

Total-Decom: Decomposed 3D Scene Reconstruction with Minimal Interaction

Xiaoyang Lyu* Chirui Chang* Peng Dai Yang-Tian Sun Xiaojuan Qi†

The University of Hong Kong

{shawlyu, chiruichang, sunyt98}@connect.hku.hk, {daipeng, xjqi}@eee.hku.hk

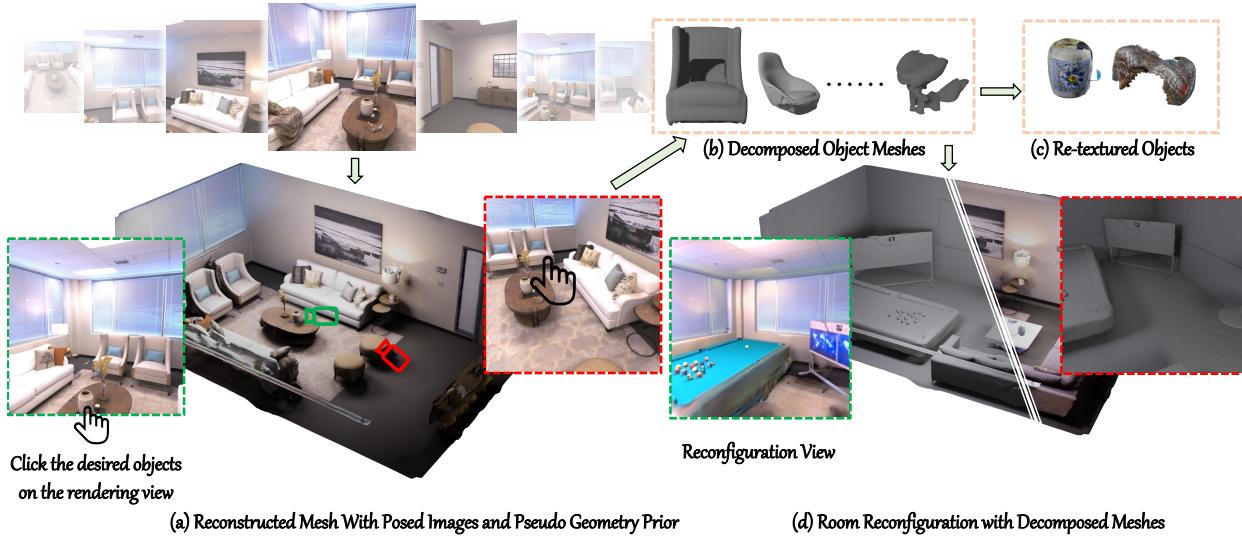


Figure 1. Indoor scenes consist of complex compositions of objects and backgrounds. Our proposed method, *Total-Decom*, (a) performs 3D reconstruction from posed multiview images, (b) decomposes the reconstructed mesh to generate high-quality meshes for individual objects and backgrounds with minimal human annotations. This approach facilitates such applications as (c) object re-texturing and (d) scene reconfiguration. For additional demonstrations, please refer to our supplementary materials and videos.

Abstract

Scene reconstruction from multi-view images is a fundamental problem in computer vision and graphics. Recent neural implicit surface reconstruction methods have achieved high-quality results; however, editing and manipulating the 3D geometry of reconstructed scenes remains challenging due to the absence of naturally decomposed object entities and complex object/background compositions. In this paper, we present Total-Decom, a novel method for decomposed 3D reconstruction with minimal human interaction. Our approach seamlessly integrates the Segment Anything Model (SAM) with hybrid implicit-explicit neural surface representations and a mesh-based

region-growing technique for accurate 3D object decomposition. Total-Decom requires minimal human annotations while providing users with real-time control over the granularity and quality of decomposition. We extensively evaluate our method on benchmark datasets and demonstrate its potential for downstream applications, such as animation and scene editing. The code is available at <https://github.com/CVMI-Lab/Total-Decom.git>.

1. Introduction

Scene reconstruction from multi-view images is a fundamental problem in computer vision and graphics [13, 14, 24, 27, 29, 31, 32]. Recently, neural implicit surface reconstruction methods such as VolSDF [43] and NeuS [39] have been proposed to address this problem and have achieved high-

*Equal contribution.

†Corresponding author.

quality results. However, editing and manipulating the 3D geometry of a reconstructed scene remains challenging due to the absence of naturally decomposed object entities and complex object/background compositions. Such functionality is, however, desired for many real-world applications, such as editing, animation, and simulation. Consequently, we are motivated to investigate decomposed 3D reconstruction, which enables the extraction of desired object-level shapes and facilitates scene manipulations such as reorganizing objects in a scene (see Fig. 1).

A few attempts have been made to decompose a reconstructed 3D scene into individual objects using separate Multi-Layer Perceptron (MLP) layers to represent specific objects [19, 22, 40, 41]. However, these approaches face scalability issues when dealing with scenes containing numerous objects [19]. Furthermore, the success of these methods heavily relies on human-annotated instance masks on multi-view images during training, which poses challenges in obtaining them for large-scale practical applications. Moreover, even with ground-truth instance labels, the existing state-of-the-art method [41] still fails to produce satisfactory results, with multiple objects missing, as shown by the second row of Fig. 7, due to the inherent difficulties in separating all objects using implicit representations.

In this paper, we introduce *Total-Decom*, a novel method designed for decomposed 3D reconstruction with minimal human interaction. At the core of our method lies the integration of the Segment Anything Model (SAM) [17]—an interactive image segmentation model—hybrid implicit and explicit surface representation, and a mesh-based region-growing approach. This integration allows the decomposition of a scene into the background and individual objects with minimal interactions and provides users with control over the granularity and quality of decomposition via real-time interactions, as shown in Fig. 1.

Specifically, our method first employs an implicit neural surface representation for its ability to achieve dense and complete 3D reconstruction from images. At this stage, we also integrate object-aware information by distilling image features from the SAM model for follow-up efficient interaction and accurate decomposition. After obtaining the learned implicit surface and features, our approach further extracts explicit mesh surfaces while distilling features into their vertices. The explicit representation provides valuable geometry topology information for scene decomposition and enables real-time neural rendering to enhance human interactions. Then, in order to identify and separate the desired object for surface decomposition, we utilize the SAM decoder and the rendered SAM feature, converting a human-annotated click on a single rendered image view into corresponding dense object masks. Thanks to the segmentation capability of SAM and our feature rendering design, this interactive process also allows users to obtain the

desired objects at different granularities while minimizing human interactions and avoiding high computational costs. Lastly, with the derived object mask from a single view with good object boundaries serving as object seeds, we propose a mesh-based region-growing module that progressively expands these seeds along the mesh surface to obtain decomposed 3D object surfaces. This process leverages distilled feature similarities of vertices and 3D mesh geometry topology for accurate object decomposition, further ensuring precision by confining the growing process to mask boundaries obtained from the SAM decoder.

We extensively validate our approach on benchmark datasets. More importantly, our high-quality decomposed 3D reconstruction enables many downstream applications in manipulating and animating objects in virtual environments, including re-texturing [4] with diffusion model, deformation, and motion. (See Fig. 1 and videos in supplementary).

In sum, our main contributions are as follows:

- We introduce a novel pipeline that seamlessly integrates the segment anything model with hybrid implicit-explicit neural surface representations for 3D decomposed reconstruction from sparse posed images. Our approach requires minimal human annotations (approximately one click per object on average) while achieving high decomposition quality.
- We propose a new mesh-based region-growing method that leverages the geometry topology of 3D mesh, feature similarities among vertices, object masks, and boundaries derived from the SAM model to accurately identify and extract object surfaces for decomposition.
- We perform an extensive evaluation of our approach on various datasets, demonstrating its superior ability to decompose objects with high accuracy. Furthermore, we showcase the potential of our results for numerous downstream tasks, such as animation and scene editing.

