Grid. The most straightforward way of parameterizing an SDF is to directly store SDF values in each cell of a discretized volume $\mathcal{G}_\theta$ with resolution of $R_H \times R_W \times R_D$ [31]. To query the SDF value $\hat{s}$ for an arbitrary point $\mathbf{x}$ from the dense SDF grid, we can use any interpolation operation:

$$\hat{s} = \texttt{interp}(\mathbf{x}, \mathcal{G}_\theta) \ . \tag{2}$$

In our experiments, we implement `interp` as trilinear interpolation.

Single MLP. The SDF function can also be parameterized by a single MLP [50] f_θ:

$$\hat{s} = f_\theta(\gamma(\mathbf{x})) \ , \tag{3}$$

where $\hat{s}$ is the predicted SDF value and γ corresponds to a fixed positional encoding [44, 70] mapping $\mathbf{x}$ to a higher dimensional space. After their introduction to novel view synthesis [44], positional encoding functions are now widely used for neural implicit surface reconstruction [49, 76, 81, 82] as they increase the expressiveness of coordinate-based networks [70].

Single-Resolution Feature Grid with MLP Decoder. We can also combine both parameterizations and use a feature-conditioned MLP f_θ together with a feature grid Φ_θ with a resolution of R^3, where each cell of the grid stores a feature vector [28, 38, 54, 69] instead of directly storing SDF values:

$$\hat{s} = f_\theta(\gamma(\mathbf{x}), \texttt{interp}(\mathbf{x}, \Phi_\theta)) \ . \tag{4}$$

Note that the MLP f_θ is conditioned on the interpolated local feature vector from the feature grid Φ_θ.

Multi-Resolution Feature Grids with MLP Decoder. Instead of using a single feature grid Φ_θ, one can also employ multi-resolution feature grids $\{\Phi_\theta^l\}_{l=1}^L$ with resolutions R_l [10, 22, 45, 69, 92]. The resolutions are sampled in geometric space [45] to combine features at different frequencies:

$$R_l := \lfloor R_{\text{min}} b^l \rfloor \qquad b := \exp\left(\frac{\ln R_{\text{max}} - \ln R_{\text{min}}}{L-1}\right) \ , \tag{5}$$

where $R_{\text{min}}, R_{\text{max}}$ are the coarsest and finest resolution, respectively. Similarly, we extract the interpolated features at each level and concatenate them together:

$$\hat{s} = f_\theta(\gamma(\mathbf{x}), \{\texttt{interp}(\mathbf{x}, \Phi_\theta^l)\}_l)) \ . \tag{6}$$

As the total number of grid cells grows cubically, we use a fixed number of parameters to store the feature grids and use a spatial hash function to index the feature vector at finer levels [45] (see supplementary for details).

Color Prediction. In addition to the 3D geometry, we also predict color values such that our model can be optimized with a reconstruction loss. Following [82], we therefore define a second function $\mathbf{c}_\theta$

$$\hat{\mathbf{c}} = \mathbf{c}_\theta(\mathbf{x}, \mathbf{v}, \hat{\mathbf{n}}, \hat{\mathbf{z}}) \tag{7}$$

that predicts a RGB color value $\hat{\mathbf{c}}$ for a 3D point $\mathbf{x}$ and a viewing direction $\mathbf{v}$. The 3D unit normal $\hat{\mathbf{n}}$ is the analytical gradient of our SDF function. The feature vector $\hat{\mathbf{z}}$ is the output of a second linear head of the SDF network as in [82]. We parameterize $\mathbf{c}_\theta$ with a two-layer MLP with network weights θ. In case of the dense grid SDF parameterization, we similarly optimize a dense feature grid and obtain the feature vector $\hat{\mathbf{z}}$ via the interpolation function `interp`.

3.2 Volume Rendering of Implicit Surfaces

Following recent work [49,76,81,82], we optimize the implicit representations described in Section 3.1 via an image-based reconstruction loss using differentiable volume rendering. More specifically, to render a pixel, we cast a ray $\mathbf{r}$ from the camera center $\mathbf{o}$ through the pixel along its view direction $\mathbf{v}$. We sample M points $\mathbf{x}_\mathbf{r}^i = \mathbf{o} + t_\mathbf{r}^i \mathbf{v}$ along the ray and predict their SDF $\hat{s}_\mathbf{r}^i$ and color values $\hat{\mathbf{c}}_\mathbf{r}^i$. We follow [81] to transform the SDF values $\hat{s}_\mathbf{r}^i$ to density values $\sigma_\mathbf{r}^i$ for volume rendering:

$$\sigma_\beta(s) = \begin{cases} \frac{1}{2\beta}\exp\left(\frac{s}{\beta}\right) & \text{if } s \le 0 \\ \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(-\frac{s}{\beta}\right)\right) & \text{if } s > 0 \end{cases} \ , \tag{8}$$

where β is a learnable parameter. Following NeRF [44], the color $\hat{C}(\mathbf{r})$ for the current ray $\mathbf{r}$ is computed via numerical integration:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, \hat{\mathbf{c}}_\mathbf{r}^i \qquad T_\mathbf{r}^i = \prod_{j=1}^{i-1}\left(1 - \alpha_\mathbf{r}^j\right) \qquad \alpha_\mathbf{r}^i = 1 - \exp\left(-\sigma_\mathbf{r}^i \delta_\mathbf{r}^i\right) \ , \tag{9}$$

where $T_\mathbf{r}^i$ and $\alpha_\mathbf{r}^i$ denote the transmittance and alpha value of sample point i along ray $\mathbf{r}$, respectively, and $\delta_\mathbf{r}^i$ is the distance between neighboring sample points. Similarly, we compute the depth $\hat{D}(\mathbf{r})$ and normal $\hat{N}(\mathbf{r})$ of the surface intersecting the current ray as:

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, t_\mathbf{r}^i \qquad \hat{N}(\mathbf{r}) = \sum_{i=1}^{M} T_\mathbf{r}^i \, \alpha_\mathbf{r}^i \, \hat{\mathbf{n}}_\mathbf{r}^i \ . \tag{10}$$

3.3 Exploiting Monocular Geometric Cues

Unifying volume rendering with implicit surfaces leads to impressive 3D reconstruction results. Yet, this approach struggles with more complex scenes especially in textureless and sparsely covered regions. To overcome this limitation, we use readily available, efficient-to-compute monocular geometric priors thereby improving neural implicit surface methods.

Monocular Depth Cues. One common monocular geometric cue is a monocular depth map, which can be easily obtained via an off-the-shelf monocular depth predictor. More specifically, we use a pretrained Omnidata model [17] to predict a depth map $\bar{D}$ for each input RGB image. Note that the absolute scale is difficult to estimate in general scenes, so $\bar{D}$ must be considered as a relative cue. However, this relative depth information is provided also over larger distances in the image.

Monocular Normal Cues. Another geometric cue we use is the surface normal. Similar to the depth cues, we apply the same pretrained Omnidata model to acquire a normal map $\bar{N}$ for each RGB image. Unlike depth cues that provide semi-local relative information, normal cues are local and capture geometric detail. We hence expect that surface normals and depth are complementary to each other.

3.4 Optimization

Reconstruction Loss. Eq. (9) provides a linkage from the 3D scene representation to 2D observations. We can therefore optimize the scene representation with a simple RGB reconstruction loss:

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\|\hat{C}(\mathbf{r}) - C(\mathbf{r})\right\|_1 \ . \tag{11}$$

Here $\mathcal{R}$ denotes the set of pixels/rays in the minibatch and $C(\mathbf{r})$ is the observed pixel color.

Eikonal Loss. Following common practice, we also add an Eikonal term [20] on the sampled points to regularize SDF values in 3D space

$$\mathcal{L}_{\text{eikonal}} = \sum_{\mathbf{x}\in\mathcal{X}} \left(\left\|\nabla f_\theta(\mathbf{x})\right\|_2 - 1\right)^2 \ , \tag{12}$$

where $\mathcal{X}$ are a set of uniformly sampled points together with near-surface points [81].

Depth Consistency Loss. Besides $\mathcal{L}_{\text{rgb}}$ and $\mathcal{L}_{\text{eikonal}}$, we also enforce consistency between our rendered expected depth $\hat{D}$ and the monocular depth $\bar{D}$:

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\|(w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r})\right\|^2 \ , \tag{13}$$

where w and q are the scale and shift used to align $\hat{D}$ and $\bar{D}$ since $\bar{D}$ is defined only up to scale. Note that these factors have to be estimated individually per batch as the depth maps predicted for different batches can differ in scale and shift. Specifically, we solve for w and q with a least-squares criterion [19, 56] which has a closed-form solution (see supplementary for details).

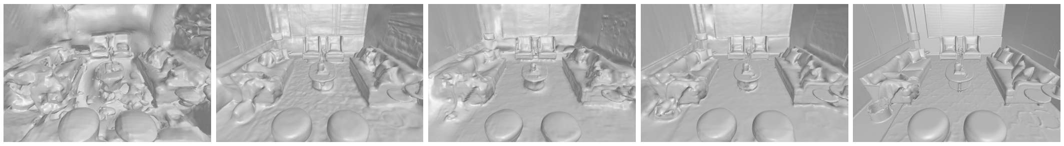
Normal Consistency Loss. Similarly, we impose consistency on the volume-rendered normal $\hat{N}$ and the predicted monocular normals $\bar{N}$ transformed to the same coordinate system with angular and L1 losses [17]:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\right\|_1 + \left\|1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r})\right\|_1 \ . \tag{14}$$

The overall loss we use to optimize our implicit surfaces jointly with the appearance network is:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_1\mathcal{L}_{\text{eikonal}} + \lambda_2\mathcal{L}_{\text{depth}} + \lambda_3\mathcal{L}_{\text{normal}} \ . \tag{15}$$

Implementation Details. We implement our method in PyTorch [52] and use the Adam optimizer [33] with a learning rate of 5e-4 for neural networks and 1e-2 for feature grids and dense SDF grids. We set λ_1, λ_2, λ_3 to 0.1, 0.1, 0.05, respectively. We sample 1024 rays per iteration and apply the error-bounded sampling strategy introduced by [81] to sample points along each ray. For MLPs and feature grids, we adapt the architecture and initialization scheme from [81] and [45], respectively. For obtaining monocular cues, we first resize each image and center crop it to 384 × 384, which we then feed as input to the pretrained Omnidata model [17]. See supplementary for more details.



Dense SDF Grid | MLP | Single-Res. Fea. Grid | Multi-Res. Fea. Grids | Ground Truth

Figure 3: **Architectural Ablation Study.** Comparing different design choices for neural implicit surface representations, we observe that a dense SDF grid leads to noisy reconstructions due to a missing smoothness bias. The MLP and the Single-Res. Fea. Grid improve results, but geometry tends to be overly smooth with missing details. The best results are obtained using Multi-Res. Fea. Grids.

4 Experiments

We first analyze different architectural design choices and perform ablation studies wrt. monocular cues and optimization time on a room-level dataset (Replica) with perfect ground truth. Next, we provide qualitative and quantitative comparisons against state-of-the-art baselines on real-world indoor scenes. Finally, we evaluate our method on object-level reconstruction for both sparse input and dense input scenarios.

Datasets. While previous neural implicit-based reconstruction methods mainly focused on single-object scenes with many input views, in this work, we investigate the importance of monocular geometric cues for scaling to more complex scenes. Thus we consider: a) Real-world indoor scans: Replica [67] and ScanNet [13]; b) Real-world large-scale indoor scenes: Tanks and Temples [34] advanced scenes; c) Object-level scenes: DTU [1] in the sparse 3-view setting from [46, 84].

Baselines. We compare against a) state-of-the-art neural implicit surfaces methods: UNISURF [49], VolSDF [81], NeuS [76], and Manhattan-SDF [21]. b) Classic MVS methods: COLMAP [62] and a state-of-the-art commercial software (RealityCapture[2]). c) TSDF-Fusion [12] with predicted monocular depth cues, where GT depth maps are used to recover the scale and shift values (cf. Eq. (13)). This baseline shows the reconstruction quality if only monocular depth cues and no implicit surface model is used.

Evaluation Metrics. For DTU, we follow the official evaluation protocol and report the Chamfer distance. For Replica and ScanNet, following [21, 42, 53, 54, 68, 92], we report the Chamfer Distance, the F-score with a threshold of 5cm, as well as a Normal Consistency measure.

4.1 Ablation Study

We first analyze different scene representation choices on the Replica dataset. Next, we ablate the impact of our geometric cues on reconstruction quality and convergence time.

	Normal C.↑	Chamfer-L_1 ↓	F-score ↑
MLP [81]	86.48	6.75	66.88
Dense SDF Grid	57.30	26.68	15.50
Single-res. Fea. Grid	86.41	6.28	64.22
Multi-res. Fea. Grids	**87.95**	**5.03**	**78.38**

Table 1: **Architectural Ablation on Replica.**

Architecture Choices for Scene Representations. We compare the four different scene geometry representations introduced in Section 3.1 and report metrics averaged over the Replica dataset in Table 1. Note that no monocular geometric cues are used here. We first observe that using a single MLP as the scene geometry representation leads to decent results, but the reconstruction tends to be over-smooth (see Table 1 and Fig. 3). For grid-based representations, optimizing a dense SDF grid leads to a significantly worse performance compared to all other neural implicit scene representations, even with careful parameter tuning. The reason is the lack of a smoothness bias: The SDF values in grid cells are all stored and optimized independently of each other, hence there is no local or global smoothness bias. In contrast, the Single-Res. Fea. Grid replaces the SDF value in each grid cell with a low-dimensional latent code, and uses a shallow MLP conditioned on these features to read out SDF values of arbitrary 3D points. This modification leads to a notable boost in reconstruction quality over the dense grid, performing similarly well as the single MLP. Using a Multi-Res. Fea. Grids as in [45] further increases performance. We observe that the Multi-Res. Fea. Grids is the best-performing grid-based model, and from now on we report results for the single MLP

[2] https://www.capturingreality.com/

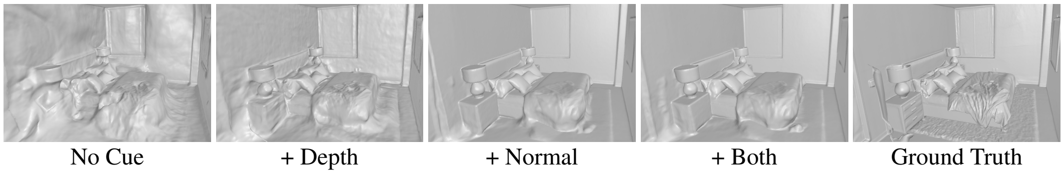
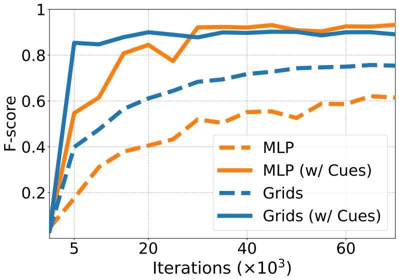


Figure 4: **Ablation of Monocular Geometric Cues.** Monocular geometric cues significantly improve reconstruction quality for both architectures (we show our MLP variant). With monocular depth cues, the recovered geometry contains more details and a better overall structure. With normal cues, missing details are added and the results become smoother. Using both cues leads to the best performance.

