3D Scene Change Modeling with Consistent Multi-View Aggregation

Anonymous 3DV submission

Paper ID 332

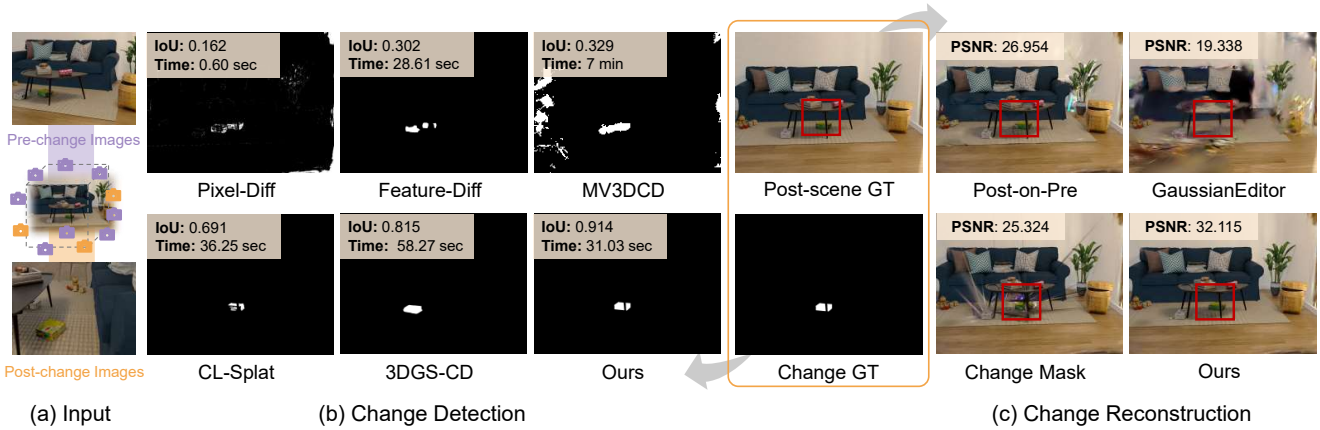

Figure 1. We propose **SCAR-3D**, a 3D scene change modeling framework that detects changes from dense-view pre-change images and sparse-view post-change images, while seamlessly reconstructing the post-change scene. **SCAR-3D** significantly outperforms existing 3D change detection methods in change mask accuracy and computational efficiency, and delivers high-quality post-change reconstructions.

Abstract

Change detection plays a vital role in scene monitoring, exploration, and continual reconstruction. Existing 3D change detection methods often exhibit spatial inconsistency in the detected changes and fail to explicitly separate pre- and post-change states. To address these limitations, we propose **SCAR-3D**, *a novel 3D scene change detection framework that identifies object-level changes from a dense-view pre-change image sequence and sparse-view post-change images. Our approach consists of a signed-distance–based 2D differencing module followed by multi-view aggregation with voting and pruning; the aggregation strategy leverages the consistent nature of 3DGS and also robustly separates pre- and post-change states. Based on the detected change regions, we further develop a continual scene reconstruction strategy that selectively updates dynamic regions while preserving the unchanged areas. We also contribute* **CCS3D**, *a challenging synthetic dataset that allows flexible combinations of 3D change types to support controlled evaluations. Extensive experiments demonstrate that our method achieves both high accuracy and efficiency, outperforming existing methods.*

1. Introduction

3D reconstruction [24, 33, 37, 42, 46] is a fundamental task in computer vision, playing a crucial role in visual perception, embodied artificial intelligence (EAI), environment monitoring, and AR/VR. Real-world environments are inherently dynamic, where objects may appear, disappear, or translate and rotate over time. Much like a Sherlockian observer piecing together a scene from the smallest clues, a robust 3D reconstruction system must detect and interpret subtle environmental changes from the sparse, new observations through the lens of its 3D representation. Thus, reliable 3D change detection is essential to maintain an up-to-date and accurate representation of the evolving scenes.

Change detection aims to identify objects in a scene that have changed between two time points, given multi-view images captured before and after the change. Previously, 2D change detection has been extensively studied, particularly in remote sensing applications such as monitoring land use changes, including the construction of buildings or roads [27]. However, these methods face significant limitations when applied directly to 3D scenes. First, most 2D approaches rely on supervised learning with annotated datasets, which are costly to create and often lack generaliz-

ability across different environments. Besides, these methods often struggle to maintain consistency across multiple views due to random noise and visual ambiguities, limiting their effectiveness in identifying coherent 3D changes.

Recent studies introduce 3D representations into change detection, with 3D Gaussian Splatting (3DGS) [24] emerging as a particularly prominent approach. 3DGS enables efficient rendering of pre-change scenes from novel viewpoints in the post-change, and its explicit and editable structure facilitates the seamless identification and modification of the changed regions. Building on this, methods [1, 17] encode change indicators directly into Gaussian primitives, yielding a unified representation of altered regions. However, such Gaussian-level representations lack holistic object-awareness, often producing fragmented change masks and view-dependent inconsistencies when representing the same object. To mitigate these issues, 3DGS-CD [28] proposes identifying pre-change object masks using segmentation confidence, followed by learning pose transformations between pre- and post-change states. However, accurately matching masks of the same object consistently across multi-views remains challenging, leading to notable performance degradation under diverse change types such as translation, insertion, and removal.

Our method, Spotting Changes and Reconstruction in 3D Scenes (**SCAR-3D**), is a multi-view voting-and-validation-based framework for efficient and consistent change detection in complicated and large-scale 3D scenes. Given two image sets captured from arbitrary viewpoints before and after scene changes, we first register their camera poses within a unified coordinate system. We then identify the feature-level difference between the pre-change and post-change observations by computing signed distance metrics. Utilizing a voting-based approach, we aggregate 2D differences from multiple perspectives, and suppress noise and ensure geometric coherence via multi-view voting and pruning operations. The pruning strategy also robustly separates the pre-change and post-change difference. Finally, we leverage EfficientSAM's segmentation capability to validate the 3D differences and extract high-confidence change masks. By integrating these masks into the 3D reconstruction pipeline, our method enables locally enhanced reconstruction in regions where changes have occurred, while preserving the integrity of unchanged areas.

