International Journal of Computer Vision (2026) 134:394
https://doi.org/10.1007/s11263-026-02941-w

# Mastering Regional 3DGS: Locating, Initializing, and Editing with Diverse 2D Priors

Lanqing Guo[1] · Yufei Wang[2] · Hezhen Hu[1] · Yan Zheng[1] · Yeying Jin[3] · Siyu Huang[4] · Zhangyang Wang[1]

Received: 7 July 2025 / Accepted: 26 June 2026


## Abstract

Many 3D scene editing tasks focus on modifying local regions rather than the entire scene, except for some global applications like style transfer, and in the context of 3D Gaussian Splatting (3DGS), where scenes are represented by a series of Gaussians, this structure allows for precise regional edits, offering enhanced control over specific areas of the scene; however, the challenge lies in the fact that 3D semantic parsing often underperforms compared to its 2D counterpart, making targeted manipulations within 3D spaces more difficult and limiting the fidelity of edits, which we address by leveraging 2D diffusion editing to accurately identify modification regions in each view, followed by inverse rendering for 3D localization, then refining the frontal view and initializing a coarse 3DGS with consistent views and approximate shapes derived from depth maps predicted by a 2D foundation model, thereby supporting an iterative, view-consistent editing process that gradually enhances structural details and textures to ensure coherence across perspectives. Experiments demonstrate that our method achieves state-of-the-art performance while delivering up to a 4× speedup, providing a more efficient and effective approach to 3D scene local editing.

**Keywords** 3D Gaussian Splatting · Image generation · 3D editing

## 1 Introduction

In many 3D scene editing applications (Zuo et al., 2025), the focus is increasingly on modifying specific aspects, such as avatar attribute editing (Ho et al., 2023; Zhu et al., 2026), adding or removing objects (Ruiz et al., 2025); Anciukevičius et al., (2023); Weber et al. (2024), and similar localized adjustments. Unlike global modifications like style transfer (Chen et al., 2024b; Kawar et al., 2023), which aim to achieve a consistent look or feel across an entire scene, localized edits allow for targeted changes that enhance specific elements without altering the overall scene context, which are also aligned with various nuanced requirements of tasks like object manipulation, detailed feature enhancement, or region-specific alterations. These targeted modifications are particularly valuable in applications requiring precision, where fine-grained control over specific parts of a scene is essential for achieving high visual fidelity. One emerging 3D modeling approach in this space, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023b), represents scenes as a series of Gaussian points, which allows for accurate regional adjustments and enhanced control over discrete areas of the 3D environment. This representation is highly flexible, enabling

Communicated by Jianfei Cai.

✉ Lanqing Guo
smurfsthen@gmail.com

Yufei Wang
yufei.wang@sparclab.ai

Hezhen Hu
hezhen.hu@austin.utexas.edu

Yan Zheng
yanzheng@utexas.edu

Yeying Jin
jinyeying@u.nus.edu

Siyu Huang
siyuh@clemson.edu

Zhangyang Wang
atlaswang@utexas.edu

[1] The University of Texas at Austin, Austin, USA

[2] SparcAI Inc., Newark, USA

[3] National University of Singapore, Singapore, Singapore

[4] Clemson University, Clemson, USA

Published online: 12 August 2026

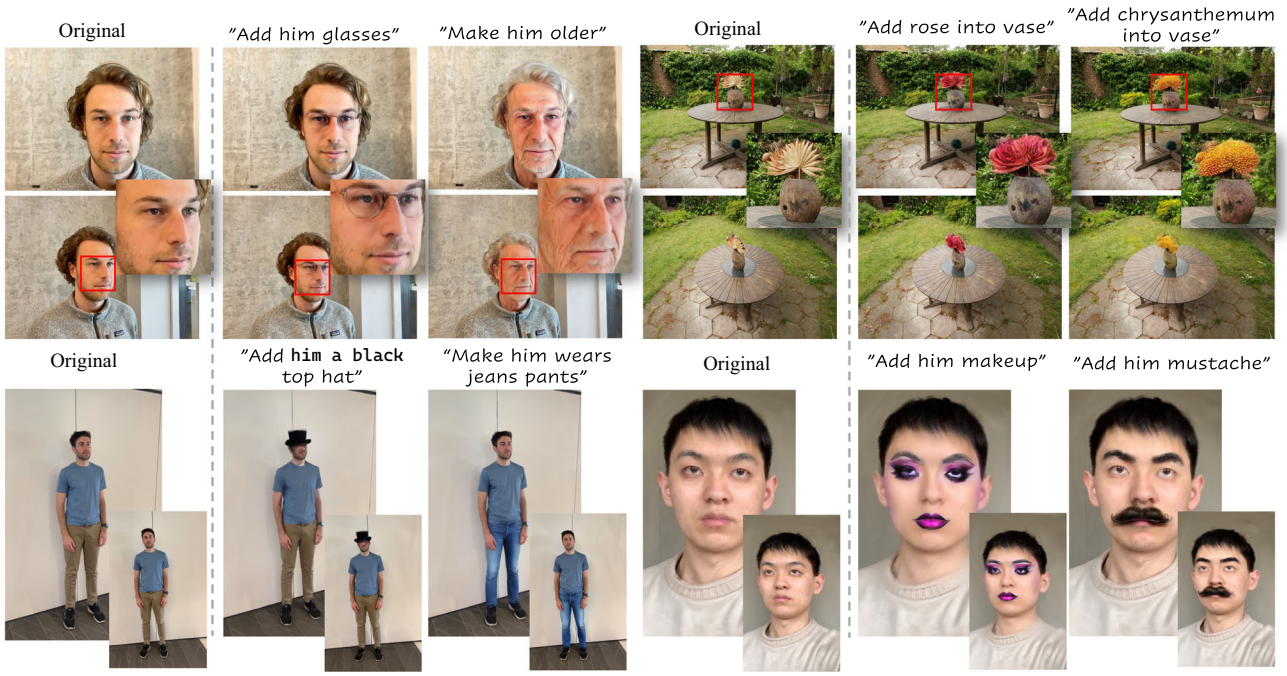


**Fig. 1** Comprehensive results demonstrate our method's effectiveness in diverse localized editing tasks, including modifications to human bodies, faces, and scenes. The leftmost column displays the original view, while the two columns on the right show rendered views of the edited 3DGS generated by our approach using the corresponding text descriptions. Please zoom in for a clearer visualization

it to facilitate localized scene modifications with greater accuracy and precision Fig. 1.

However, despite the advantages offered by 3DGS in terms of localization, there remain considerable challenges in achieving high-quality, consistent edits across all views in a 3D space. Many current methods for 3D editing struggle with two primary issues: time efficiency and detail fidelity. Existing approaches (Chen et al., 2024c; Haque et al., 2023) often rely on time consuming diffusion model editing rendered frames and iterative optimization across multiple views, resulting in extended processing times that hinder real-time application. Additionally, these methods frequently exhibit detail loss, particularly in intricate structures or textures. The underperformance of 3D semantic parsing relative to 2D parsing further complicates the editing process, as it becomes difficult to reliably identify and target specific regions within the 3D space without sacrificing the quality of edits. As a result, achieving high fidelity without compromising edit quality remains difficult, especially in intricate scenes where detail is critical to preserving visual coherence and realism.

To address these issues, we propose a novel approach that combines 2D diffusion editing with a re-rendering process to achieve efficient, high-fidelity localization within 3D environments. Our method begins with 2D diffusion-based region identification, enabling precise targeting of edit regions in each view without the need for time-consuming 3D parsing steps. Once these regions are identified in 2D, we re-render the scene and use Gaussian point expansion to locate the corresponding areas within the 3D space. This approach not only accelerates the editing process but also circumvents the limitations of conventional 3D parsing by leveraging the higher accuracy of 2D parsing models. Following this step, we initialize a coarse 3D Gaussian Splatting representation using consistent views and approximate shapes derived from depth maps predicted by a 2D foundation model. This initialization serves as a strong foundation for an iterative, view-consistent editing process that progressively refines both structural details and textures, ensuring coherence and detail preservation across all perspectives. Experimental results demonstrate that our method achieves state-of-the-art performance, providing substantial improvements in both accuracy and speed.

Our main contributions are concluded as follows:

- We propose a new 3DGS editing framework that separates the process into sequential localizing, initializing, and editing stages, ensuring explicit Gaussian points localization and precise modification.
- Within the located Gaussians, we initialize a coarse 3DGS with approximate shapes based on the depth map predicted by the 2D foundation model. Then we refine the structural details and textures of sequential views by conditional sequential diffusion editing.