		Normal C.↑	Chamfer-L_1 ↓	F-score ↑
MLP	No Cues	86.48	6.75	66.88
	Only Depth	90.56	4.26	76.42
	Only Normal	91.35	3.19	85.84
	Both Cues	**92.11**	**2.94**	**86.18**
Multi-Res. Grids	No Cues	87.95	5.03	78.38
	Only Depth	90.87	3.75	80.32
	Only Normal	89.90	3.61	81.28
	Both Cues	**90.93**	**3.23**	**85.91**

(a) Different Cues



(b) Optimization Time

Table 2: **Ablation of Monocular Geometric Cues.** a.) We report reconstruction results on Replica for MLP and Multi-Res. Grids with and without the monocular geometric cues. We observe that monocular cues improve reconstruction quality for both architectures, and using both cues in combination leads to the best performance. b.) The optimization speed becomes significantly faster when incorporating monocular cues. Comparing the two architectures, we observe that the grid approach yields faster convergences while the MLP with both cues leads to the best results.

and the Multi-Res. Feature Grids. For simplicity, we will refer to the multi-resolution feature grids as *Multi-Res. Grids* or *Grids* in the following.

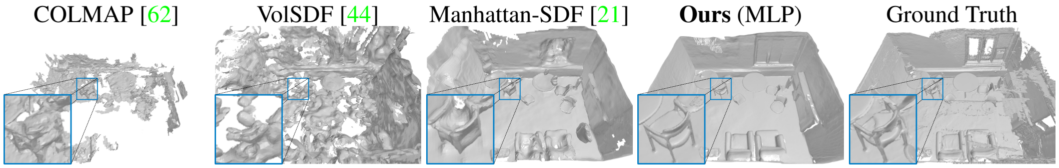
Ablation of Different Cues. We now investigate the effectiveness of different monocular geometric cues for the two chosen representations. Table 2 (a) and Fig. 4 show that, for both representations, using either one or both monocular cues significantly boosts reconstruction quality. We also find both cues to be complementary, with the best performance being achieved when using both. Similar behavior can be observed for the other two representations (cf. supplementary material). It is worth noting that the differences between the two representations become negligible when using monocular cues, indicating that those serve as a general drop-in to improve reconstruction quality.

Optimization Time. Table 2 (b) shows optimization time for the two scene representations with and without cues. We see that the Multi-Res. Grids converge faster than the single MLP model. Further, adding the monocular cues significantly speeds up the convergence process. After only 10K iterations, both representations perform better than the converged models without monocular cues. Note that the overhead required for incorporating the monocular cues into the optimization process is small and can be neglected. An extended version of Table 2 (b) can be found in the supplementary materials.

4.2 Real-world Large-scale Scene Reconstruction

To show the effectiveness of our method for large-scale scene reconstruction, we compare against various baselines on two challenging large-scale indoor datasets.

ScanNet. On ScanNet, we use the test split from [21] and also follow their evaluation protocol in which depth maps are rendered from input camera poses and then re-fused using TSDF Fusion [12] to evaluate only observed areas. We observe in Table 3 that our MLP variant outperforms all baselines achieving smoother reconstructions with more fine details. Note that we outperform concurrent work [75]. Further, we find that the MLP variant performs significantly better than using Multi-Res. Grids. ScanNet's RGB images contain motion blur and the camera poses are also noisy. This can be harmful to the local geometry updates in grid-based representations, while MLPs are more robust to this noise due to their smoothness bias.



	COLMAP [62]	UNISURF [49]	NeuS [76]	VolSDF [81]	M-SDF [21]	NeuRIS [75]	Ours (Grids)	Ours (MLP)
Chamfer-L_1 ↓	0.141	0.359	0.194	0.267	0.070	0.050	0.064	**0.042**
F-score↑	0.537	0.267	0.291	0.364	0.602	0.692	0.626	**0.733**

Table 3: **Scene-level Reconstruction on ScanNet.** Colmap and VolSDF do not lead to competitive reconstructions. Manhatten-SDF achieves compelling results, but less-observed areas are noisier and details are missing. In contrast, our approaches reconstruct smooth and details surfaces, achieving the best results. Further, MLPs are more robust to the motion blur and noise in camera poses.

Tanks & Temples. To further investigate the scalability of our method to larger-scale scenes, we conduct experiments on the Tanks and Temples advanced sets. The qualitative results in Fig. 1 show that the monocular cues significantly boost the performance of VolSDF [81], making MonoSDF the first neural implicit model achieving reasonable results on such a large-scale indoor scene. See the supplementary material for more visual comparisons and discussions.

4.3 Object-level Reconstruction from Sparse Views

We now evaluate our method on another challenging task: reconstructing single objects from sparse input views. We adopt the test split from [81, 82] on DTU and choose *three* input views following [46].

We first observe in Table 4 and Fig. 1 that without the usage of the monocular geometric cues, neither the MLP (VolSDF [81]) nor the Multi-Res. Grids work well with only 3 input views. When incorporating the cues, the results for both representations are significantly improved. Interestingly, the grid-based representations perform inferior to a single MLP as they are updated locally and do not benefit from the inductive bias of a monolithic MLP representation.

Comparing against TSDF Fusion [12] that fused predicted depth cues from all views into a TSDF volume without any optimization, we observe that this baseline has difficulties in reconstructing meaningful details due to inconsistencies in the monocular depth cues. Note that this baseline uses the GT depth maps from [16] to compute scale and shift for the depth cues. Classic MVS methods perform well quantitatively, but they heavily rely on dense matching, and in case of three input images, this inevitably leads to incomplete reconstructions (see supplementary material). In contrast, our approach combines neural implicit surface representations with the benefits from monocular geometric cues that are more robust to less-observed regions.

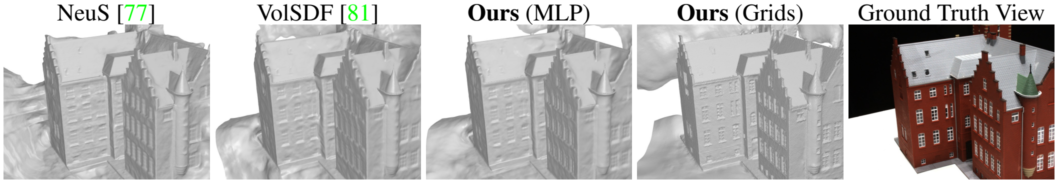
	Chamfer-L_1 ↓
TSDF-Fusion [12]	4.80
COLMAP [62]	2.56
RealityCapture	2.84
Grids	6.47
Grids w/ cues	3.68
MLP [81]	4.21
MLP w/ cues	**1.86**

Table 4: **Reconstruction on DTU (3 Views).** We report the average over the test split from [81] (see supplementary for per-object results).

4.4 Object-level Reconstruction from Dense Views

To further investigate the effectiveness and flexibility of our method, we evaluate our approach on the DTU dataset with all input views, which is a common setting in recent work [49, 77, 81]. In this experiment, we simply resize the low-resolution monocular cues to full resolution (from 384×384 to 1200×1200 pixels) while keeping the image ratio. As the original image is of size 1200×1600, the monocular cues are missing in the left and right part of the image. Therefore, we only use the monocular cues where they are available.

As shown in Table 5, our approach with MLP architecture achieves reconstruction quality similar to state-of-the-art methods [49, 77, 81]. This is reasonable as the dense input views provide enough constraints and the prior information from monocular cues is negligible. However, our method with multi-resolution feature grid architecture outperforms previous work by a large margin. We attribute this to the expressiveness of multi-resolution feature grids where monocular cues are still effective to



NeuS [77] VolSDF [81] **Ours** (MLP) **Ours** (Grids) Ground Truth View

Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
COLMAP	0.81	2.05	0.73	1.22	1.79	1.58	1.02	3.05	1.40	2.05	1.00	1.32	0.49	0.78	1.17	1.36
NeRF [44]	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49
UniSurf [49]	1.32	1.36	1.72	0.44	1.35	0.79	0.80	1.49	1.37	0.89	0.59	1.47	0.46	0.59	0.62	1.02
NeuS [77]	1.00	1.37	0.93	0.43	1.10	**0.65**	**0.57**	1.48	**1.09**	0.83	**0.52**	1.20	0.35	**0.49**	0.54	0.84
VolSDF [81]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86
Ours (MLP)	0.83	1.61	0.65	0.47	0.92	0.87	0.87	1.30	1.25	0.68	0.65	**0.96**	**0.41**	0.62	0.58	0.84
Ours(Grids)	**0.66**	**0.88**	**0.43**	**0.40**	**0.87**	0.78	0.81	**1.23**	1.18	**0.66**	0.66	**0.96**	**0.41**	0.57	**0.51**	**0.73**

Table 5: **Object-level Reconstruction on DTU Dataset will All Input Views.** We compare Chamfer distance with state-of-the-art methods. Our approach with MLP achieves similar results to previous methods, while our method with multi-resolution feature grids leads to more detailed surfaces and outperforms previous work by a large margin.

suppress noise and therefore can reconstruct smooth and detailed surfaces. We kindly refer the reader to the supplementary material for additional visual comparisons.

5 Conclusion

We have presented MonoSDF, a novel framework that systematically explores how monocular geometric cues can be incorporated into the optimization of neural implicit surfaces from multi-view images. We show that such easy-to-obtain monocular cues can significantly improve 3D reconstruction quality, efficiency, and scalability for a variety of neural implicit representations. When using monocular cues, a simple MLP architecture performs best overall, demonstrating that MLPs in principle are able to represent complex scenes, albeit being slower to converge compared to grid-based representations. Multi-resolution feature grids in general can converge fast and capture details, but are less robust to noise and ambiguities in the input images.

Limitations. The performance of our model depends on the quality of the monocular cues. Filtering strategies to handle failures of the monocular predictor are thus a promising direction to further improve reconstruction quality. We kindly refer the reader to the supplementary material for additional analysis. While we demonstrated that integrating depth and normal cues significantly improves reconstruction, exploring other cues such as occlusion edges, plane, or curvature [17, 87] is an interesting future direction. We are currently limited by the low-resolution (384 × 384 pixels) output of the Omnidata model [17] and plan to explore different ways of using higher-resolution cues. We provide some preliminary results of using high-resolution cues in the supplementary. Joint optimization of scene representations and camera parameters [4, 92] is another interesting direction, especially for multi-resolution grids, in order to better handle noisy camera poses.

Acknowledgments and Disclosure of Funding

This work was supported by an NVIDIA research gift. We thank the Max Planck ETH Center for Learning Systems (CLS) for supporting SP and the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting MN. ZY is supported by BMWi in the project KI Delta Learning (project number 19A19013O). AG is supported by the ERC Starting Grant LEGO-3D (850533) and DFG EXC number 2064/1 - project number 390727645. TS is supported by the EU Horizon 2020 project RICAIP (grant agreeement No.857306), and the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15_003/0000468). We thank the authors of Manhattan-SDF and NeuRIS for sharing results on ScanNet. We also thank Christian Reiser and Zijian Dong for proofreading.

References

[1] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision (IJCV)*, 120(2):153–168, 2016. 3, 7, 18

[2] M. Agrawal and L. S. Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001. 3

[3] M. Atzmon and Y. Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 17

[4] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies. Neural rgb-d surface reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 10

[5] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[6] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2011. 3

[7] J. D. Bonet and P. Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 1999. 3

[8] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2001. 3

[9] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[10] J. Chibane, T. Alldieck, and G. Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 5

[11] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 1999. 3

[12] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM Trans. on Graphics*, 1996. 7, 8, 9, 23

[13] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 7, 18

[14] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 23

[15] T. Do, K. Vuong, S. I. Roumeliotis, and H. S. Park. Surface normal estimation of tilted images via spatial rectifier. In *Proc. of the European Conference on Computer Vision*, Virtual Conference, August 23–28 2020. 21

[16] S. Donne and A. Geiger. Learning non-volumetric depth fusion using successive reprojections. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 9

[17] A. Eftekhar, A. Sax, J. Malik, and A. Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2, 3, 6, 10, 17, 19, 21

[18] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 2

[19] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 6, 16

[20] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman. Implicit geometric regularization for learning shapes. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020. 6

[21] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 7, 8, 9, 18, 22, 26

[22] S. Hadadan, S. Chen, and M. Zwicker. Neural radiosity. *ACM Trans. Graph.*, 2021. 2, 3, 5

[23] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, and K. Schindler. Learned multi-patch similarity. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 3

[24] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision (IJCV)*, 80:3–15, 2008. 2

[25] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. on Graphics*, 2005. 2

[26] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2005. 2

[27] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision (IJCV)*, 75(1):151–172, October 2007. 2

[28] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5

[29] P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[30] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2, 3

[31] Y. Jiang, D. Ji, Z. Han, and M. Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4

[32] M. M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Trans. on Graphics*, 32(3):29, 2013. 22

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Machine learning (ICML)*, 2015. 6

[34] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. on Graphics*, 36(4), 2017. 3, 7, 18

[35] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218, 2000. 3

[36] V. Leroy, J. Franco, and E. Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 3

[37] B. Li, Y. Huang, Z. Liu, D. Zou, and W. Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 21

[38] L. Liu, J. Gu, K. Z. Lin, T. Chua, and C. Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 5

[39] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[40] W. Luo, A. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[41] J. N. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. In *ACM Trans. on Graphics*, 2021. 3

[42] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 7, 18

[43] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *CVPR*, 2021. 23

[44] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 1, 3, 4, 5, 9, 10, 26

[45] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 2022. 2, 3, 5, 6, 7, 16

[46] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 7, 9

[47] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1

[48] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[49] M. Oechsle, S. Peng, and A. Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 1, 2, 3, 5, 7, 9, 10, 22

[50] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 4

[51] D. Paschalidou, A. O. Ulusoy, C. Schmitt, L. van Gool, and A. Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 6

[53] S. Peng, C. M. Jiang, Y. Liao, M. Niemeyer, M. Pollefeys, and A. Geiger. Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 7, 18

[54] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2, 3, 5, 7, 18

[55] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 2

[56] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 2, 6, 16, 21

[57] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger. OctNetFusion: Learning depth fusion from data. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2017. 3

[58] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[59] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems (NIPS)*, 2006. 2

[60] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV)*, 76:53–69, 2008. 2

[61] A. Saxena, M. Sun, and A. Y. Ng. Make3D: learning 3D scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31:824–840, 2009. 2

[62] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 3, 7, 9, 22, 23, 26

[63] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[64] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1997. 3

[65] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006. 3

[66] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 1

[67] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D.

Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv.org*, 1906.05797, 2019. 3, 7, 18

[68] E. Sucar, S. Liu, J. Ortiz, and A. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 7, 18

[69] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 5

[70] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4, 5

[71] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, and M. Gross. Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of VMV'03, Munich, Germany*, 2003. 16

[72] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[73] A. O. Ulusoy, A. Geiger, and M. J. Black. Towards probabilistic volumetric reconstruction using ray potentials. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2015. 3

[74] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[75] J. Wang, P. Wang, X. Long, C. Theobalt, T. Komura, L. Liu, and W. Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *ECCV*, 2022. 3, 8, 9, 21, 22

[76] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3, 5, 7, 9, 22

[77] S. Wang, M. Mihajlovic, Q. Ma, A. Geiger, and S. Tang. Metaavatar: Learning animatable clothed human models from few depth images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 9, 10

[78] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural fields in visual computing and beyond. In *EUROGRAPHICS*, 2022. 3

[79] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 3

[80] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[81] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3, 5, 6, 7, 9, 10, 22, 23, 27, 28, 34, 35

[82] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 1, 2, 3, 5, 9

[83] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to recover 3d scene shape from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021. 21

[84] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[85] Z. Yu and S. Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *CVPR*, 2020. 3

[86] Z. Yu, L. Jin, and S. Gao. P^{2}net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *ECCV*, 2020. 21

[87] Z. Yu, J. Zheng, D. Lian, Z. Zhou, and S. Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *CVPR*, pages 1029–1037, 2019. 10

[88] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[89] J. Zhang, Y. Yao, S. Li, Z. Luo, and T. Fang. Visibility-aware multi-view stereo network. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2020. 22, 31

[90] K. Zhang, G. Riegler, N. Snavely, and V. Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 2

[91] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *arXiv preprint arXiv:2106.01970*, 2021. 2

[92] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5, 7, 10, 18

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss potential negative societal impacts in our supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code and data are released.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We describe details of our computational resources in supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material for MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction

In this **supplementary document**, we first discuss architectural and implementation details in Section A. Next, we provide additional ablation studies of our monocular geometric cues for four different scene representations in Section B and report additional quantitative and qualitative results in Section C. Finally, we discuss potential negative impact of this work in Section D.

A Implementation Details

In this section, we first present an overview of 4 different architectures for neural implicit scene representations and details of Multi-Res. Grids in Section A.1 and provide details of the depth loss computation in Section A.2. Next, we describe additional details regarding our parameterizations and optimization in Section A.3 and discuss evaluation metrics in Section A.4.

A.1 Architectures

In the main paper, we investigate four different architectures as our scene representation: *Dense SDF Grid*, *Single MLP*, *Single-Res. Grid*, and *Multi-Res. Grids* . See Fig. 5 for an overview over the architectures. In the following, we provide details for Multi-Res. Feature Grids.

Multi-Res. Grids. Following Instant-NGP [45], we use L levels of feature grids with resolutions sampled in geometric space to combine features at different frequencies:

$$R_l := \lfloor R_{\text{min}} b^l \rfloor \qquad b := \exp\left(\frac{\ln R_{\text{max}} - \ln R_{\text{min}}}{L-1}\right) \quad , \tag{16}$$

where $R_{\text{min}}, R_{\text{max}}$ are the coarsest and finest resolutions, respectively. As the total number of grid cells grows cubically, we use a fixed number of parameters to store the feature grids and use a spatial hash function to index the feature vector at finer levels. More specifically, each grid contains up to T feature vectors with dimensionality F. At the coarse level where $R_l^3 \leq T$, the feature grid is stored densely. At the finer level where $R_l^3 > T$, a spatial hash function [71] is used to index the corresponding feature vector:

$$h(\mathbf{x}) = \left(\bigoplus_{i=1}^{3} \mathbf{x}_i \pi_i\right) \bmod T \quad , \tag{17}$$

where $\bigoplus$ is the bit-wise XOR operation and π_i are unique, large prime numbers. We use the default values $R_{\text{min}} = 16$, $R_{\text{max}} = 2048$, $L = 16$, $F = 2$, and $T = 2^{19}$ similar to [45] in all experiments.

A.2 Depth Consistency Loss

We enforce consistency between our rendered expected depth $\hat{D}$ and the monocular depth $\bar{D}$ with a scale invariant loss function:

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| (w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r}) \right\|^2 \quad , \tag{18}$$

where w and q are the scale and shift used to align $\hat{D}$ and $\bar{D}$ since $\bar{D}$ is given only up to scale. Specifically, we solve w and q with a least-squares criterion [19, 56]:

$$(w, q) = \arg\min_{w,q} \sum_{\mathbf{r} \in \mathcal{R}} \left(w\hat{D}(\mathbf{r}) + q - \bar{D}(\mathbf{r}) \right)^2 . \tag{19}$$

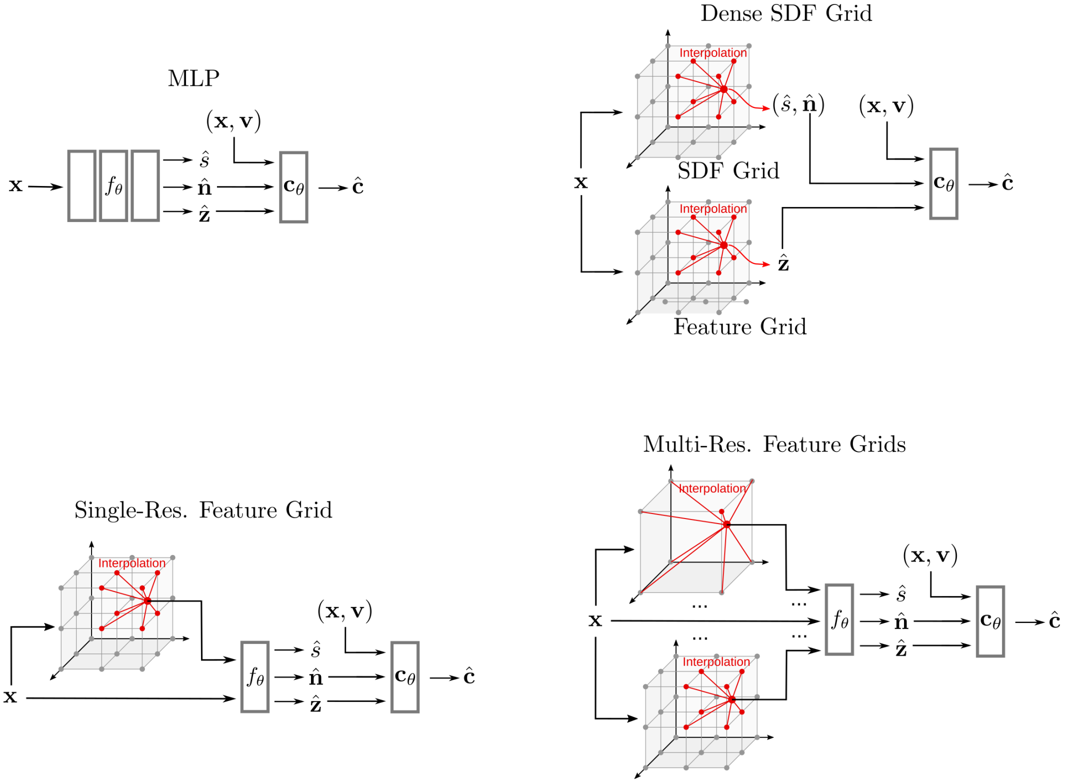


Figure 5: **Architectures.** We show an overview over four different scene representations considered in this paper.

w and q can be efficiently computed as follows: Let $\mathbf{h} = (w, q)^T$ and $\mathbf{d_r} = (\hat{D}(\mathbf{r}), 1)^T$, then Eq. (19) can be rewrite as:

$$\mathbf{h}^{\text{opt}} = \arg\min_{\mathbf{h}} \sum_{\mathbf{r} \in \mathcal{R}} \left(\mathbf{d}_\mathbf{r}^T \mathbf{h} - \bar{D}(\mathbf{r}) \right)^2 . \tag{20}$$

which has the closed-form solution:

$$\mathbf{h} = \left(\sum_{\mathbf{r}} \mathbf{d_r} \mathbf{d}_\mathbf{r}^T \right)^{-1} \left(\sum_{\mathbf{r}} \mathbf{d_r} \bar{D}(\mathbf{r}) \right) . \tag{21}$$

Note that we estimate w and q individually at each iteration for a batch of randomly sampled rays within a single image because depth maps predicted by the monocular depth predictor can differ in scale and shift and the underlying scene geometry changes at each iteration.

A.3 Additional Details

For our single MLP architecture, we use an 8-layer MLP with hidden dimension 256. We use a two-layer MLP with hidden dimension 256 for the SDF prediction for both, Single-Res. Grid and Multi-Res. Grids. We implement the color network with a two-layer MLP with hidden dimension 256 and use it for all architectures. We use Softplus activation for geometric network and use ReLU activation for the color network. We explicitly initialize the SDF grid with a sphere and use the geometric initialization from [3] for other architectures. For obtaining monocular cues, we first resize each image and center crop it to 384 × 384, which we then feed as input to the pretrained Omnidata model [17]. The output depth and normal maps have the same resolution of 384 × 384. As a result, we use the same resolution for RGB images, depth cues and normal cues and adjust camera intrinsics accordingly for all experiments. We optimize our model for 200k iterations which takes about 6 hours and 11 hours for our Multi-Res. Grids and MLP, respectively, on a single NVIDIA RTX3090 GPU.

A.4 Evaluation Metrics

For the DTU dataset [1], we follow the official evaluation protocol and report the reconstruction quality with: *Accuracy*, *Completeness* and *Chamfer Distance*. *Accuracy* measures how close the reconstructed points are to the ground truth and is defined as the mean distance of the reconstructed points to the ground truth. *Completeness* measures to what extent the ground truth points are recovered and is defined as the mean distance of the ground truth points to the reconstructed points. *Chamfer Distance* is the mean of *Accuracy* and *Completeness*. It measures the overall reconstruction quality. For efficiency, we use the Python script[3] to compute these evaluation metrics.

Metric	Definition
Acc	$\underset{\mathbf{p}\in P}{\text{mean}} \left(\underset{\mathbf{p}^*\in P^*}{\min} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1 \right)$
Comp	$\underset{\mathbf{p}^*\in P^*}{\text{mean}} \left(\underset{\mathbf{p}\in P}{\min} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1 \right)$
Chamfer	$\frac{\text{Acc}+\text{Comp}}{2}$
Precision	$\underset{\mathbf{p}\in P}{\text{mean}} \left(\underset{\mathbf{p}^*\in P^*}{\min} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1 < 0.05 \right)$
Recall	$\underset{\mathbf{p}^*\in P^*}{\text{mean}} \left(\underset{\mathbf{p}\in P}{\min} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1 < 0.05 \right)$
F-score	$\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$
Normal-Acc	$\underset{\mathbf{p}\in P}{\text{mean}} \left(\mathbf{n}_\mathbf{p}^T \mathbf{n}_{\mathbf{p}^*} \right) \ \text{s.t.}\ \mathbf{p}^* = \underset{p^*\in P^*}{\text{argmin}} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1$
Normal-Comp	$\underset{\mathbf{p}^*\in P^*}{\text{mean}} \left(\mathbf{n}_\mathbf{p}^T \mathbf{n}_{\mathbf{p}^*} \right) \ \text{s.t.}\ \mathbf{p} = \underset{p\in P}{\text{argmin}} \Vert\mathbf{p}-\mathbf{p}^*\Vert_1$
Normal-Consistency	$\frac{\text{Normal-Acc}+\text{Normal-Comp}}{2}$

Table 6: **Evaluation Metrics.** We show the evaluation metrics with their definitions that we use to measure reconstruction quality. P and P^* are the point clouds sampled from the predicted and the ground truth mesh. $\mathbf{n}_\mathbf{p}$ is the normal vector at point $\mathbf{p}$.

For Replica [67] and ScanNet [13], we report *Accuracy*, *Completeness*, *Chamfer Distance*, *Precision*, *Recall*, and *F-score* with a threshold of 5cm following [21, 68, 92]. We further report *Normal Consistency* for the Replica dataset following [21, 42, 53, 54, 68, 92] as near-perfect ground truth is available. These metrics are defined in Table 6.

For the Tanks and Temples dataset [34], we submit our reconstruction results to the official evaluation server[4] and report the provided F-score.

