High-Fidelity Mask-free Neural Surface Reconstruction for Virtual Reality

Haotian Bai*
HKUST(GZ)

Yize Chen †
University of Alberta

Lin Wang‡
HKUST(GZ)
HKUST

Project homepage: `https://vlislab22.github.io/Hi-NeuS/`

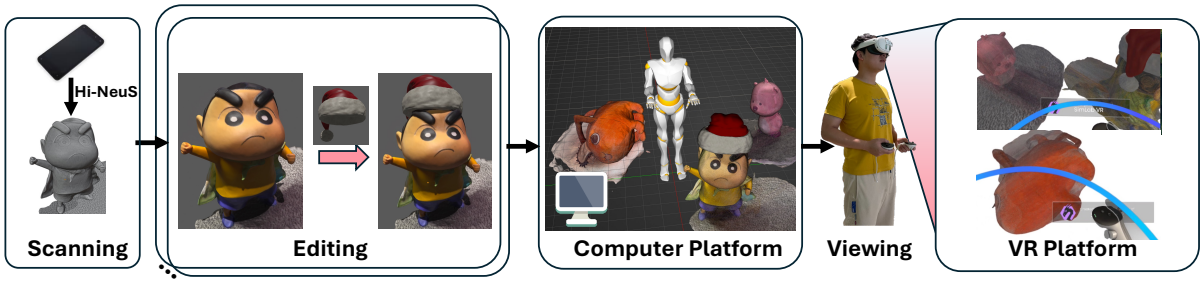


Figure 1: **Overview of our Hi-NeuS Framework:** It converts multi-view images captured from phone cameras into digitized mesh objects. The meshes can be edited and integrated into both computer and VR for viewing and editing. This framework can be potentially employed for direct geometry capturing and subsequent viewing and creation in VR/AR.

Abstract

Object-centric surface reconstruction from multi-view images plays a crucial role in creating editable digital assets for AR/VR. Due to the lack of geometric constraints, existing methods, *e.g.*, NeuS [38] necessitate annotating the object masks to reconstruct compact surfaces in mesh processing. Mask annotation, however, is labor-intensive due to its cumbersome nature, and its absence may lead to noisy surfaces. This paper presents **Hi-NeuS**, a novel rendering-based framework for neural implicit surface reconstruction, aiming to recover *compact and precise surfaces without multi-view object masks*. Our key insight is that the overlapping regions in the object-centric views naturally highlight the object of interest as the camera orbits around objects. The object of interest can be essentially specified by estimating the distribution of the rendering weights accumulated from multiple views, which implicitly identifies the surface that a user intends to capture. This inspires us to design a geometric refinement approach, which takes multi-view rendering weights to guide the signed distance functions (SDF) of neural surfaces in a self-supervised manner. Specifically, it retains these weights to resample a pseudo surface based on their distribution. This facilitates the alignment of the SDF to the object of interest. We then regularize the SDF's bias for geometric consistency. Moreover, we propose to use unmasked Chamfer Distance(CD) to measure the extracted mesh without post-processing for more precise evaluation. Our approach's effectiveness has been validated through NeuS and its variant Neuralangelo, demonstrating its adaptability across different NeuS backbones. Extensive benchmark on the DTU dataset shows that our method reduces surface noise by about 20%, and improves the unmasked CD by around 30%, meanwhile, achieving better surface details. The superiority of Hi-NeuS is further validated on the BlendedMVS dataset, as well as the real-world applications using *handheld* camera captures for content creation.

Index Terms: 3D and volumetric display and projection technology; Neural Surface Reconstruction; Signed Distance Field

*email: haotianwhite@outlook.com
†email: yize.chen@ualberta.ca
‡Corresponding author, email: linwang@ust.hk

1 Introduction

Imagine the last time you captured photos of an object by walking around it. Now, you want to integrate that object into a 3D virtual environment, such as an AR/VR world, ideally in a mesh format for both viewing and content creation, as depicted at Fig. 1. Traditionally, this process relied on classical stereo-based methods [35, 36, 3, 23, 2, 13, 34]. However, recent advancements in 3D reconstruction using neural volume rendering [28, 27] have transformed it, enabling the recovery of high-fidelity details and more complex structures. Compared to explicit representations like Gaussian Splatting [10, 16, 47, 19], which consists of dense collections of 3D Gaussians, the implicit NeRF is commonly used for surface reconstruction due to its maturity in terms of conversion and compatibility, as demonstrated by industry practices [14]. Neural implicit surface reconstruction typically employs multi-layer perceptrons (MLPs) to implicitly represent scenes as occupancy fields [30], SDF[44, 38], or hybrid grids[25]. Due to the inherent continuity of implicit representations, these methods can synthesize plausible novel view images. However, they lack sufficient surface constraints and struggle to extract high quality surfaces [30].

To tackle these issues, some works [44, 38, 30] integrate implicit representations in volume rendering to reduce inherent geometry errors. Among them, NeuS [38] is one of the pioneering works that adopt SDF-based volume rendering to model geometric surfaces. It integrates SDF into the density field in volume rendering to constrain the scene, yielding unbiased surface reconstruction with occlusion awareness. Notably, NeuS reduces the reliance on multi-view object masks as training supervision. Despite NeuS's superiority, its neural surface representation via SDF remains under-constrained. Specifically, when the SDF is not fully trained, it struggles to accurately represent the underlying geometry, resulting in a biased SDF distribution [8]. This bias causes geometry errors, leading the predicted surface to deviate from the expected geometry as the noise to be reduced at the middle of (a) in Fig. 2. Therefore, to extract compact mesh objects from learned SDF representations, NeuS and its follow-up works [25, 38, 39] often require annotated masks for each camera pose during training or to remove background mesh noise in post-processing. However, the *annotation process* is labor-intensive and prone to human error. Considering that all perspectives must be annotated, this approach becomes increasingly cumbersome and costly. In scenes with complex or overlapping objects, these masks frequently struggle to de-

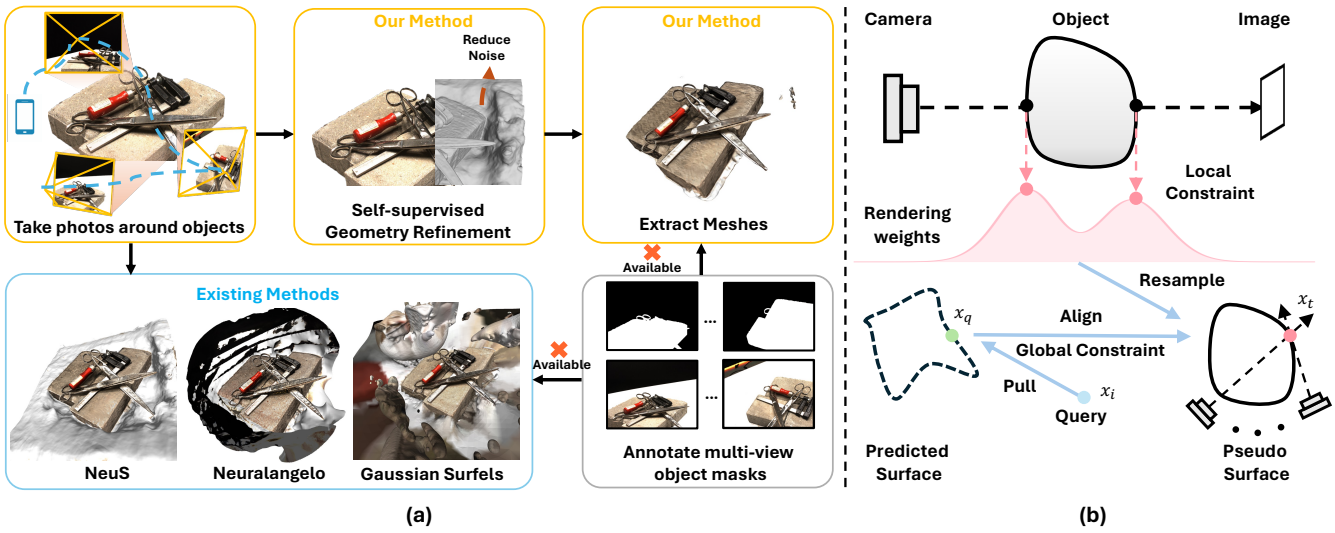


Figure 2: **(a) Comparison on surface reconstruction without masks.** We aim to reduce the noise in surface reconstruction without relying on multi-view object masks. Compared to existing methods, including NeuS [38], Neuralangelo [25], and Gaussian Surfels [10], Hi-NeuS produces compact and more precise mesh results, enhancing its utility for downstream applications in virtual reality. **(b) Self-supervised geometry refinement.** The rendering weights from multiple views are accumulated, corresponding to resampled target surface points x_t(red). Based on this supervision, we advance query points x_i (blue) to obtain the predicted surface points x_q (green). We then align them using Chamfer Distance (CD) with global geometric constraints related to SDF.

lineate boundaries accurately, even when using methods like the Segment Anything Model(SAM) [21]. On the other hand, direct filtering with the annotation masks is not plausible, as it leads to geometry artifacts, such as jagged edges or holes as depicted Fig. 5. We refer readers to the further discussion in our *suppl.* materials.

Despite recent efforts to enhance geometric accuracy, many approaches continue to prioritize improving neural representation power [11] or assist surface reconstruction through auxiliary point cloud supervision [48] and pretrained models [7]. The challenge of surface reconstruction using solely multi-view images, without multi-view object masks, remains under-explored. As shown in (a) of Fig. 2, the absence of these masks can lead to significant mesh artifacts in existing methods. This makes it hard to achieve a direct mesh reconstruction such as the pipeline shown at Fig. 1.

To recover more compact and precise surfaces without object mask, *our key inspiration is that, as the camera orbits around objects, the overlapping regions in the captured views naturally highlight the object of interest*. This overlap implicitly identifies the subject the photographer intends to capture. Similarly, during volume rendering, as illustrated in the upper part at (b) of Fig. 2, the rendering weights peak near the object's surface when the camera rays intersect with it, which delineates the surface boundaries. However, this ray-wise local constraint alone is insufficient to obtain the surfaces due to the biased SDF distribution. To address this, we propose leveraging the accumulated weights from multiple views to more effectively capture the global surface regions.

This idea builds on the findings of NeuS [38], which demonstrates that, with an unbiased rendering weight function, surface points contribute more to their corresponding ray pixels than other ray samples. Furthermore, occlusion awareness ensures that the first intersection along a ray holds a higher value than subsequent intersections. Compared with this local geometric constraint, accumulating multi-view rendering weights may potentially delineate the object's surface with similar peak weights regardless of ray directions. Given this surface supervision, we can mitigate the geometry bias introduced during training by aligning SDF globally towards the generated pseudo-surfaces from these rendering weights.

Specifically, as the geometric refinement process depicted at the lower part at (b) of Fig. 2, for a given arbitrary ray query, upon the inference of SDF via MLPs, we employ the differentiable neural pulling operation [5] to pull the ray sample towards the object's surface, guided by the predicted signed distances and their gradients. This approach allows gradients from these predicted surface points to be back-propagated into the neural SDF. Hi-NeuS aligns these points with the target surface supervision, which can be resampled based on the accumulated multi-view weights distribution to locate the underlying surface. Thus, this supervision serves as the reference, imposing geometric constraints on the SDF to approximate a globally consistent surface through a self-supervised methodology.