We introduce a new synthetic dataset tailored for 3D scene change detection, featuring complex and diverse indoor environments beyond simple tabletop settings. The dataset is fully editable, allowing flexible combinations of change types to support controlled evaluations. To assess the effectiveness of our method, we conduct experiments on both real-world datasets and synthetic datasets. Compared to existing methods, our approach produces more accurate and view-consistent change masks with higher efficiency.

In summary, our key contributions are as follows.

- We propose a novel 3D scene change detection framework that leverages a 3D difference map and a multi-view consistency validation mechanism to accurately and efficiently identify object-level changes from two sequences captured under arbitrary viewpoints.
- We construct a high-quality synthetic dataset, **CCS3D**, comprising editable indoor scenes for controlled evaluation of various 3D change types in complex environments.
- Extensive experiments demonstrate that our method outperforms previous approaches in terms of detection accuracy, change mask quality, and computational efficiency.

2. Related Work

2.1. Change Detection

Change detection involves identifying regions or objects that exhibit differences by comparing images taken before and after the changes occur. 2D change detection from paired images has been a long-studied problem, with traditional methods such as CVA [4], PCA [7], image regression [29], and deep learning approaches [2, 13, 15, 20]. 3D change detection has traditionally relied on aligned street-view image pairs [36]. With the emergence of NeRF [33] and 3DGS [24], it can now operate on well-reconstructed scenes. For instance, Huang et al. [21] and Martinson et al. [32] train separate NeRFs on pre- and post-change images to detect changes from aligned views. Lu et al. [28] aggregates 2D change masks into a 3D point cloud to learn pose changes, while Galappaththige et al. [17] embeds change channels in 3DGS. Our method introduces an effective voting strategy to initialize a 3D difference map on 3DGS, validated by multi-view checks and segmentation confidence, enabling fast and accurate change detection from arbitrary viewpoints.

2.2. Continual Scene Reconstruction

Continual 3D reconstruction aims to model a continuously updated 3D scene or its static background from an image sequence taken in dynamic environments [11]. However, directly training a scene representation over the sequence causes catastrophic forgetting [12, 26] and degradation [41]. To mitigate this, Li et al. [26] and Cai et al. [5] introduce keyframe database for historical image replaying. Another key topic for continual 3D reconstruction is to identify the transient regions to be excluded during model update. Traditional methods rely on depth residuals [35] and pixel difference [16]. Learning based methods include [5, 25, 26] masking the transient objects with a learned classifier to maintain reconstruction consistency. Others [1, 25] exploits off-the-shelf vision model [6, 44] to identify the dynamic region. These methods highlight that effective change detection is essential for maintaining accurate and up-to-date 3D

scene reconstructions over time, motivating our approach to integrate change detection with continual reconstruction.

2.3. 3D Editing

3D editing refers to modifying specific parts of a reconstructed scene. Traditional 3D editing relies on human-operated tools such as Maya and Blender. For neural implicit representations such as NeRF and 3DGS, existing approaches primarily focus on text-driven [14, 19, 34, 39, 43, 45, 48] and image-based [3, 22, 39, 48] 3D editing. While existing methods provide stable edits and user-friendly interaction, they lack precise object insertion capabilities and depend on manual initiation. An alternative paradigm for 3D editing involves segmenting all objects in the scene, followed by selective editing of the targeted objects [8, 18, 19, 23, 47]. However, when only a few objects in a cluttered scene require editing, this approach leads to significant computational overhead. Our method leverages scene change detection to automatically trigger precise 3D edits, enabling efficient and targeted modifications by localizing updates to the detected change regions.

3. Method

An overview of our method is shown in Fig. 2. Given pre-change and post-change image sets of a scene, we aim to detect object-level 3D changes. We first estimate camera poses and render the paired pre-post images (Sec. 3.2). We then compute signed-distance maps for coarse 2D differences (Sec. 3.3), which are aggregated into 3D differences through multi-view voting and validation (Sec. 3.4). The resulting change masks then guide localized 3D updates, enabling stable and accurate reconstruction (Sec. 3.5).

3.1. Problem Setup

The input consists of two image sets: $\mathcal{I}_{pre} = \{I_i \mid i = 1, \ldots, n_{pre}\}$ captured from the pre-change scene under n_{pre} arbitrary viewpoints, and $\mathcal{I}_{post} = \{I'_i \mid i = 1, \ldots, n_{post}\}$ from the post-change scene under n_{post} viewpoints. We emphasize that $\mathcal{I}_{pre}$ represents a densely sampled set of views, whereas $\mathcal{I}_{post}$ corresponds to a sparsely sampled one. Our goal is to generate a set of change masks $\mathcal{C} = \{C_i \mid i = 1, \ldots, n_{test}\}$ under specified target viewpoints, and reconstruct the 3D scene $\mathcal{G}_{post}$ in 3DGS.

3.2. Image Registration

For $\mathcal{I}_{pre}$ and $\mathcal{I}_{post}$, we first leverage the structure-from-motion (SFM) algorithm [37], *e.g.*, COLMAP [38], to simultaneously estimate their camera poses $\mathcal{P}_{pre}$ and $\mathcal{P}_{post}$. Assuming that the majority of scene features remain unchanged, we jointly register both image sets in a single SfM process to ensure that all estimated poses lie within a unified coordinate system. Additional implementation details are provided in *supplementary*.