B Ablation

In this section, we first conduct several ablation studies to verify the effectiveness of our method, including using geometric cues with different scene representations in Section B.1, different architecture configurations in Section B.2, different number of input views in Section B.3, different cues predictors in Section B.4. Next, we analyze the optimization time of our framework in Section B.5.

		Test Split			Train Split		
		Normal C.↑	Chamfer-L_1 ↓	F-score ↑	Normal C.↑	Chamfer-L_1 ↓	F-score ↑
Dense SDF Grid	No Cues	57.30	26.68	15.50	60.86	17.34	26.34
	Only Depth	71.81	12.60	30.09	73.15	13.09	30.30
	Only Normal	73.95	13.62	33.34	77.80	11.30	**42.45**
	Both Cues	**76.47**	**11.39**	**37.27**	**80.05**	**10.09**	41.57
MLP	No Cues	86.48	6.75	66.88	86.69	7.48	63.24
	Only Depth	90.56	4.26	76.42	91.80	3.59	85.67
	Only Normal	91.35	3.19	85.84	92.85	4.23	85.58
	Both Cues	**92.11**	**2.94**	**86.18**	**93.86**	**2.63**	**92.12**
Single-Res. Grids	No Cues	86.41	6.28	64.22	86.54	6.63	67.26
	Only Depth	90.50	3.94	78.42	91.3	3.29	86.34
	Only Normal	89.60	4.07	76.47	**91.87**	3.13	85.96
	Both Cues	**90.59**	**3.56**	**83.34**	**91.87**	**2.98**	**88.23**
Multi-Res. Grids	No Cues	87.95	5.03	78.38	87.15	5.83	72.13
	Only Depth	90.87	3.75	80.32	91.25	3.41	**87.04**
	Only Normal	89.90	3.61	81.28	91.11	3.59	84.02
	Both Cues	**90.93**	**3.23**	**85.91**	**91.41**	**3.14**	86.87

Table 7: **Ablation of Monocular Geometric Cues on Replica.** Our monocular geometric cues significantly improve reconstruction quality across all architectures.

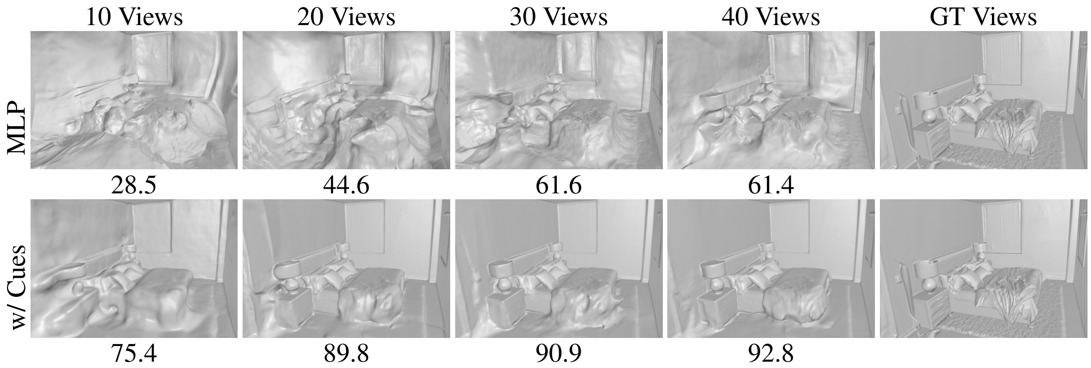


Figure 6: **Ablation of Different Number of Input Views on the Replica Dataset.** We show F-score under each image. We observe that using more input views for training improves reconstruction quality. Further, adding monocular geometric cues improves reconstruction quality. When using only 10 input views, the MLP fails to reconstruct reasonable results while using monocular geometric cues significantly improves results.

B.1 Ablation of Different Cues

To evaluate the effectiveness of our monocular geometric cues for different scene representations, we conduct ablation studies on the Replica dataset with our four different scene representations. Note that as the Replica dataset is part of the training set of Omnidata (making up 0.46% of the entire training data) [17], we split the evaluation into the train/test split of Omnidata [17].

As shown in Table 2 and Fig. 8, our geometric cues improve reconstruction quality significantly independent of the underlying scene representations. We observe that using both, depth cues and normal cues, leads to the best results, indicating the complementary nature of the different cues. We further observe that the reconstruction quality as well as the improvements from adding geometric cues are similar for the train and test split of Omnidata, showing that the monocular predictor did not overfit to the training data.

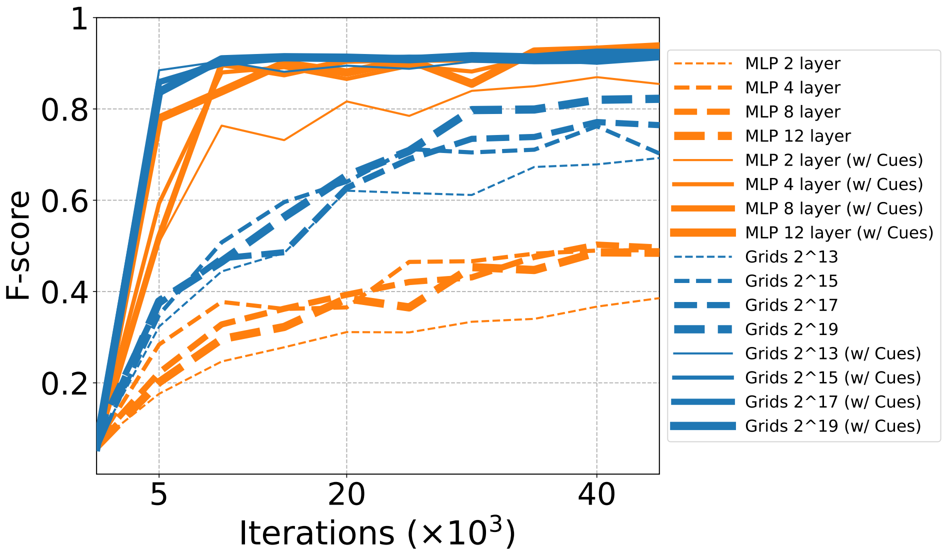


Figure 7: **Optimization Processes Using Different Architecture Configurations.** Using monocular geometric cues improves reconstruction quality and convergence speed independent of the network configurations.

Model configuration	Num. Params
MLP (2 layers)	0.15M
MLP (4 layers)	0.26M
MLP (8 layers)	0.53M
MLP (12 layers)	0.8M
Multi-res. Feature Grids (hash table size 2^{13})	0.41M
Multi-res. Feature Grids (hash table size 2^{15})	1.11M
Multi-res. Feature Grids (hash table size 2^{17})	3.67M
Multi-res. Feature Grids (hash table size 2^{19})	12.67M

Table 8: **Number of Learnable Parameters Using Different Architecture Configurations.**

B.2 Ablation of Different Architecture Configurations

In order to evaluate the performance with different model capacities, we consider MLPs with a different number of layers and Multi-res. Feature Grids with different sizes of the hash table. We list the number of learnable parameters using different architecture configurations in the Table 8, and show their performance over the optimization processes in Fig. 7. Our experiments show that using monocular geometric cues improves reconstruction quality and convergence speed independent of the network configuration.

B.3 Ablation of Different Numbers of Input Views

We ran experiments with a different number of input images and monocular geometric cues. As shown in Fig. 6, adding the monocular geometric cues leads to consistent improvements across different numbers of input views.

Method	F-score
MLP	64.2
w/ MiDaS [56]	68.6
w/ LeReS [83]	72.6
w/ Omnidata [17]	86.7

(a) Different Depth

Method	F-score
MLP	64.2
w/ Tilted [15]	45.6
w/ Omnidata [17]	92.2

(b) Different Normal

Method	F-score
MLP	64.2
w/ Self-supervised [37, 86]	45.6
w/ Omnidata [17]	86.7

(c) Self-supervised Depth

Table 9: **Ablation of Different Monocular Cues Predictors.** a.) Adding monocular depth improves performance over a single MLP without cues. Unsurprisingly, better depth predictors lead to better performance, with the state-of-the-art Omnidata model giving the best results. b.) Adding monocular normal improve the results. Similarly, using normals predicted by the state-of-the-art Omnidata model leads to the best performance. c.) Using self-supervised depth estimator degrades performance. We hypothesize that this is due to the weaker performance of the self-supervised model which is also trained with an RGB loss and hence suffers from the under-constrained problem of recovering geometry from multi-view images.

B.4 Ablation of Different Monocular Cues Predictors

To further analyze the robustness of our approach to monocular geometric cues of different levels of quality, we further tested our model with different supervised depth predictors [56, 83], normal predictors [15], and self-supervised depth predictors [37, 86]. The result is shown in Table 9. We found that using the state-of-the-art Omnidata model leads to the best results, indicating that the development of better geometric cues will further improve the performance of our approach.

B.5 Optimization Time

Adding monocular geometric cues to the optimization introduces a small overhead to our overall optimization pipeline. First, predicting these cues with a pretrained Omnidata model is very efficient (36 FPS with an NVIDIA RTX3090 GPU). For example, it takes less than 26 seconds to predict both depth maps and normal maps for 464 images for one of the ScanNet scene. Note that this only needs to be done once and that we measure FPS with a batch size of one; using a larger batch size will result in a speed up. Second, we volume render depth and normals during optimization in order to apply a loss against these monocular cues. This overhead is also small and can be neglected since the most expensive part wrt. compute is the inference of the network. For our MLP variant, the additional flops for volume rendering depth and normal is only 0.0002% of the MLP inference time. While adding monocular geometric cues introduce a small overhead, the improvements in terms of reconstruction quality and converge speed are significant. As shown in Table 2 (b) in the main paper, with only 5k iterations, our Multi-Res. Grids representation with cues performs better than the converged models without geometric cues, which implies a $40\times$ speed up (5k vs. 200k).

C Additional Results

In this section, we provide more qualitative and quantitative results for three datasets: ScanNet (Section C.1), Tanks and Temples (Section C.2), and DTU (Section C.4).

C.1 ScanNet

We report quantitative results with all metrics for ScanNet in Table 10 and show more visualizations in Fig. 9. Compared to state-of-the-art methods, our approach with MLP architecture produces significantly better reconstructions both visually as well as quantitatively. It's worth noting that we perform better than concurrent work [75] even though they have some filtering mechanism.

	Acc↓	Comp↓	Chamfer-L_1 ↓	Prec↑	Recall↑	F-score↑
COLMAP [62]	0.047	0.235	0.141	0.711	0.441	0.537
UNISURF [49]	0.554	0.164	0.359	0.212	0.362	0.267
NeuS [76]	0.179	0.208	0.194	0.313	0.275	0.291
VolSDF [81]	0.414	0.120	0.267	0.321	0.394	0.346
Manhattan-SDF [21]	0.072	0.068	0.070	0.621	0.586	0.602
NeuRIS [75]	0.050	0.049	0.050	0.717	0.669	0.692
Ours (Multi-Res. Grids)	0.072	0.057	0.064	0.660	0.601	0.626
Ours (MLP)	**0.035**	**0.048**	**0.042**	**0.799**	**0.681**	**0.733**

Table 10: **Scene-level 3D Reconstruction on ScanNet.** We report reconstruction results for our methods and baselines on ScanNet (baselines from [21]). We find that our approaches outperform previous state-of-the-art, highlighting the effectiveness of the use of monocular geometric priors. As ScanNet's RGB images contain motion blur and the camera poses are partially noisy, we further observe that the MLP architecture is more robust to this noise and achieves the best results. It's worth noting that we perform better than concurrent work [75] even though they have some filtering mechanism.

	Grid	Grid w/ cues	MLP [81]	MLP w/ cues
Auditorium	1.36	**3.17**	1.60	3.09
Ballroom	2.67	**3.70**	2.04	2.47
Courtroom	7.84	**13.75**	8.03	10.00
Museum	4.12	**5.68**	2.96	5.10
mean	4.00	**6.58**	3.66	5.165

Table 11: **Evaluation Results on the Tanks and Temples Dataset Advanced Set.** We evaluate the reconstructed meshes using the official server and report the F-score with 10mm. Our monocular geometric cues improve the reconstruction quality for all scenes.

C.2 Tanks and Temples

We show quantitative results for Tanks and Temples in Table 11. Qualitative comparisons of with or without monocular cues of our MLP variant are shown in Fig. 10 and Fig. 11. Fig. 12 and Fig. 13 show qualitative comparison of our Mulit-Res. Grids. Our monocular geometric cues significantly improve the reconstruction quality.

We further show an additional comparison against state-of-the-art MVS methods in Fig. 14. We use a pretrained Vis-MVSNet [89] to predict depth maps for the input images and fuse them to point clouds follow the official code.[5] Next, we use Meshlab's screened Poisson reconstruction [32] to reconstruct a mesh from point clouds with default parameters. We observe that our reconstructions are more complete which is useful for many applications. Further, reconstructing a mesh from point clouds involves lossy post-processing, leading to floating artifacts and bloated areas in less-observed areas.