Our key contributions are as follows: **(1)** We propose Hi-NeuS to create more precise and compact surfaces without requiring multi-view object mask annotations. This framework can potentially be used for direct geometry recovery; subsequent viewing and content creation in VR/AR. **(2)** We propose using a geometric constraint to regularize the surface globally. This method leverages rendering weights accumulated from multiple views to align with the surface that a user intends to capture. **(3)** Comprehensive experiments using the NeuS backbone on the DTU dataset [17] and the BlendedMVS dataset [43] demonstrate that our method significantly reduces noise while enhancing geometric detail. Additionally, our method's versatility is validated across NeuS and its variant Neuralangelo [25] and exhibits superior performance in challenging *real-world* scenarios involving handheld phone camera captures.

2 Related Works

3D content creation and interaction. Advances in neural rendering and real-time graphics have significantly enhanced 3D content viewing and interaction in AR/VR, enabling more immersive and interactive experiences across devices. Recent technologies, such as Re-ReND [32] and RT-NeRF [24], have demonstrated real-time rendering of NeRFs in VR/AR headsets using standard graphics pipelines, offering high-quality visual experiences. Additionally, FoV-NeRF [12] improves rendering by focusing on the user's gaze and optimizing computational resources. Furthermore, the VR-GS [18] integrates physical dynamics for realistic, responsive interaction with 3D contents represented with Gaussian Splatting, en-

suring a comprehensive virtual experience.

Despite the rapid progress in integrating NeRF/GS into VR/AR, acquiring high-quality 3D assets remains a significant challenge in populating virtual worlds with 3D content. Unlike generation methods [37] that rely on text or images, which prioritize creativity, 3D reconstruction focuses on accuracy and realism. Consequently, the resulting meshes often exhibit intricate details and better align with the user's desired outcome. To facilitate the process, several approaches have been explored, including few-shot novel view synthesis[49, 26], more efficient NeRF representations[42], and reducing training time [31]. In contrast, our approach focuses on recovering compact and precise surfaces without relying on object masks, thereby reducing the need for costly annotation.

Surface reconstruction from multi-view images. Before the deep learning era, traditional multi-view stereo (MVS) methods dominated the field of surface reconstruction from multi-view images [3]. These techniques primarily reconstructed 3D shapes by matching features across adjacent frames [22, 23, 2], employing discretized frameworks like voxel grids [23, 2], and point clouds [13, 34]. However, they often struggled to capture fine geometric details due to the limited resolution of their cost volumes. The recent advent of Neural Radiance Fields (NeRFs) [28] brings a paradigm shift with its continuous volumetric representation. NeRFs utilize an MLP to encode 3D scenes, correlating spatial locations with their corresponding colors and densities for photo-realistic volumetric rendering. To further enhance implicit geometry, a variety of methods [11, 30, 38] have been introduced. They aim to revise the rendering procedure to handle occlusions and sudden depth changes. Additionally, other methods [40, 25] focus on enhancing representational capabilities and training strategies to improve surface estimation accuracy. Notably, Neuralangelo [25] proposes multi-resolution 3D hash grids with coarse-to-fine optimization integrated with SDF-based rendering, yielding state-of-the-art (SOTA) geometry accuracy and rendering capability.

On the other hand, Gaussian Splitting (GS) has gained significant attention for representing complex scenes using 3D Gaussians, with GS-based surface reconstruction demonstrating impressive performance compared to NeRF, particularly in terms of training and inference efficiency [10, 16, 47, 10]. However, NeRF offers more geometric detail while maintaining compactness and reducing overfitting, even when dealing with intricate geometries, textures, and material properties. The adoption of NeRF for geometry has also been validated in recent industry practices [14], owing to its maturity in terms of conversion and compatibility.

Unlike previous approaches, our work aims to recover more compact and precise surfaces in neural implicit representations, eliminating the need for multi-view object masks. By doing so, we mitigate geometry artifacts, such as jagged edges and holes, which can potentially reduce the requirement for additional annotations. Although recent advancements [25, 38, 30, 45] have enabled surface reconstruction without auxiliary object masks as training supervision, they may still rely on them to refine meshes, necessitating time-consuming annotation or manual editing to eliminate geometric noise. This limitation hinders the development of surface reconstruction methods with solely multi-view images.

Geometrical constraints for neural representations. To represent the scene geometry, implicit functions such as occupancy grids [30, 29] and SDFs [25, 38] are preferred due to their continuous representation with low memory consumption. Recent works [30, 38, 11] employ implicit functions to enforce geometric consistency across multi-view images, thereby imposing geometrical constraints on the learned object representation. For instance, NeuS [38] parameterizes the volume density and integrates it into the volume rendering process, achieving unbiased surface reconstruction with occlusion awareness. NeuralWarp [11] enhances geometry accuracy by warping views to learn from high-frequency image textures. Neuralangelo [25] enhances surface smoothness with continuous numerical gradients in its hash grids and predicted curvature. However, these geometric constraints are limited to the image, ray, or patch level, imposing only regularization based on partial visual cues. This approach may lack the 3D spatial awareness needed for more consistent geometry regularization.

To mitigate this gap between 2D and 3D, follow-up works use auxiliary information to enhance global geometrical consistency. For instance, NeuralWarp [11] and RegSDF [48] leverage information from structure-from-motion to guide surface optimization. D-NeuS [8] utilizes depth maps to correct geometrical deviations in the SDF values at surface intersection points. Additionally, it employs a pre-trained model to enhance feature representation ability with RGB inputs. Similarly, Chen *et al.* [7] introduces a probability mask to refine pixel sampling on the foreground object, complemented by segmentation masks. The recent works using Gaussian Splatting for surface reconstruction, *i.e.*, Gaussian Surfels [10] also rely on external object masks and surface norms to assist geometry learning during training. These approaches' reliance on external models or data can be a considerable constraint when such resources are unavailable or their information is inaccurate, limiting their applicability to general cases. In contrast, our work introduces a self-supervised geometric refinement approach that leverages auto-generated rendering weights for surface refinement, establishing a global 3D geometrical constraint to regularize the geometry representation automatically.

3 Hi-NeuS Framework

3.1 Overview

As shown in Fig. 3, given a set of posed multi-view images, surface reconstruction seeks to reconstruct the 3D object surfaces. The Hi-NeuS training framework uses the same SDF-based volume rendering as our baseline NeuS [38] and Neuralangelo [25]. This approach integrates a geometry representation SDF $f(x)$ into a color MLP $g(x)$ to generate images. Specifically, given an arbitrary point sample x_i and its corresponding ray direction $\mathbf{d_i}$, SDF-based volume rendering aims to convert SDF values into a volume density field α_i using the logistic function. Then, the activated density with $\mathbf{d_i}$ is sent to $g(x)$ to infer the color $\hat{\mathbf{c}}_\mathbf{i}$. At last, it predicts the corresponding pixel color $\hat{\mathbf{C}}(\mathbf{r_i})$ by accumulating ray samples along rays supervised by the ground truth $\mathbf{C}(\mathbf{r_i})$. Throughout this process, Hi-NeuS records rendering weights w_i from multiple views. Then w_i are transformed to form a probability distribution, from which target surface points x_t are resampled from the pseudo surface as supervisory signals. The predicted surface points x_q is obtained via the Neural Pulling operation [5] to allow gradient updates to be back-propagated into the SDF. Finally, Hi-NeuS aligns x_q towards x_t in a self-supervised way, while regularizing geometric consistency. We refer readers to the algorithm's pseudo-code in our *suppl.*.

3.2 Preliminary: Volume Rendering with Geometry Learning

As shown in Fig. 3(a), we introduce the volume rendering process with input posed multi-view images to infer the rendering color and multi-view rendering weights. Those weights are then sent into the global geometric refinement to reduce geometric noise.

Neural Volume Rendering. The neural volume rendering process involves learning the parameters of two implicit functions, $f(x)$ and $g(x)$. To infer the color $\hat{C}(\mathbf{r}_i)$ of a ray $\mathbf{r}_i$, we integrate N samples x_i along the ray. $\hat{C}(\mathbf{r}_i)$ for the corresponding pixel is computed by summing the weighted colors $\hat{\mathbf{c}}_\mathbf{i}$ of the samples x_i along the ray, where each sample x_i is weighted by w_i:

$$\hat{C}(\mathbf{r_i}) = \sum_{i=1}^{N} w_i \hat{\mathbf{c}}_\mathbf{i}, \text{ where } w_i = T_i \alpha_i, \tag{1}$$

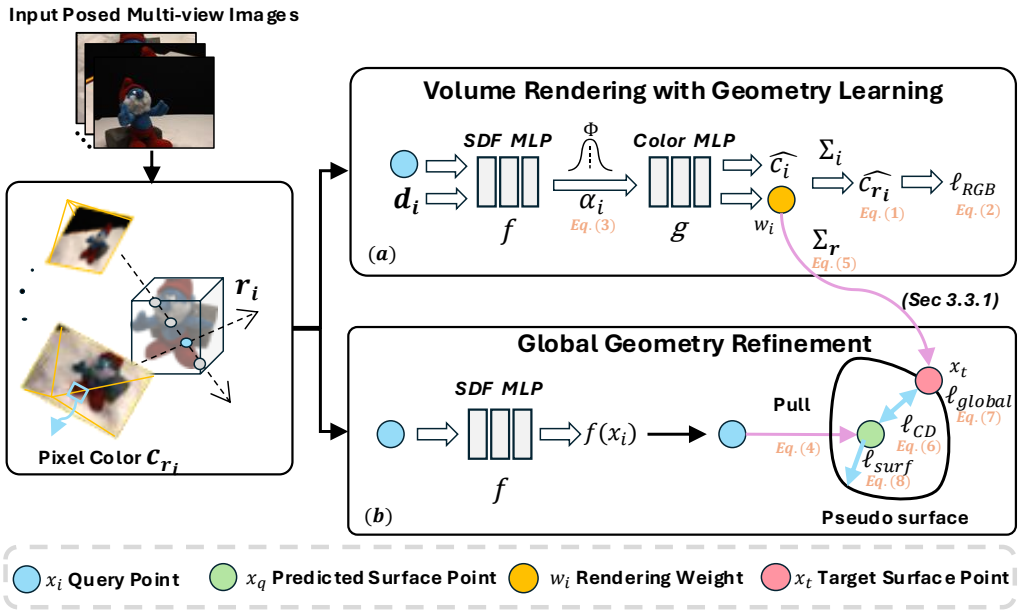


Figure 3: **Our proposed Hi-NeuS training framework**: In volume rendering combined with geometry learning, we capture rendering weights from multiple views. Hi-NeuS then resamples based on the weight distribution to obtain supervisory surface points. Finally, global geometric refinement is applied using geometric constraints.

where $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ denotes the opacity of the i-th ray segment, and the light accumulated through any ray $\mathbf{r_i}$ to sample i is represented by $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$. Here, σ_i denotes the volume density, and $\mathbf{c}_i$ is the color vector in the form of RGB or spherical harmonics (SH). The predicted color $\hat{C}(\mathbf{r_i})$ is optimized to approximate the ground truth (GT) color $C(\mathbf{r_i})$ by minimizing the mean squared error (MSE) loss:

$$\mathscr{L}_{RGB} = \sum_i \|C(\mathbf{r_i}) - \hat{C}(\mathbf{r_i})\|_2^2 \tag{2}$$

Neural Rendering with SDF. The surface $\mathbb{S}$ of an SDF is defined by its zero-level set: $\mathbb{S} = \{x \in \mathbb{R}^3 | f(x) = 0\}$. NeuS [38] converts the SDF into volume density α using a logistic function $\phi_s(f(x))$, which is the derivative of the sigmoid function Φ_s. The discretized approximation of volume rendering is computed similarly to Eq. (1), with the revised opacity given by:

$$\alpha_i = \max\left(\frac{\Phi_s(f(x_i)) - \Phi_s(f(x_{i+1}))}{\Phi_s(f(x_i))}, 0\right). \tag{3}$$

Neural Pulling with neural SDF. To enhance the quality of SDF representations, Baorui et al. [5] proposed a differentiable method that pulls a query 3D location x_i toward its closest surface intersection. As shown in Fig. 2, this predicted surface intersection is denoted as x_q. It is calculated using the predicted signed distance value $f(x_i)$ and its gradient $\nabla f(x_i)$:

$$x_q = x_i - f(x_i)\frac{\nabla f(x_i)}{\|\nabla f(x_i)\|}. \tag{4}$$

This operation facilitates effective gradient updates to the SDF by aligning x_q with the pseudo surface, which consists of target surface points x_t resampled from the multi-view rendering weights.