We then train a 3DGS model using the pre-change image set $\mathcal{I}_{pre}$ to obtain a pre-change 3DGS $\mathcal{G}_{pre}$. We render $\mathcal{G}_{pre}$ from the post-change camera poses $\mathcal{P}_{post}$, producing $\mathcal{I}_{ren}$, where each rendered image in $\mathcal{I}_{ren}$ is paired with its corresponding real image in $\mathcal{I}_{post}$.

3.3. 2D Difference Generation

Feature Extraction We utilize EfficientSAM [44] to extract image features f from given image I:

$$f = \mathcal{F}(I), \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the image encoder of EfficientSAM and $f \in R^{h \times w \times d}$. We bilinearly upsample EfficientSAM's raw feature maps to the image resolution while keeping the embedding dimension d.

Signed Distance-Based Change Localization To capture the directionality of changes, we adopt a signed distance formulation in the feature space. Specifically, for each feature map pair (f_i, f'_i), obtained from pairs of rendered pre-change image and post-change image via Eq. (1), we apply Principal Component Analysis (PCA) [31] to determine the dominant direction of variation. All pixel-wise feature vectors from f_i and f'_i are collected, and the first principal component vector v is extracted as the direction of maximum variance. For each pixel p, the signed distance between pre- and post-change features is computed by projecting the feature difference onto v:

$$D_i^p = \frac{(f_i^p - f_i'^p) \cdot v}{\|v\|}. \tag{2}$$

Since the signed distance D_i^p separates foreground and background [10, 30], we threshold it to obtain two directional binary change masks:

$$\mathcal{M}_{i,1} := \mathbf{1}\{\, D_i^p > \epsilon_1 \,\}, \tag{3}$$
$$\mathcal{M}_{i,2} := \mathbf{1}\{\, D_i^p < \epsilon_2 \,\}, \tag{4}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, and $\epsilon_1 \geq 0 \geq \epsilon_2$ are thresholds.

3.4. 3D Difference Aggregation

Although the 2D difference masks $\mathcal{M}_{i,1}$ and $\mathcal{M}_{i,2}$ incorporate image-level semantic features and are more robust than raw pixel-level comparisons, they still suffer from noise and viewpoint-induced ambiguities. To mitigate these issues, we aggregate the 2D differences into a unified 3D representation, leveraging spatial consistency across multiple views.

Multi-view Voting To aggregate the 2D differences into 3D, we initialize a 3D difference representation based on the pre-trained pre-change 3DGS model $\mathcal{G}_{pre}$. Specifically, we embed an additional difference channel into each Gaussian to indicate whether it has changed.

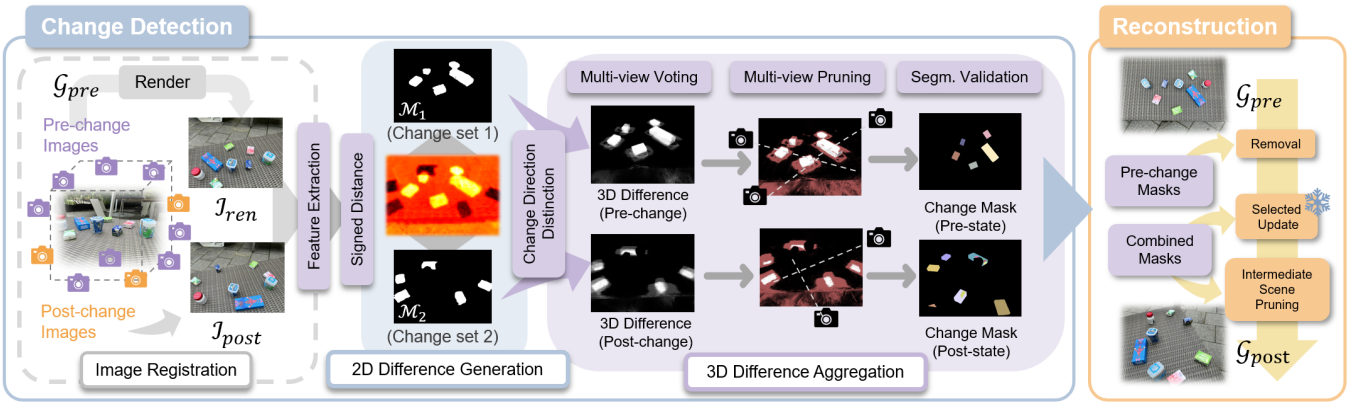

Figure 2. **Overview of SCAR-3D.** We first employ COLMAP for image registration, producing paired pre-change renders and post-change captures. In the *2D Difference Generation* stage, features are extracted and a signed distance metric is applied to separate the change regions into two sets. After that, the *3D Difference Aggregation* stage integrates multi-view differences through voting, pruning, and segmentation validation. Finally, the change masks are applied to the reconstruction process to selectively update the 3D scene.

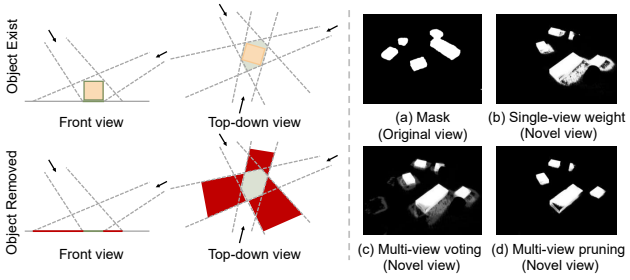

Figure 3. **Multi-view pruning.** Left: When objects exist (top), masks are scattered and few Gaussians are removed; when objects are removed (bottom), background tracing leads to extensive pruning. Right: Novel-view 3D difference visualization for (b) single-view weighting in Eq. (5), (c) multi-view voting in Eq. (7), and (d) multi-view pruning.