- Experimental results demonstrate that our 3DGS local editing method achieves both higher efficiency and better structural details compared to existing methods, achieving up to 4× speedup comparing to 3DGS-based editing.

## 2 Related Work

### 2.1 Text-based Image Editing

Given the limited availability of large-scale datasets specifically designed for training 3D editors from scratch, most existing methods circumvent this challenge by adapting techniques originally developed for 2D image editing. In particular, the field of text-to-image editing–driven by the success of diffusion models–has witnessed rapid progress in recent years (Quan et al., 2024; Wang et al., 2025b; Zheng et al., 2024; Xiao et al., 2024; Wang et al., 2025a, 2024; Zhang et al., 2024). Early methods in this line of work typically apply noise to the source image and then re-generate it based on a new prompt, aiming to achieve the desired edit through conditional denoising. However, such approaches often fall short in preserving data fidelity; the edited results frequently suffer from layout distortions or the loss of critical structural and semantic content. To address these shortcomings, subsequent methods have explored techniques to better preserve the spatial and semantic integrity of the original image. Some approaches attempt to align attention maps during generation (Hertz et al., 2023; Mokady et al., 2023), ensuring that important regions remain stable throughout the editing process. Others modify internal feature representations (Couairon et al., 2023; Tumanyan et al., 2022), effectively steering the generation path closer to the original content. These methods have also inspired instruction-based editing pipelines, such as InstructPix2Pix (Brooks et al., 2023), which leverages large-scale synthetic data conditioned on both images and editing instructions to train a dedicated diffusion model. Building on this paradigm, further efforts have refined InstructPix2Pix via human-annotated supervision to improve edit controllability and consistency. In a different direction, another class of methods directly fine-tunes the diffusion model on the source image itself (Kawar et al., 2023)[?], enabling the model to better internalize the structure and appearance of the specific image. This approach significantly improves content alignment, ensuring that both high-level semantics and fine-grained visual details are faithfully preserved throughout the editing process. Together, these advancements form the foundation for recent progress in generative editing, which is increasingly being adapted and extended to the more complex domain of 3D content manipulation.

### 2.2 Text-driven 3D Editing

Recently, image editing techniques have expanded into 3D scene consistent editing. NeRF (Mildenhall et al., 2020) and 3DGS (Kerbl et al., 2023b) are among the most popular 3D models for neural novel view synthesis: NeRF encodes scene geometry and color implicitly through a Multi-Layer Perceptron (MLP), while 3DGS represents the scene explicitly with Gaussian ellipse point clouds. IN2N (Haque et al., 2023) pioneered NeRF-based 3D editing with text instructions, utilizing 2D image editing tool, *i.e.*, IP2P (Brooks et al., 2023), on rendered views and iteratively updating the 3D scene until convergence. Subsequently, ViCA-NeRF (Dong and Wang, 2023) introduced cross-view blending to improve view consistency, and DreamEditor (Zhuang et al., 2023) incorporated SDS loss and DreamBooth (Ruiz et al., 2023) to address object consistency issues. However, NeRF-based editing faces challenges with slow processing speeds and limited control over specific scene regions. With advances in 3DGS, new 3DGS-based editing approaches have emerged. For example, GaussianEditor (Chen et al., 2024c) leverages 2D IP2P as a prior and introduces semantic tracing to track the editing target throughout training. Later methods, such as DGE (Chen et al., 2024) and VcEdit (Wang et al., 2025b), propose techniques like the episolar constraint and Cross-attention Consistency Modules to enhance view consistency in edits. EditSplat (Lee et al., 2025) introduces a multi-view fusion guidance strategy to enforce view consistency in 2D diffusion models, along with an attention-based pruning mechanism that trims pretrained Gaussians and selectively optimizes regions with high attention scores. Another work, TIP-Editor (Zhuang et al., 2024), presents a flexible 3D editing framework that accommodates both text prompts and reference images. It emphasizes accurate 3D localization through a dedicated localization loss and employs LoRA-based adaptation to integrate reference image features into the editing process. Nonetheless, semantic-level consistency constraints still suffer from low-level structural inconsistencies, limiting the fidelity of edits. Besides, the editing process still face challenge for practical applications. In this paper, we delve into the realm of 3DGS to explore a highly efficient approach for 3D local editing. We propose a coarse-to-fine pipeline that initializes 3DGS using a predicted coarse Gaussian points by predicted depth maps and refines structural details through conditional sequential 2D view editing.

### 2.3 Local Editing

Local editing refers to modifications confined to specific regions within 2D or 3D scenes, encompassing a wide range of tasks such as object manipulation, attribute adjustment (Zhu et al., 2016), spatial transformation (Jaderberg et al., 2015), and inpainting (Lugmayr et al., 2022), among

others (Shuai et al., 2024). For certain explicit local editing tasks, a binary mask indicating the region of interest is often provided as an auxiliary input to guide the model (Isola et al., 2017), ensuring edits are spatially constrained. To further improve precision and avoid unintended alterations, recent works (Meng et al., 2021; Lugmayr et al., 2022) have introduced diffusion-based editing frameworks that effectively control modification strength and localization. Notably, (Brooks et al., 2023) extends this paradigm to support multi-region editing by applying a DDS loss to designated areas of the image, thus enabling localized pixel optimization while preserving surrounding content. In contrast to the substantial progress in 2D, local editing in 3D scenes–particularly at the semantic or attribute level–remains relatively underexplored due to challenges in region localization, multi-view consistency, and the lack of fine-grained 3D supervision. Some early efforts have made progress in this direction: for instance, Xiao et al. (2025) explores object-level insertion and removal operations to coarsely infer editable regions in 3D space, while (Mirzaei et al., 2025) proposes using relevance maps from 2D projections as editing masks, which, however, may lead to view inconsistency and suboptimal control in the 3D domain. In this paper, we propose to extend the concept of localization from 2D to 3D by operating on semantically meaningful regions tied to specific attributes, and we build upon the 3D Gaussian Splatting (3DGS) representation to enable precise positioning and manipulation of Gaussians within the identified regions, supporting coherent and fine-grained local edits in 3D scenes.

## 3 Preliminary

### 3.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) models scenes using anisotropic 3D neural Gaussians for efficient rendering via tile-based rasterization. Each Gaussian $G$ is characterized by its mean $\mu \in \mathbb{R}^3$, covariance $\Sigma$, color $c \in \mathbb{R}^3$, and opacity $\alpha \in \mathbb{R}$. To optimize, the covariance $\Sigma$ is decomposed as $\Sigma = RSS^TR^T$, with scaling $S \in \mathbb{R}^3$ and rotation $R \in \mathbb{R}^{3\times3}$. A Gaussian centered at $\mu$ is given by $G(x) = \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$, where $x$ is the 3D coordinates. The 3DGS $\mathcal{G}$ can be efficiently projected to 2D space, where the color $C(x')$ depth $D(x')$ for each pixel $x'$ can be obtained via point-based volumetric rendering:

$$C(x') = \sum_{i\in N} c_i\sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \tag{1}$$

$$D(x') = \sum_{i\in N} d_i\sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \quad \sigma_i = \alpha_i G(x') \tag{2}$$

where $N$ is the ordered set of Gaussians overlapping the pixel $x'$, and $d_i$ is the distance between 3D Gaussian center $\mu_i$ and camera center.

### 3.2 Diffusion-based Editing

A diffusion-based image editing method samples the target image $\mathbf{x}'$ from a conditional distribution $p(\cdot|\mathbf{x}, c)$, where the condition $c$ can be a textual prompt for text-based editing and $\mathbf{x}$ is the source image. We adopted IP2P (Brooks et al., 2023) as the 2D prior for view editing, which is built upon Stable Diffusion (SD) (Podell et al., 2023) and utilizes the classifier-free guidance (Ho and Salimans, 2021) during inference. Specifically, the encoder $\mathcal{E}$ first converts image into latent code $\mathbf{z}'_0 = \mathcal{E}(\mathbf{x}_0)$, and the sampling with the guidance of $\mathbf{c}$ is defined as follows

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{z}'_0, \mathbf{c}) = (1+w)\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{z}'_0, \mathbf{c}) - w\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{z}'_0), \tag{3}$$

where $\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{z}'_0, \mathbf{c})$ and $\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{z}'_0)$ are the predicted noise w/ and w/o the condition guidance, and $w$ is the weight of the guidance. The difference to Ho and Salimans (2021) is that IP2P additionally conditioned on the latent of the input image $\mathbf{z}'_0$ to support image editting. During the forward diffusion process, noise is gradually added to the latent code:

$$q(\mathbf{z}'_t|\mathbf{z}'_0) = \mathcal{N}(\mathbf{z}'_t; \sqrt{\bar{\alpha}_t}\mathbf{z}'_0, (1-\bar{\alpha}_t)\mathbf{I}) \ . \tag{4}$$

where $\bar{\alpha}_t$ controls the noise level at timestep $t$.

