GS-Checker: Tampering Localization for 3D Gaussian Splatting

Haoliang Han[1], Ziyuan Luo[1], Jun Qi[1], Anderson Rocha[2], Renjie Wan[1]*

[1]Department of Computer Science, Hong Kong Baptist University
[2]Institute of Computing, University of Campinas
{haolianghan, ziyuanluo}@life.hkbu.edu.hk, tsinghua.qijun@gmail.com, arrocha@unicamp.br, renjiewan@hkbu.edu.hk

Abstract

Recent advances in editing technologies for 3D Gaussian Splatting (3DGS) have made it simple to manipulate 3D scenes. However, these technologies raise concerns about potential malicious manipulation of 3D content. To avoid such malicious applications, localizing tampered regions becomes crucial. In this paper, we propose ***GS-Checker***, a novel method for locating tampered areas in 3DGS models. Our approach integrates a 3D tampering attribute into the 3D Gaussian parameters to indicate whether the Gaussian has been tampered. Additionally, we design a 3D contrastive mechanism by comparing the similarity of key attributes between 3D Gaussians to seek tampering cues at 3D level. Furthermore, we introduce a cyclic optimization strategy to refine the 3D tampering attribute, enabling more accurate tampering localization. Notably, our approach does not require expensive 3D labels for supervision. Extensive experimental results demonstrate the effectiveness of our proposed method to locate the tampered 3DGS area.

Code — https://github.com/haolianghan/GS-Checker

1 Introduction

Methods for editing 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) have gained significant attention. Despite the impressive capabilities of these advanced editing techniques (Chen et al. 2024a; Chen, Laina, and Vedaldi 2024; Wu et al. 2024), they could be exploited to alter 3DGS content maliciously and misuse tampered models. Therefore, detecting such tampering is essential to prevent malicious applications.

In our scenario, illustrated in Figure 1, a malicious user uses advanced editing tools to alter specific properties of a 3DGS model originally created by its owner. This tampering raises two main concerns: **1)** it may undermine the original owner's rights over the 3DGS model, and **2)** the tampered model could convey meanings with potentially negative societal impacts. Our objective is to detect such unauthorized modifications, safeguarding the integrity and rightful ownership of 3DGS models.

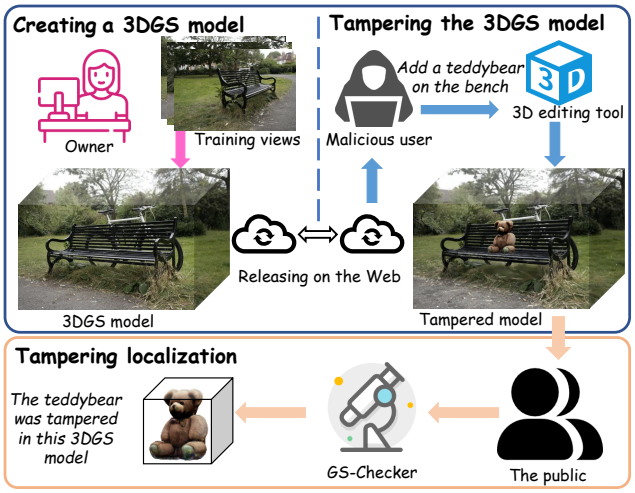

Figure 1: Our proposed scenario for 3DGS tampering localization. When the owner creates a 3DGS model and releases it publicly over the web, malicious users can manipulate it using advanced 3D editing tools for illegal purposes. Upon the release of a tampered 3DGS, the public can use our proposed GS-Checker to identify the tampered areas, thereby protecting the integrity and ownership of the 3DGS model.

A straightforward approach is to leverage established tampering localization mechanisms. While tampering localization has been well-researched for decades, most approaches (Dong et al. 2022; Guo et al. 2023; Guillaro et al. 2023; Zhang et al. 2024a; Ma et al. 2023) focus primarily on 2D data like images or video frames. When applied to 3DGS (Kerbl et al. 2023), these mechanisms can only detect tampering in rendered images. However, without considering 3D spatial correlations across viewpoints, 2D methods struggle to accurately locate tampered areas in each viewpoint. For example, as shown in Figure 5, directly applying the 2D image tampering localization approach can mistakenly identify authentic areas as tampered.

A major reason for the above limitations is that it overlooks the specific challenge in our scenario: ***the tampering occurs within the 3DGS model***. 2D tampering localization (Dong et al. 2022; Wang et al. 2022b; Guo et al. 2023;

*Corresponding author. This work was carried out at the Renjie Group, Hong Kong Baptist University.

Guillaro et al. 2023; Zhang et al. 2024a; Ma et al. 2023) can only partially identify tampering traces from rendered 2D images, and these traces do not accurately represent the tampering operations in 3D. We need a 3D tampering localization method capable of directly detecting tampering at the model level. Then, regardless of the viewpoint rendered, we can identify the tampered areas.