For every single view, following the semantic tracing method introduced in GaussianEditor [14], we identify and update the relevant Gaussians for each 2D mask by computing their contribution:

$$w_i = \sum_p o_i(p) \cdot T_i(p) \cdot M(p), \tag{5}$$

where w_i represents the weight of the i-th Gaussian, $o_i(p)$, and $T_i(p)$ denote the Gaussian's opacity, transmittance from pixel p, and $M(p)$ the 2D mask of pixel p. To normalize the weights, we define $\tilde{w}_i = \frac{w_i}{w_{\max}}$, where $w_{\max}$ is the maximum weight across all Gaussians in the current view, ensuring that $\tilde{w}_i \in [0, 1]$.

For the multi-view setting, a straightforward approach is to aggregate weights from all post-change views. Let $S_i^k = \sum_p o_i^k(p)\, T_i^k(p)\, M^k(p)$ denote the aggregated contribution of the i-th Gaussian from the k-th view. A simple normalization by the total number of post-change views is:

$$w_i = \frac{1}{n_{\text{post}}} \sum_{k=1}^{n_{\text{post}}} \frac{S_i^k}{w_{\max}^k} \tag{6}$$

However, this uniform normalization by n_{post} introduces bias against Gaussians visible in fewer views due to occlusions or restricted fields of view, assigning them disproportionately low weights. To address this, we adopt a visibility-aware strategy, normalizing each Gaussian's weight by the actual number of views in which it is observed, n_i^{seen}:

$$w_i = \frac{1}{n_i^{\text{seen}}} \sum_{k=1}^{n_i^{\text{seen}}} \frac{S_i^k}{w_{\max}^k}. \tag{7}$$

To visualize the aggregated 3D difference map, the original view-dependent colors represented by spherical harmonics are replaced during rendering with these computed difference weights, which are mapped from a normalized [0, 1] range to [0, 255] grayscale values, as shown in Fig. 3.

Multi-view Pruning We observe from Fig. 3 that the accumulated weights lack object-level awareness. In some cases, the weights erroneously bleed through foreground objects and are projected onto the background, leading to inconsistent aggregation. To mitigate this issue, we perform a multi-view consistent pruning step after the voting process. Specifically, we remove Gaussians whose centers are projected outside the 2D masks in more than τ out of the total n_{post} views.

Let μ_i denote the center of the i-th Gaussian in 3D space, and p_k^i be its projection, *i.e.*, the pixel location, onto the k-th view. For each view k, we define an indicator $\delta_k^i = 1$ if the 2D mask value $M^k(p_k^i)$ at pixel p_k^i is zero (*i.e.*, outside the mask), and $\delta_k^i = 0$ otherwise. The total number of views in which the Gaussian center lies outside the mask is:

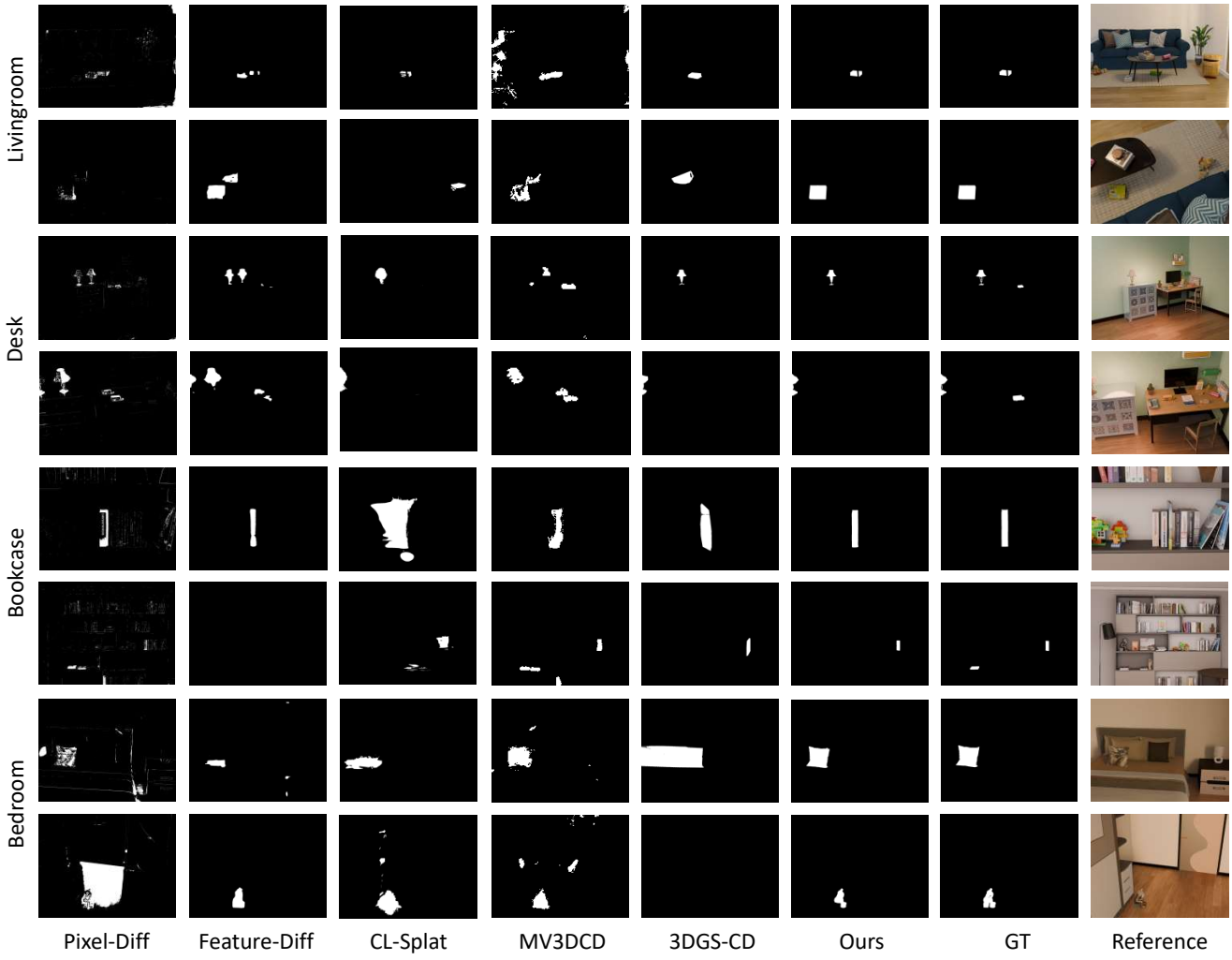

Figure 4. **Qualitative results on the CCS3D dataset.** Each pair of rows corresponds to a single scene captured from different viewpoints. The last column, labeled *Reference*, shows the post-change images from the matched viewpoints.