## 4 Methodology

In this section, we present the overall pipeline of our proposed method. First, we perform 3D localization to identify Gaussians requiring edits based on accurate 2D location information (Sect 4.1). Next, we initialize a coarse 3D Gaussian points within the identified 3D regions using predicted depth (Sect 4.2). Finally, we apply conditional sequential view refinement to enhance texture, color, and structural details (Sect 4.3). The complete framework is illustrated in Fig 2.

### 4.1 From 2D to 3D Localization

The essence of the editing task can be decomposed into locating the modification regions based on user requirements and then adjusting these regions while preserving information from the original image. In the first step, we explicitly localize the target Gaussian points in 3D space, identifying the specific points to be modified.

Inspired by previous work (Mirzaei et al., 2025), given several rendered views $\mathbf{X} = (\mathbf{x}_v)_{v=1}^V$ from the source 3DGS

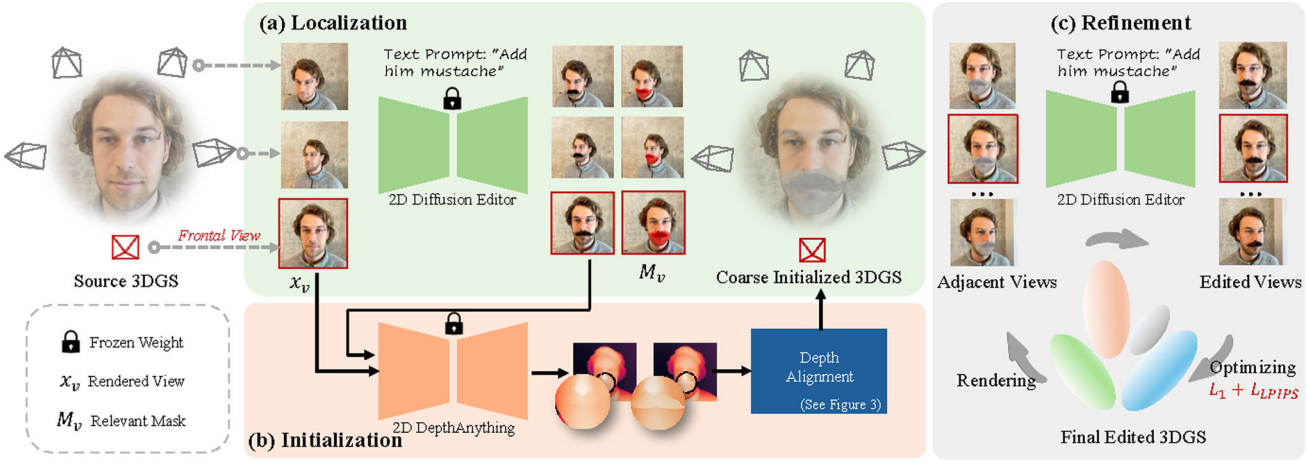


**Fig. 2** The overall framework of the proposed method that enables efficient fine-grained editing to 3DGS scenes. Localization: Given the text prompt, we first isolate relevant regions for each view by evaluating noise differences with and without text guidance (Eq.5). The resulting 2D masks are inverse-rendered to 3D (Chen et al., 2024c; Wang et al., 2025b) to obtain a view-consistent 3D mask, which has better quality in the side views that originally IP2P fails. Initialization: Local editing not only affects the existing points in the source 3DGS, but it may also require newly added Gaussian points, *e.g.*, added mustache. To address this, we propose leveraging the frontal view's depth prior from a pre-trained monocular depth estimator. Then, we add new Gaussian points by mapping the edited pixels in the aforementioned 2D mask to 3D space using the monocular depth. Refinement: We refine views adjacent to the frontal view conditioned on the coarsely edited scene, where the 3DGS scene is iteratively finetuned using refined views. After several finetuning iterations, a new frontal view is selected, and the adjacent refinement and optimization steps are repeated. (We visualize mask $M_v$ as the mixed picture with corresponding image for visualization)

$\mathcal{G}$ and a textual prompt $c$ describing the desired edits, we can perform IP2P to isolate regions relevant to the prompt. We achieve the 2D edited areas $M_v$ for each view by calculating the difference between the noise predicted by IP2P's UNet, $\epsilon_\theta$, when conditioned on both the image and text, and the noise predicted when conditioned only on the image with an empty text prompt as follows

$$M_v = \mathcal{F}(\left\|\epsilon_\theta(\mathbf{z}_{v,\tau}, \mathbf{x}_v, \mathbf{c}) - \epsilon_\theta(\mathbf{z}_{v,\tau}, \mathbf{x}_v, \emptyset)\right\|_c, \gamma), \tag{5}$$

where $\tau$ is a hyper-parameter determines the timestep that we use, $\mathbf{z}_{v,\tau} = \sqrt{\alpha_\tau}\,\mathcal{E}(\mathbf{x}_v) + \sqrt{1-\alpha_\tau}\,\epsilon$ is the noised latent code in timestep $\tau$, and $\|\cdot\|_c$ is the mean absolute value in the channel dimension. We obtain the binary mask through $\mathcal{F}$ by applying a threshold $\gamma \in [0, 1]$, followed by Gaussian filtering to reduce isolated regions and noise.

After that, we inverse render $\{M_v\}_{v=1}^V$ to 3D space (Wang et al., 2024, 2025b) to achieve the sparse 3D mask $\mathcal{M}$ with the number of Gaussian points in the source 3DGS $\mathcal{G}$.

## 4.2 Incremental Initialization with Predicted Depth

Although 2D diffusion generation/editing models achieve more promising results than 3D models, they usually only perform well on specific views, such as the frontal view, due to the higher amount of training data for these perspectives. Therefore, we propose to randomly select a few views and start by editing the view $v_{\text{first}} = \arg\max_v \sum_{i,j} M_{v,i,j}$, which has the largest correlation with the text prompt. To provide better initialization of the Gaussian points, which was demonstrated to be crucial especially for sparse view settings (Zhu et al., 2025), we propose to utilize the depth prior from DepthAnything model (Yang et al., 2024a, b). Specifically, since the output of the DepthAnything is the disparity instead of the real depth as shown in Fig. 3, we need to calibrate it first before usage. Inspired by Kerbl et al. (2024), we propose to first utilize the depth rendered by the unedited Gaussian points using Eq. 2 to obtain the unknown scale $a$ and bias $b$ as follows which we found has better robustness than directly using the linear regression

$$a = \frac{s(\delta_{3DGS})}{s(\delta_{mono})}, b = f(\delta_{3DGS}) - a \cdot f(\delta_{mono}), \tag{6}$$

where $\delta_{mono}$ and $\delta_{3DGS}$ are the disparities from depth anything and 3DGS, respectively, *i.e.*, $\mathbf{d} = \frac{1}{\delta}$, $f(\delta) = \text{median}(\delta)$ and $s(\delta) = \frac{1}{M}\sum_i |\delta_i - f(\delta)|$. We could obtain the depth after calibration from monocular depth estimator as $\mathbf{d}_{mono} = \frac{1}{a\delta_{mono}+b}$. After that, we can obtain the Gaussian points to be added as follows

$$\Delta\mathcal{G} = \mathcal{M} \odot P[\frac{\mathbf{d}_{mono}(\mathbf{x}_{edited})}{\mathbf{d}_{mono}(\mathbf{x}_{unedit})}\mathbf{d}_{3dgs}, \mathbf{x}^{edit}] \tag{7}$$

where $P$ is the mapping from 2D pixels to 3D points given the predicted depth and image, and $\mathbf{d}_{mono}(\mathbf{x})$ is the calibrated monocular depth from image $\mathbf{x}$. The motivation behind Eq. 7 is that it uses the same estimator to identify the relative depth variation, enhancing the robustness of the results. The added