C.3 Preliminary Results of Using High-resolution Monocular Cues

In the main paper, we center-crop each image and resize it to 384 × 384. Then, we use a pretrained Omnidata model to predict depth maps and normal maps which are also of size 384 × 384. While we have shown that training at a resolution of 384 × 384 produces impressive results, we believe that exploring different ways to generate and integrate higher resolution cues could further improve reconstruction quality. Here, we provide a proof-of-concept experiment for generating higher resolution monocular cues and integrating them into our model. We use a divide-and-conquer method for generating high-resolution cues. First, we partition a high-resolution image to multiple overlapping sub-images, and we predict monocular depth and normal for each sub-image. Next, we merge these predictions. We use Eq. 21 to align the depth maps and solve the rotation for the normal

[3] https://github.com/jzhangbs/DTUeval-python

[4] https://www.tanksandtemples.org/

[5] Available at https://github.com/jzhangbs/Vis-MVSNet

	TSDF [12]	COLMAP	RealityCapture	MLP [81]	MLP w/ cues	Multi-Res. Grids	Multi-Res. Grids w/ cues
scan24	5.01	4.45	4.19	5.24	**3.47**	6.46	5.24
scan37	5.28	4.67	3.85	5.09	**3.61**	8.30	6.37
scan40	5.09	2.51	2.26	3.99	**2.10**	7.03	2.52
scan55	4.63	1.90	2.49	1.42	**1.05**	5.87	1.95
scan63	5.03	2.81	3.49	5.10	**2.37**	6.92	6.64
scan65	4.50	2.92	3.97	4.33	**1.38**	3.09	2.05
scan69	4.55	2.12	1.91	5.36	**1.41**	5.34	4.25
scan83	4.88	2.05	2.49	3.15	**1.85**	6.03	1.81
scan97	6.22	2.93	2.37	5.78	**1.74**	6.93	5.27
scan105	3.89	2.05	2.27	2.07	**1.10**	6.01	2.54
scan106	5.67	2.01	2.90	2.79	**1.46**	6.14	3.85
scan110	3.80	N/A	4.60	5.73	**2.28**	7.62	3.89
scan114	4.67	1.10	1.38	**1.20**	1.25	6.27	1.90
scan118	4.51	2.72	2.57	5.64	**1.44**	7.59	3.12
scan122	4.35	1.64	1.76	6.20	**1.45**	6.47	3.84
mean	4.80	2.56	2.84	4.21	**1.86**	6.47	3.68

Table 12: **Evaluation Results on the DTU Dataset with 3 Input Views.** Note the COLMAP fails on scan110 so we take the average over the remaining 14 scenes. We find that without geometric cues, neither Grids nor MLP works well with only 3 input views. When incorporating the monocular geometric cues, the results for both representations are significantly improved. Interestingly, the grid-based representations perform inferior to a single MLP as they are updated only locally and do not have an inductive smoothness bias compared to a monolithic MLP representation.

maps. An example of the resulting high-resolution monocular cues is shown in Fig. 15. We found that our high-resolution cues contain more fine details compared to low-resolution cues. Note that using other methods for generating high-resolution depth maps is also possible, e.g., [43]. We then use the high-resolution cues to train our model, and the results are shown in Fig. 16. We observe significant improvements when using high-resolution monocular cues.

C.4 DTU

Geometry. We show per-scene quantitative results on the DTU dataset with 3 input-views in Table 12 and more qualitative results in Fig. 17. We find that without the monocular geometric cues, both MLP and Multi-Res. Grids fail to produce satisfying reconstructions, while with our monocular cues, both methods are improved and are able to reconstruct high-quality meshes. We further show more visualizations on the DTU dataset using all input views in Fig. 19. Compared to state-of-the-art methods, our approach with multi-resolution feature grids produces more accurate reconstructions.

Novel View Synthesis. We further compare our novel view synthesis results on the DTU dataset with three input views. As shown in Table 13 and Fig. 18, using monocular geometric cues improves novel view synthesis results significantly.

	PSNR
MLP [81]	17.65
MLP w/ cues	**23.64**

Table 13: **Novel view synthesis results on DTU (3 Views).**

Weight Annealing. As the monocular depth and normal predictor is not perfect, we exponentially anneal the loss weight for the monocular depth consistency and normal consistency loss, λ_2 and λ_3, to 0 during the first 200 epochs of optimization. Qualitative comparison in Fig. 20 verifies the importance of weight annealing.

Failure cases. We show a failure case on DTU with 3 input views in Fig. 21. The reconstructed mesh duplicates the object in front of each camera frustum. One reason is that the monocular depth cues that we use are only up to scale so they do not guarantee multi-view consistency. Therefore, the optimization is still underconstrained since the input RGB images and monocular cues can be explained by individual objects in front of the image plane. One possible solution would be incorporating explicit multi-view constraints such as using sparse point clouds from COLMAP [62] as an additional supervision [14].

D Societal Impact

Our method can faithfully reconstruct a 3D scene which can be used for application ranging from virtual reality to robotics. However, it can also have potential negative societal impact. First, our method relies on a general purpose monocular geometric predictor that needs to be trained on large amounts of data and with large computational resources, which potentially has a negative impact on global climate change. Second, accurate reconstruction of a scene may raise privacy concerns that need to be addressed carefully. Finally, accurate geometry reconstructed by our method can potentially be used for malicious purposes.

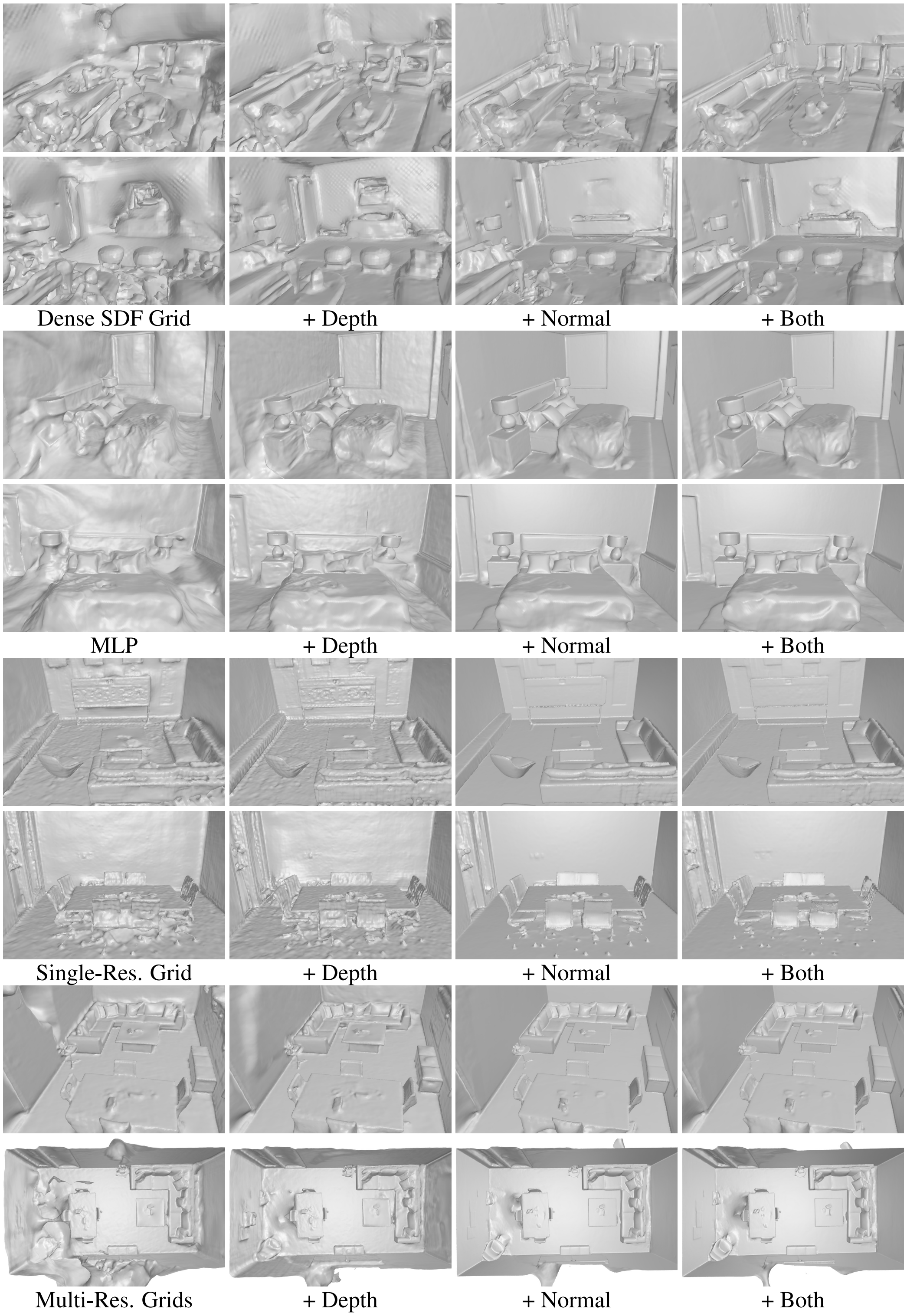


Figure 8: **Ablation of Monocular Geometric Cues on the Replica Dataset.** Monocular geometric cues significantly improve reconstruction quality for all architectures. With monocular depth cues, the recovered geometry contains more details and a better overall structure. Similarly, with our normal cues, missing details are added and the results become smoother. Using both cues leads to the best performance. Zoom in for details.

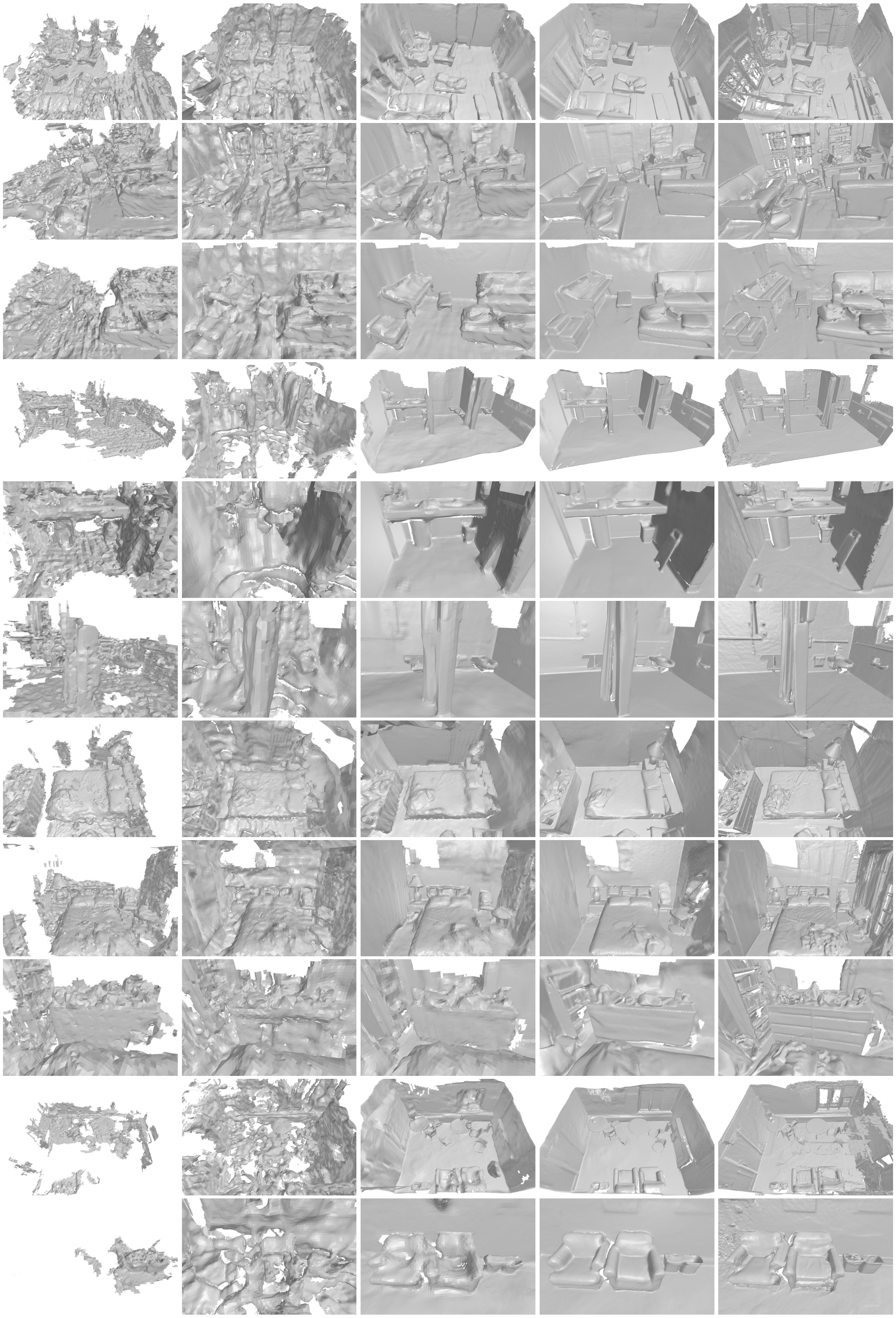


Figure 9: **Qualitative Comparison on ScanNet.** We show different views for each scene. Our method leads to better results containing smooth surfaces and detailed reconstructions compared against state-of-the-art neural implicit methods.