$$n^i_{\text{out}} = \sum_{k=1}^{n_i^{seen}} \delta^i_k. \tag{8}$$

Finally, the i-th Gaussian is pruned if $n^i_{\text{out}} > \tau \cdot n_i^{seen}$.

Change Direction Distinction As introduced in Sec. 3.3, we use signed distances to separate change regions into two directional categories. With the multi-view pruning strategy in Sec. 3.4, we can further infer whether each change corresponds to a pre-change or post-change object state.

Let N denote the total number of Gaussians selected during the voting process, and let N_p denote the number of Gaussians pruned during the consistency check. We define the retention rate as $R = \frac{N-N_p}{N}$. As illustrated in Fig. 3, when an object exists at the hypothesized location, the projected mask aligns well with the actual object surface, resulting in a higher weight concentration and a higher retention rate R. Conversely, when the object is absent, weights are more likely to scatter and get pruned, leading to a lower R. Applying 3D difference tracing to the pre-change Gaussians ($\mathcal{G}_{pre}$) associates a higher retention rate (R) with the pre-change state and a lower rate with the post-change state.

Segmentation Validation To estimate the reliability of the 3D difference map and refine it into a high-quality 2D mask, we validate the difference against predictions from EfficientSAM. For each view, we compute the Intersection-over-Union (IoU) between the projected 3D difference and all masks generated by EfficientSAM in its segment-everything mode. If the maximum IoU exceeds a predefined threshold τ, the corresponding EfficientSAM mask is considered valid and selected as the final segmentation result.

Table 1. **Quantitative change detection results on the CCS3D dataset.** The best, second-best, and third-best scores are highlighted in red, orange, and yellow, respectively. Our method demonstrates the best overall performance.

	Livingroom		Desk		Bookcase		Bedroom		Average	
Method	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Pixel-Diff	0.273	0.162	0.398	0.254	0.315	0.201	0.286	0.176	0.318	0.198
Feature-Diff	0.420	0.302	0.480	0.323	0.320	0.256	0.705	0.584	0.450	0.343
CL-Splat	0.789	0.657	0.567	0.399	0.294	0.199	0.501	0.341	0.538	0.399
MV3DCD	0.478	0.329	0.291	0.178	0.449	0.295	0.547	0.413	0.441	0.304
3DGS-CD	0.897	0.815	0.525	0.408	0.477	0.353	0.148	0.089	0.512	0.416
Ours	0.955	0.914	0.610	0.477	0.423	0.377	0.909	0.834	0.724	0.650

3.5. Continual Scene Reconstruction

Updating $\mathcal{G}_{pre}$ to $\mathcal{G}_{post}$ presents several challenges. First, given the sparsity of post-change views, it is crucial to prevent degradation of Gaussians that are unseen in the post-change images. A straightforward approach of mixing post-change images with pre-change images can mitigate this issue, but it significantly increases computational cost and introduces view-dependent ambiguities within the changed regions. Alternatively, freezing Gaussians outside the change masks is a common strategy; however, insufficient supervision near the boundaries of the 2D change masks often leads to the accumulation of drifting Gaussians along the edges of changed areas.

To mitigate these problems, we adopt a 2D change mask–based strategy that replaces pre-change objects with their post-change counterparts during reconstruction. Specifically, we first apply the pre-change masks to $\mathcal{G}_{pre}$ to remove the corresponding objects. Then, we use the masked post-change images $\mathcal{I}_{post}$ to locally update $\mathcal{G}_{pre}$, producing an intermediate scene. In this process, Gaussians outside the 2D change masks are frozen, while loss computation and gradient descent are restricted to the masked regions in screen space. Finally, leveraging the voting and pruning method introduced in Sec. 3.4, we replace the change region in $\mathcal{G}_{pre}$ with the corresponding regions from the intermediate scene. This results in the updated post-change scene $\mathcal{G}_{post}$ with fewer drifting artifacts from view-dependent degradation while accurately relocating the changed objects to their new position. The localized update approach further improves reconstruction fidelity within the changed regions, as it may circumvent sub-optimal results.

4. Experiments

4.1. Experiment Setup

Datasets We first introduce a new dataset, Controllable Change in 3D Scenes (**CCS3D**), which comprises four diverse and comprehensive synthetic scenes: Desk, Bookcase, Livingroom, and Bedroom, constructed with Blender [9]. Compared with existing 3D change detection datasets, **CCS3D** is not restricted to tabletop scenarios with simple camera trajectories (*e.g.*, face-forward or fixed 360° rotation). The Bookcase scene features a full-wall, multi-floor bookshelf, where the camera navigates from a distant view to a close-up inspection, sequentially exploring each shelf. The Livingroom and Bedroom scenes offer complete 360° environments containing both large-scale furniture (*e.g.*, chairs, tables) and small tabletop items (*e.g.*, books, pencil cases), designed to support fine-grained change detection. These scenes also incorporate complex, human-like camera navigation patterns, simulating natural walking and exploration. Furthermore, our dataset enables controlled experiments on change detection with varying numbers of objects and change types. In contrast, existing 3D change detection datasets are primarily based on real-world scenes, where such control is difficult to achieve.