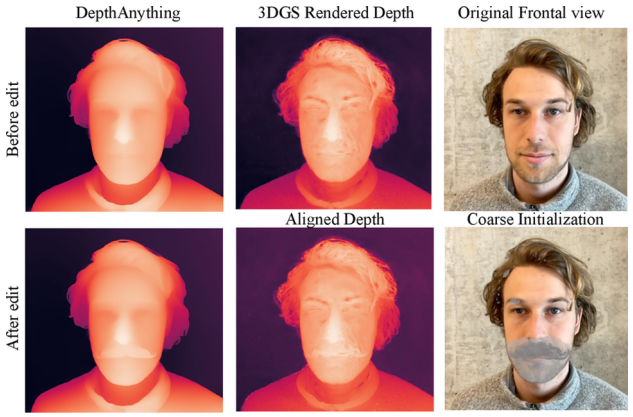


**Fig. 3** Illustration of the local initialization process using depth maps predicted by DepthAnything (Yang et al., 2024a, b). According to the predicted depth maps by DepthAnything on before and after edit 2D view images, we convert the 3DGS rendered depth map into aligned depth, adjusted based on the residual differences between the before and after DepthAnything predictions. This yields a regional initialized Gaussian points and its corresponding coarse rendered edit view

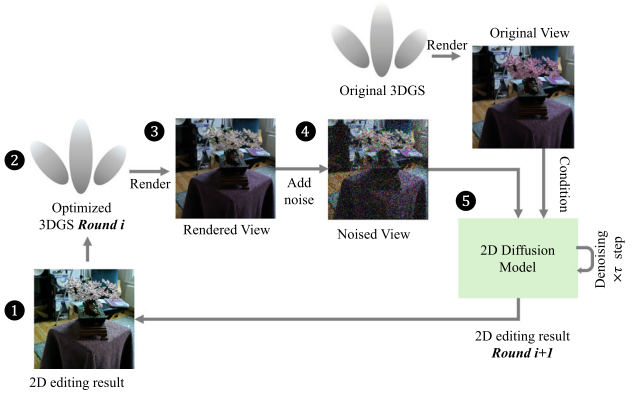


**Fig. 4** Detailed workflow of conditional sequential view refinement. ①→② In each round $i$, the 2D editing results are used to optimize the 3DGS representation. ②→③ The updated 3DGS is then rendered to obtain views with improved multi-view consistency, though artifacts may still persist (only one view is shown for clarity). ③→④ To mitigate these artifacts, the rendered image is not directly used as the diffusion condition; instead, it is perturbed with noise to a timestep $\tau$ ($0 < \tau < T$). ⑤ The diffusion model is always conditioned on the unaltered original view, ensuring faithful guidance. It then performs $\tau$-step denoising to produce the refined view $\mathbf{x}'_{j_k}$, which supervises the 3DGS optimization in round $i+1$

Gaussian points $\Delta\mathcal{G}$ are added to the source 3DGS to provide better initialization of areas with depth change.

### 4.3 Conditional Sequential View Refinement

After the initialization, for viewpoints near the frontal view, the 3D shape already has a substantial initialization and only requires refinement of texture and color. To this end, our approach utilizes several rounds of conditional sequential refinement to improve frame quality; each individual round is depicted in Fig. 4.

To effectively propagate edited information from the frontal view, we first select $m$ adjacent views $\{\mathbf{x}_{j_k}\}_{k=1}^{m}$ of the frontal view $\mathbf{x}_{v_{\text{first}}}$:

$$\{\mathbf{x}_{j_k}\}_{k=1}^{m} = \arg\operatorname{top}_m^{min}\left(\|\mathbf{p}_{v_{\text{first}}} - \mathbf{p}_i\| \mid v_{\text{first}} \neq i\right), \tag{8}$$

where $\|\mathbf{p}_{v_{\text{first}}} - \mathbf{p}_i\|$ represents the Euclidean distance between the camera positions of views $\mathbf{x}_{v_{\text{first}}}$ and $\mathbf{x}_i$. Following the pipeline illustrated in Fig. 4, we employ a multi-round refinement process to systematically eliminate inconsistencies. In each round, rather than performing standard image editing, we treat the selected rendered views $\tilde{\mathbf{x}}_{j_k}$ from the current 3DGS as an informative initialization. Specifically, we perturb the latent representation of these rendered coarse views $\tilde{\mathbf{x}}_{j_k}$ with noise to a specific timestep $\tau$ (as defined in Eq. 4), yielding $\tilde{\mathbf{z}}_{j_k,\tau}$. The IP2P (Haque et al., 2023) denoising process then commences directly from this noisy latent $\tilde{\mathbf{z}}_{j_k,\tau}$. Crucially, to ensure content preservation and faithful guidance, the diffusion model is always conditioned on the original unedited views $\mathbf{x}_{j_k}$ to sample the refined image $\mathbf{x}'_{j_k}$ from the distribution $p(\mathbf{x}'_{j_k}|\mathbf{x}_{j_k}, \tilde{\mathbf{z}}_{j_k,\tau}, \mathbf{c})$. This informed starting point enforces structural alignment from the outset, significantly improving multi-view consistency and optimization stability. The resulting refined frames are subsequently used to optimize the 3DGS scene:

$$\mathcal{G}^{\text{edit}} = \arg\min_{\mathcal{G}} \sum_{v \in V} \mathcal{L}_{\text{edit}}(\mathcal{R}(\mathcal{G}, v), \mathbf{x}'_v), \tag{9}$$

where $\mathcal{L}_{\text{edit}}$ incorporates L1 and LPIPS (Zhang et al., 2018b) losses. To accommodate large-scale geometric edits and correct any potential initialization inaccuracies, we optimize the entire set of Gaussian parameters throughout this iterative cycle.

After several iterations of finetuning the 3DGS for each round, we designate a new frontal view from $\{\mathbf{x}'_v\}_{v \in V}$ as the updated reference and reselect $m$ adjacent views to initiate a fresh round of conditional editing. The workflow for a single round can be summarized as follows: randomly select a frontal view → select $m$ adjacent views of the chosen frontal view → apply 2D conditional editing → perform 3DGS finetuning. Each cycle progressively improves coherence across perspectives, ensuring consistency and alignment with the updated reference viewpoint. To improve efficiency, we progressively reduce the IP2P sampling starting timestep $\tau$ in experiments. We perform two or three such cycles in experiments. The detailed hyper-parameters can be found in supplementary materials.

# 5 Experiments

## 5.1 Implementation Details

The proposed method is implemented using PyTorch and trained on one NVIDIA A6000 GPUs. We use the optimized renderer from Kerbl et al. (2023b) for Gaussian rendering, with our implementation based on Threestudio (Guo et al., 2023). All 3D Gaussians in this study are trained using the method described in Kerbl et al. (2023b). For 2D editing, we employ IP2P (Brooks et al., 2023) following most prior methods (Chen et al., 2024, c; Haque et al., 2023) to ensure fair comparison. A selection of 3D scenes from publicly available datasets, such as LLFF and Mip-NeRF360 (Barron et al., 2022; Mildenhall et al., 2019), is used for evaluation.The 3DGS model is finetuned for 1,000-1,500 iterations, with a default classifier-free guidance strength of 7.5. For certain cases, we adjust the edit strength consistently across all competing methods.

**Training details.** For the training iterations, the 3DGS model is fine-tuned for 1,000 iterations on the "kitchen" scene, as shown in Fig. 7, and for 1,500 iterations on other scenes. For every 500 iterations, we repeat the cycle mentioned in Section 4.3 of main paper and re-select a new frontal view. We select a linearly decreasing starting timestep for IP2P, as high-level semantics are adjusted after some iterations, leaving only low-level structural details to be refined in subsequent cycles. Specifically, $t = [750, 500, 250]$ is used as the starting denoising steps for three training cycles, and $t = [750, 500]$ is used for two training cycles. For each cycle, we select 20 adjacent views for 2D editing. Besides, a default classifier-free guidance strength of 7.5 is used, except for the "face" scene in Figure 1 of main paper with prompts such as "make him older" and "turn him into a werewolf." A classifier-free guidance strength of 7.5 effectively handles most attribute and accessory editing cases. For the localization part, we set $\tau = 600$ steps and the mask threshold $\gamma = 0.6$ for most scenes.