2. Related Work

Object Compositional Reconstruction In order to better reconstruct the geometry of objects in complex scenes, many research efforts have been devoted to exploring better object-level compositional scene representations. For instance, ObjSDF [40] proposes a compositional scene representation to assist in geometry optimization in highly composite scenes and better object-level extraction with the help of multi-view consistent instance labels. Kong *et al.* [19] build an object-level scene model from a real-time RGB-D input stream for object-compositional SLAM. Wu *et al.* build upon ObjSDF and further mitigate object occlusions in complex scenes through an object distinction regularization term. Additionally, a new occlusion-aware object opacity rendering scheme is introduced to reduce the negative impact of occlusion on scene reconstruction. How-

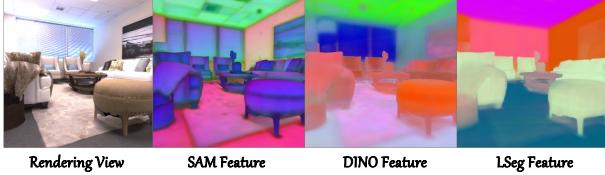


Figure 2. Visualization for distilled generalized features.

ever, these methods rely heavily on accurate multi-view consistent ground-truth instance-level labels and cannot effectively preserve all objects within the reconstruction.

Decompositional Neural Rendering Neural radiance fields (NeRFs) offer a unified representation to represent the appearance [25, 44], and other spatially varying properties [20]. In the field of decompositional neural rendering, many research efforts investigate the distillation of generalized features [8, 16, 18, 35], or segmentation predictions [1, 33] from large-scale pre-trained image backbones, such as DINO [2, 28], CLIP features [30] and panoptic segmentation [5], to neural fields. Specifically, Semantic NeRF [47] explores incorporating regressing semantic labels into the process of Novel View Synthesis. Building upon this, distilled feature fields (DFFs) [18] and neural feature fusion fields (N3F) [35] have emerged as pioneering approaches for distilling [12] semantic features from pretrained models, such as LSeg [21] and DINO [2], into NeRF for decomposition. ISRF [8] does not rely solely on feature matching, but instead obtains the final results through region growing. These approaches leverage feature similarities or predicted instance/semantic codes for grouping and implicitly decomposing a scene, requiring no additional human annotation efforts and offering greater scalability for large-scale scenes. However, when directly applied to the reconstruction task, these strategies often produce decomposed outputs that are incomplete and exhibit poorly defined contours (see Fig. 3) due to ill-defined object boundaries [8] or multiview inconsistent and low-quality segmentation predictions [5, 11, 37] used during training. In contrast to existing methods, our work not only avoids the need for extensive manual interaction through a geometry-guided feature approach but also enables the extraction of any constituent part of a complex scene.

NeRF with SAM Recently, segmentation methods based on different representations [5, 17, 21, 23] have developed rapidly. SAM [17], as an emerging vision foundation model, achieves efficient 2D interactive segmentation capabilities through extensive high-quality annotated supervision. It brings forth more possibilities in this field and has the potential to help achieve thorough decomposition of complex scenes with plausible 2D segmentation capabilities. Recently, some concurrent works have explored the combination of SAM and NeRF for decomposition. SA3D [3] incorporates cross-view self-prompting technique to obtain multi-view consistent masks. However, running

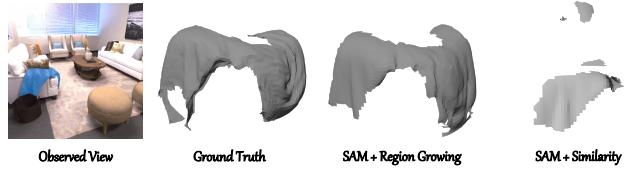


Figure 3. Comparison on different decomposition methods with SAM feature. SAM + region growing represents object extraction with our method. SAM + similarity indicates object extraction with similarity matching in 3D space, following [18, 35].

SAM multiple times incurs high computational costs, and the segmentation results are sensitive to interaction trajectories, leading to unstable performance.

3. Empirical Study on General Visual Features

A central challenge in decomposed 3D scene reconstruction is incorporating object-aware knowledge to accurately separate individual objects and backgrounds. While existing methods have explored the use of ground-truth multi-view consistent instance-level annotations [19, 22, 40, 41], these approaches suffer from high annotation costs and scalability issues. Motivated by recent advancements in vision foundation models providing generic features, we investigate their potential for object decomposition and reducing human annotation requirements. Although this strategy has been examined in neural rendering [8, 18, 35], it remains underexplored in decomposed 3D reconstruction, which necessitates more precise boundary information. Accordingly, we investigate features from three foundation models: CLIP-LSeg [21], DINO [2], and SAM [17].

We utilize the MonoSDF [45] for implicit neural surface reconstruction, augment it with a feature rendering head, and distill features from the above 2D backbones. The rendered features after distillation are depicted in Fig. 2. We observe that: (1) distilled CLIP-LSeg features cannot distinguish objects of the same categories; (2) distilled DINO features lack accurate object boundaries; and (3) distilled SAM features preserve object boundaries. However, none of these features are discriminative enough to accurately separate different object instances. For example, while SAM features perform the best, directly grouping 3D objects based on similar feature responses leads to the merging of distant areas due to the absence of geometry and object boundary information, as shown in Fig. 3.

Despite the above shortcomings, we observe that the distillation brings the features of the same object from different views closer than the original 2D features from SAM, suggesting that high feature similarity often indicates a high likelihood of belonging to the same object, as shown in Fig. 4. Consequently, we propose a novel approach that leverages SAM features and a mesh-based region-growing method to decompose a 3D scene with minimal human an-

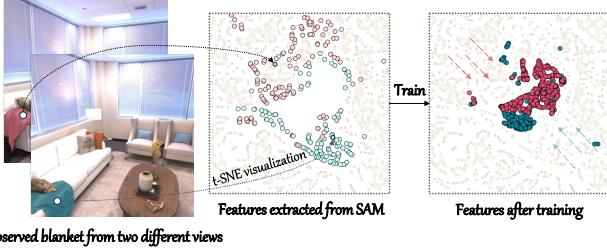


Figure 4. Visualization of the SAM feature for the same object in different views with t-SNE [36]. All the features are in the same feature space.

notations, typically requiring only one click per object.

4. Overview

Our objective is to reconstruct a 3D scene from multi-view images and decompose it into individual object entities and the background while minimizing the need for human annotations. To achieve this, we propose a novel pipeline that integrates SAM into a hybrid implicit-explicit surface representation, combined with a mesh-based region-growing method to effectively identify and decompose arbitrary 3D objects within a scene. The overview of our approach is illustrated in Fig. 5.

As shown in Fig. 5 (Left) and detailed in Sec. 5, we first adopt an implicit neural surface representation to achieve dense and complete 3D reconstruction from images while incorporating object-aware information by distilling image features from the SAM model into an implicit neural feature field (see Fig. 5: bottom). Importantly, we introduce geometry-guided regularization integrated with semantic priors to disentangle foreground objects and backgrounds, including occluded and invisible background regions with details unfolded in Sec. 5.