We also evaluate our model on the real-world dataset, 3DGS-CD dataset, which focuses on tabletop scenes with complex object changes, including object removal, insertion and movement in cluttered environments.

Baselines and Metrics We compare our change detection method with one 2D-based approach, MTP [40], and three 3D-based methods: 3DGS-CD [28], MV3DCD [17], and the change detection module of CLSplat [1]. In addition, we evaluate two baseline strategies based on pixel difference (Pixel-Diff) and feature difference (Feature-Diff) for comparison. Following the evaluation protocols in C-NeRF [21] and 3DGS-CD [28], we report Precision, Recall, F1-score, and IoU as our quantitative metrics.

4.2. Evaluation Results

CCS3D Results in Tab. 1 show that our method consistently achieves the highest Precision and IoU across all scenes. Intermediate steps like 2D pixel and feature differences yield stable but lower results due to noise from Gaussian splatting artifacts. MV3DCD's Gaussian-wise change representation often leads to fragmented detections lacking object integrity. Furthermore, 3DGS-CD's performance

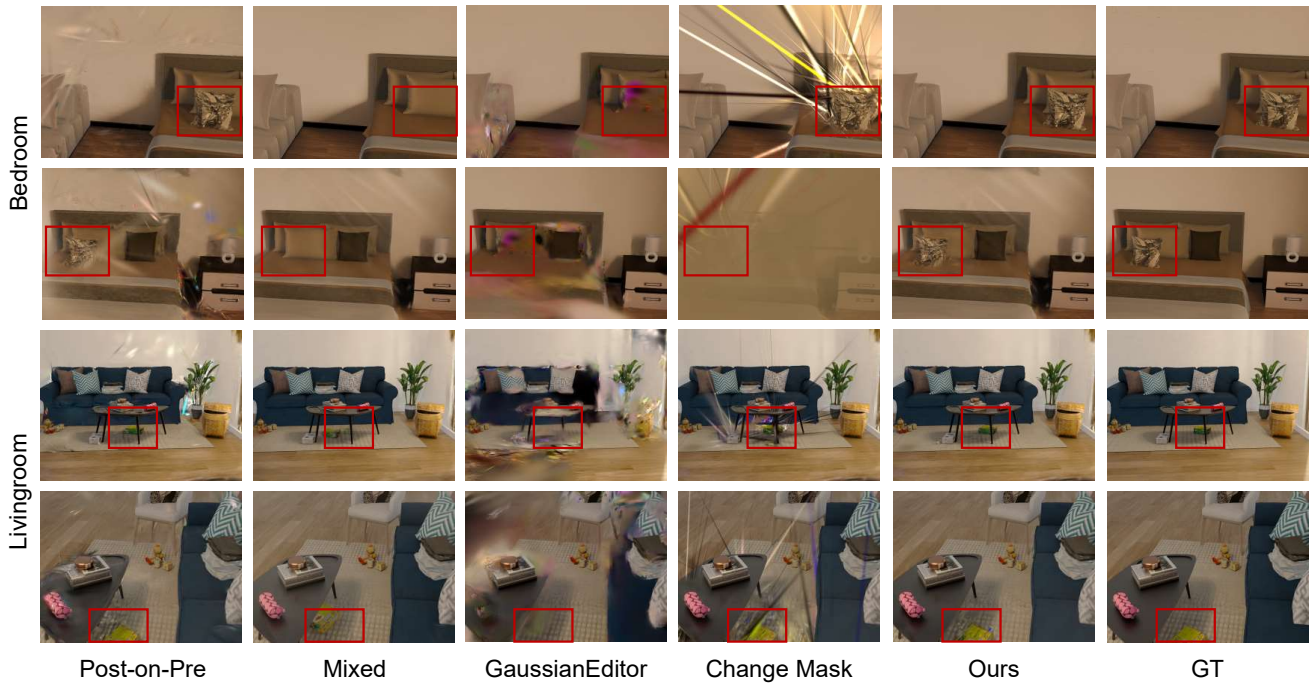

Figure 5. **Qualitative reconstruction results on novel views.** Each pair of rows corresponds to a single scene captured from different novel viewpoints. The changed regions are highlighted with red boxes.

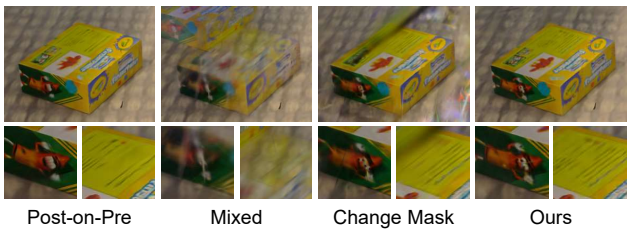

Figure 6. **Reconstruction results on post-change views.**

Table 2. **Controlled Evaluation on the Livingroom scene.** We report F1/IoU across change types and number of changed objects.

	# Obj = 1		# Obj = 2		# Obj = 4	
Change Type	**F1**	**IoU**	**F1**	**IoU**	**F1**	**IoU**
In/Out	0.953	0.911	0.770	0.742	0.764	0.732
Translation	0.985	0.970	0.613	0.498	0.585	0.472
Rotation	0.982	0.964	0.690	0.613	0.383	0.253
Mixed	–	–	0.390	0.307	0.408	0.260

proves highly sensitive to its object matching and pose estimation steps, especially in challenging scenarios. For instance, it fails in the Bedroom scene, where limited viewpoints lead to critical matching errors, and struggles to distinguish visually similar books in the Bookcase scene, causing a significant drop in IoU. In contrast, our approach uses 3D difference aggregation and segmentation validation to effectively suppress such noise and recover complete, accurate change masks.