## 5.2 Baselines and Evaluation Metrics

We compare our approach with a set of competing methods, including two NeRF-based editing techniques–Instruct-N2N (Haque et al., 2023) and ViCA-NeRF (Dong and Wang, 2023)–and five 3D Gaussian Splatting (3DGS)-based methods: GaussianEditor (Chen et al., 2024c), DGE (Chen et al., 2024), Gaussctrl (Wu et al., 2024), EditSplat (Lee et al., 2025), and TIP-Editor (Zhuang et al., 2024). All methods, including ours, utilize IP2P as the 2D editor, except for Gaussctrl, which uses ControlNet (Zhang et al., 2023). To evaluate these methods, we use CLIP Text-Image Direction Similarity (CTIDS) (Radford et al., 2021) to assess semantic alignment after editing, along with Image-Image Similarity (IIS) and FID (Heusel et al., 2017) to measure fidelity. We also measure editing time, following the evaluation metrics used in (Wang et al., 2024). Furthermore, we conduct a user study to assess subjective aspects, specifically measuring the detailed structural quality and consistency across different views, results and details of which are included in the supplementary.

## 5.3 Qualitative Evaluation

Figure 5 shows a comparison between our method and competing methods across different views on two scenes, each accompanied by the corresponding text prompt. GaussianCtrl (Wu et al., 2024) utilizes ControlNet as the 2D editor, which is more suitable for global editing, offering better consistency and aesthetic quality. However, it struggles to retain original scene information, often altering features such as face ID and skin tone during editing. Noted that we followed the GaussianCtrl using pre-processing data to a $512^2$ resolution, and we manually padded the visual examples in Fig. 5 for better visualization. DGE (Chen et al., 2024) and GaussianEditor (Chen et al., 2024c) can maintain high-level information consistency; however, artifacts may appear in boundary details, such as around the edges of an added mustache. Additionally, fine-grained elements like clothing tags and zippers may lose some high-frequency details, resulting in slight distortions. Apart from the issue of detail preservation, the second example highlights an information leakage problem. Existing methods may inadvertently modify unintended regions that are closely related to the text prompt, such as background elements like grass and leaves. This can lead to inconsistencies across views and the appearance of unwanted artifacts. In contrast, towards local editing tasks, our proposed method demonstrates the ability to achieve enhanced fine-grained structural details, particularly evident in the reduced occurrence of artifacts along the editing boundaries. Moreover, our method excels in maintaining high fidelity to the original scene, effectively preserving key identity-related attributes, such as face ID, clothing details, and hair style after editing. We include additional qualitative results in Fig. 6, showcasing multiple edited views. The results demonstrate the effectiveness of our method, exhibiting strong consistency across different viewpoints. We primarily focus on editing attributes such as color, texture, and facial features, as well as accessories like hats, clothing, and pants. For local editing tasks, our proposed method showcases its capability to achieve refined fine-grained structural details, with a notable reduction in artifacts along editing boundaries. Additionally, it demonstrates strong fidelity to the original scene by effectively preserving key identity-related attributes such as facial features, clothing details, and hairstyle after editing.

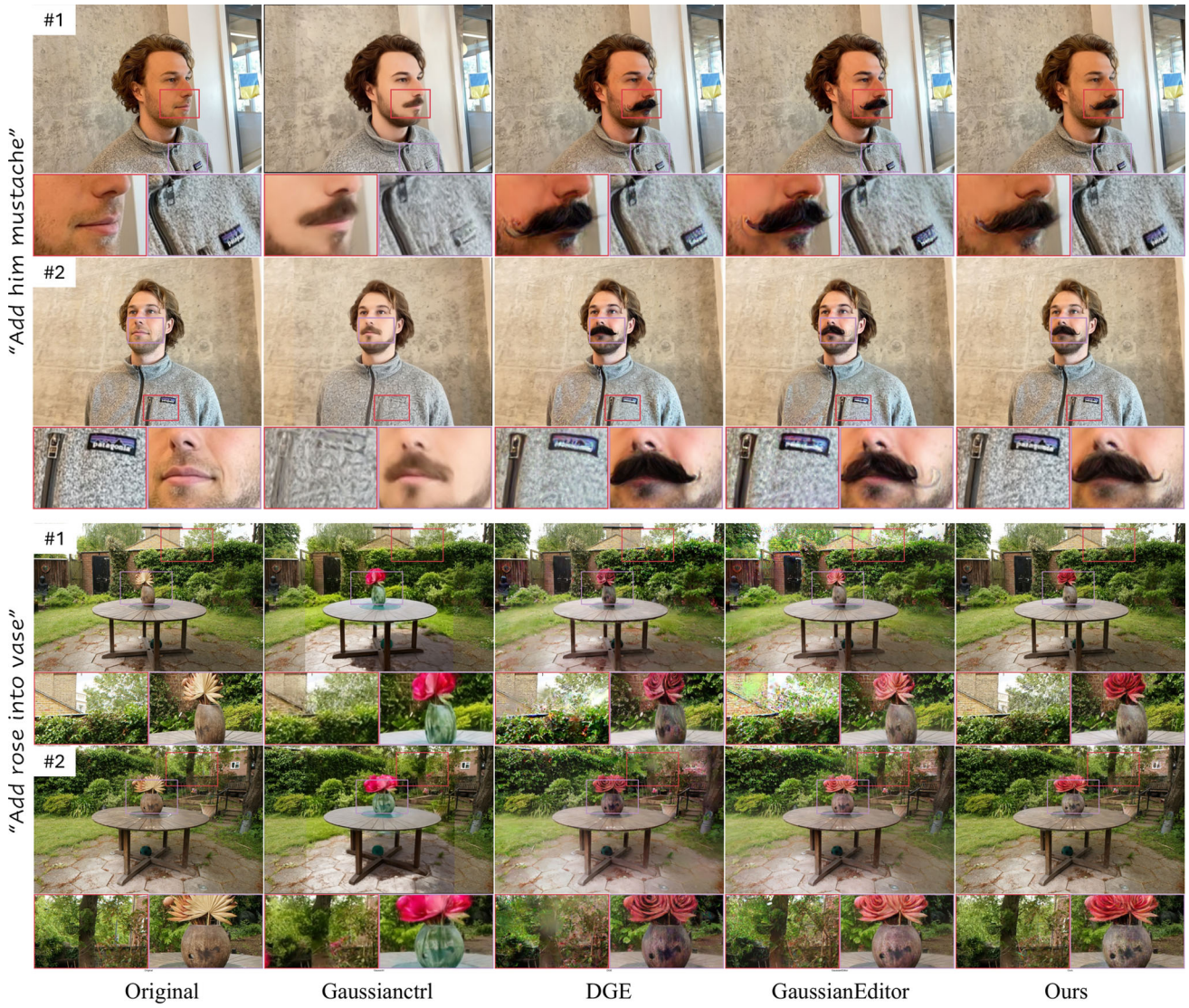


**Fig. 5** Qualitative comparison with three 3DGS-based competing methods: The leftmost column demonstrate the original views, while the right four columns show the rendering view of edited 3DGS by Gaussianctrl (Wu et al., 2024), DGE (Chen et al., 2024), GaussianEditor (Chen et al., 2024c), and our method, respectively. Our method effectively preserves fidelity and structural details while delivering artifact-free and view-consistent edited results

**Table 1** When comparing different editing methods, Gaussian Splatting-based techniques are considerably quicker than NeRF-based approaches.

| Method | 3D Model | CTIDS↑ | IIS↑ | FID↓ | Editing Time |
|---|---|---|---|---|---|
| IN2N (Haque et al., 2023) | NeRF | **0.2128** | **0.7591** | 150.80 | ∼ 51min |
| ViCA-NeRF (Dong and Wang, 2023) | NeRF | 0.2103 | 0.7205 | 161.35 | ∼ 28min |
| WatchYourSteps (Mirzaei et al., 2025) | NeRF | 0.2117 | 0.7368 | **129.68** | ∼ 51min |
| GaussianEditor (Chen et al., 2024c) | 3DGS | 0.2043 | 0.8212 | 70.14 | ∼ 7min |
| DGE (Chen et al., 2024) | 3DGS | **0.2057** | 0.8183 | <u>68.60</u> | ∼ 4min |
| Gaussctrl (Wu et al., 2024) | 3DGS | 0.1882 | 0.7862 | 132.76 | ∼ 9min |
| EditSplat (Lee et al., 2025) | 3DGS | <u>0.2093</u> | <u>0.8301</u> | 79.24 | ∼ 6min |
| TIP-Editor (Zhuang et al., 2024) | 3DGS | 0.2013 | 0.8248 | 74.54 | ∼ 8min |
| Ours | 3DGS | 0.2053 | **0.8308** | **65.96** | ∼ 2min |