3.3 Self-supervised Global Geometry Refinement

3.3.1 Self-supervision via multi-view rendering weights

Recent research [4] has demonstrated that rendering weights can effectively highlight regions of interest for sample allocation. As shown in Fig. 3(a), these weights are automatically generated during the volume rendering process, they can serve as a form of self-supervision, providing valuable cues for localizing overlapping parts during training. Moreover, NeuS [38] demonstrates that, firstly, with an unbiased rendering weight function, surface points contribute more to their corresponding ray pixels than other ray samples; Secondly, occlusion awareness ensures that the first intersection along a ray holds a higher value than subsequent intersections. Consequently, the multi-view weights can be leveraged to generate a pseudo-surface for the global supervision.

As depicted in Fig. 3(b), Hi-NeuS leverages this supervision by aggregating rendering weights from multiple views, enabling more frequent evaluations and thereby reducing uncertainty. More importantly, differs from the ray-wise SDF constraint of NeuS in Eq. (3), our method imposes geometrical constraints on the 3D space, allowing reducing the neural SDF bias from a global scale.

To obtain this pseudo-surface, we utilize a temporary grid buffer to accumulate the multi-view rendering weights averaged across all camera poses. We periodically refresh the buffer to update the training statistics. Upon refreshing the grid buffer, we apply a global voting scheme, formulated as follows:

$$w_t = \sum_r w_i \frac{1}{n_i}; \;\; f(x_t) = \sum_r f(x_i)\frac{1}{n_i}; \tag{5}$$

where n_i denotes the number of ray hits at the grid unit; t denotes the buffer's refreshing times. In addition, we also record signed distance values to facilitate subsequent refinement stages.

Once w_t is obtained, we apply a pulling operation to predict surface points x_q that are expected to distribute around the pseudo-surface from w_t when training converges. To achieve it, we resample target surface points x_t based on the recorded normalized ren-

dering weights w_{t-1}. Our problem is then reformulated as the alignment of point clouds $\vec{x}_q$ and $\vec{x}_t$, subject to geometric constraints.

3.3.2 Global Geometric Refinement

As illustrated in the bottom branch of Fig. 3, we propose three geometric constraints to mitigate neural bias on a global scale: point cloud alignment loss $\mathscr{L}_{cd}$, global geometry consistency loss $\mathscr{L}_{global}$, and surface regularization loss $\mathscr{L}_{surf}$, which are introduced as follows:

(1) Point cloud alignment loss (CD $\rightarrow$ 0): To align the predicted surface with $\vec{x}_t$, Hi-NeuS employs the bidirectional Chamfer Distance(CD). Specifically, CD is defined as $d(A,B) = \frac{1}{|\vec{x}_q|}\sum_{x\in A}\min_{x'\in B}\|x-x'\|_2^2$. This formula quantifies the similarity between two point sets, A and B, by calculating the averaged nearest distance between each point in A and any point in B. By minimizing the bidirectional CD, Hi-NeuS ensures both sets are matched as closely as possible in both directions. The point cloud alignment loss is formulated as:

$$\mathscr{L}_{cd}(\vec{x}_q,\vec{x}_t) = \frac{1}{2}(d(\vec{x}_q,\vec{x}_t) + d(\vec{x}_t,\vec{x}_q)). \tag{6}$$

By minimizing $\mathscr{L}_{cd}(\vec{x}_q,\vec{x}_t)$, we guide the learnable surface points $\vec{x}_q$ to align toward $\vec{x}_t$, allowing for the refinement of the surface through the backpropagation of gradients into $\vec{x}_q$ and subsequently into the SDF representation.

(2) Global geometry consistency loss (|SDF| < CD): The SDF measures measures the orthogonal distance, representing the shortest distance from a given point to the surface boundary. However, during training, the learnable surface points $\vec{x}_q$ may bring an inconsistent CD due to training dynamics and unstable learning. To maintain global geometry consistency, we require that the absolute SDF value of $\vec{x}_q$ remains consistently smaller than the CD of the point cloud. To enforce this constraint, we introduce a novel global geometry consistency loss,

$$\mathscr{L}_{global} = \big||f(\vec{x}'_q)| - d(\vec{x}'_q,\vec{x}_t)\big|, \tag{7}$$

where $\vec{x}'_q \in \vec{x}_q$ is a filtered subset of points, ensuring that all predicted surface points lie within valid coordinate spaces. Specifically, we derive $f(\vec{x}'_q)$ from the grid buffer recorded at Eq. (5), excluding outliers that do not correspond to valid buffer units, to ensure stable training.

(3) Surface regularization loss (SDF $\rightarrow$ 0): Since $\vec{x}_q$ represents the surface points, the ideal signed distance values $f(\vec{x}_q)$ should ideally be zero. To reduce surface geometry bias, we introduce a penalty for the absolute SDF error on all valid learnable surface points $\vec{x}'_q$. This is captured by the surface SDF regularization loss $\mathscr{L}_{surf}$, defined as:

$$\mathscr{L}_{surf} = f(\vec{x}'_q) = \sum_i \frac{1}{n_i}|f(x'_{q,i})|. \tag{8}$$

3.4 Optimization

To further verify the adaptability, we extend our framework to Neuralangelo [25], which shares the same SDF-based rendering as NeuS [38]. In our implementation, we incorporate two additional losses: the eikonal loss $\mathscr{L}_{eik}$ and the curvature loss $\mathscr{L}_{curv}$. The eikonal loss $\mathscr{L}_{eik}$ is based on the eikonal equation and ensures that the gradient magnitude is normalized throughout the entire space to reduce SDF truncation. Meanwhile, the curvature loss $\mathscr{L}_{curv}$ guarantees that the analytical gradients of hash encoding are zero everywhere when using trilinear interpolation.

$$\mathscr{L}_{eik} = \frac{1}{N}\sum_{i=1}^{N}(\|\nabla f(x_i)\|_2 - 1)^2;\ \mathscr{L}_{\text{curv}} = \frac{1}{N}\sum_{i=1}^{N}\Big|\nabla^2 f(x_i)\Big|. \tag{9}$$

Our global geometric constraints $\mathscr{L}_{geo}$ are integrated as an add-on module, comprising a weighted sum of three components:$\mathscr{L}_{cd}$, $\mathscr{L}_{global}$, and $\mathscr{L}_{surf}$:

$$\mathscr{L}_{geo} = \mathscr{L}_{cd} + w_{surf}\mathscr{L}_{surf} + w_{global}\mathscr{L}_{global}; \tag{10}$$

We then combine $\mathscr{L}_{geo}$ with other losses to form the total loss functions for NeuS and Neuralangelo:

$$\mathscr{L}_{neus} = \mathscr{L}_{RGB} + w_{eik}\mathscr{L}_{eik} + w_{geo}\mathscr{L}_{geo}; \tag{11}$$

$$\mathscr{L}_{neuralangelo} = \mathscr{L}_{neus} + w_{curv}\mathscr{L}_{\text{curv}}. \tag{12}$$

We apply these loss functions, $\mathscr{L}_{neuralangelo}$, $\mathscr{L}_{neus}$ in the original implementations for Neuralangelo and NeuS respectively.

4 Experiments

Datasets. We conducted experiments on the DTU dataset [17], which includes 15 object-centric scenes. Each scene comprises 49 or 64 images captured by a robot-held monocular RGB camera, with ground truth obtained using a structured light scanner. Additionally, following NeuS, we performed experiments on 7 challenging scenes from the low-resolution subset of the BlendedMVS dataset [43]. Each of these scenes contains 31 to 143 images at 768×576 pixels with corresponding masks. Moreover, we capture hand-held phone videos using an iPhone 13, recording short sequences of 30 to 60 seconds in duration, with 80-160 frames sampled uniformly throughout each sequence. All mesh reconstructions in this study were processed using the marching cubes algorithm, with the resolution set to 512.

Evaluation criteria. We calculate the Peak Signal-to-Noise Ratio (PSNR) on all masked parts of images from each scene, following the same method as Neuralangelo. We report the masked CD on the observed regions with the masked-out meshes to ensure fair comparison against previous works [25, 38, 11, 48, 8, 44, 28]. We propose evaluating meshes using the unmasked CD, which assesses the extracted raw mesh without any post-processing. As illustrated in Fig. 5, our empirical study reveals that due to the limited number of camera views, the visual hull may inadvertently cover desirable object parts, resulting in an incomplete filtered mesh. To address this issue, we consider using the raw mesh directly for a more comprehensive and error-free evaluation of meshes on a global scale. For evaluating geometry noise, we introduce a mesh noise metric that calculates the ratio of filtered faces to the total number of mesh faces during mesh filtering as depicted in Fig. 5. This metric effectively indicates the majority of regions that users do not intend to have uncovered by the multi-view object masks.

Baselines. The baselines are as follows: **1)** NeuS [38], a pioneering work that first developed SDF-based volume rendering, has had a profound impact on the surface reconstruction field. It has inspired subsequent research, including Neuralangelo. The NeuS's superior performance justifies its exclusion from direct comparison with contemporaries like UNISURF[30] and IDR [45] in Tab. 2. However, it is worth noting that NeuS's SDF representation introduces a significant amount of noise, approximately 40%, which underscores our argument that geometry is still underconstrained when relying solely on ray-based geometry constraints. **2)** Neuralangelo [25] is a state-of-the-art surface reconstruction framework that can recover high-quality surfaces. However, due to the lack of publicly available per-scene configurations, we apply the same configuration to all DTU scenes, without performing per-scene fine-tuning. To ensure a fair comparison, our Hi-NeuS framework, which builds upon Neuralangelo, also utilizes this same configuration. Unfortunately, Neuralangelo introduces considerable noise, with approximately 60%. These artifacts necessitate the use of foreground masks for cleanup in DTU datasets, which can be problematic as visual hulls alone may not accurately capture the intended geometry. (See Fig. 5 for an illustration of this issue.)