Controlled Experiments As shown in Tab. 2, we evaluate our method under varying change types and object counts. Results show that performance for simple scenarios (*e.g.*, single-object cases) are strong across all change types. As the number of changed objects increases, performance degrades, especially for Rotation and Translation. Mixed changes are the most challenging overall, yielding low scores and indicating compounded difficulty under heterogeneous transformations. This also validates the merits of our dataset in examining the robustness of algorithms in more challenging cases.

3DGS-CD Dataset Results on the 3DGS-CD dataset (Tab. 3) show our method achieves significantly higher F1 and IoU scores than prior approaches. While MTP attains high precision, its lower recall limits performance. The 3DGS-CD method localizes changes accurately but suffers from instability due to reliance on 2D detection and object association steps, especially in cluttered scenes like Mustard and Bench. In contrast, our approach delivers more consistent results by leveraging 3D difference voting and validation to reduce errors from 2D difference detection.

Continual Scene Reconstruction We further assess the image quality of our continual reconstruction approach. As

Table 3. **Quantitative change detection results on the 3DGS-CD dataset.** Our method consistently achieves the best performance in terms of F1 score and IoU.

Scene	Method	Precision	Recall	F1	IoU
Mustard	MTP	0.949	0.231	0.371	0.228
	3DGS-CD	0.315	0.104	0.155	0.085
	Ours	0.794	0.573	0.583	0.507
Desk	MTP	0.957	0.344	0.506	0.339
	3DGS-CD	0.967	0.961	0.964	0.930
	Ours	0.995	0.968	0.981	0.964
Swap	MTP	0.942	0.246	0.390	0.243
	3DGS-CD	0.983	0.989	0.986	0.973
	Ours	0.998	0.992	0.995	0.990
Bench	MTP	0.902	0.887	0.895	0.809
	3DGS-CD	0.851	0.796	0.817	0.691
	Ours	0.995	0.867	0.915	0.863
Sill	MTP	0.483	0.308	0.376	0.232
	3DGS-CD	0.981	0.974	0.977	0.956
	Ours	0.998	0.972	0.982	0.970
Average	MTP	0.846	0.403	0.508	0.370
	3DGS-CD	0.819	0.765	0.780	0.727
	Ours	0.956	0.874	0.891	0.859

Table 4. **Quantitative reconstruction results.** Scenes are rendered from both post-change and novel viewpoints.

Method	PSNR↑	LPIPS↓	SSIM↑
Post-change views			
Post-on-Pre	34.496	0.07	0.945
Mixed	26.665	0.231	0.855
GaussianEditor	20.664	0.446	0.722
Change Mask	25.708	0.262	0.826
Ours	33.095	0.083	0.935
Novel views			
Post-on-Pre	26.954	0.166	0.888
Mixed	30.345	0.095	0.936
GaussianEditor	19.338	0.366	0.765
Change Mask	25.324	0.202	0.874
Ours	32.115	0.087	0.938

shown in Tab. 4, our method attains the best rendering quality on novel views and the second-highest on post-change views. Qualitative comparisons are provided in Fig. 5 and Fig. 6. Directly fine-tuning the pre-change Gaussians $\mathcal{G}_{pre}$ using post-change images $\mathcal{I}_{post}$ achieves the best performance on post-change views but suffers from severe view-dependent degradation on novel views. The same issue is also observed in GaussianEditor. Training with a mixed set of pre-change and post-change images yields balanced but suboptimal results for both novel and post-change views. As illustrated in Fig. 5, it fails to accurately relocate the green-boxed object to its new position.

4.3. Ablation Study

Table 5. **Ablation study on difference maps.** We evaluate pixel-, feature-, and SSIM-based differences, as well as their combinations (pixel+feature as in [16], and SSIM+feature as in [17]). These are compared against the 3D difference and our full method. For 2D methods, input images are pre-aligned to the test views.

Method	Precision	Recall	F1	IoU
2D difference				
Pixel-Diff	0.303	0.458	0.318	0.198
Feature-Diff	0.436	0.538	0.450	0.343
SSIM-Diff	0.316	0.637	0.394	0.256
Pixel+Feature	0.519	0.241	0.308	0.199
SSIM+Feature	0.481	0.355	0.388	0.271
3D difference				
3D-Diff	0.522	0.617	0.487	0.366
Ours (Full)	0.836	0.680	0.724	0.650

Ablation on Difference Module Tab. 5 presents the results of our ablation study, averaged over the **CCS3D** dataset. The first five rows correspond to vanilla 2D change detection methods commonly adopted in prior works [1, 16, 17, 28], indicating that feature difference is more robust than pixel- and SSIM-based difference, while combining them via multiplication does not yield significant improvement. Compared to 2D difference methods, the 3D difference approaches consistently achieve better performance, demonstrating the effectiveness of multi-view voting in suppressing noise. Our full method, which integrates 2D difference, 3D difference, and segmentation validation, attains the highest accuracy overall.

5. Conclusion

We propose **SCAR-3D**, a multi-view voting-and-validation-based 3D change detection and reconstruction framework designed for complex and large-scale 3D scenes. By integrating 2D signed-distance differencing with multi-view aggregation and pruning, our method generates accurate and consistent change masks, enabling localized updates to dynamic 3D scenes. Extensive experiments, together with the proposed dataset **CCS3D**, demonstrate that **SCAR-3D** outperforms state-of-the-art methods in both accuracy and efficiency, yielding high-fidelity reconstructions with fewer artifacts. Future work may address current limitations, including handling a greater number of changed objects within a scene and mitigating the effects of varying lighting conditions and shadows. Furthermore, exploring how to effectively model non-rigid deformations or significant topological changes within the 3D Gaussian framework remains a key challenge.