Our method achieves comparable performance, particularly in preserving data fidelity, while offering the fastest editing speed
Bold indicates the best result, and underlined values indicate the second-best result


(a) Examples on "person" scene from IN2N (Haque et al.,)

(b) Examples on "fanzhou" scene from Nerf-Art (Wang et al., 2023)

**Fig. 6** Illustrative visual examples showcasing the application of our proposed method for local editing on 3D scenes, demonstrating its effectiveness in handling fine-grained modifications

### 5.4 Quantitative Evaluation

Following previous works (Haque et al., 2023; Chen et al., 2024), we evaluate the alignment of edited 3D models and target text prompts with CLIP similarity score, *i.e.*, the cosine similarity between the text and image embeddings encoded by CLIP. We also evaluate the fidelity preservation between original and edited 3D models by calculating the cosine similarity between original image and edited image as well as the FID between original and edited image sets. Table 1 presents a detailed comparison of our method against several competing approaches including editing results on three 3D scenes, where the details can be found in supplementary. Our method demonstrates comparable CLIP Text-Image Direction Similarity scores to the current state-of-the-art methods, indicating its effectiveness in preserving textual alignment. Additionally, our approach outperforms competing methods in preserving directional consistency with the original scene, ensuring high-fidelity editing that selectively modifies only the intended characteristics of the scene. Apart from evaluating effectiveness, we also assess the efficiency of our approach. By leveraging coarse initialization in Sect. 4.2 and skipping diffusion steps during iterative refinement in Sect. 4.3, our method achieves more efficient editing, offering up to a 4× speedup compared to existing 3DGS-based methods and a 25× speedup compared to NeRF-based methods. To evaluate background preservation, we report masked PSNR (40.73 dB), SSIM (0.9903), and LPIPS (Zhang et al., 2018a) (0.0089) calculated strictly outside the predicted 3D masks. These metrics, averaged across all validation views, quantitatively confirm that our method leaves unedited regions virtually untouched, consistent with our qualitative results.

### 5.5 User Study

We conducted a user study focusing on two key aspects: view consistency and fine-grained details. Our proposed method was compared against GaussCtrl (Wu et al., 2024), DGE (Chen et al., 2024), and GaussianEditor (Chen et al., 2024c) on the several scenes, with the participation of 30 users. Our proposed method also achieves a voting percentage of 61.67%, with GaussCtrl, DGE, and GaussianEditor receiving 8.33%, 18.33%, and 11.67%, respectively.

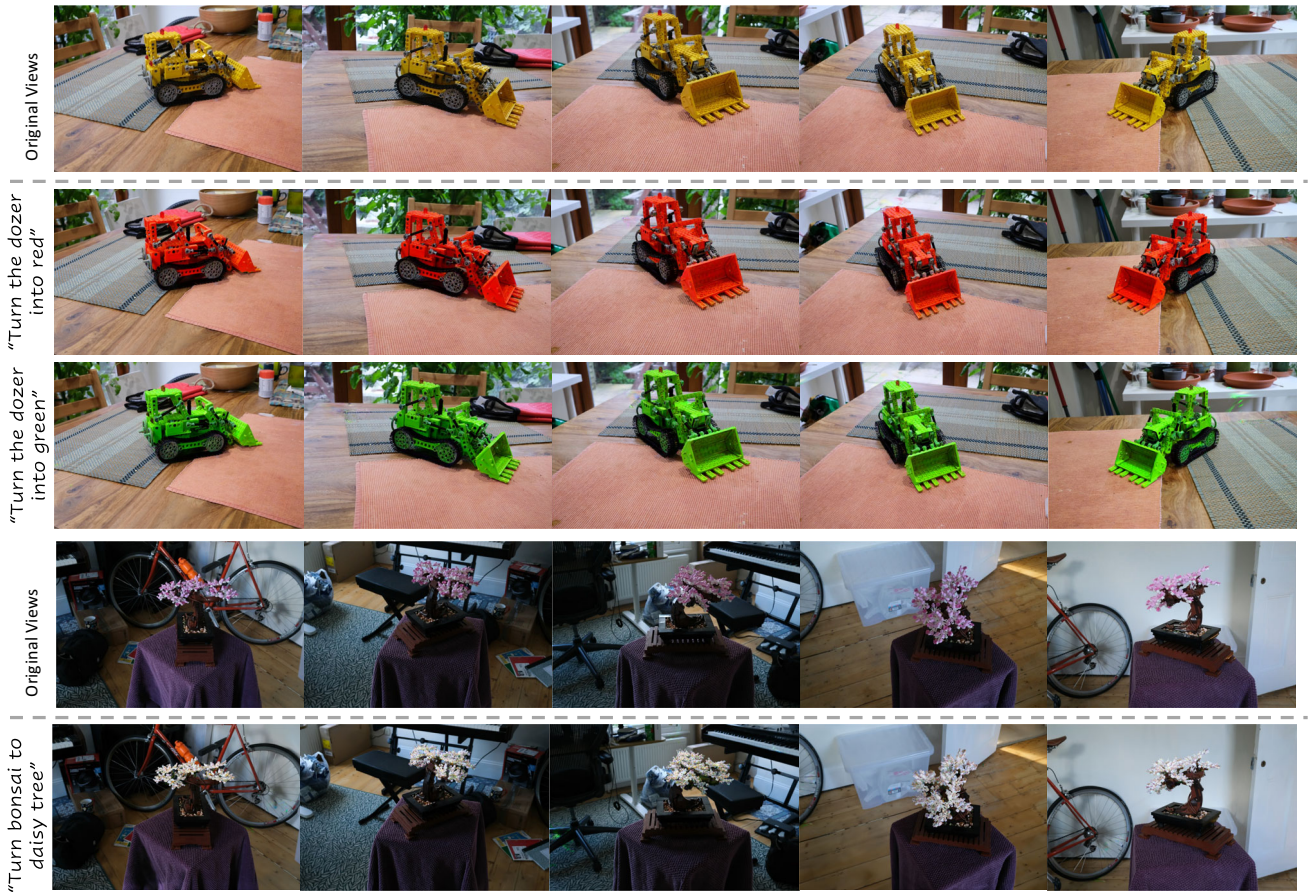


**Fig. 7** Illustrative visual examples on "kitchen" and "bonsai" scenes from Mip-Nerf 360 (Barron et al., 2022) dataset with various text prompts showcasing the application of our proposed method for local editing on a 3D scene, demonstrating its effectiveness in handling fine-grained modifications


**Fig. 8** Visualization of our 3D localization and the corresponding edited result compared to the whole region editing and directly using SAM with extra prompt for localization

## 5.6 Ablation Study

**The effect of 3D localization.** To validate the effect of the 3D localization, we conduct experiments on a variant where the 3D localization component is removed from our framework. As shown in the comparison between the first column and third column in Figure 8, removing the 3D localization results in distortion and blurry in background and unedited regions, a significant decrease in the overall scene reconstructed quality, especially in scenarios with complex spatial configurations. Color information can leak into unrelated or undesired areas, leading to unintended color changes in objects like the vase, table, and floor. For instance, they might turn somewhat red due to the influence of a prompt focused on "rose". Besides, our 3D localization is distinct from most previous works (Chen et al., 2024, c; Wang et al., 2025b) used SAM with a corresponding prompt to locate the edited regions in object level. As shown in the comparison between the second and third column in Figure 8, some editing scenarios make it challenging to accurately target a specific object, such as object replacement or adding extra acces-

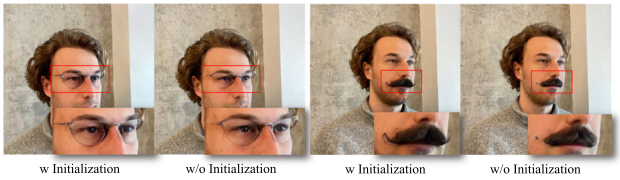


**Fig. 9** Two editing examples of our method with and without initialization. Without proper initialization, if parts of these accessories aren't set to slightly extend beyond the character's edges, they may be omitted in certain views, resulting in inconsistencies and undesired outcomes

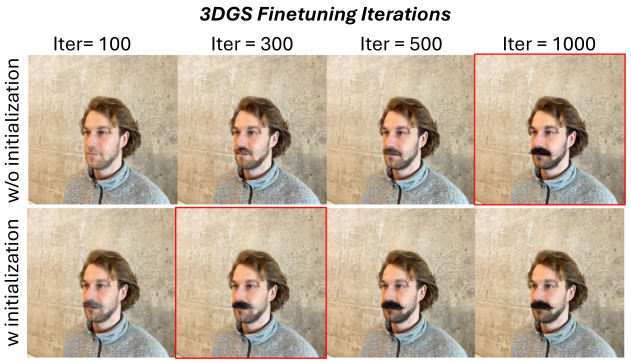


**Fig. 10** Visualization of intermediate results during 3DGS finetuning with and without initialization. Proper initialization effectively reduces the required finetuning iterations

sories. Sometimes, SAM may fail even if you provide a fairly precise prompt. In contrast, our 3D localization identifies the edited regions without requiring additional prompts and more effectively prevents information leakage into unedited areas.