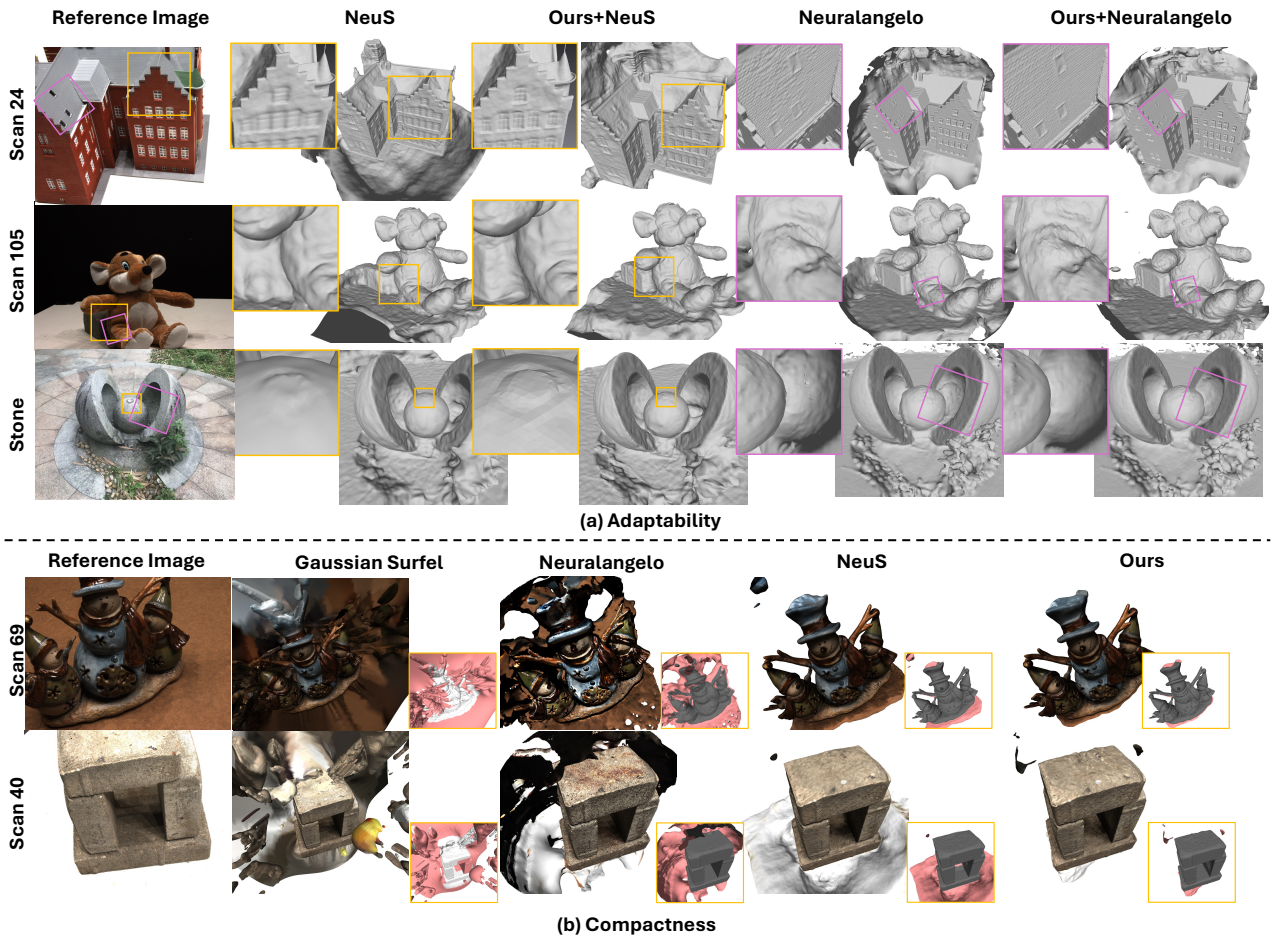


Figure 4: **Qualitative comparison of Hi-NeuS**: **(a) Adaptability.** We integrate our geometry refinement with NeuS and Neuralangelo to prove its adaptability. The magnified boxes reveal the recovered details. **(b) Compactness.** We compare our Hi-NeuS with the NeuS backbone against existing methods. The corner box of each image displays the highlighted areas outside the visual hull, denoted as mesh noise.

Implementation details. We adopt the same experimental setup and framework architectures as NeuS and Neuralangelo. For detailed hyperparameter configurations, we refer readers to the original implementations of NeuS and Neuralangelo. In our method, we refine learnable surface points $\vec{x}_q$ to a subset $\vec{x}'_q$, comprising only the valid points within the normalized coordinate space $[-1,1]$. The resampled point cloud supervision, $\vec{x}_t$, matches the set size of $\vec{x}_q$, as described in Eq. (6). This enables global surface refinement to have equal strength for both directions between the two point clouds. We utilize a grid buffer with a resolution of $(32,64,128)$ implemented using [46], which benefits from customized CUDA acceleration for efficient spatial queries. The grid buffer is reset to 0 after considering a certain ratio of views, allowing it to record the most recent training statistics after refreshing.

4.1 Performances

Quantitative & Qualitative Results. Hi-NeuS achieves comparable or superior performance in terms of both masked and CD for most scenes in DTU. Specifically, as shown in Tab. 2 and Fig. 4, the CD of scene 24 is reduced by approximately 17%, allowing for more detailed architectural features to be captured on the building. Especially, as shown in (b) of Fig. 7, Hi-NeuS successfully recovers the missing hand of the clock, which NeuS fails to reconstruct, likely due to a lack of front-facing photos. This demonstrates Hi-NeuS's ability to effectively capture the 3D structure and depict the correct geometry when views are limited, proving the effectiveness of the global scale geometric constraint. Besides, our method recovers more intricate details in scenes such as 105, where the brick structures behind the toy are more accurately represented. However, the masked CD calculation does not fully reflect this improvement, as it is limited by the visual hull coverage. In contrast, our proposed unmasked CD provides a more comprehensive evaluation of the raw mesh geometry, revealing the overall geometry accuracy. In terms of visual quality, Hi-NeuS demonstrates improved results on the NeuS backbone and achieves comparable PSNR values on Neuralangelo. This suggests that the corrected geometry may have a positive impact on color learning, potentially leading to more accurate and realistic color representations. Regarding noise reduction, Hi-NeuS achieves superior performance, reducing noise by approximately 37% compared to the NeuS backbone and by around 20% compared to Neuralangelo. This improvement is visually evident, with a significant reduction in noise scale. In general, Hi-NeuS demonstrates its ability to recover compact, accurate, and high-fidelity surfaces, showcasing its adaptability and versatility when integrated with NeuS backbones. Our study's full quantitative and qualitative result is attached at Tab.1 in *suppl.*.

4.2 Ablation Studies and Analysis

Loss Effectiveness. To gain insight into the impact of our proposed geometrical constraints on reconstruction results, we evaluate the

Table 1: Quantitative unmasked CD, PSNR, mesh noise results on DTU dataset [17].

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
unmasked CD ↓	Gaussian Surfels*† [10]	1.00	1.97	1.06	1.74	2.32	2.35	2.02	3.48	2.45	2.55	2.31	8.13	1.49	2.69	3.48	2.60
	NeuS [38]	1.59	1.98	1.44	0.95	1.82	0.74	0.64	1.63	1.30	1.41	0.59	1.33	0.44	0.51	0.54	1.13
	Neuralangelo* [25]	0.62	1.63	0.66	0.56	1.51	1.38	2.60	2.03	2.15	1.11	0.46	1.31	0.48	0.95	1.25	1.25
	Hi-NeuS(NeuS)	0.96	0.93	0.71	0.47	1.37	0.71	0.66	1.45	1.02	1.07	0.58	1.27	0.44	0.50	0.54	0.81
	Hi-NeuS(Neuralangelo)	0.55	1.55	0.61	0.60	1.51	0.77	2.25	1.19	1.52	1.09	0.43	1.20	0.43	0.88	1.36	1.06
unmasked CD ↓	RegSDF† [48]	0.60	1.41	0.64	0.43	1.34	0.62	0.60	0.90	0.92	1.02	0.60	0.59	0.30	0.41	0.39	0.72
	NeuralWarp† [11]	0.49	0.71	0.38	0.38	0.79	0.81	0.82	1.20	1.06	0.68	0.66	0.74	0.41	0.63	0.51	0.68
	D-NeuS† [8]	0.44	0.79	0.35	0.39	0.88	0.58	0.55	1.35	0.91	0.76	0.40	0.72	0.31	0.39	0.39	0.61
	NeRF [28]	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
	VolSDF [44]	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
	NeuS [38]	0.93	1.07	0.81	0.38	1.02	0.60	0.58	1.42	1.15	0.78	0.57	1.16	0.35	0.45	0.46	0.78
	Neuralangelo* [25]	0.39	0.72	0.35	0.33	0.82	0.74	1.70	1.34	1.95	0.71	0.47	1.00	0.33	0.82	0.78	0.83
	Hi-NeuS(NeuS)	0.77	0.90	0.73	0.37	1.00	0.59	0.59	1.42	1.19	0.79	0.56	1.93	0.35	0.45	0.48	0.81
	Hi-NeuS(Neuralangelo)	0.39	0.71	0.36	0.33	0.92	0.55	1.42	1.25	1.44	0.73	0.45	0.99	0.33	0.70	0.73	0.75
PSNR ↑	RegSDF† [48]	24.78	25.31	23.47	23.06	22.21	28.57	25.53	21.81	28.89	26.81	27.91	24.71	25.13	26.84	21.67	28.25
	VolSDF [44]	26.28	25.61	26.55	26.76	31.57	31.50	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.90	34.75	30.38
	NeRF [28]	26.24	25.74	26.79	27.57	31.96	31.50	29.58	32.78	28.35	32.08	33.49	31.54	31.00	35.59	35.51	30.65
	NeuS [38]	25.82	23.64	26.64	25.60	27.68	30.83	27.68	34.04	26.61	31.35	29.29	28.08	28.55	31.28	33.68	28.79
	Neuralangelo* [25]	30.90	28.01	31.60	34.18	36.15	36.30	34.10	38.84	31.28	37.15	35.73	33.60	31.80	38.19	38.42	34.13
	Hi-NeuS(NeuS)	26.24	23.79	26.98	25.70	30.21	31.65	29.27	34.94	26.59	32.31	32.37	29.30	28.73	34.15	33.69	29.73
	Hi-NeuS(Neuralangelo)	30.80	28.01	31.50	29.82	36.12	36.17	34.06	39.04	31.13	37.18	35.62	33.71	31.53	38.01	38.07	34.05
Noise% ↓	Gaussian Surfels*† [10]	43.09	45.46	50.04	61.64	25.07	60.11	58.98	62.56	54.89	56.93	75.41	99.69	77.05	74.77	84.89	62.04
	NeuS [28]	40.75	60.50	56.83	72.60	32.27	28.69	26.07	75.41	43.14	64.46	57.33	17.35	15.47	8.53	11.03	39.13
	Neuralangelo* [25]	36.24	52.32	55.62	66.63	56.77	57.84	77.97	76.70	57.71	63.60	39.52	84.71	49.13	35.34	51.41	57.44
	Hi-NeuS(NeuS)	34.02	3.74	5.90	49.12	27.58	29.52	22.04	67.33	26.98	61.79	32.90	19.47	14.71	17.10	15.33	28.50
	Hi-NeuS(Neuralangelo)	32.36	44.25	39.16	59.96	43.01	34.23	68.31	61.89	58.68	60.71	36.15	17.45	21.54	28.42	57.60	45.67

* † denotes auxiliary data inputs, including 3D points from SFM or other pretrained models. * denotes our evaluated results with the available open-source configuration. The best performance is highlighted in red, while the second best is marked in orange for each measure and scene. Hi-NeuS(backbone) refers to the backbone architecture used in conjunction with our proposed geometric constraints.