Upon obtaining the learned implicit surface and features, our approach further extracts explicit mesh surfaces for foreground objects and background; see Fig. 5: Background $d_B(p)$ and Foreground $d_F(p)$; and distills features into their vertices; see Fig. 5: Neural Mesh. The explicit mesh surface provides valuable geometry topology information for follow-up scene decomposition and enables real-time neural rendering to enhance human interactions. Then, given rendered images and features from the mesh surface, our method employs the SAM decoder to convert image clicks into dense object masks to precisely identify the target object in one rendered view (see Fig. 5: Interactive Decomposition) and allow users control over granularity and quality while minimizing human interactions. Details are elaborated in Sec. 6. Notably, by leveraging the rendered SAM features from our model, we only require the decoding process of the SAM model at this stage, thus circumventing high computational costs.

Lastly, using the vertices corresponding to dense object masks as seeds, we propose a mesh-based region-growing module that progressively expands the seeds along the mesh surface to obtain object surfaces as detailed in Sec. 6. This module harnesses the feature similarities of vertices and the geometry topology of the 3D mesh to achieve accurate object decomposition. The growing process is also confined by the vertices corresponding to mask boundaries, which further ensures the precision of object decomposition.

5. Neural Implicit Feature Distillation and Surface Reconstruction

In this stage, we employ an implicit neural field for reconstruction from posed images, disentangle foreground and background using geometric priors, and distill features from the SAM encoder to incorporate object-aware information. An illustration of our implicit reconstruction is depicted in Fig. 5. In the following, we first elaborate on our reconstruction network and rendering formula, and then detail our core design for achieving background and foreground separation while reconstructing occluded foreground areas.

Reconstruction network and SDF-based neural implicit surface representation. We use the signed distance function $d(\mathbf{p})$ to represent the geometry of the surface at each point \mathbf{p} . Considering a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}$ from a camera position \mathbf{o} in the direction of \mathbf{v} , we calculate the signed distance function of each sampled point \mathbf{p} using the geometry MLP and use three MLPs to predict the color $C(\mathbf{p}, v)$, semantic logits $S(\mathbf{p})$, and generalized feature $F(\mathbf{p})$ distilled from the SAM encoder, respectively; refer to Fig. 5 for details. To apply the volume rendering formula, we follow VolSDF [43] to convert the signed distance function to volume density $\sigma(\mathbf{p})$:

$$\sigma(\mathbf{p}) = \begin{cases} \frac{1}{2\beta} \exp\left(\frac{-d(\mathbf{p})}{\beta}\right) & \text{if } d(\mathbf{p}) \geq 0 \\ \frac{1}{\beta} - \frac{1}{2\beta} \exp\left(\frac{d(\mathbf{p})}{\beta}\right) & \text{if } d(\mathbf{p}) < 0 \end{cases}, \quad (1)$$

where $\beta > 0$ is a learnable parameter to decide the sharpness of the surface density. Then, we use the volume rendering formula [15] to obtain outputs \mathbf{E} of the target pixel,

$$\hat{\mathbf{E}}(r) = \sum_{i=1}^M T_i^r \alpha_i \hat{e}_i^r, \quad (2)$$

where $\hat{e} \in \{\hat{c}, \hat{n}, \hat{d}, \hat{s}, \hat{f}\}$ represent the predicted color, normal, depth, semantic logits, generalized feature. T_i^r and α_i represent the transmittance and alpha value (a.k.a opacity) of the sample point, and their values can be computed by

$$T_i^r = \prod_{j=1}^{i-1} (1 - \alpha_j), \quad \alpha_i = 1 - \exp(-\sigma_i^r \delta_i^r), \quad (3)$$

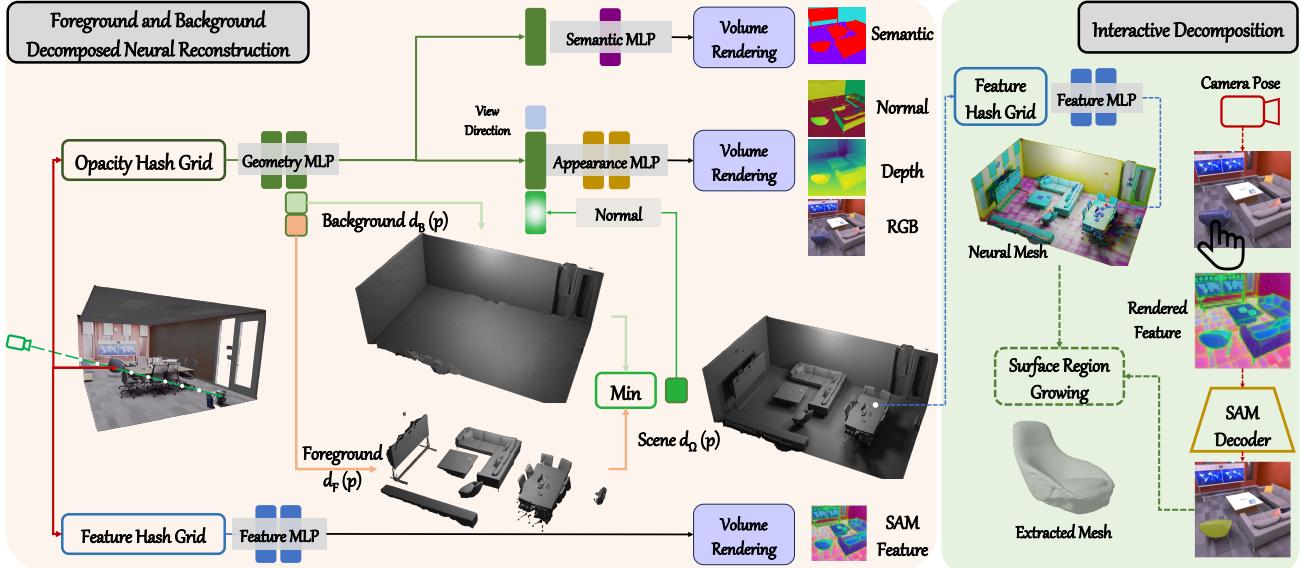


Figure 5. Overview of Total-Decom. (1) **Foreground and background decomposed neural reconstruction.** We have four networks in this stage to predict the geometry, appearance, semantic, and SAM features per point. We follow the ObjSDF++ [41] to use the foreground and background compositional representation with pseudo geometry priors and apply min operation to construct the whole scene. Notably, the foreground is constrained with object distinct loss (Eq. (6)) and the background is regularized with Manhattan loss (Eq. (7)) and floor reflection loss (Eq. (8)). Furthermore, we also train a solely feature network to render the generalized features. (2) **Interactive Decomposition.** We firstly extract the SAM feature from the feature network into the vertices of the reconstruction mesh. Subsequently, for any given pose, we can render a color image and a feature image. Passing the feature image and user-selected prompt into the SAM decoder allows us to obtain the 2D mask of the regions of interest. Utilizing our newly proposed surface region-growing algorithm, we can then acquire the 3D mesh corresponding to these regions. Our method enables the user to select objects with varying levels of granularity, requiring just one or two clicks.

where δ_i^r is the distance between adjacent sample points.

We follow the loss function \mathcal{L}_{rgb} and \mathcal{L}_{geo} in MonoSDF [45] to optimize the rendered color, depth, and normal. For the rendered semantic $\hat{S}(r)$, we use the cross-entropy loss defined as

$$\mathcal{L}_{\text{sem}} = -\mathbb{E}_{\mathbf{r} \in \mathcal{R}} \left[\sum_{l=1}^L P_l(r) \log \hat{P}_l(r) \right], \quad (4)$$

where $P_l(r), \hat{P}_l(r)$ are the multi-class semantic probability as class l of the ground truth map and rendering map for ray r , respectively. Additionally, we use the $L2$ loss \mathcal{L}_f to optimize the rendered generalized feature $\hat{F}(r)$ for distilling the $F(r)$ from the SAM encoder.