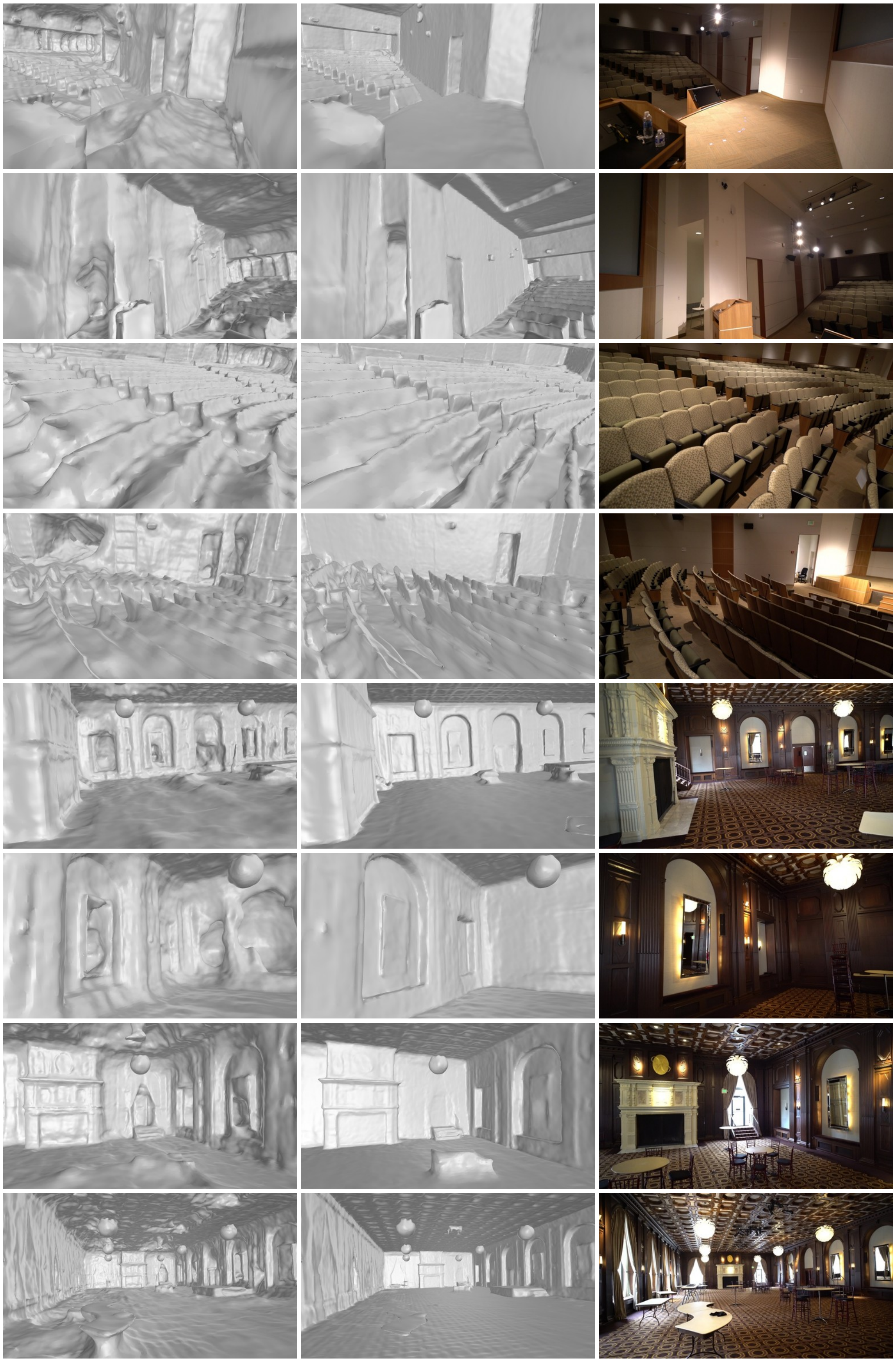


Figure 10: **Qualitative Comparison on Tanks & Temples.** We use a single MLP as the scene geometry representation [81] and compare the reconstruction when using monocular cues or not on Auditorium and Ballroom.

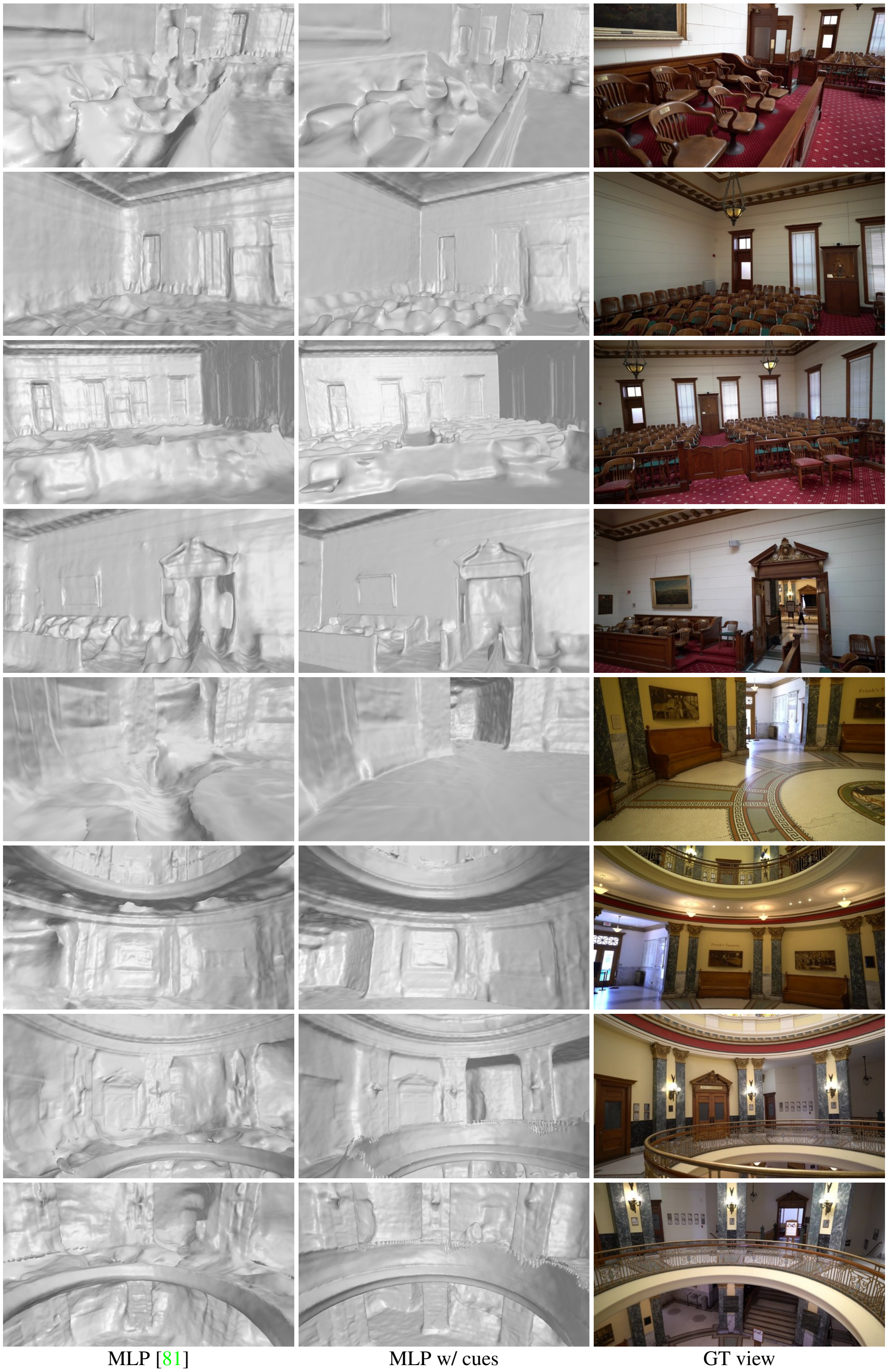


Figure 11: **Qualitative Comparison on Tanks & Temples Dataset.** We use a single MLP as the scene geometry representation [81] and compare the reconstruction quality when using monocular cues or not on Courtroom and Museum.

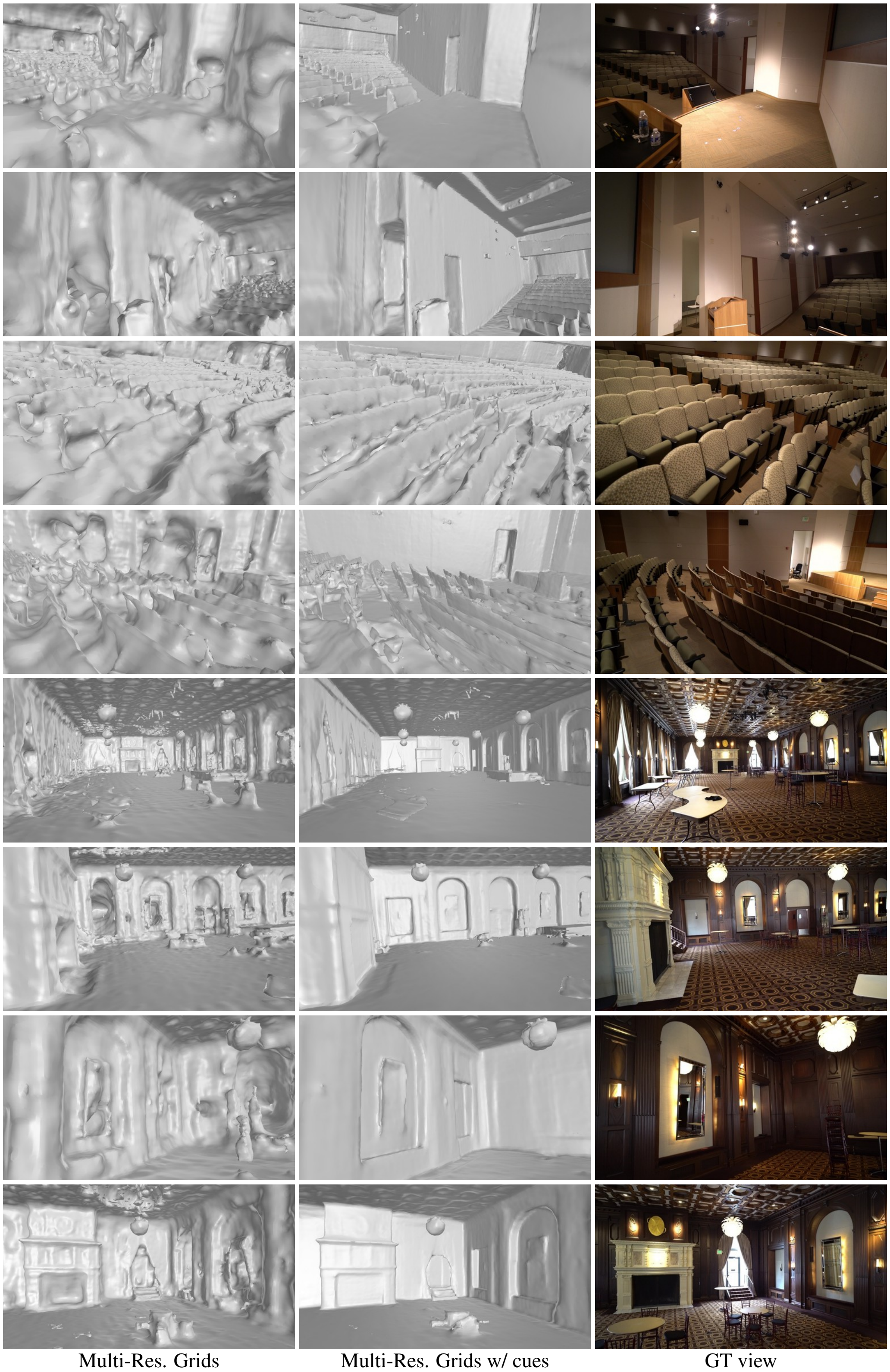


Figure 12: **Qualitative Comparison on Tanks & Temples.** We use Multi-Res. Grids as the scene geometry representation and compare the reconstruction when using monocular cues or not on Auditorium and Ballroom.

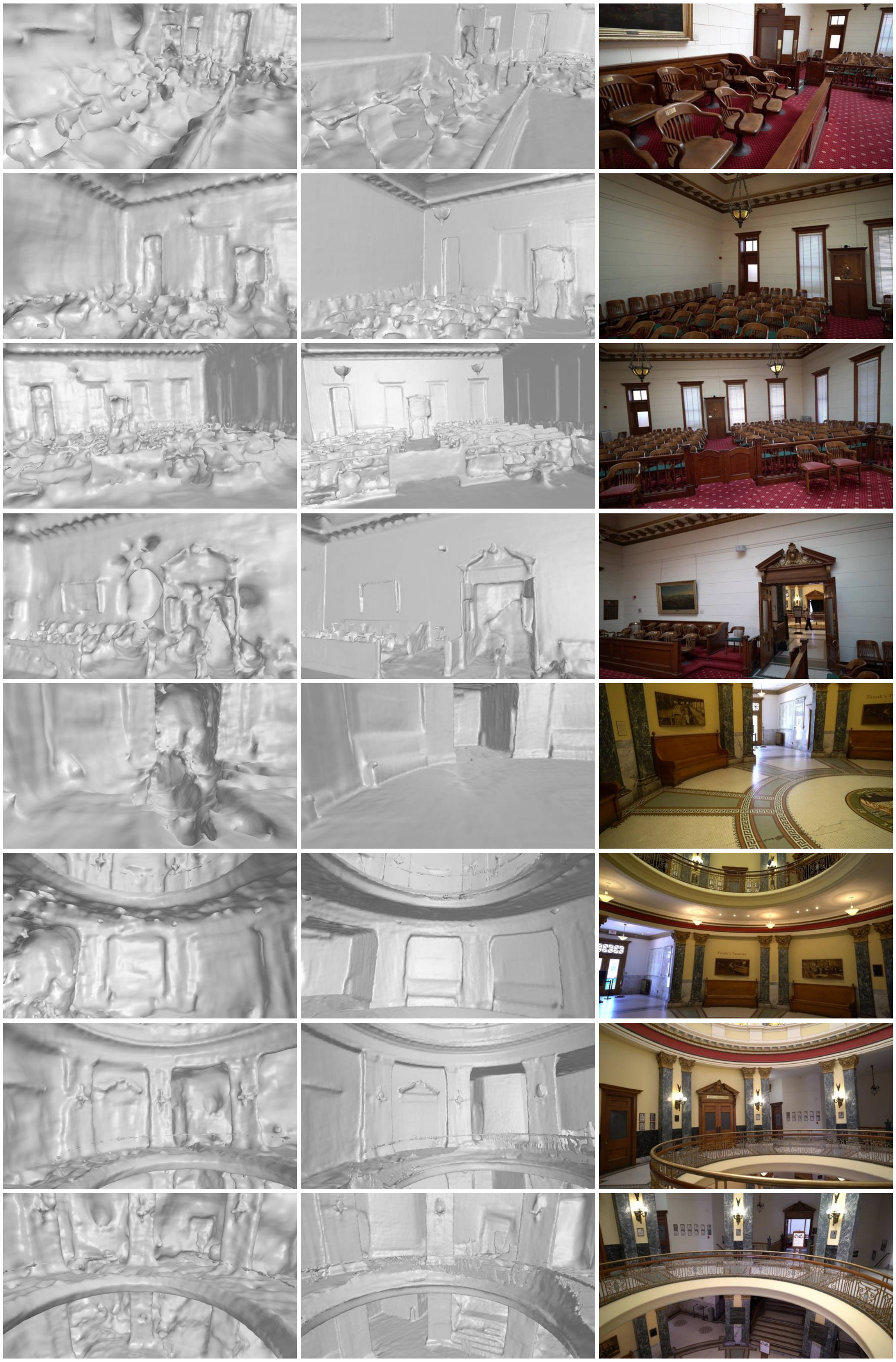


Figure 13: **Qualitative Comparison on Tanks & Temples.** We use Multi-Res. Grids as the scene geometry representation and compare the reconstruction when using monocular cues or not on Courtroom and Museum.