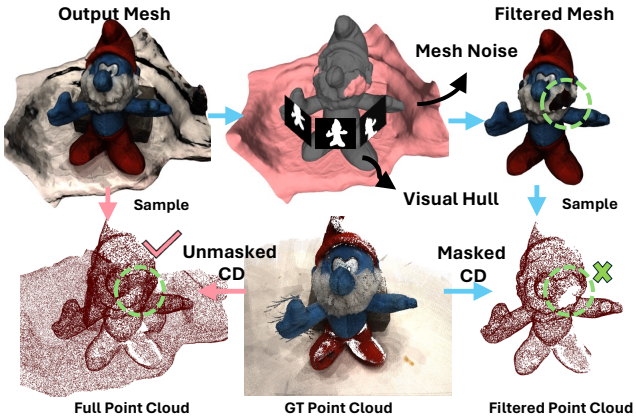
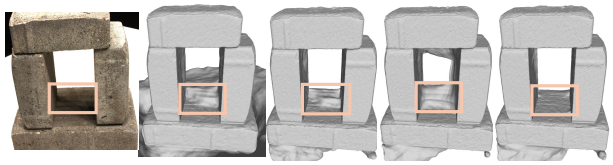


Figure 5: **The mesh post-processing and its evaluation**: Mesh noise refers to the space ratio outside the 3D visual hull created by silhouettes. The dashed circles highlight the areas where space is missing and must be evaluated. To evaluate this, we use sampled point clouds and GT point clouds to calculate the CD between the two. We compare the range of space to be evaluated between our proposed unmasked CD (red arrows) and the masked CD used in previous methods [25, 38, 30, 45] (blue arrows).



	NeuS	Hi-NeuS (w/o $\mathcal{L}_{global}$)	Hi-NeuS (w/o $\mathcal{L}_{surf}$)	Hi-NeuS (full)
↓ masked CD	0.81	0.88(+0.07)	0.98(+0.17)	0.68(-0.13)
↓ masked CD (Avg)	0.78	0.91(+0.13)	0.79(+0.01)	0.77(-0.01)
↑PSNR	26.64	26.72(+0.08)	26.76(+0.12)	26.88(+0.24)
↑PSNR(Avg)	28.79	29.06(+0.27)	29.57(+0.78)	29.72(+0.93)
↓ Noise%	56.83	6.91(-49.92)	13.12(-43.71)	8.47(-48.36)
↓ Noise%(Avg)	39.13	30.42(-8.71)	31.70(-7.43)	29.50(-9.63)

Figure 6: **Ablation study on proposed losses:** performance evaluation on scan 40 in the DTU dataset and the average results across the DTU dataset, with their performance comparisons relative to NeuS. The boxes emphasize the difference in mesh quality.

individual loss components in Eq. (10) and present the results in Fig. 6 for the challenging Scene 40 from the DTU dataset. The highlighted area reveals the differences in texture inside the opening. Our analysis shows that the brick structure with an opening in the center is reconstructed with high fidelity, capturing its fine geometric details. Notably, the global geometry consistency loss $\mathcal{L}_{global}$ effectively aligns surface points to more accurate positions, resulting in a 15% improvement in masked CD. However, this improvement comes at the cost of increased noise, with increased 22% noise observed after adding $\mathcal{L}_{global}$, as exemplified by artifacts at the bottom of the brick structure. In contrast, the surface regularization loss $\mathcal{L}_{surf}$ efficiently captures the surface boundary by penalizing absolute SDF errors on its zero-level set. This leads to a 6% reduction in noise and a marginal improvement in the CD measure. Combining both $\mathcal{L}_{global}$ and $\mathcal{L}_{surf}$, our full model achieves a more detailed reconstruction while maintaining compactness. Notably, the rendering fidelity is significantly enhanced by reducing geometric errors for both loss components, demonstrating the effectiveness of our proposed geometrical constraints.

Analysis on the training process. As shown in (a) of Fig. 7, we averaged each iteration across all scenes in the DTU dataset using the Neuralangelo backbone and compared it with our Hi-NeuS model. Our findings indicate that the masked CD for Hi-NeuS is

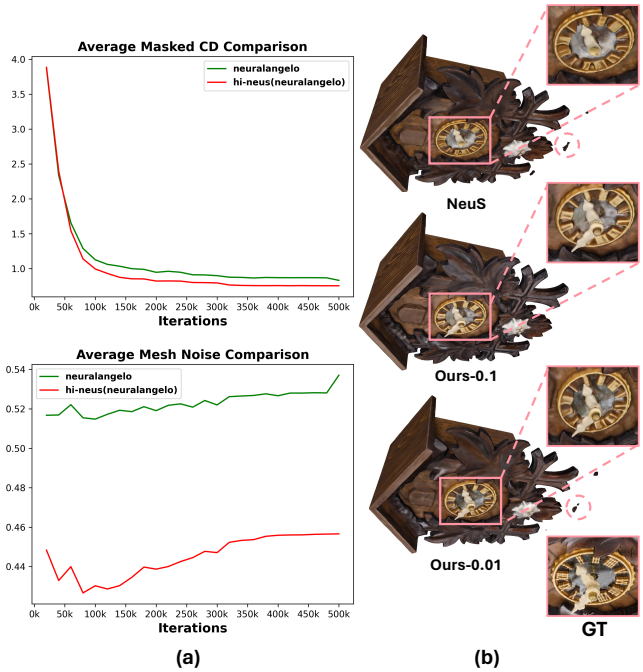
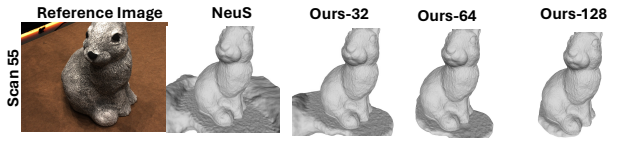


Figure 7: **(a)** Average masked CD and mesh noise on Neuralangelo and our revised Neuralangelo across DTU datasets during training. **(b)** Effectiveness of loss scale refinement on the BlendedMVS dataset with varying $\mathscr{L}_{geo}$ scales. The magnified box provides a detailed comparison with GT, while the dashed circles highlight the reduction of artifacts.

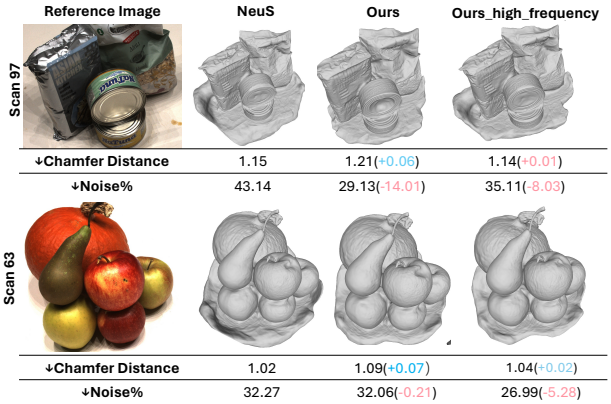


↓Masked CD	0.38	0.37(-0.01)	0.38(+0.00)	0.37(-0.01)
↑PSNR	25.60	26.98(+1.38)	26.88(+1.28)	26.90(+1.30)
↓Noise%	72.60	49.12(-23.48)	33.06(-39.54)	15.61(-56.99)

Figure 8: **The effectiveness of grid resolution**: refinement on DTU scan 55 by varying grid resolutions.

consistently lower than that of the baseline, especially during the early training stages between 50k and 200k iterations. Additionally, our model exhibits significantly less noise throughout the training process, with the noise level reaching its minimum around 100k iterations and then gradually increasing. In contrast, Neuralangelo experiences a sharp noise increase, especially at the end of training. These observations demonstrate that our model is more accurate and stable and converges faster compared with the baseline. We refer interested readers to Sec.4.1 of our *suppl.* for more analysis and visual demonstrations.

Strength of Geometric Refinement. We investigate the effectiveness of Hi-NeuS in adjusting the strength of geometric refinement through the grid buffer resolution and loss scale. Increasing the grid resolution enhances the accuracy of the grid in recording more fine-grained rendering weights, leading to a more compact structure. As shown in Fig. 8, scene 55 achieves a more compact result without compromising PSNR or noise. However, it is worth noting that increasing the resolution may not be universally beneficial, particularly for scenes with highly reflective materials, where it may lead to uneven distribution of rendering weights. As shown in Fig. 9, the metallic material in scene 97 exhibits CD degradation, as depicted in the distorted tin surface. Specifically, the high reflectivity in one direction causes the SDF to blend toward the reflected directions, resulting in a less noisy but more distorted surface. On the other hand, as depicted in (b) of Fig. 7, adjusting the loss scale has a similar effect on the compactness and accuracy of the mesh output. Increasing the loss scale produces more compact mesh outputs with reduced noise. In comparison to a loss scale of 0.01, a higher loss scale like 0.1 may result in less noise given more intensive constraints. Overall, our results indicate that Hi-NeuS effectively scales the global refinement with varying grid resolutions and loss scales, ensuring optimal performance in different scenarios.



↓Chamfer Distance	1.15	1.21(+0.06)	1.14(+0.01)
↓Noise%	43.14	29.13(-14.01)	35.11(-8.03)
↓Chamfer Distance	1.02	1.09(+0.07)	1.04(+0.02)
↓Noise%	32.27	32.06(-0.21)	26.99(-5.28)

Figure 9: **To solution on the highly reflective case**: We increase our proposed gird buffer refreshing frequency and compare the results with the previous version.

Our solution for highly reflective scenarios. As previously discussed, our method can be challenging to handle in scenarios with high reflectance variations, such as metals in scene 97 and smooth fruit surfaces in scene 63. To address this, we introduce a hyperparameter to regulate the buffer update frequency. Instead of collecting images from all camera views, we refresh the buffer from scratch after a certain number of views to record upcoming values. This approach is particularly effective in scenarios with challenging light reflections, as illustrated in Fig. 9. For example, Scan 97 in the DTU dataset features highly reflective surfaces, resulting in intensive light contributions from accumulated perspectives. This leads to distortion in the mesh despite lower noise levels. To mitigate this issue, we employ a higher frequency update, using fewer images to accumulate rendering weights, which provides more instant feedback on the ongoing training status. As shown in Fig. 9, the higher frequency update substantially reduces distortion. This approach strikes a balance between maintaining consistency across multiple views and mitigating distortion introduced by buffer delay.

Training time and potential improvements. Our proposed global geometry refinement module requires additional time compared to the NeuS backbone. On average, training time across scenes is approximately 30% slower. However, testing time remains unchanged, as the SDF and color fields have the same parameter size and we use the same marching cube resolution. In the NeuS's setup, our training time for one case was around 10 hours, compared to the 8-hour baseline. To further accelerate *real-world* applications, we integrate our adaptable geometrical constraints into the Instant-NGP implementation with improved CUDA parallelism [15]. This integration significantly reduces training time, from *around 10 minutes* per case for our adapted algorithms in the new module.

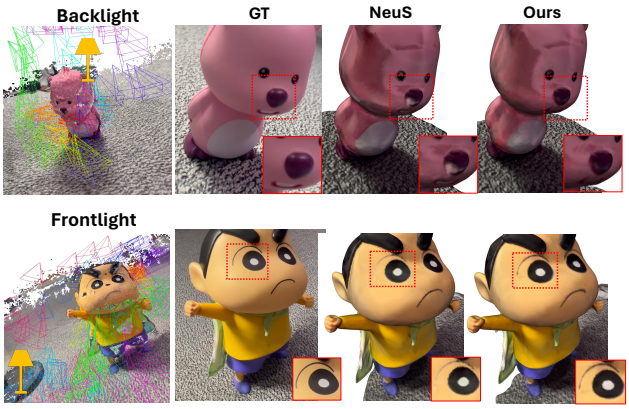


Figure 10: **Real-world application with our handheld phone captures**: We compare different lighting conditions, including back and front lights. The yellow icon denotes where our laptop is set. The magnified boxes reveal the visual quality differences.