Modeling foreground and background compositional scene geometry. To represent foreground and background geometry separately, we construct two different SDF fields $\mathcal{S} = \{\mathcal{F}, \mathcal{B}\}$ following [41]. The single scene Ω is the composition of the $\Omega = \mathcal{F} \cup \mathcal{B}$. The scene SDF can be calculated as the minimum of two fields SDFs $d_\Omega(\mathbf{p}) = \min\{d_{\mathcal{F}}(\mathbf{p}), d_{\mathcal{B}}(\mathbf{p})\}$. To learn the geometry from the supervision of foreground and background masks, we adopt occlusion-aware opacity rendering [41] to guide the learning of different field surfaces. The loss function is defined

as:

$$\mathcal{L}_O = \mathbb{E}_{\mathbf{r} \in \mathcal{R}} \left[\sum_{\mathcal{S}_i \in \mathcal{S}} \left\| \hat{O}_{\mathcal{S}_i}(\mathbf{r}) - O_{\mathcal{S}_i}(\mathbf{r}) \right\| \right], \quad (5)$$

where $\hat{O}(r) = \int_{t_n}^{t_f} T(\mathbf{r}(t)) \alpha(\mathbf{r}(t)) dt$ to formulate the occlusion-aware object opacity in the depth range $[t_n, t_f]$.

For reconstructing the clean foreground mesh, we follow the object distinction regularization term [41] forcing each point in a single scene to be only located inside one field, which is defined as follows:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{\mathbf{p}} \left[\sum_{d_{\mathcal{S}_i}(\mathbf{p}) \neq d_{\Omega}(\mathbf{p})} \text{ReLU}(-d_{\mathcal{S}_i}(\mathbf{p}) - d_{\Omega}(\mathbf{p})) \right], \quad (6)$$

Compared with the foreground, the background is more difficult to reconstruct because it has many occluded areas that are not visible in all captured views. To regularize the reconstruction of these areas, we follow the Manhattan world assumption [10, 46], i.e., the surfaces of man-made scenes should be aligned with three dominant directions. We use this to regularize the reconstruction of the floor and the wall using,

$$\begin{aligned} \mathcal{L}_{\text{man}} = & \mathbb{E}_{r \in \mathfrak{F}} (\hat{p}_f(r) |1 - \hat{\mathbf{n}}(r) \cdot \mathbf{n}_f|) + \\ & \mathbb{E}_{r \in \mathfrak{W}} \left(\min_{i \in \{-1, 0, 1\}} \hat{p}_w(r) |i - \hat{\mathbf{n}}(r) \cdot \mathbf{n}_w| \right), \end{aligned} \quad (7)$$



Figure 6. The effect of different constraint on Replica *room-1*.

where \hat{p}_f, \hat{p}_w represent the probabilities of the pixel being floor and wall derived from the semantic MLP, $\mathfrak{F}, \mathfrak{W}$ are the sets of camera rays of image pixels that are labeled as floor and wall regions, \hat{n}_r is the rendering normal of rays r , $n_f = \langle 0, 0, 1 \rangle$ represent the assumed normal direction in the floor regions and n_w is a learnable normal to represent the normal direction in the wall regions.

As shown in Fig. 6, applying this constraint to regularize background reconstruction will yield more regular geometry but there still exist many undesired structures due to the inaccurate semantic information of these invisible regions. Fortunately, the majority of occlusions occur on the ground, and the ceiling corresponds one-to-one with the structure of the floor. Therefore, we can utilize the structural information of the ceiling to constrain the unknown areas on the ground. Specifically, when we have a frame of an image that observes the ceiling, we can start from points on the ceiling \mathbf{p}_c , emitting a ray along the direction of gravity $\mathbf{n}_g = \langle 0, 0, -1 \rangle$. In the background domain, the first point \mathbf{p}_f hit by this ray is considered as the ground. By employing the root finding method [27], we locate this point and constrain its normal vector by

$$\mathcal{L}_{\text{floor}} = |1 - \mathbf{n}(\mathbf{p}_f) \cdot \mathbf{n}_f|. \quad (8)$$

As shown in Fig. 6, this constraint can help us get the complete and regular background.

6. Interactive Decomposition

Upon obtaining the implicit foreground surface $d_{\mathcal{F}}$ and features, we focus on decomposing the foreground into individual objects and allowing users to control the decomposition through interactions in this stage, as shown in Fig. 5. We propose to use an explicit mesh surface as it can provide geometry information for better decomposition and allow efficient rendering by using rasterization or combining with Gaussian splatting. We provide details on the integration of Gaussian splatting in the supplementary material. Hence, before conducting decomposition, we extract foreground mesh $\mathcal{M}_{\mathcal{F}}$ and distill features to its vertices $\mathcal{V} = \{v_1, \dots, v_n\}$. Then, each vertex v_i is associated with a distilled feature f_{v_i} and a 3D location p_i . The vertices on the mesh are connected via edges \mathcal{E} , which are determined by the geometry topology of the mesh. As shown in Sec. 3, it is still challenging to rely solely on distilled features f_{v_i} to decompose 3D objects. Thus, we introduce

human annotations to identify each object and aim to minimize human interactions. The following details how we realize object decomposition based on a designed mesh-based region-growing method with human interactions using our designed method. We will first introduce how we obtain seed points for an object by combining SAM, rendered features, and human clicks. Then we elaborate on our new mesh-based region-growing algorithm for acquiring object-level meshes to realize a complete decomposition.

Object Seed Generation Given a $\mathcal{M}_{\mathcal{F}}$, the goal is to obtain a set of initial seed points for each object o by using human annotations. As shown in Fig. 5, with one image I containing o with corresponding feature map f_I rendered from \mathcal{M} , the user will produce a click c on the image identifying the desired object. Based on the click c , serving as the prompt, and f_I , a 2D mask m_o that describes the object can be efficiently obtained through the lightweight mask decoder of SAM. Users are also allowed to adjust its clicks according to m_o to refine it. Our experiment shows that most of the objects' mask m_o can be obtained with just one click. The pixels in mask m_o are then mapped to their corresponding vertices to yield the 3D object seeds denoted as \mathcal{S}_o for the follow-up region-growing algorithm. This process efficiently turns a single click into a set of vertices to enhance the accuracy of identifying object o . Besides, to locate object boundaries, we also extract the contour pixels c_o of m_o and map them to their corresponding vertices \mathcal{B}_o , which forms the boundary condition for region growing.

Region-growing on Mesh for Decomposition Given the seeds \mathcal{S}_o and boundary condition \mathcal{B}_o , we design a region-growing method to obtain the corresponding mesh M_o for object o . As illustrated in Algorithm 1 in the supplementary material, the seeds \mathcal{S}_o are progressively expanded along the mesh $\mathcal{M}_{\mathcal{F}}$ to include their connected neighboring vertices with high feature similarities. The boundary vertices \mathcal{B}_o constrain the propagation process by enforcing it to stop if including vertices that will be outside of mask m_o . This helps ensure the boundary accuracy of the decomposed object. It is worth mentioning that the geometric relationship between vertices is leveraged in this progressive expanding process, benefiting the extraction of objects and reducing computation. Compared to region-growing algorithm that purely relies on spatial location [8] or features, incorporating edges from the mesh as growth paths introduces a constraint from topological structure, thereby making the growth process more consistent with the geometric structure and avoiding including vertices that are not geometrically related with seed vertices but share similar features with them, making the extraction of object be accurate. We present more analysis on the region-growing algorithm in the supplementary material.