**The effect of initialization by predicted depth.** We utilize the edited first view as a reference, generating its depth map using the DepthAnything model. This depth map is then inversely rendered into 3D space to produce a coarse point cloud for initialization. This initial coarse representation effectively reduces the overall training time and enables the edited shape and accessories in the first view to be consistently rendered across other views, serving as a condition for subsequent view edits. As illustrated in Figure 9, without a robust initialization, adding new accessories or altering the shape of an existing object may encounter significant challenges in localization. This is especially true for details like the edges of accessories, which are difficult to generate if they extend beyond the object's boundary in certain views. For example, in our case with glasses and a mustache, if parts of these accessories are not well-initialized to extend slightly beyond the character's edges, they may be omitted in certain views, leading to inconsistencies. Thus, a good initialization serves as a crucial solution to mitigate these localization limitations and ensure cohesive multi-view rendering.

## 5.7 Network Analysis

**Efficiency analysis.** With the proposed 3D localization and regional initialization, both the diffusion process and 3DGS fine-tuning are accelerated. On one hand, for IP2P sampling steps, we employ a linearly decreasing starting timestep, as high-level semantics are adjusted in the early iterations, leaving only low-level structural details to be refined in later stages. This approach effectively reduces the diffusion refinement time. (Details refer to supplementary.) On the other hand, since the coarse shape of the edited region (e.g., the mustache) is well-initialized, the necessary Gaussian points are already properly positioned. Only specific properties, such as texture and color, require further refinement using the diffusion model. As a result, the number of iterations for 3DGS fine-tuning is significantly reduced. As shown in Figure 10, with proper initialization, noticeable modifications occur after just 300 iterations, whereas without initialization, up to 1,000 iterations may be required.

**Robustness analysis.** Owing to our incremental initialization design, object removal tasks rely heavily on the localization and refinement stages. We demonstrate the effectiveness of our approach for object removal in Figure 12, featuring challenging cases such as 'make him bald' and 'remove the shoes on the carpet'. Besides, we also include some cases with extensive edited regions and simultaneous modification of multiple components as shown in Figure 13. Our method achieves precise localization of the regions to be erased and modified, ensuring that the edit mask accurately covers the target object while preserving the integrity of the surrounding background.

**Global editing.** Our method focuses on viewpoint consistency and high fidelity in localized editing, making it particularly effective for attribute modifications. However, for global edits like style transfer, 3D localization can be treated as an all-one mask. In this case, the required 3DGS finetuning iterations increase, as the initialized 3DGS may lack the precision of localized editing, necessitating further refinement to achieve the desired outcome. Figure 11 shows a style transfer example on the "face" scene, requiring 2,000 3DGS finetuning iterations over four cycles.

**Compared to direct 2D editing.** We present a visual comparison between standalone 2D diffusion editing and our complete 3D editing framework, using the instructions "change the bonsai to a blossom daisy tree" and "make the person wear jeans pants", as illustrated in Figure 14. The 2D diffusion results are obtained directly from InstructPix2Pix (Brooks et al., 2023), rather than from intermediate outputs of our pipeline, since our approach introduces an additional depth-aligned initialization stage to mitigate cross-view inconsistencies prior to refinement. Applying 2D diffusion independently to each view frequently results in discrepancies in geometry, texture, and color across view-


**Fig. 11** An example of global editing is setting $\gamma = 0$ for an all-one 3D localization


**Fig. 12** Examples of 3D editing removal tasks. Cases include "make him bald" and "remove the shoes on the carpet." Top row: Two viewpoints of the original 3DGS. Bottom row: Corresponding edited results demonstrating multi-view consistency

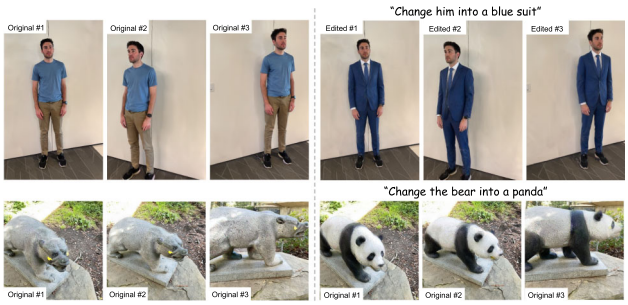


**Fig. 13** Additional 3D editing examples featuring extensive edited regions and simultaneous modification of multiple components. Cases include "change him into a blue suit" and "change the bear into a panda". Left: Three viewpoints of the original 3DGS. Right: Corresponding edited results demonstrating multi-view consistency

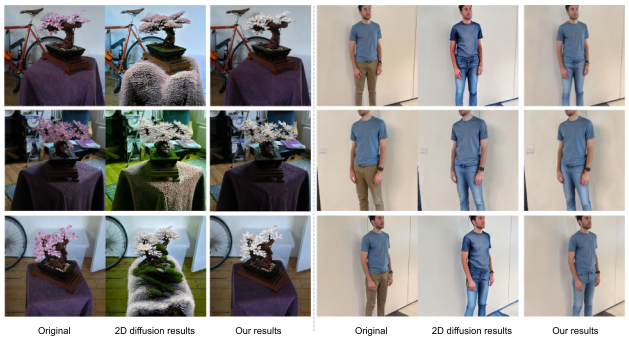


**Fig. 14** Visualization comparing standalone 2D diffusion edits across different views with our final 3D editing results, using the instructions "change the bonsai to a blossom daisy tree" and "make the person wear jeans pants." Note that the 2D diffusion results are directly generated by Instruct-Pix2Pix, rather than the intermediate outputs in our pipeline, since our method further performs depth-aligned initialization to reduce cross-view discrepancies before refinement

points. Moreover, enforcing more prominent edits in 3D space often requires a higher CFG scale, which can produce unnatural artifacts in certain views (see the first and third rows of the left example in Figure 14). In contrast, our method effectively enforces cross-view coherence through 3D optimization, leading to substantially improved multi-view consistency.

**Limitations.** Our method relies on an additional depth estimation module to provide an initial 3D reconstruction of the scene, which serves as the starting point for the subsequent optimization process. This initialization is critical to guiding the editing process toward geometrically consistent and view-aligned results. However, in scenarios where the depth estimation is highly inaccurate, such as under extreme lighting conditions, low-texture regions, reflective surfaces, or occlusions, the initialization quality deteriorates significantly. As a result, the optimization is more prone to start from an incorrect geometry and requires substantially more iterations to converge to a satisfactory solution. The model must effectively compensate for these inaccuracies through iterative refinement, gradually correcting the structural inconsistencies and recovering a faithful reconstruction. While our pipeline is still capable of handling such challenging cases, the overall computational cost increases and the editing fidelity may initially be compromised due to the suboptimal guidance from the erroneous depth.

## 6 Conclusion

In conclusion, our proposed framework for 3D Gaussian Splatting (3DGS) offers a highly efficient and effective solution for localized editing in 3D scenes. By integrating 2D diffusion-based editing to identify and guide region-specific modifications across multiple views, and leveraging depth-based initialization derived from a pretrained 2D foundation model, our method significantly improves both the precision and coherence of edits in the 3D space. This design enables an iterative, view-consistent refinement process that preserves structural integrity and textural fidelity across viewpoints. Extensive experiments demonstrate that our approach not

only achieves state-of-the-art editing quality but also delivers a substantial speedup–up to 4× faster–compared to existing 3DGS-based editing pipelines, making it a practical and scalable choice for real-world applications.