4.3 Phone Capturing and Reconstruction Pipeline

We present our pipeline for capturing and reconstructing objects from phone-captured videos to 3D meshes. The pipeline consists of three stages: (1) preprocessing videos with COLMAP to estimate camera poses, (2) training Hi-NeuS to learn the underlying geometry, (3) extracting meshes from the learned SDF and color fields, and (4) editing meshes on software for artistic creation. In this study, we focus on object-centric capturing. For more challenging real-world scenarios, such as forward-facing and aerial circling, please refer to Section 5 of our *suppl.* material.

(1) Preprocessing videos with COLMAP. Given a short video, we use COLMAP[33] to estimate camera poses. Before running COLMAP, we evenly sample the video frames to around 80 or 160 frames. As shown in Fig. 10, we visualize estimate camera trajectories and the dense map output of COLMAP. Note that in practice, only the sparse mapping is required for pose estimation. The COLMAP processing takes approximately 2 minutes using the exhaustive matching method for optimal pose quality.

(2) Launch Hi-NeuS training. With posed RGB frames, we perform geometry refinement using NeuS or its variants. As illustrated in Fig. 3, we first gather multi-view rendering weights and set a grid buffer to manage them during training. Our global geometry refinement process then reduces geometry bias during training.

(3) Extract meshes. Given a batch of grid samples and camera poses, we evaluate each sample's signed distance values and vertex colors. We then perform marching cubes on the signed distance values to extract the surface and assign each surface vertex with its corresponding color.

(4) Content creation in computer platform. Most extracted meshes can be used directly or with minor edits, thanks to our proposed algorithm. As shown at Fig. 11 (a), we further refine them using software such as Meshlab[9] and Blender[6] for mesh editing. For instance, as shown in Fig. 1, we added a Santa hat to the toy's head and adjusted the lighting in the scene to enhance coherence during rendering.

(5) Application in VR. The reconstructed objects can then be combined in SimLab Composer [1] from the computer platform like Fig. 11 (b) and put into the headset for an immersive VR/AR experience like Fig. 11 (c). The pipeline streamlines the surface reconstruction from object-centric views.

Discussion on the light condition. In our phone-capturing process, we find that lighting conditions are a critical factor in obtaining high-quality meshes. Specifically, when comparing backlight and frontlight conditions in Fig. 10, we observe that backlight conditions often result in dark artifacts or shallow details, whereas frontlight conditions tend to capture more realistic details. To achieve optimal results, users are recommended to capture objects under well-lit conditions, ideally with the light source positioned in front of the object. Furthermore, in terms of quality validation, our geometric refinement approach provides more realistic textures compared to NeuS, thanks to the refined geometry.

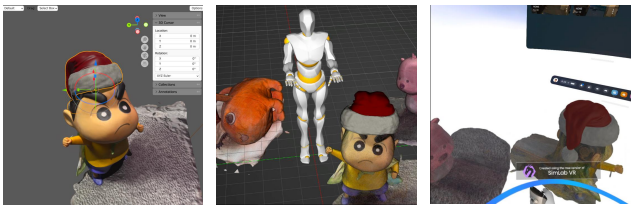


Figure 11: **Cross-platform viewing and editing**: **(a)** The mesh editing in Blender. **(b)** The mesh composition by SimLab Composer. **(c)** The mesh viewing with VR headset.

Supplemental Materials

We direct readers to our *suppl.* materials for the video demonstration, which includes the complete processing pipeline and visual comparisons with rotating camera views. Additionally, the paper's other discussions are also included.

5 Conclusion and Future Work

In this paper, we introduced Hi-NeuS, a novel rendering-based neural implicit surface reconstruction framework that leverages SDF-based volume rendering with our proposed global geometrical constraint. Our algorithm enabled recovering more compact and precise surfaces without relying on multi-view object masks. The capability and performance of our framework have been rigorously tested against the SOTA models with various datasets, demonstrating superior generalized performance in reducing geometry errors and recovering intricate details. By streamlining the geometry-capturing process, our framework has the potential to enable the geometry extraction directly from phone-captured data to meshes. This reduces the need to annotate multi-view object masks, facilitating seamless viewing and content creation in VR/AR.

Future work. We plan to adapt Hi-NeuS to more baselines and other datasets to further verify its ability. Furthermore, we would like to explore and execute more effective geometry constraints to boost our geometry accuracy.

Acknowledgments

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ.

References

[1] Simlab composer. `https://www.simlab-soft.com/3d-products/simlab-composer-main.aspx`. Accessed: [Insert Date].

[2] M. Agrawal and L. Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Aug 2005. doi: 10.1109/cvpr.2001.990999

[3] A. Andrew. Multiple view geometry in computer vision. *Kybernetes*, p. 1333–1341, Dec 2001. doi: 10.1108/k.2001.30.9_10.1333.2

[4] H. Bai, Y. Lin, Y. Chen, and L. Wang. Dynamic plenoctree for adaptive sampling refinement in explicit nerf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8785–8795, 2023.

[5] M. Baorui, Z. Han, Y. Liu, and M. Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *Cornell University - arXiv,Cornell University - arXiv*, Nov 2020.

[6] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2022.

[7] D. Chen, H. Lu, I. Feldmann, O. Schreer, and P. Eisert. Dynamic multi-view scene reconstruction using neural implicit surface. Feb 2023.

[8] D. Chen, P. Zhang, I. Feldmann, O. Schreer, and P. Eisert. Recovering fine details for neural implicit surface reconstruction. Nov 2022.

[9] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. Meshlab: an open-source mesh processing tool. In *European Interdisciplinary Cybersecurity Conference*, 2008.

[10] P. Dai, J. Xu, W. Xie, X. Liu, H. Wang, and W. Xu. High-quality surface reconstruction using gaussian surfels, 2024.

[11] F. Darmon, B. Bascle, J.-C. Devaux, P. Monasse, and M. Aubry. Improving neural implicit surfaces geometry with patch warping. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022. doi: 10.1109/cvpr52688.2022.00616

[12] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-nerf: Foveated neural radiance fields for virtual reality, 2022.

[13] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2007. doi: 10.1109/cvpr.2007.383246

[14] Z. Group. Z potentials interview with luma ai: Revolutionizing 3d content creation with multimodal ai and photorealistic capture, 2023. Accessed: 2024-08-12.

[15] Y.-C. Guo. Instant neural surface reconstruction, 2022. https://github.com/bennyguo/instant-nsr-pl.

[16] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields, 2024.

[17] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanaes. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014. doi: 10.1109/cvpr.2014.59

[18] Y. Jiang, C. Yu, T. Xie, X. Li, Y. Feng, H. Wang, M. Li, H. Lau, F. Gao, Y. Yang, and C. Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality, 2024.

[19] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023.

[20] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

[22] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Jan 1999. doi: 10.1109/iccv.1999.791235

[23] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 150–162, Jan 1994. doi: 10.1109/34.273735

[24] C. Li, S. Li, Y. Zhao, W. Zhu, and Y. Lin. Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering, 2022.

[25] Z. Li, T. Müller, A. Evans, R. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin. Neuralangelo: High-fidelity neural surface reconstruction.

[26] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views, 2022.

[27] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00459

[28] 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*, p. 405–421. Jan 2020. doi: 10.1007/978-3-030-58452-8_24

[29] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.00356

[30] M. Oechsle, S. Peng, and A. Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. doi: 10.1109/iccv48922.2021.00554

[31] B. Peng, J. Hu, J. Zhou, X. Gao, and J. Zhang. Intrinsicngp: Intrinsic coordinate based hash encoding for human nerf, 2023.

[32] S. Rojas, J. Zarzar, J. C. Perez, A. Sanakoyeu, A. Thabet, A. Pumarola, and B. Ghanem. Re-rend: Real-time rendering of nerfs across devices, 2023.

[33] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

[34] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. *Pixelwise View Selection for Unstructured Multi-View Stereo*, p. 501–518. Jan 2016. doi: 10.1007/978-3-319-46487-9_31

[35] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Nov 2002. doi: 10.1109/cvpr.1997.609462

[36] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, p. 903–920, Sep 2012. doi: 10.1007/s00138-011-0346-8

[37] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao. Nerf-art: Text-driven neural radiance fields stylization, 2022.

[38] 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. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, Jun 2021.

[39] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. Dec 2022.

[40] Y. Wang, I. Skorokhodov, and P. Wonka. Improved surface reconstruction using high-frequency details. Jun 2022.

[41] J. Xie, A. Yuille, and et al. Accurate segmentation in large-scale datasets. *Journal of Computer Vision*, 127(2):252–265, 2018.

[42] G.-W. Yang, W.-Y. Zhou, H.-Y. Peng, D. Liang, T.-J. Mu, and S.-M. Hu. Recursive-nerf: An efficient and dynamically growing nerf, 2021.

[43] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.00186

[44] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. *Neural Information Processing Systems,Neural Information Processing Systems*, Dec 2021.

[45] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, R. Basri, and Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. Jan 2020.

[46] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. doi: 10.1109/iccv48922.2021.00570

[47] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes, 2024.

[48] J. Zhang, Y. Yao, S. Li, T. Fang, D. McKinnon, Y. Tsin, and L. Quan. Critical regularizations for neural surface reconstruction in the wild. Jun 2022.

[49] H. Zhu, T. He, and Z. Chen. Cmc: Few-shot novel view synthesis via cross-view multiplane consistency, 2024.

In this supplementary document, we provide (1) The algorithm overview along with other implementation details; (2) further discussion on the multi-object annotation; (3) additional ablation studies, different types of real-world capturing, and full quantitative and qualitative results; (4) the video demos.

6 Additional Implementation Details

In the global surface searching module illustrated in Sec.3.3.1, , we add contrast on the collected rendering weights based on buffer statistics values, $w_i' = max(w_i + \delta(w_i - \frac{1}{n}\sum_i w_i), 0)$, where w_i is volume rendering weight, and δ is a hyperparameter to balance the strength of the contrast adjustment. This method can reduce the ambient noise while making it more possible to sample on more valuable surface regions.

Algorithm 1 Iteration t of Hi-NeuS Training

Input: camera poses, RGB pixels, grid buffer; **Output**: predicted pixel color $\hat{c}_{r_i}$, global geometric constraints $\mathscr{L}_{geo}$.

if Iter t > 1 **then**
  Access buffer for weights $\vec{w_{t-1}}$ and SDF $f(\vec{x_{t-1}})$.
  Resample supervision $\vec{x_t} \sim P(x_{t-1}|w_{t-1})$
end if
for pose $= 0$ **to MaxPose do**
  Sample $H \times W$ rays from a the given camera pose.
  for $i = 0$ **to** $H \times W$ **do**
    Calculate ray-grid intersection for ray r_i to get $n_{r,i}$ hits.
    Sample N points with normalized location along r_i.
    Calculate volumetric rendering color $\hat{C}_{r_i}$.
    Recording w_i and $f(x_i)$ in to grid buffer at x_i.
  end for
end for
Normalize the buffer for $\vec{w_t}$ and $f(\vec{x_t})$.
Predict surface points $\vec{x_q}$ given queries $\vec{x_t}$.
Calculate $\mathscr{L}_{geo}(\vec{x_q}, \vec{x_t})$ and other losses.
Update network parameters via optimizer.
Refresh buffer with $\vec{w_t}$ and $f(\vec{x_t})$.