7. Experiments

7.1. Experiment Setup

Implementation Details. Our method is implemented using Pytorch and uses the Adam optimizer with a learning rate of $5e - 4$ for the tiny MLP part (2 layers with 256 channels for the geometry, appearance, and feature prediction, 1 layer with 128 nodes for the semantic prediction) In the reconstruction strategy, we minimize the loss

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{geo}} + \lambda_1 \mathcal{L}_O + \lambda_2 \mathcal{L}_{\text{reg}} + \\ & \lambda_3 \mathcal{L}_{\text{man}} + \lambda_4 \mathcal{L}_{\text{floor}} + \lambda_5 \mathcal{L}_{\text{sem}} + \lambda_6 \mathcal{L}_f, \end{aligned} \quad (9)$$

to optimize our implicit neural surface, where we set $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ as $0.1, 0.1, 0.01, 0.01, 0.5, 0.1$, respectively. More details can be found in the supplementary.

Dataset and Metrics Our experiments are mainly conducted on the **Replica** [34] dataset, which is a synthetic dataset with each providing accurate geometry, HDR textures and 3D instance annotations. We follow the selection of ObjSDF++ [41] to evaluate the effectiveness of our method. We report both instance-level and holistic reconstruction results on this dataset. The reconstruction results are mainly evaluated by Chamfer- L_1 and F-Score. To further demonstrate the robustness of our method, we also use the **ScanNet** [6] as the real-world dataset which provides 1513 scenes. Due to its lack the object object-level ground truth, we show the visualized assessment in the main paper. Besides the public dataset mentioned before, we also evaluate the performance of our method on the self-captured data, one is the room from the NICE-SLAM [48] and another is the self-captured billiard room. More details and results can be seen in the supplementary.

Compared Methods. The compared methods are mainly divided into two categories. The first one is the object compositional reconstruction method that uses multiple fields to represent each object with the supervision from ground truth instance masks, like ObjSDF++ [41]. We compare the instance level and holistic scene reconstruction quality with them. The second category is the volume density based methods that decompose each scene with generalized features, like ISRF [8], DFF [18]. Since this type of method does not introduce geometric constraints, we mainly compare the way of decomposition.

7.2. Results

Method	Scene Reconstruction		Decomposed Reconstruction	
	Chamfer- $\mathcal{L}_1 \downarrow$	F-score \uparrow	Chamfer- $\mathcal{L}_1 \downarrow$	F-score \uparrow
ObjSDF++	3.58	85.69	3.84 ± 0.02	79.49 ± 0.08
<i>Ours</i>	3.53	85.82	3.58 ± 0.01	81.70 ± 0.08

Table 1. Quantitative assessments from Replica dataset on scene and decomposed reconstruction.

Scene reconstruction and object decomposition on the Replica dataset.

To evaluate the decomposed reconstruction accuracy of *Total-Decom*, we conduct experiments on the Replica dataset as it provides ground-truth objects' meshes for evaluation. We mainly compared our approach with the ObjSDF++, the state-of-the-art method that decomposes the scene structure with pseudo geometry priors as far as we know. Because the number of decomposed objects from ObjSDF++ is limited (around 25 objects per scene), we only evaluate the foreground objects that ObjSDF++ can extract for a fair comparison. In reality, our approach can generally yield a more complete decomposition of the scene with more objects. The results are shown in Table 1. Although our approach doesn't rely on ground-truth instance masks for decomposition, our method still surpasses ObjSDF++ in both scene and decomposed object reconstruction results. It is worth noting that our approach only requires 1.41 clicks on average per object, while ObjSDF++ requires dense human instance annotations on multi-view images. Our reconstructed results also outperform ObjSDF++ qualitatively. As shown in Fig. 7, ObjSDF++ tends to inaccurately reconstruct the vase (*Room_0*) and trash bin (*Office_4*) within the background field or may fail to recover the structure of the chair (*Office_4*). With an elaborately mesh-based region-growing method facilitated by SAM, our method can deliver decomposed results of higher qualities.

Background reconstruction. We demonstrate the quality of background reconstruction in Fig. 6 and Fig. 7. Our approach consistently delivers high-quality clean background reconstructions and reconstructs occluded areas.

Comparison of Decomposition Capabilities We compare the decomposition capabilities of our model with existing methods in NeRF and 3D reconstruction in terms of whether they can support geometric decomposition, multi-grained decomposition, scene-level decomposition, and interactive selection. Tab. 2 presents the comparison results, indicating that our method is the sole approach encompassing all capabilities. Additional results and applications ex-

Method	Geometry	Multi-grained	Scene-level	Single-View Interaction
LSeg + DFF [18]	✗	✗	✗	✗
DINO + DFF/N3F [35]	✗	✓	✗	✗
ISRF [8]	✗	✓	✗	✗
SAMNeRF [3]	✗	✓	✗	✗
Panoptic Lifting [33]	✗	✗	✓	✗
AssetField [42]	✗	✗	✓	✗
vMAP [19]	✓	✗	✓	✗
ObjSDF++ [41]	✓	✗	✓	✗
<i>Ours</i>	✓	✓	✓	✓

Table 2. Comparison of decomposition capabilities between different methods.