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/s11263-026-02941-w.

**Author Contributions** Lanqing Guo conducted the main research, including conceptualization, methodology development, experiments, and writing of the original draft. Yan Zheng assisted with part of the comparative experiments and validation. Yufei Wang, Hezhen Hu, and Zhangyang Wang contributed to idea discussion and feasibility analysis. Yeying Jin and Siyu Huang contributed to manuscript polishing and verification of experimental settings. Zhangyang Wang also provided supervision, project guidance, and contributed to manuscript revision. All authors reviewed and approved the final manuscript.

**Funding** This research has been supported by the Sony Research Award Program and computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at UT Austin.

**Data Availability** The datasets analyzed during the current study are available in the repository at the following link: https://instruct-gs2gs.github.io/.

## Declarations

**Ethical Approval** Ethical approval was not required as this study uses only publicly available datasets.

**Conflict of Interest** The authors declare that they have no Conflict of interest.

**Consent for Publication** Not applicable.

**Consent to Participate** Not applicable.



## References

Anciukevičius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N.J., & Guerrero, P. (2023). Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12608–12618.

Barron, J. T. Mildenhall, B., Verbin, D., Srinivasan, P. P., & Hedman, P. (2022). Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5470–5479.

Brooks, T., Holynski, A., & Efros, A. A. (2023). Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402

Chen M, Laina I, Vedaldi A. (2024). Dge: Direct gaussian 3d editing by consistent multi-view editing. In *European conference on computer vision*. Cham: Springer Nature Switzerland. pp. 74–92.

Chen, Y., Yuan, Q., Li, Z., Liu, Y., Wang, W., Xie, C., Wen, X., & Yu, Q. (2024b). Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene. *IEEE Transactions on Visualization and Computer Graphics*.

Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., & Lin, G. (2024c). Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21476–21485.

Couairon, G., Careil, M., Cord, M., Lathuiliere, S., & Verbeek, J. (2023). Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2174–2183.

Dong, J., & Wang, Y.-X. (2023). Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Advances in Neural Information Processing Systems* (p. 36)

Guo, Y.-C., Liu, Y.-T., Shao, R., Laforte, C., Voleti, V., Luo, G., Chen, C.-H., Zou, Z.-X., Wang, C., & Cao, Y.-P. (2023). Threestudio: A unified framework for 3d content generation threestudio: A unified framework for 3d content generation.

Haque, A., Tancik, M., Efros, A. A., Holynski, A., & Kanazawa, A. (2023). Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750.

Hertz, A., Mokady, R., Tenenbaum, et al. (2023). Prompt-to-prompt image editing with cross-attention control. The Eleventh International Conference on Learning Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5291–5300.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637.

Ho, H.-I., Xue, L., Song, J., & Hilliges, O. (2023). Learning locally editable virtual humans. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Ho J, Salimans T. (2021). Classifier-Free Diffusion Guidance[C]//NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1125–1134.

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2017–2025.

Kawar, B., Zada, S., Lang, O., et al., (2023). Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al. (2023). 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph., 42*(4), 139–1.

Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., & Drettakis, G. (2024). A hierarchical 3d gaussian representation for

real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG), 43*(4), 1–15.

Lee, D. I., Park, H., Seo, J., Park, E., Park, H., Baek, H. D., Shin, S., Kim, S., & Kim, S. (2025). Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11135–11145.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timm, F., & Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11471.

Meng, C., Song, Y., Song, J., Wu, J., & Ermon, S. (2021). Sdedit: Image synthesis and editing with stochastic differential equations. preprint arXiv:2108.01073.

Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., & Kar, A. (2019). Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG), 38*(4), 1–14.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 405–421.

Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M. A. Kelly, J., Levinshtein, A., Derpanis, K. G., & Gilitschenski, I. (2025). Watch your steps: Local image and scene editing by text instructions. In *European Conference on Computer Vision*, pp. 111–129. Springer.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., & Cohen-Or, D. (2023). Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6038–6047.

Parmar, G. et al. (2023). Zero-shot image-to-image translation. ACM SIGGRAPH 2023 conference proceedings.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., & Rombach, R.Sdxl: Improving latent diffusion models for high-resolution image synthesis. *The Twelfth International Conference on Learning Representations.*

Quan, W., Chen, J., Liu, Y., Yan, D.-M., & Wonka, P. (2024). Deep learning-based image and video inpainting: A survey. *International Journal of Computer Vision, 132*(7), 2367–2400.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. Proceedings of the International Conference on Machine Learning (ICML)

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2023). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510.

Ruiz, N., Li, Y., Wadhwa, N., Pritch, Y., Rubinstein, M., Jacobs, D. E., & Fruchter, S. (2025). Magic insert: Style-aware drag-and-drop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 15971–15981.

Shuai, X., Ding, H., Ma, X., Tu, R., Jiang, Y.-G., & Tao, D. (2024). A survey of multimodal-guided image editing with text-to-image diffusion models. preprint arXiv:2406.14555

Tumanyan, N., Bar-Tal, O., Bagon, S., & Dekel, T. Splicing vit features for semantic appearance transfer. (2022) In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10738-10747.

Wang, C., Jiang, R., Chai, M., He, M., Chen, D., & Liao, J. (2023). Nerf-art: Text-driven neural radiance fields stylization. IEEE Transactions on Visualization and Computer Graphics

Wang, J., Yue, Z., Zhou, S., Chan, K. C. K., & Loy, C. C. (2024). Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision, 132*(12), 5929–5949.

Wang, J., Fang, J., Zhang, X., Xie, L., & Tian, Q. (2024b). Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20902–20911.

Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., et al. (2025). Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision, 133*(5), 3059–3078.

Wang, Y., Guo, L., Li, Z., Huang, J., Wang, P., Wen, B., & Wang, J. (2025b). Training-free text-guided image editing with visual autoregressive model. preprint arXiv:2503.23897.

Wang, Y., Guo, L., Li, Z., Huang, J., Wang, P., Wen, B., & Wang, J. (2025b).Training-free text-guided image editing with visual autoregressive model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 17577–17586.

Weber, E., Holynski, A., Jampani, V., Saxena, S., Snavely, N., Kar, A., & Kanazawa, A. (2024). Nerfiller: Completing scenes via generative 3d inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20731–20741.

Wu, J., Bian, J.-W., Li, X., Wang, G., Reid, I., Torr, P., & Prisacariu, V. (2024). GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. European conference on computer vision

Xiao, G., Yin, T., Freeman, W. T., Durand, F., & Han, S. (2024). Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision*, pp. 1–20.

Xiao, H., Chen, Y., Huang, H., Xiong, H., Yang, J., Prasad, P., & Zhao, Y. (2025). Localized gaussian splatting editing with contextual awareness. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5207–5217. IEEE.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., & Zhao, H. (2024a). Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10371–10381.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything v2. *Advances in Neural Information Processing Systems, 37*, 21875–21911.

Zhang, D. J., Wu, J. Z., Liu, J.-W., Zhao, R., Ran, L., Gu, Y., Gao, D., & Shou, M. Z. (2024). Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, pp. 1–15.

Zhang L, Rao A, Agrawala M. (2023). Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 3836–3847.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595.

Zheng, Y., Liang, Z., Cong, X., Wang, Y., Wang, P., Wang, Z. et al. (2024). Oscillation inversion: Understand the structure of large flow model through the lens of inversion method. preprint arXiv:2411.11135.

Zhu J, Yang L, Yao A. (2026). Instructhumans: Editing animated 3d human textures with instructions. *IEEE Transactions on Multimedia*.

Zhu, J.-Y., Krahenbuhl, P., Shechtman, E., & Efros, A. A. (2016). Generative visual manipulation on the natural image manifold. *European Conference on Computer Vision (ECCV)*, pp. 597–613.

Zhu, Z., Fan, Z., Jiang, Y., & Wang, Z. (2025). Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pp. 145–163. Springer.

Zhuang, J., Wang, C., Lin, L., Liu, L., & Li, G. (2023). Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10.

Zhuang, J., Kang, D., Cao, Y.-P., Li, G., Lin, L., & Shan, Y. (2024). Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *ACM Transactions on Graphics (ToG), 43*(4), 1-12.

Zuo, X., Samangouei, P., Zhou, Y., Di, Y., & Li, M. (2025). Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision, 133*(2), 611-627.