7 Discussion on Mult-view Image Annotation

As shown in Fig. 12, the object masks vary significantly in their level of detail. For instance, Scene 37 requires intricate binary segmentation for elements like scissors with thin edges. Each scene in the DTU dataset includes either 49 or 64 images, making the per-scene annotation process extremely labor-intensive to achieve precise segmentation. Therefore, manual annotation becomes cumbersome in this context. Additionally, Scenes 83 and 115 lack detailed masks for the bricks supporting toys, which affects the accuracy of performance evaluation on recovered bricks. This omission highlights the need for an approach like Hi-NeuS to handle segmentation without relying on foreground masks. Obtaining such masks is not only cumbersome but also poses challenges even for advanced models like the Segment Anything Model (SAM). For instance, the SAM and SAM-E results in Fig. 12 show noticeable artifacts, which may pose issues for maintaining multi-view consistency when filtering meshes. The complexity and effort required for manual annotation in datasets like DTU have been documented in various studies. For example, [41] discusses the difficulties in achieving accurate segmentation in large-scale datasets due to high annotation costs and the time-consuming nature of the process. Similarly, [20] highlights the limitations of automatic segmentation models when dealing with fine-grained details in objects like thin edges and intricate shapes. By addressing these challenges, Hi-NeuS aims to provide a more efficient solution for object masking in complex scenes without the need for exhaustive manual annotation or heavy reliance on pre-existing foreground masks.

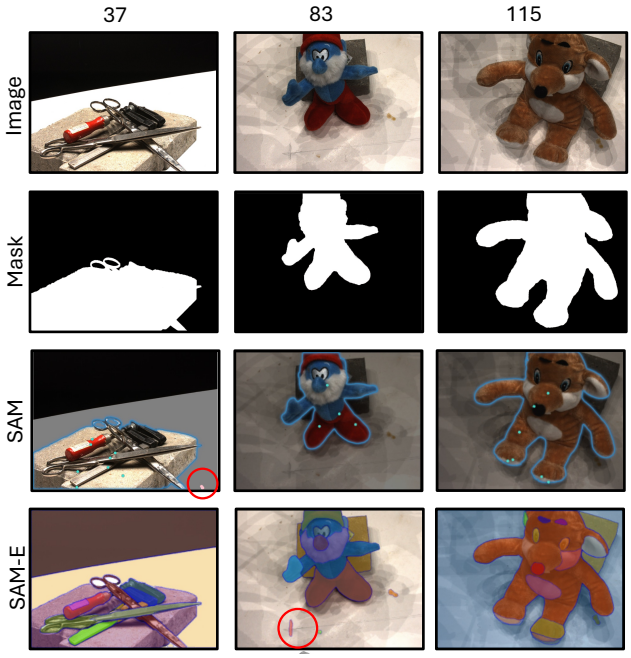


Figure 12: **Object masks for scenes in DTU.** We denote the scene number above. SAM and SAM-E are extracted mask results by hover & click and everything modes, respectively. The artifacts of SAM predictions are highlighted by red circles.

8 Performance

8.1 Additional ablation studies

Training quality and convergence. In Fig. 13, we show the norm maps during the training process of Hi-NeuS and compare them with NeuS. We observe that the compact result appears at a very early stage of training, for example, at 20k/300k or 10k/500k iterations. This indicates that Hi-NeuS's SDF representation remains compact, focusing on objects rather than surrounding noise, which demonstrates improved geometry accuracy at earlier stages. Throughout the training process, we maintain this compactness, whereas our baselines tend to accumulate noise, potentially due to uncertainty accumulation. Notice that Neuralangelo has more noise in both the surroundings and objects. In contrast, Hi-NeuS successfully produces a more compact structure, achieving compactness and geometry accuracy throughout training.

8.2 Overall performance on the selected model variants

In Tab. 2, we identify the optimal model variants across different grid resolutions, where all selected variants are highlighted for each scene. During our model selection, we prioritize the ones with less mesh noise rather than geometric accuracy and rendering quality. In Fig. 15 and Fig. 16, we list all uncolored mesh results for NeuS and Neuralangelo.

8.3 Training cost

We use an NVIDIA A800-SXM4-40GB GPU to evaluate the training cost, averaging the results on the DTU dataset. For NeuS, we report memory consumption as follows: 10.25GB for a resolution of $128 \times 128 \times 128$, 9.58GB for $64 \times 64 \times 64$, 9.03GB for $32 \times 32 \times 32$, and 8.23GB for our NeuS baseline. For Neuralangelo, the memory

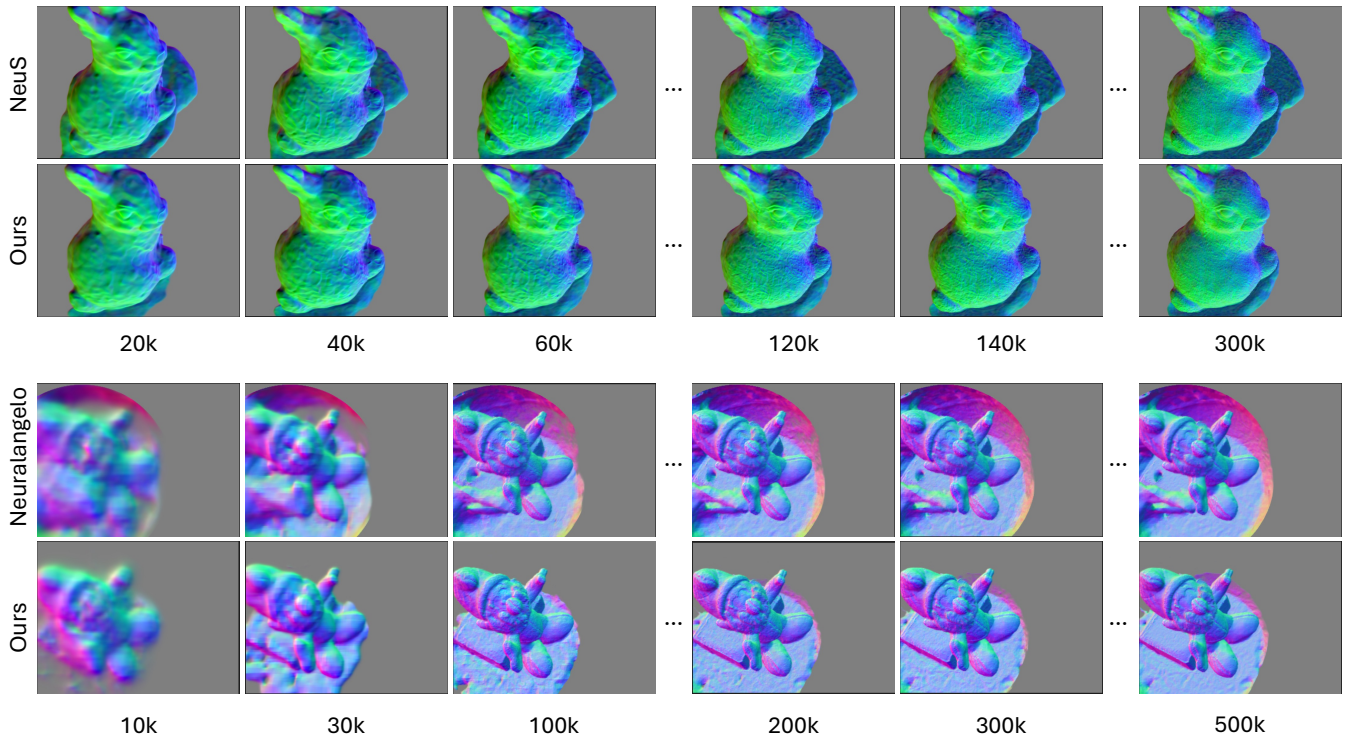


Figure 13: **The quality comparison between Hi-NeuS and NeuS based on different baselines during Training.**

Table 2: Quantitative results on DTU dataset [17]. Proposed method consistently boosts the performance on the NeuS and Neuralangelo.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
unmasked CD ↓	NeuS [38]	1.59	1.98	1.44	0.95	1.82	0.74	0.64	1.63	1.30	1.41	0.59	1.33	0.44	0.51	0.54	1.13
	Hi-NeuS(NeuS)-32	0.96	0.93	0.76	0.78	1.43	0.71	0.66	1.90	1.02	1.18	0.58	2.00	0.45	0.52	0.56	0.96
	Hi-NeuS(NeuS)-64	0.97	0.95	0.71	0.56	1.37	0.71	1.02	1.45	1.05	1.11	0.59	1.27	0.44	0.51	0.55	0.88
	Hi-NeuS(NeuS)-128	0.95	1.10	1.07	0.47	1.37	0.68	0.66	1.47	1.25	1.07	0.60	3.86	0.44	0.50	0.54	1.07
	Gaussian Surfels [10]	1.00	1.97	1.06	1.74	2.32	2.35	2.02	3.48	2.45	2.55	2.31	8.13	1.49	2.69	3.48	2.60
	Neuralangelo [25]	0.62	1.63	0.66	0.56	1.51	1.38	2.60	2.03	2.15	1.11	0.46	1.31	0.48	0.95	1.25	1.25
	Hi-NeuS(Neuralangelo)-64	0.55	1.55	0.61	0.60	1.51	0.77	2.25	1.19	1.52	1.09	0.43	1.20	0.43	0.88	1.36	1.06
masked CD ↓	NeRF [28]	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
	VolSDF [44]	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
	RegSDF† [48]	0.60	1.41	0.64	0.43	1.34	0.62	0.60	0.90	0.92	1.02	0.60	0.59	0.30	0.41	0.39	0.72
	NeuralWarp† [11]	0.49	0.71	0.38	0.38	0.79	0.81	0.82	1.20	1.06	0.68	0.66	0.74	0.41	0.63	0.51	0.68
	D-NeuS† [8]	0.44	0.79	0.35	0.39	0.88	0.58	0.55	1.35	0.91	0.76	0.40	0.72	0.31	0.39	0.39	0.61
	NeuS [38]	0.93	1.07	0.81	0.38	1.02	0.60	0.58	1.42	1.15	0.78	0.57	1.16	0.35	0.45	0.46	0.78
	Hi-NeuS(NeuS)-32	0.77	0.90	0.73	0.37	1.00	0.59	0.59	1.42	1.19	0.79	0.56	1.93	0.35	0.45	0.48	0.81
	Hi-NeuS(NeuS)-64	0.85	0.92	0.68	0.38	1.09	0.57	0.65	1.40	1.21	0.80	0.57	1.11	0.34	0.44	0.47	0.77
	Hi-NeuS(NeuS)-128	0.76	1.07	0.95	0.37	0.99	0.56	0.59	1.45	1.25	0.89	0.59	3.78	0.33	0.45	0.46	0.98
	Neuralangelo [25]	0.39	0.72	0.35	0.33	0.82	0.74	1.70	1.34	1.95	0.71	0.47	1.00	0.33	0.82	0.78	0.83
	Hi-NeuS(Neuralangelo)-64	0.39	0.71	0.36	0.33	0.92	0.55	1.42	1.25	1.44	0.73	0.45	0.99	0.33	0.70	0.73	0.75
PSNR ↑	RegSDF† [48]	24.78	25.31	23.47	23.06	22.21	28.57	25.53	21.81	28.89	26.81	27.91	24.71	25.13	26.84	21.67	28.25
	VolSDF [44]	26.28	25.61	26.55	26.76	31.57	31.50	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.90	34.75	30.38
	NeRF [28]	26.24	25.74	26.79	27.57	31.96	31.50	29.58	32.78	28.35	32.08	33.49	31.54	31.00	35.59	35.51	30.65
	NeuS [38]	25.82	23.64	26.64	25.60	27.68	30.83	27.68	34.04	26.61	31.35	29.29	28.08	28.55	31.28	33.68	28.79
	Hi-NeuS(NeuS)-32	26.24	23.79	26.98	25.70	30.21	31.65	29.27	34.94	26.59	32.31	32.37	29.30	28.73	34.15	33.69	29.73
	Hi-NeuS(NeuS)-64	26.25	23.76	26.88	25.63	30.50	31.57	29.14	34.90	26.55	32.27	32.27	29.43	28.83	34.00	33.89	29.72
	Hi-NeuS(NeuS)-128	26.14	23.56	26.90	25.48	30.22	31.38	29.23	35.06	26.65	32.56	31.89	24.30	28.86	34.02	34.08	29.66
	Neuralangelo [25]	30.90	28.01	31.60	34.18	36.15	36.30	34.10	38.84	31.28	37.15	35.73	33.60	31.80	38.19	38.42	34.13
	Hi-NeuS(Neuralangelo)-64	30.80	28.01	31.50	29.82	36.12	36.17	34.06	39.04	31.13	37.18	35.62	33.71	31.53	38.01	38.07	34.05
Noise% ↓	NeuS [28]	40.75	60.50	56.83	72.60	32.27	28.69	26.07	75.41	43.14	64.46	57.33	17.35	15.47	8.53	11.03	39.13
	Hi-NeuS(NeuS)-32	34.02	3.74	5.90	49.12	27.58	29.52	22.04	67.33	26.98	61.79	32.90	19.47	14.71	17.10	15.33	28.50
	Hi-NeuS(NeuS)-64	34.02	4.18	8.47	33.06	32.06	30.76	43.08	66.89	29.13	61.40	32.63	17.45	14.44	17.75	17.19	29.50
	Hi-NeuS(NeuS)-128	45.72	3.39	5.23	15.61	25.56	34.72	10.76	65.59	34.78	53.71	32.41	15.34	14.31	5.52	12.94	24.90
	Gaussian Surfels [10]	43.09	45.46	50.04	61.64	25.07	60.11	58.98	62.56	54.89	56.93	75.41	99.69	77.05	74.77	84.89	62.04
	Neuralangelo [25]	36.24	52.32	55.62	66.63	56.77	57.84	77.97	76.70	57.71	63.60	39.52	84.71	49.13	35.34	51.41	57.44
	Hi-NeuS(Neuralangelo)-64	32.36	44.25	39.16	59.96	43.01	34.23	68.31	61.89	58.68	60.71	36.15	17.45	21.54	28.42	57.60	45.67