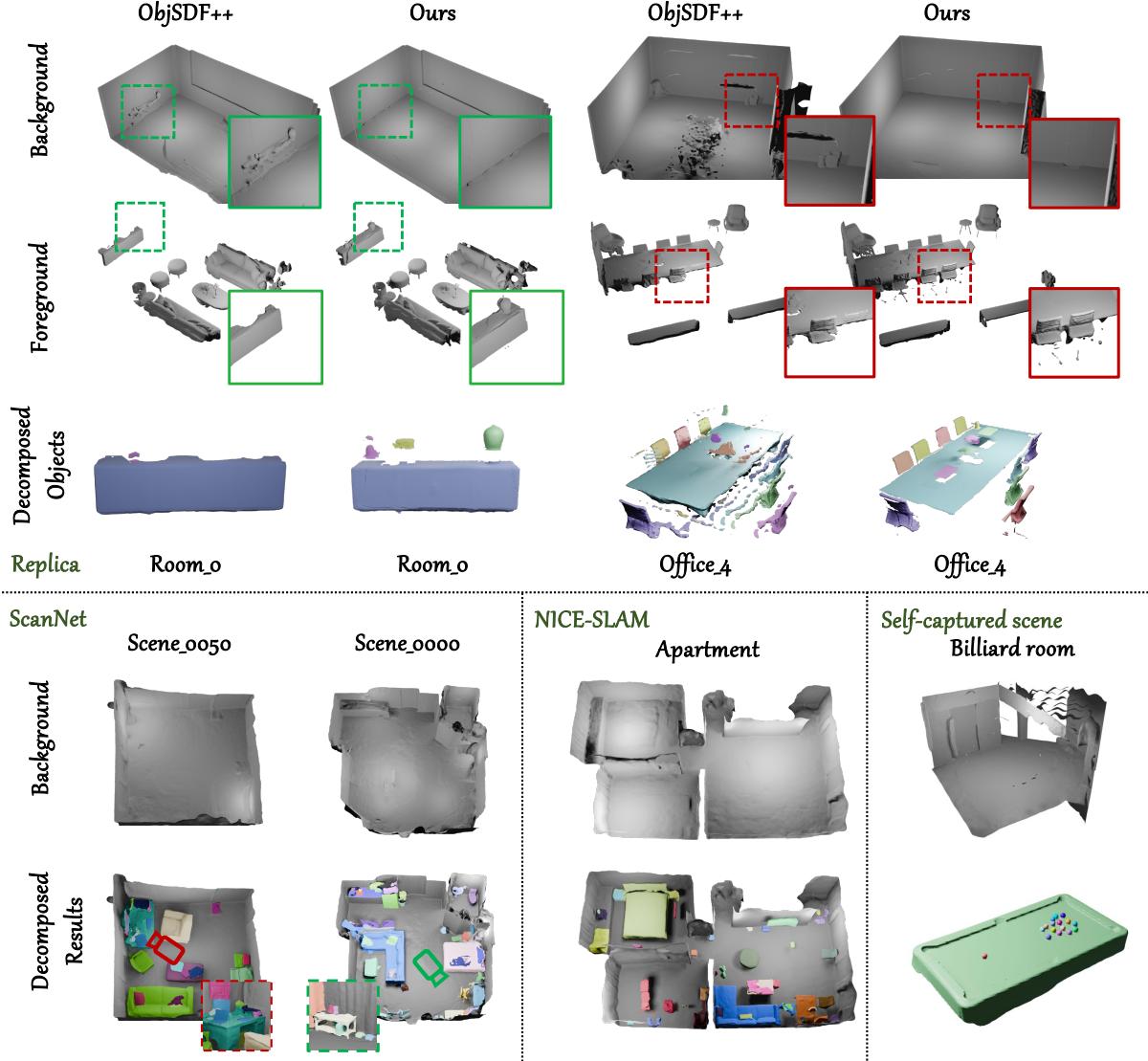


Figure 7. **Visualized assessments on different datasets.** We present the reconstruction results for the background, foreground and decomposed objects on Replica [34], ScanNet [6], NICE-SLAM [48] and our self-captured billiard room. To clearly visualize the decomposed objects, we use different color for the different instances.

ploiting these capabilities can be found in the supplementary material.

8. Conclusion

We presented *Total-Decom*, a novel framework for reconstructing a 3D surface and decomposing it into individual objects and backgrounds, significantly reducing the reliance on object annotations. Our approach centers on integrating SAM with hybrid implicit-explicit surface representations and a mesh-based region-growing algorithm. SAM provides object-aware features and facilitates the acquisition of more accurate object seeds for region growth. Simultaneously, the region-growing algorithm effectively combines geometric topology from explicit mesh and object-

aware features from SAM to accurately decompose desired objects with minimal human annotations. Qualitative and quantitative evaluations indicate that our method can yield not only precise geometry but also extract as many objects as possible. We hope our proposed method will facilitate the development of environment simulators in the future.

Acknowledgments: This work has been supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant No. 17202422), and RGC Matching Fund Scheme (RMGS). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

Supplementary

In the supplementary file, we provide more implementation details and more results not elaborated in our paper due to the paper length limit:

- Sec. S9: more implementation details.
- Sec. S10: more qualitative comparisons and qualitative results.
- Sec. S11: more ablation results
- Sec. S12: limitation analysis.

S9. Implementation Details

S9.1. Hash Grid in Implicit Reconstruction

Our entire pipeline primarily consists of four distinct networks: geometry network, appearance network, semantic network, and feature network. The geometry, appearance, and semantic networks share the same hash grid, while the feature network utilizes another. We follow instant-NGP [26] to construct the hash grid as a replacement for the frequency position encoding used in vanilla NeRF [25] to accelerate model convergence. Specifically, the 3D space is represented by multi-level feature grids:

$$R_l := \lfloor R_{\min} b^l \rfloor, b := \exp \left(\frac{\ln R_{\max} - \ln R_{\min}}{L-1} \right), \quad (10)$$

where $R_{\min} = 16$ and $R_{\max} = 2048$ represent the coarsest and finest resolutions, respectively. Each level grid has $T = 2$ dimensional features. Both the feature network and geometry network grids share the same structure.

S9.2. Loss Function

Our training objective consists of different losses including \mathcal{L}_{rgb} , \mathcal{L}_{geo} , $\mathcal{L}_{\mathcal{O}}$, \mathcal{L}_{reg} , \mathcal{L}_{man} , $\mathcal{L}_{\text{floor}}$, \mathcal{L}_{sem} , \mathcal{L}_f simultaneously, following [10, 41, 45]. Their detailed formulations are shown below. (1) \mathcal{L}_{rgb} is the color loss function defined as

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r} \in \mathcal{R}} \|C(\mathbf{r}) - \hat{C}(\mathbf{r})\|, \quad (11)$$

where $C(r)$ is the ground truth color value along the ray r and $\hat{C}(r)$ is the rendered color along the ray r . \mathcal{L}_{geo} is the geometry loss function to constrain the geometry of the implicit surface which includes three different parts $\mathcal{L}_{\text{depth}}$, $\mathcal{L}_{\text{normal}}$, \mathcal{L}_{eik} , defined as

$$\mathcal{L}_{\text{geo}} = 0.1\mathcal{L}_{\text{depth}} + 0.05\mathcal{L}_{\text{normal}} + 0.05\mathcal{L}_{\text{eik}}. \quad (12)$$

$\mathcal{L}_{\text{depth}}$ is the scale-invariant depth loss to regularize the rendering depth $\hat{D}(r)$ by the pseudo depth $\bar{D}(r)$ from Omnidata [7], which is defined as

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

Algorithm 1 Mesh-based Region Growing

```

1: Input:  $\mathcal{V}$  associated with  $f_{v_i}$  for each vertex  $v_i$ ,  $\mathcal{S}_o$ ,  $\mathcal{B}_o$ ,  $\mathcal{E}$ , a similarity threshold  $\tau$ , an attenuation parameter  $\theta$  and a tolerance  $\epsilon$ 
2: Output: the vertex set  $\mathcal{V}_o$  for the object mesh  $\mathcal{M}_o$ 
3: Initialize the target vertex set  $\mathcal{V}_o \leftarrow \mathcal{S}_o$ 
4: Initialize the intermediate target vertex set  $\mathcal{V}'_o \leftarrow \mathcal{V}_o$ 
5: initialize the candidate seed vertex set  $\mathcal{S}'_o \leftarrow \emptyset$ 
6: while  $\mathcal{S}_o \neq \emptyset$  do
7:   for each vertex  $s$  in  $\mathcal{S}_o$  do
8:     Find all the neighbors of  $s$  with  $\mathcal{E}$ , as  $\mathcal{N}$ 
9:     for each vertex  $n$  in  $\mathcal{N}$  do
10:       $\text{sim}(f_s, f_n) \leftarrow \frac{f_s \cdot f_n}{\|f_s\| \|f_n\|}$ 
11:      if  $\text{sim}(f_s, f_n) > \tau$  then
12:        add  $n$  to  $\mathcal{V}'_o$ , add  $n$  to  $\mathcal{S}_o$ 
13:      else
14:        add  $s$  to  $\mathcal{S}'_o$ 
15:      end if
16:    end for
17:  end for
18:  if  $\frac{|\mathcal{V}'_o \cap \mathcal{B}_o|}{|\mathcal{B}_o|} > \epsilon$  then
19:    return  $\mathcal{V}_o$ 
20:  else
21:     $\mathcal{V}_o \leftarrow \mathcal{V}'_o$ ,  $\mathcal{S}_o \leftarrow \mathcal{S}'_o$ ,  $\mathcal{S}'_o \leftarrow \emptyset$ 
22:  end if
23:   $\tau \leftarrow \tau - \theta$ 
24: end while