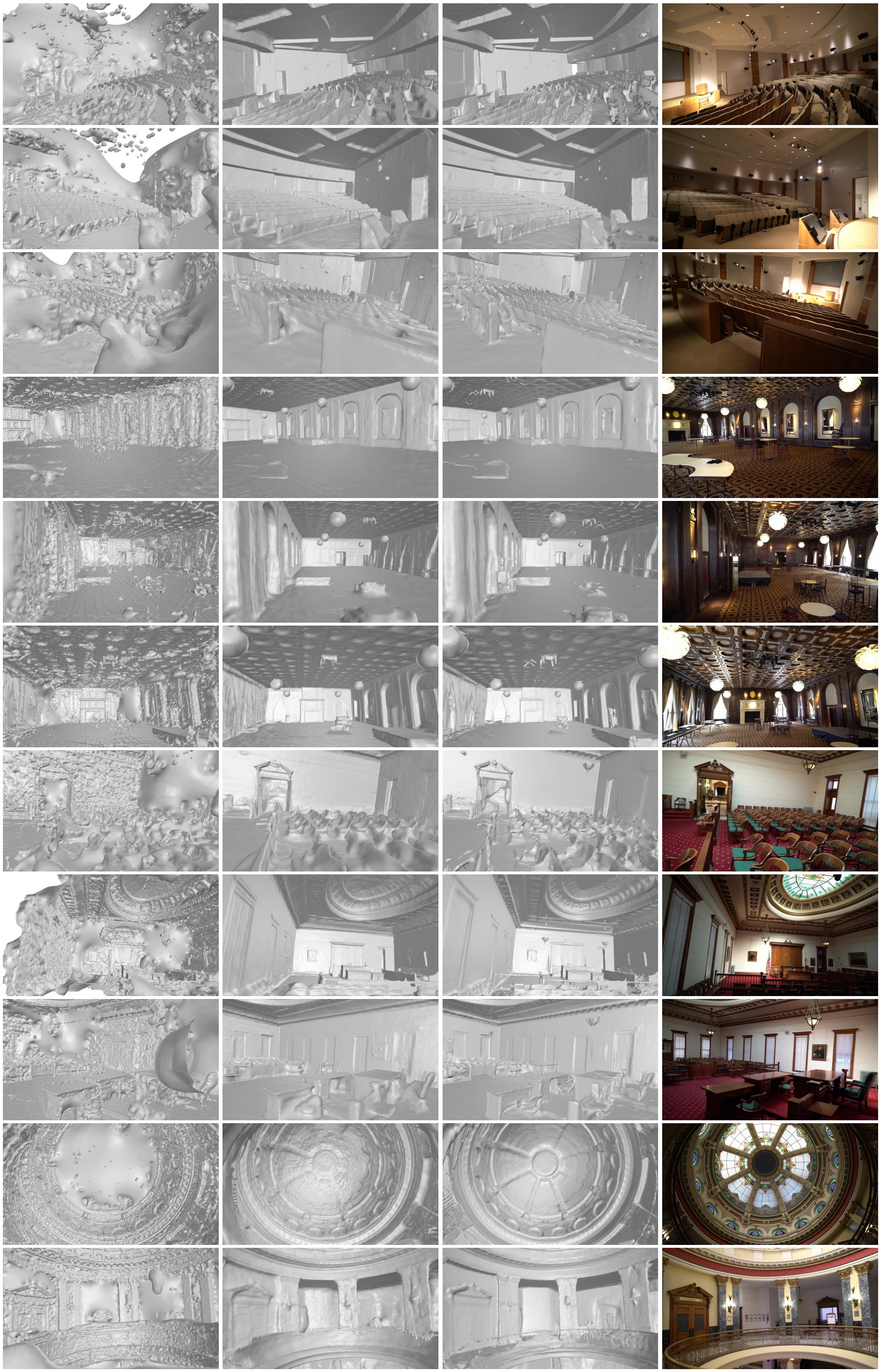
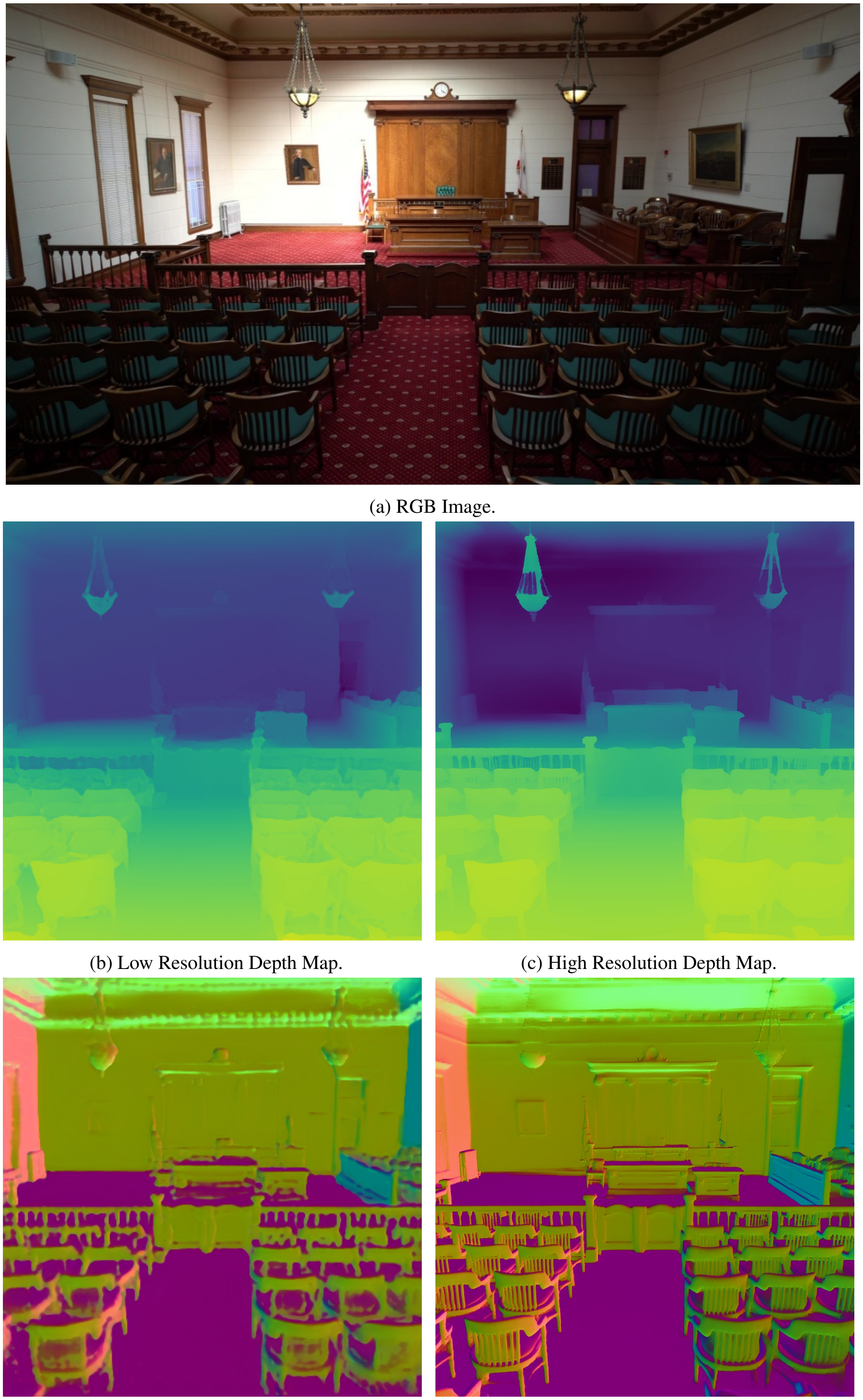


Figure 14: **Qualitative Comparison on Tanks & Temples.**



(a) RGB Image.

(b) Low Resolution Depth Map.

(c) High Resolution Depth Map.

(d) Low Resolution Normal Map.

(e) High Resolution Normal Map.

Figure 15: **Visual Comparison of Different Resolution Monocular Cues.**

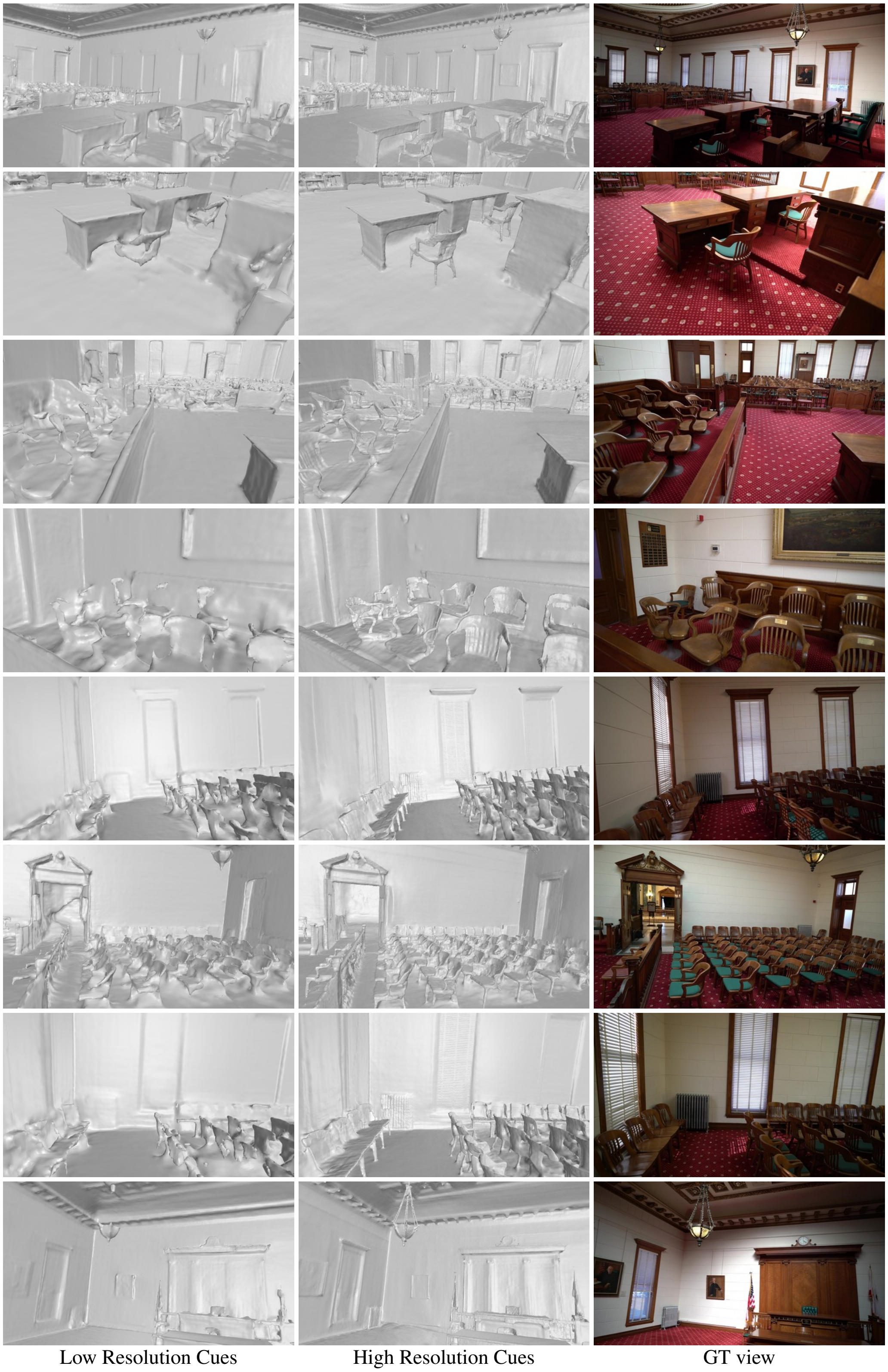


Figure 16: **Qualitative Comparison of Low Resolution Cues and High Resolution cues on Tanks & Temples.** We use Multi-Res. Grids as the scene geometry representation and compare the reconstruction when using different resolution of monocular cues.