* † denotes auxiliary data inputs, including 3D points from SFM or other pretrained models. We denote our models as Hi-NeuS(backbone)-grid resolution, with selected variants highlighted. Compared to the baselines, our models demonstrate superior performance, highlighted in red, while the sub-optimal is marked in blue for each measure and scene.

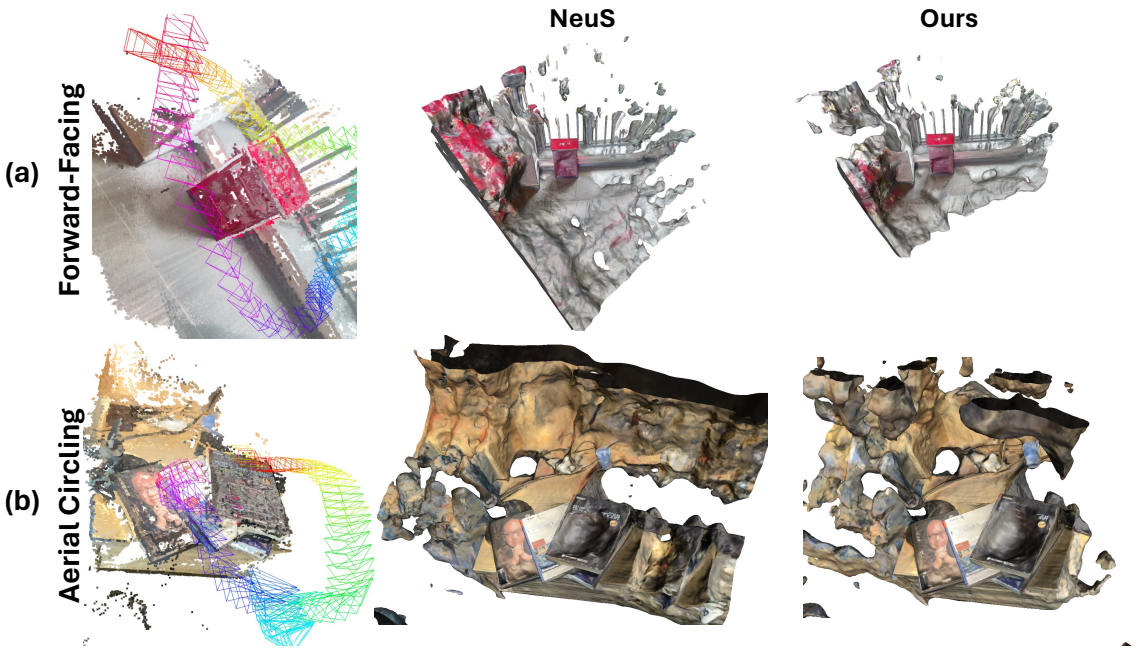


Figure 14: **More challenging real-world capturing.** The first column depicts the COLMAP results with our predicted camera poses. The other two columns compare the reconstructed scenes with the same resolution and camera viewpoints on their quality.

cost at a resolution of $64 \times 64 \times 64$ is 19.47GB, compared to its baseline of 18.98GB. The inference speed is 0.08 seconds per iteration, compared to the original 0.05 seconds per iteration.

8.4 More Challenging Real-world Capturing

To assess our model's capability in surface reconstruction in real-world scenarios, where videos may not cover sufficient views as people prefer to shoot videos while walking freely. We categorize human capturing scenarios into two main types, excluding the object-centric approach:

(1) Forward-Facing: Camera poses primarily focus on the front parts of objects, leaving back regions under-explored.

(2) Aerial Circling: When cameras are placed above objects, views are mostly concentrated on the upper regions, potentially neglecting the bottom and side views.

As illustrated in Fig. 14, we evaluate the performance of Hi-NeuS in comparison to its NeuS backbone. The results show that the reconstructed scenes exhibit improved compactness with reduced artifacts. This suggests that similar to the object-centric approach, our method can focus on the regions that capture most of the overlaps from diverse view perspectives. This can align with users' intentions on the region of interest and mitigate geometry bias, particularly when predicted camera poses are not sufficiently accurate.

9 Video Demo

We attach the video demo of our full capturing and reconstruction process, the key idea illustration, and visualization with the quality comparison with the rotating camera views.

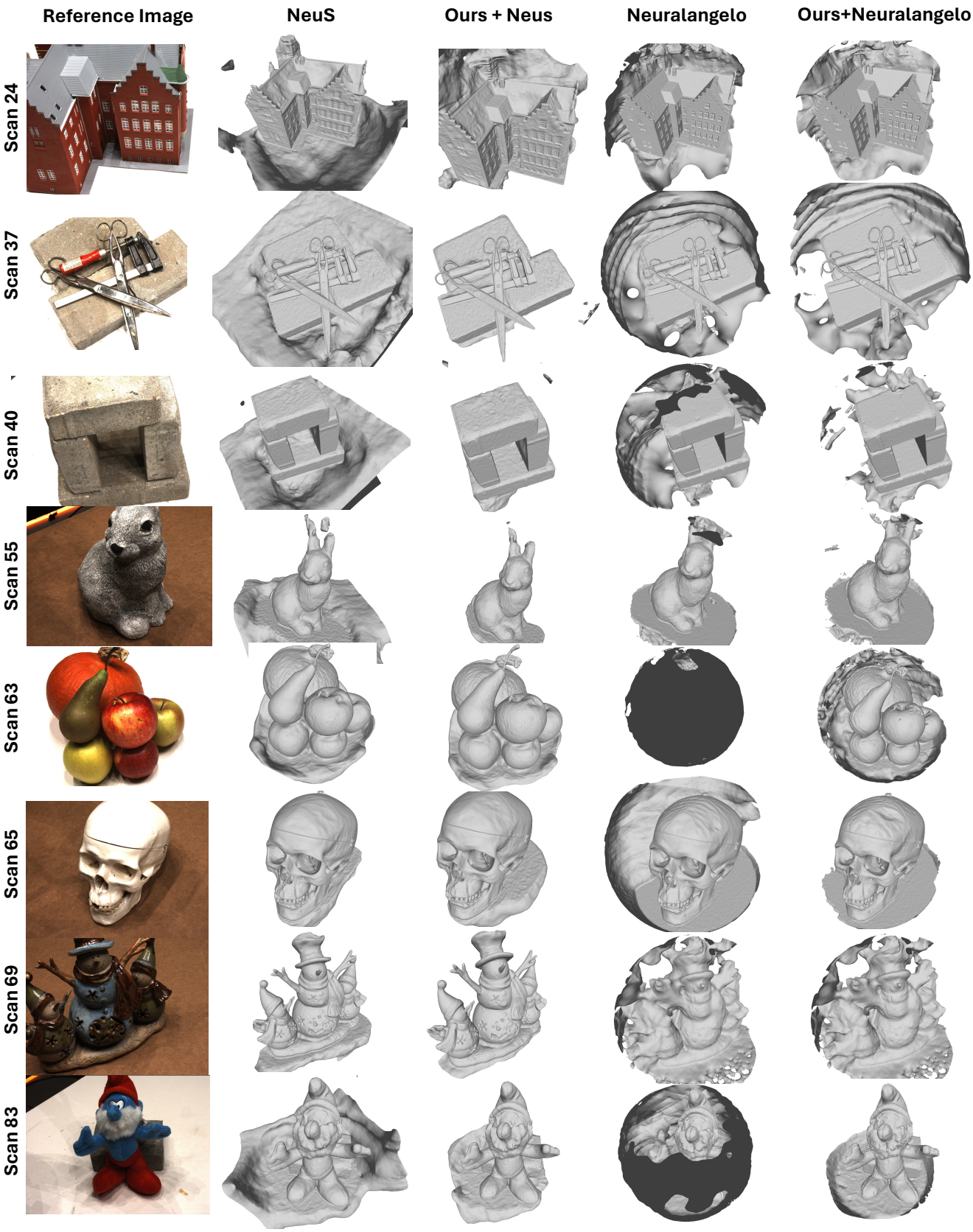

Reference Image
NeuS
Ours + Neus
Neuralangelo
Ours+Neuralangelo
Scan 24
Scan 37
Scan 40
Scan 55
Scan 63
Scan 65
Scan 69
Scan 83

Figure 15: **The full evaluation result on the DTU dataset.**

Figure 16: **The full evaluation result on the BlendedMVS dataset.**