```

where w, q represents scale and shift solved by the least square method, and \mathcal{R} is the unit of all rays. $\mathcal{L}_{\text{normal}}$ is the normal loss defined as:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \|1 - \hat{N}(\mathbf{r})\bar{N}(\mathbf{r})\|_1, \quad (14)$$

where $\hat{N}(r)$ is the predicted normal from Omnidata [7] and $\bar{N}(r)$ is the rendered normal. \mathcal{L}_{eik} is the eikonal loss proposed by [9] to regularize the signed distance field, which is defined as

$$\mathcal{L}_{\text{eik}} = \sum \mathbb{E}_{d_\Omega} (\|\nabla d_\Omega(\mathbf{p})\| - 1)^2. \quad (15)$$

S9.3. Analysis of Region Growing

The detailed algorithm of mesh-based region growing is illustrated in Algorithm 1. The success of the mesh-based region growing algorithm hinges on propagating 2D guidance into the 3D space. The accurate mesh extraction can be attributed to the following reasons:

(1) The distilled 3D features are both view-consistent and instance-aware. As demonstrated in Fig. 4 of the main paper, for features of the same object under different viewpoints, the inter-class distance of the distilled feature pairs

is significantly smaller than that of the feature pairs derived from the teacher model [17]. Consequently, with the distilled features used in both 2D and 3D spaces, 2D seed pixels and boundary pixels can serve as reliable references for 3D seed vertices and boundary vertices.

(2) The SAM decoder, on top of rendered SAM features from one view, produces a dense mask that serves as the seeds for the region-growing process and constrains the boundary for growth.

(3) The explicit geometry information, i.e., vertices and edges, constrains the growing process by considering the topology of meshes, effectively ruling out vertices that have high feature similarities or are spatially adjacent but not geometrically connected.

(4) The separation of the foreground mesh \mathcal{M}_F allows the algorithm to operate effectively in low-noise environments by removing conflicting mesh vertices from the background. With a pure foreground mesh for growth, the absence of surfaces originating from the background minimizes interference during the extraction process.

In summary, these factors collectively enable our method to extract the 3D mesh of a specified object using only a single viewpoint and a few clicks.

S10. More Qualitative Comparisons and Qualitative Results

S10.1. ScanNet Results

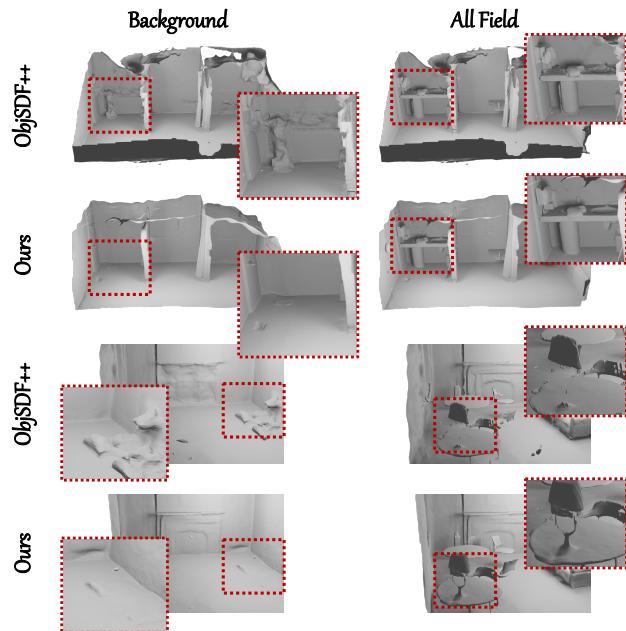


Figure S8. Visualization assessments of the reconstruction results on the ScanNet dataset.

To demonstrate the effectiveness of the foreground and

background reconstruction method. We also compare our method against various recent works on the ScanNet dataset as shown in Table S3. Our method achieves state-of-the-art on both Chamfer- \mathcal{L}_1 and F-score. In particular, our method obtains a significant increase on "Comp" and "Recall" because of the constraint on the invisible regions. However, the "Acc" and "Prec" are slightly decreasing because the ground truth mesh doesn't have the complete background mesh which will influence these two metrics.

As shown in Fig. S8, our method can obtain a cleaner background mesh and detailed foreground mesh on the real scene dataset which is much easier for cross-scene editing and getting the mesh asset for many other downstream applications.

Method	Acc ↓	Comp ↓	C- \mathcal{L}_1 ↓	Prec ↑	Recall ↑	F-score ↑
COLMAP [32]	0.047	0.235	0.141	71.1	44.1	53.7
UNISURF [27]	0.554	0.164	0.359	21.2	36.2	26.7
VolSDF [43]	0.414	0.120	0.267	32.1	39.4	34.6
NeuS [39]	0.179	0.208	0.194	31.3	27.5	29.1
Manhattan-SDF [10]	0.072	0.068	0.070	62.1	56.8	60.2
NeuRIS [38]	0.050	0.049	0.050	71.7	66.9	69.2
MonoSDF [45]	0.035	0.048	0.042	79.9	68.1	73.3
ObjSDF++	0.039	0.045	0.042	78.1	70.6	74.0
Ours	0.044	0.040	0.042	74.7	74.8	74.7

Table S3. Quantitative assessments of the proposed model against previous works on the ScanNet dataset.

S10.2. Real Time Interaction on the Replica dataset

In the main paper, we introduce *Total-Decom*, a method capable of decomposing an entire scene at any granularity level. We have also designed a graphical user interface (GUI) to interactively decompose desired objects for downstream applications. The foreground and background reconstructed meshes are loaded simultaneously, and the vertices are used as fixed initialization points to train the Gaussian Splatting model for real-time rendering. Regarding feature rendering, we utilize the trained grid and apply the rasterization method to obtain the feature map of the observed view. The rendered features and selected prompts are then passed into the SAM decoder to generate the mask for region-growing. Further details are showcased in the accompanying video.

S10.3. More results on scenes defying Manhattan assumption

We tested the influence of Manhattan constraints on the TNT auditorium, which features sloped ground. As shown in Fig. S9, our constraint effectively eliminates floaters and yields smooth floors. This success is because the optimization focuses on minimizing overall loss (Eq.(9)), and the regularization term (Eq.(7~8)) only penalizes heavily reconstructions that substantially violate the constraints, such

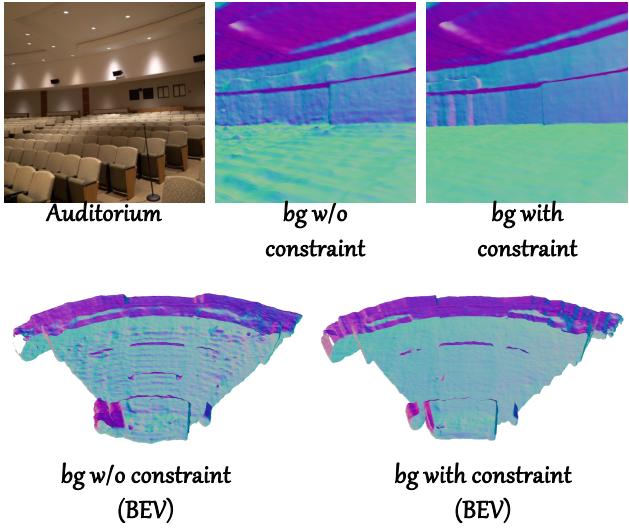


Figure S9. Background reconstruction results for Auditorium scene, consider moving this to the supp



Figure S10. Visualization of mask2former results.