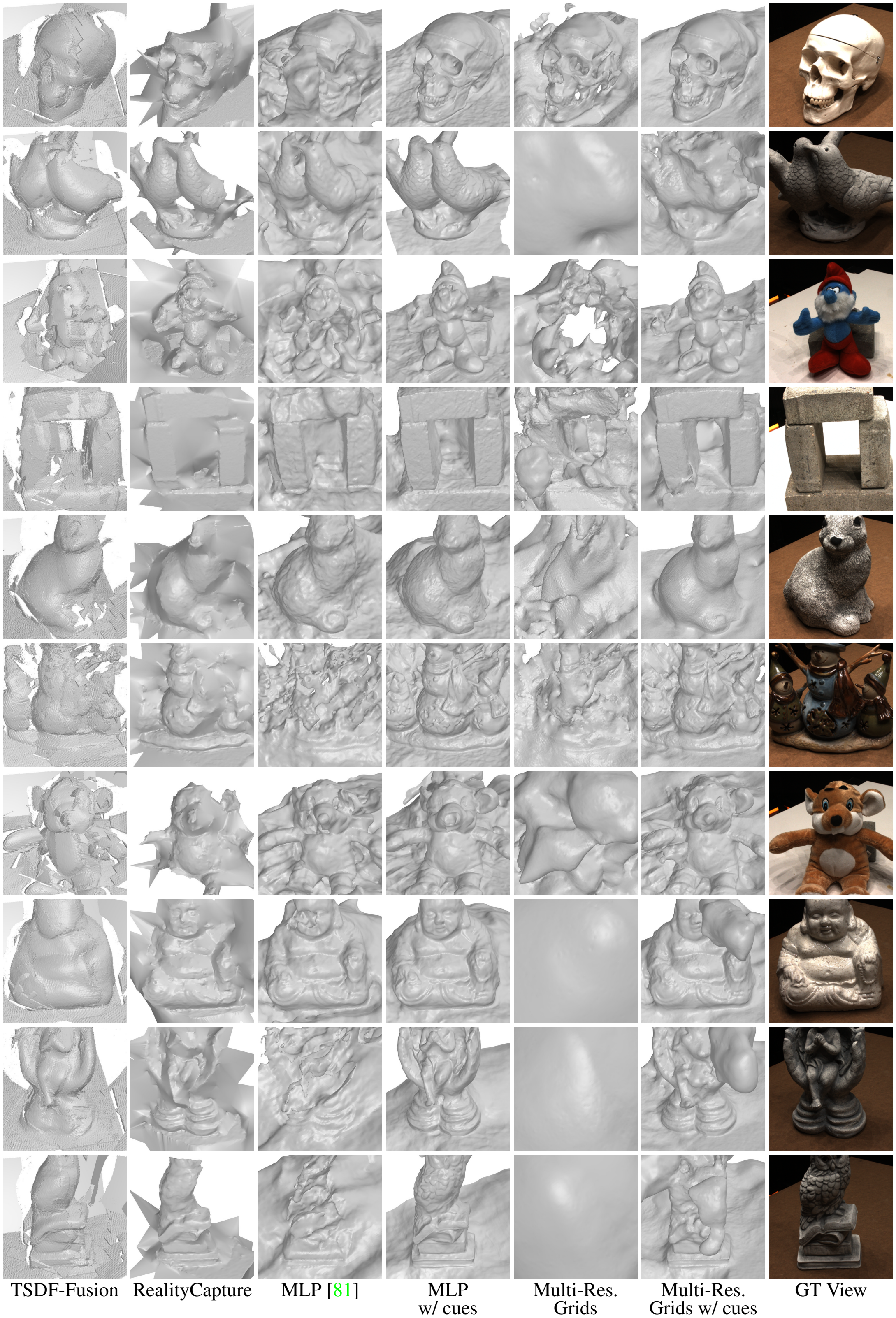


Figure 17: **Qualitative Comparison on the DTU Dataset with 3 Input Views.** Adding monocular geometric cues improves 3D reconstruction quality for both MLP and Multi-Res. Grids. We show a failure case on the last row.

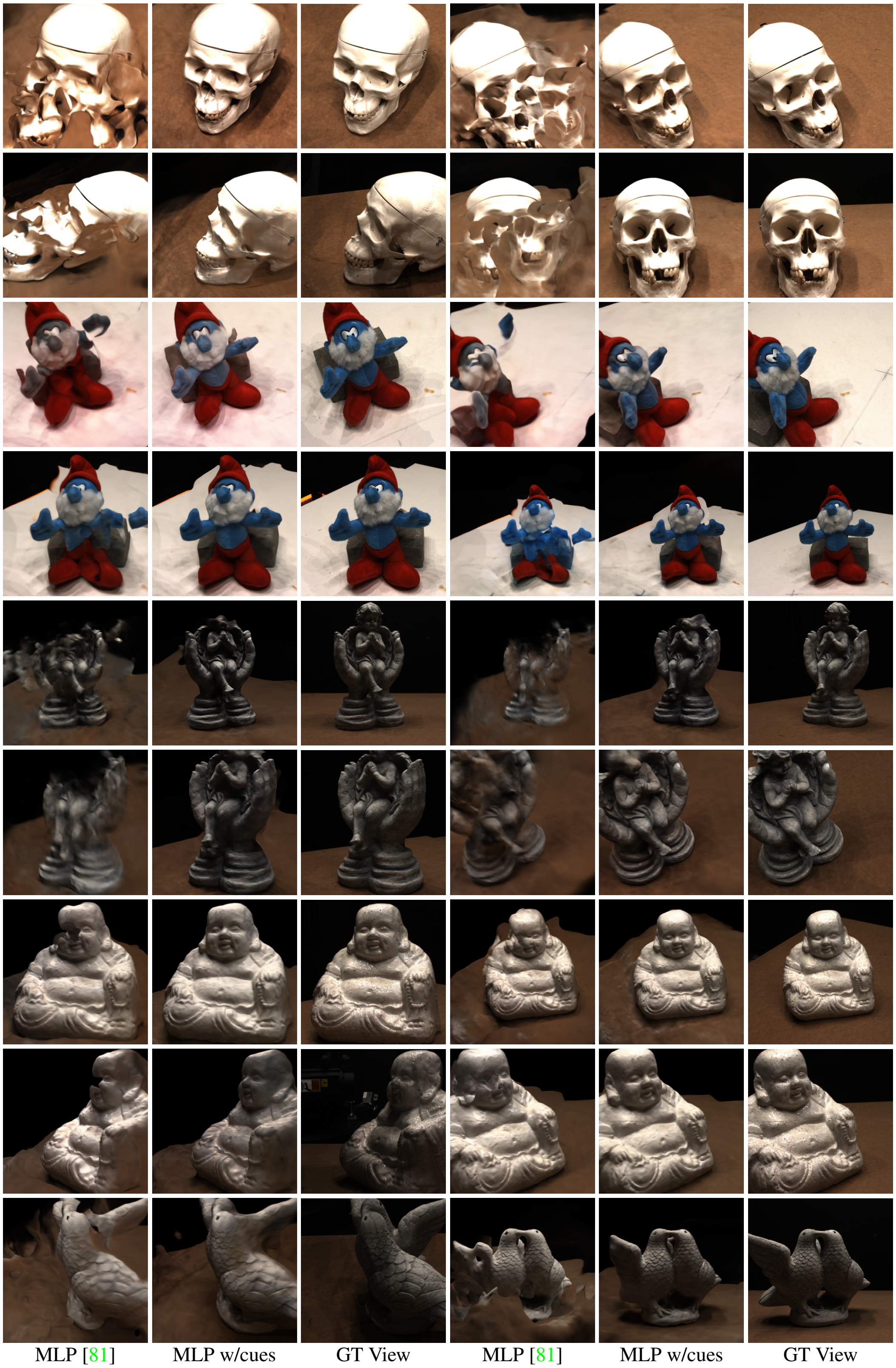


Figure 18: **Qualitative Comparison of Novel View Synthesis on the DTU Dataset with 3 Input Views.** Adding monocular geometric cues improves novel view synthesis quality.

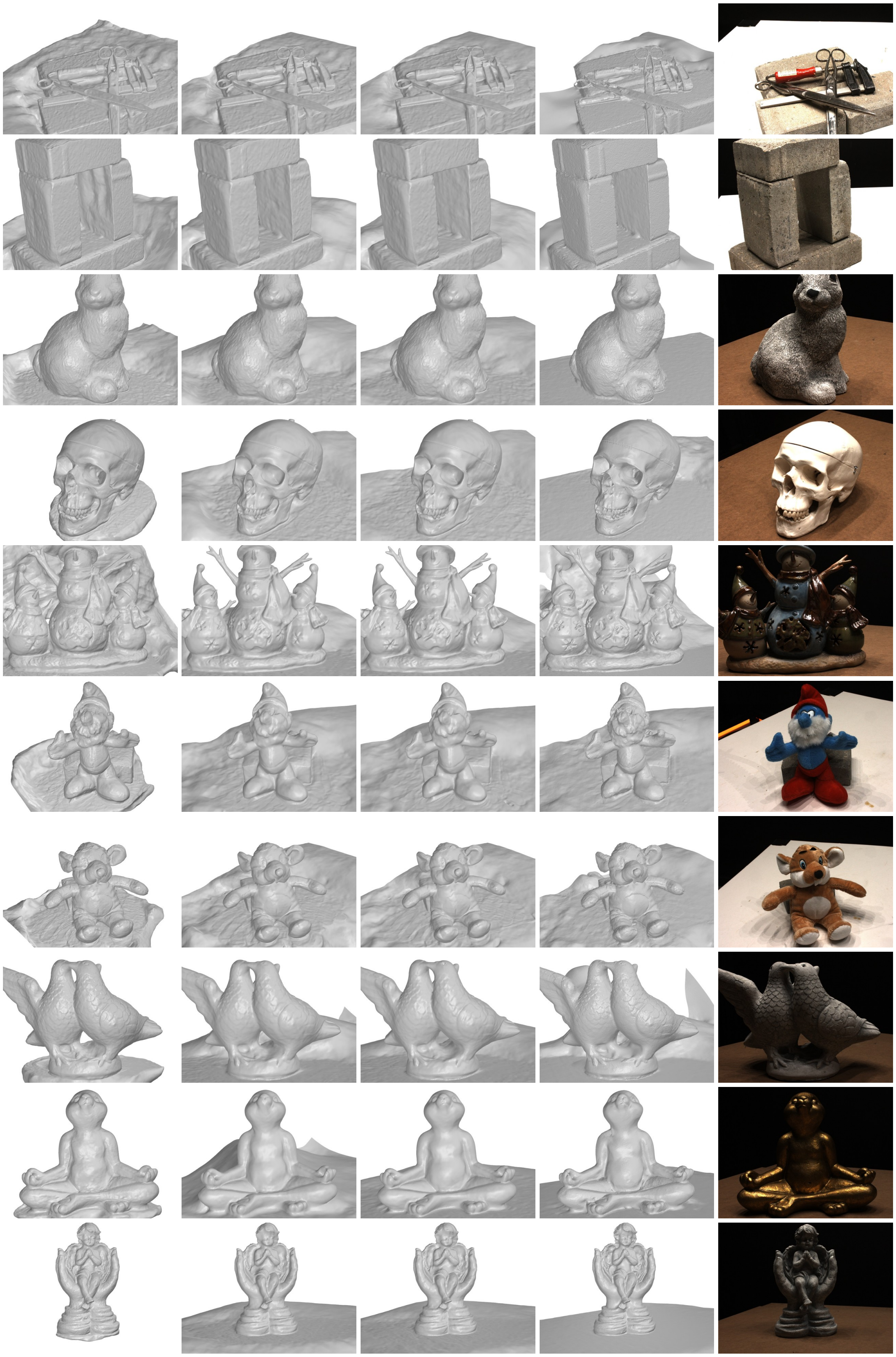


Figure 19: **Qualitative Comparison on DTU Dataset with all input views.** Our approach with MLP achieves similar results with previous method, while our method with Multi-Res. Fea. Grids reconstruct more detailed surface.

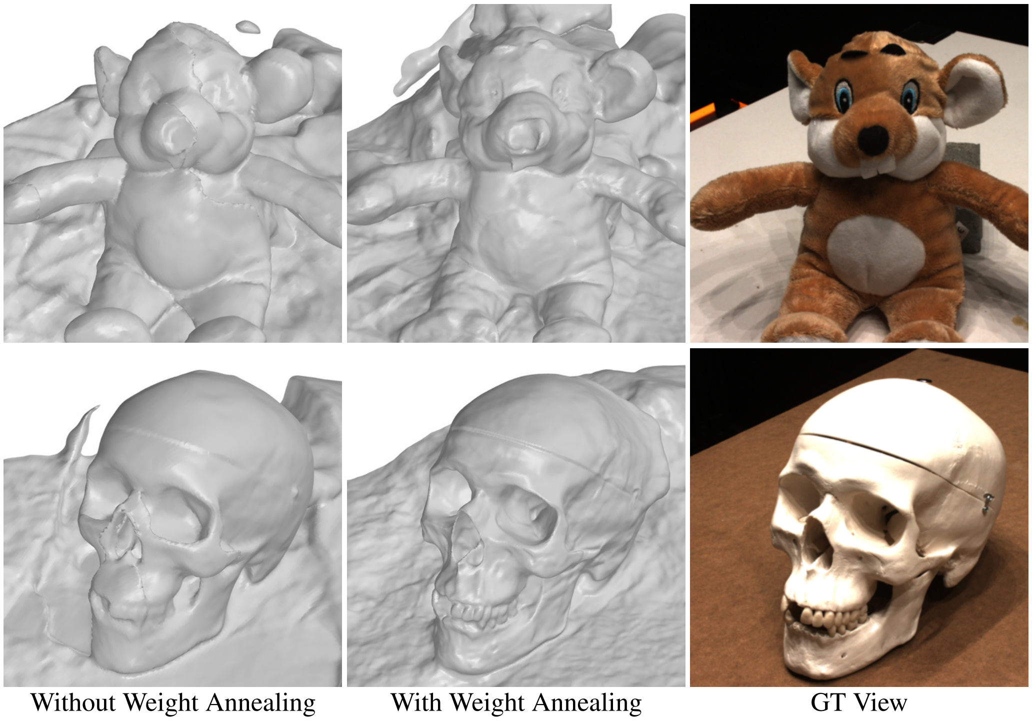


Figure 20: **Ablation of Weight Annealing on the DTU Dataset with 3 Input Views.** Using weight schedule improves reconstruction quality.

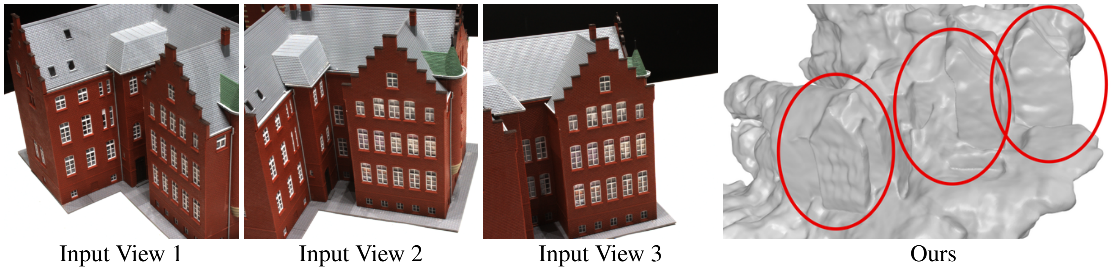


Figure 21: **Failure Case on DTU Dataset with 3 Input Views.** The reconstructed mesh duplicate the object in front of each camera frustum.