References

[1] Jan Ackermann, Jonas Kulhanek, Shengqu Cai, Xu Haofei, Marc Pollefeys, Gordon Wetzstein, Leonidas Guibas, and Songyou Peng. Cl-splats: Continual learning gaussian splatting with local optimization. *ICCV*, 2025. 2, 6, 8

[2] Wele Gedara Chaminda Bandara and Vishal M Patel. A transformer-based siamese network for change detection. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210. IEEE, 2022. 2

[3] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 3

[4] Francesca Bovolo and Lorenzo Bruzzone. A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain. *IEEE Transactions on Geoscience and Remote Sensing*, 45(1):218–236, 2006. 2

[5] Zhipeng Cai and Matthias Müller. Clnerf: Continual learning meets nerf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23185–23194, 2023. 2

[6] 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 International Conference on Computer Vision (ICCV)*, 2021. 2

[7] Turgay Celik. Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE geoscience and remote sensing letters*, 6(4): 772–776, 2009. 2

[8] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1971–1979, 2025. 3

[9] Blender Online Community. Blender - a 3d modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 6

[10] Jisheng Dang, Huicheng Zheng, Xiaohao Xu, Longguang Wang, and Yulan Guo. Beyond appearance: Multi-frame spatio-temporal context memory networks for efficient and robust video object segmentation. *IEEE Transactions on Image Processing*, 2024. 3

[11] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021. 2

[12] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. 2

[13] Sijun Dong, Libo Wang, Bo Du, and Xiaoliang Meng. Changeclip: Remote sensing change detection with multimodal vision-language representation learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208:53–69, 2024. 2

[14] Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024. 3, 4

[15] Sheng Fang, Kaiyu Li, and Zhe Li. Changer: Feature interaction is what you need for change detection. *arXiv preprint arXiv:2209.08290*, 2022. 2

[16] Bin Fu, Jialin Li, Bin Zhang, Ruiping Wang, and Xilin Chen. Gs-lts: 3d gaussian splatting-based adaptive modeling for long-term service robots, 2025. 2, 8

[17] Chamuditha Jayanga Galappaththige, Jason Lai, Lloyd Windrim, Donald Dansereau, Niko Sunderhauf, and Dimity Miller. Multi-view pose-agnostic change localization with zero labels. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 11600–11610, 2025. 2, 6, 8

[18] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. *arXiv preprint arXiv:2403.18118*, 2024. 3

[19] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting, 2024. 3

[20] Zipeng Qi Hao Chen and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2021. 2

[21] Rui Huang, Binbin Jiang, Qingyi Zhao, William Wang, Yuxiang Zhang, and Qing Guo. C-nerf: Representing scene changes as directional consistency difference-based nerf. *arXiv preprint arXiv:2312.02751*, 2023. 2, 6

[22] Vishnu Jaganathan, Hannah Hanyun Huang, Muhammad Zubair Irshad, Varun Jampani, Amit Raj, and Zsolt Kira. Ice-g: Image conditional editing of 3d gaussian splats. *arXiv preprint arXiv:2406.08488*, 2024. 3

[23] Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. Editable free-viewpoint video using a layered neural representation. In *ACM SIGGRAPH*, 2021. 3

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2

[25] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *NeurIPS*, 2024. 2

[26] Baicheng Li, Zike Yan, Dong Wu, Hanqing Jiang, and Hongbin Zha. Learn to memorize and to forget: A continual learning perspective of dynamic slam. In *European Conference on Computer Vision*, pages 41–57. Springer, 2024. 2

[27] Kaiyu Li, Xiangyong Cao, Yupeng Deng, Chao Pang, Zepeng Xin, Deyu Meng, and Zhi Wang. Dynamicearth: How far are we from open-vocabulary change detection? *arXiv preprint arXiv:2501.12931*, 2025. 1

[28] Ziqi Lu, Jianbo Ye, and John Leonard. 3dgs-cd: 3d gaussian splatting-based change detection for physical object rearrangement. *IEEE Robotics and Automation Letters*, 2025. 2, 6, 8

[29] Luigi T Luppino, Filippo M Bianchi, Gabriele Moser, and Stian N Anfinsen. Unsupervised image regression for heterogeneous change detection. *arXiv preprint arXiv:1909.05948*, 2019. 2

[30] Yunqiu Lv, Jing Zhang, Nick Barnes, and Yuchao Dai. Weakly-supervised contrastive learning for unsupervised object discovery. *IEEE Transactions on Image Processing*, 33: 2689–2702, 2024. 3

[31] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19 (3):303–342, 1993. 3

[32] Eric Martinson and Paula Lauren. Meaningful change detection in indoor environments using clip models and nerf-based image synthesis. In *2024 21st International Conference on Ubiquitous Robots (UR)*, pages 603–610. IEEE, 2024. 2

[33] 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. In *ECCV*, 2020. 1, 2

[34] Francesco Palandra, Andrea Sanchietti, Daniele Baieri, and Emanuele Rodola. Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting. *arXiv preprint arXiv:2403.05154*, 2024. 3

[35] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019. 2

[36] Ken Sakurada, Mikiya Shibuya, and Wang Weimin. Weakly supervised silhouette-based semantic scene change detection. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 2

[37] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[38] 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. 3

[39] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *arXiv preprint arXiv:2112.05139*, 2021. 3

[40] Di Wang, Jing Zhang, Minqiang Xu, Lin Liu, Dongsheng Wang, Erzhong Gao, Chengxi Han, Haonan Guo, Bo Du, Dacheng Tao, et al. Mtp: Advancing remote sensing foundation model via multi-task pretraining. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. 6

[41] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9065–9076, 2023. 2

[42] 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

[43] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision*, pages 55–71. Springer, 2024. 3

[44] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv:2312.00863*, 2023. 2, 3

[45] Ziyang Yan, Lei Li, Yihua Shao, Siyu Chen, Zongkai Wu, Jenq-Neng Hwang, Hao Zhao, and Fabio Remondino. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. *arXiv preprint arXiv:2412.01583*, 2024. 3

[46] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1

[47] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 3

[48] Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *ACM Transactions on Graphics (TOG)*, 43(4):1–12, 2024. 3