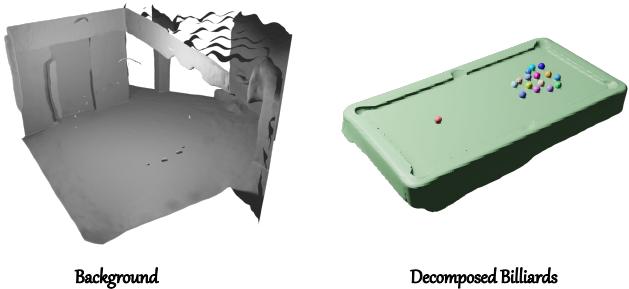


Figure S11. Visualization decomposed results on the self-captured dataset.

as floaters on the floor. The combinatorial effects of all loss terms make the optimization robust toward corner cases.

S10.4. More Demonstrations and Evaluations on Self-captured Data

We utilize self-captured data to demonstrate the generalization and practicality of our method. The data are captured using Apple ARKit. Our method can successfully separate the background and foreground, as well as interactively decompose the desired objects, as illustrated in Fig. S11. Moreover, with 16 clicks, we can successfully obtain the 16 very small billiards. Note that different instances are

	ObjSDF++	Ours
Room0	34	41
Room1	22	24
Room2	24	30
Office0	21	23
Office1	13	20
Office2	28	38
Office3	29	38
Office4	21	28

Table S4. The number of decomposed reconstructed foreground objects on different scenes from Replica following the ObjSDF++ splits (100 images).

labeled with different colors. To our knowledge, no existing works can achieve this level of separation without relying on exhaustive human annotations, such as ObjSDF and ObjSDF++. Even when provided with dense annotations across frames, decomposing all small objects remains a challenging task when relying solely on implicit reconstruction in existing methods as illustrated in Fig. 7 in the main paper.

We are aware that there are some existing methods for panoptic segmentation, such as Mask2Former [5], which can generate semantic and instance labels for each frame. However, when applied to such a challenging scenario, they fail to segment these small objects, as seen in Fig. S10. This suggests that they cannot be readily used even for segmenting a single frame, let alone ensuring view consistency across multiple frames for preparing the data required by existing methods such as ObjSDF, ObjSDF++, and vMAP.

S10.5. Number of Selected Objects

Table S4 presents the number of extracted objects on the Replica dataset. Although ObjSDF++ employs ground-truth instance labels for scene decomposition, it fails to extract certain objects, as illustrated in Fig. 7 of the main paper. Consequently, ObjSDF++ decomposes fewer objects with many objects missing compared to our proposed method. This highlights that achieving complete decomposition of 3D objects is challenging when solely relying on implicit representations, whereas our combined implicit and explicit design attains higher performance. Additionally, our method attains such decomposition qualities with merely one or two human clicks per object, rather than relying on dense masks across multiple views for a single object, as required in ObjSDF and ObjSDF++.

S11. Ablation Studies

Component-wise Study We conduct ablation studies on the Replica dataset to evaluate the effectiveness of our designed modules, as this dataset provides instance-level ground-

Method	Decomposed Reconstruction	
	Chamfer- $\mathcal{L}_1 \downarrow$	F-score \uparrow
Full Model	2.59	87.35
w/o Foreground and Background Decomposition	3.10	84.75
w/o Region Growing	6.98	54.01

Table S5. Ablation study assessing the influence of different components to the decomposed reconstruction results on Replica room0.

truth geometry. We examine the influence of foreground and background decomposition and the region-growing algorithm on object selection. Table S5 summarizes the results of our ablation study. “w/o Foreground and Background Reconstruction” refers to selecting objects on the whole mesh, while “w/o Region Growing” indicates the use of a simple cosine similarity to segment objects instead of our designed algorithm. Foreground and background decomposition methods enhance selection quality, as the foreground mesh naturally prevents the region-growing method from selecting the background mesh. The carefully designed region-growing method achieves an overall improvement of **33.34** in F-score and **4.39** in Chamfer- \mathcal{L}_1 . These experiments demonstrate the effectiveness of our proposed method.

S12. Limitations

While our method is capable of decomposing scenes with minimal human interaction, it still faces some limitations in handling occluded foreground areas. For instance, our approach cannot complete the occluded areas of foreground objects due to the absence of training supervision. In the future, we plan to explore the integration of generative methods to complete such invisible 3D objects and obtain high-quality object meshes even in the presence of occlusions.

References

- [1] WANG Bing, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. In *The Eleventh International Conference on Learning Representations*, 2022. 3
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3
- [3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. *arXiv preprint arXiv:2304.12308*, 2023. 3, 7
- [4] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023. 2
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 3, 11
- [6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 7, 8
- [7] Ainaz Eftekhari, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 9
- [8] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan. Interactive segmentation of radiance fields. *arXiv preprint arXiv:2212.13545*, 2022. 3, 6, 7
- [9] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 9
- [10] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022. 5, 9, 10
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3
- [13] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 1
- [14] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1
- [15] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 4
- [16] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 3
- [17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3, 10
- [18] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 3, 7

- [19] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J Davison. vmap: Vectorised object mapping for neural field slam. *arXiv preprint arXiv:2302.01838*, 2023. [2](#), [3](#), [7](#)
- [20] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J. Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12871–12881, 2022. [3](#)
- [21] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. [3](#)
- [22] Zizhang Li, Xiaoyang Lyu, Yuanyuan Ding, Mengmeng Wang, Yiyi Liao, and Yong Liu. Rico: Regularizing the unobservable for indoor compositional reconstruction. In *ICCV*, 2023. [2](#), [3](#)
- [23] Jiahui Liu, Chirui Chang, Jianhui Liu, Xiaoyang Wu, Lan Ma, and Xiaojuan Qi. Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9372–9381, 2023. [3](#)
- [24] Xiaoyang Lyu, Peng Dai, Zizhang Li, Dongyu Yan, Yi Lin, Yifan Peng, and Xiaojuan Qi. Learning a room with the occ-sdf hybrid: Signed distance function mingled with occupancy aids scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8940–8950, 2023. [1](#)
- [25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. [3](#), [9](#)
- [26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. [9](#)
- [27] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. [1](#), [6](#), [10](#)
- [28] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. [3](#)
- [29] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. [1](#)
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [3](#)
- [31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [1](#)
- [32] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016. [1](#), [10](#)
- [33] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. *arXiv preprint arXiv:2212.09802*, 2022. [3](#), [7](#)
- [34] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Muegler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. [7](#), [8](#)
- [35] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022. [3](#), [7](#)
- [36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. [4](#)
- [37] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. [3](#)
- [38] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. *arXiv preprint arXiv:2206.13597*, 2022. [10](#)
- [39] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. [1](#), [10](#)
- [40] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *Computer Vision-ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 197–213. Springer, 2022. [2](#), [3](#)
- [41] Qianyi Wu, Kaisiyuan Wang, Kejie Li, Jianmin Zheng, and Jianfei Cai. Objectsdf++: Improved object-compositional neural implicit surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21764–21774, 2023. [2](#), [3](#), [5](#), [7](#), [9](#)
- [42] Yuanbo Xiangli, Lining Xu, Xingang Pan, Nanxuan Zhao, Bo Dai, and Dahua Lin. Assetfield: Assets mining and re-

- configuration in ground feature plane representation. *arXiv preprint arXiv:2303.13953*, 2023. 7
- [43] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1, 4, 10
- [44] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3
- [45] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 3, 5, 9, 10
- [46] A. Yuille and J. M. Coughlan. Manhattan world: Compass direction from a single image by bayesian inference. In *Computer Vision, IEEE International Conference on Computer Vision*, page 941, Los Alamitos, CA, USA, 1999. IEEE Computer Society. 5
- [47] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 3
- [48] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 7, 8