To achieve this goal, we propose ***GS-Checker***, a framework that can conduct the tampering localization for the 3DGS model. In traditional 2D methods, tampered pixels often deviate from normal statistical distributions, revealing structural or contextual inconsistencies. This phenomenon also extends to the 3D domain, where tampered 3D Gaussians exhibit certain anomalies in their parameter distributions. These anomalies reflect differences between tampered and authentic 3D Gaussians, providing critical cues for identifying tampering. By analyzing these deviations, we can effectively detect the tampering trace across the entire 3D representation. In our whole design, we fully make use of the parameters within 3D Gaussians to identify the tampering trace. Specifically, we introduce a *3D contrastive mechanism* that compares the similarity of key attributes between 3D Gaussians to seek tampering cues at the 3D level. To better exploit the properties of the 3DGS model, a *unique tampering attribute* is integrated into each 3D Gaussian parameters. This tampering attribute is initialized by extracting tampering cues at the 2D level and projecting them into 3D. Besides, we further introduce a *cyclic optimization strategy* to update the 3D tampering attribute by projecting it into 2D. This allows us to optimize at both 2D and 3D levels jointly, resulting in precise localization of 3DGS tampering.

Figure 2 illustrates our framework. The tampering attribute is integrated within the 3D Gaussians parameters, which is initialized via 3D voting. The 3D contrastive mechanism seeks tampering cues at 3D level by comparing the similarity of key attributes between 3D Gaussians. The cyclic optimization strategy updates the tampering attribute iteratively by rendering it back into 2D. Our approach fully exploits the unique properties of 3D Gaussians, performing joint optimization across 2D and 3D levels to precisely localize tampered regions in 3DGS models. To sum up, our key contribution can be concluded as follows:

- A pioneering method, ***GS-Checker***, for locating manipulated areas in the 3DGS model by leveraging 3D Gaussian properties.
- A 3D contrastive mechanism that seeks tampering cues by analyzing the similarity of key attributes among 3D Gaussians.
- A cyclic optimization strategy to refine the 3D tampering attribute for precise localization.

Our GS-Checker operates without requiring label-intensive 3D annotations for supervision. We evaluate GS-Checker under various settings, and the results demonstrate its effectiveness.

2 Related Work

3D editing. In recent years, 3D technology has made significant progress (Liu et al. 2021; Xu and Harada 2022; Chen et al. 2024a; Zhang et al. 2024b; Song et al. 2025; Li and Cheung 2024, 2025). Some approaches (Bao et al. 2023; Gao et al. 2023; Wang et al. 2022a, 2023) leverage the CLIP model (Radford et al. 2021) to facilitate editing using text prompts or reference images. Moreover, some approaches (Chen et al. 2024a; Chen, Laina, and Vedaldi 2024; Wu et al. 2024) develop editing techniques for 3DGS models, which can effectively avoid the shortcomings of slow speed and limited control of NeRF-based methods. For instance, GaussianEditor (Chen et al. 2024a) leverages Gaussian semantic tracing and Hierarchical Gaussian Splatting (HGS) for precise and stable 3D editing, and designs a specialized 3D inpainting algorithm to streamline object removal and integration. With these methods, malicious users may easily use them to manipulate 3DGS models for negative applications. This motivates us to develop tamper localization techniques for 3DGS model, thus avoiding these 3D editing methods from being used for malicious purposes.

Tampering localization. Recently, AI security technologies (Zhang et al. 2024a; Song et al. 2024b; Huang et al. 2025; Song et al. 2024a; Luo et al. 2025; Huang et al. 2024) have made some progress. Early image forensic techniques primarily aim at particular types of manipulations (Islam et al. 2020; Li and Zhou 2018; Li and Huang 2019). Recently, some general tamper localization methods also endeavor to detect artifacts and anomalies within manipulated images (Dong et al. 2022; Guillaro et al. 2023; Ma et al. 2023; Zhang et al. 2025). For example, MVSS-Net (Dong et al. 2022) leverages multi-view feature learning and multi-scale supervision to simultaneously exploit boundary artifacts and the noise perspective of images. SAFIRE (Kwon et al. 2025) employs point prompting to segment forged image regions, allowing for the partitioning of images into multiple source regions and naturally focusing on the uniform characteristics within each region. IML-ViT (Ma et al. 2023) builds a ViT-based image manipulation localization model with high-resolution capacity, multi-scale feature extraction, and manipulation edge supervision. However, these methods are unable to achieve the tampering localization for 3DGS.

3 Preliminary

3D Gaussian Splatting. 3DGS (Kerbl et al. 2023) represents an advanced technique for modeling 3D scenes. Beginning with a sparse set of points derived from Structure-from-Motion (SfM) (Snavely, Seitz, and Szeliski 2006), the primary aim of 3DGS is to refine these points into a set of 3D Gaussians to achieve high-quality novel view synthesis. The scene is constructed as a collection of 3D Gaussians:

$$\mathcal{G}(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}. \quad (1)$$

In this equation, x represents any position in the 3D scene, μ is the mean position of the 3D Gaussian, and Σ is the covariance matrix of the 3D Gaussian. By utilizing a scaling matrix S and a rotation matrix R, the covariance matrix $\Sigma = RSS^TR^T$ can be derived, ensuring its positive semi-definiteness. These 3D Gaussians are then projected onto 2D Gaussians for rendering through volume splatting. During the rendering process, 3DGS employs a conventional neural

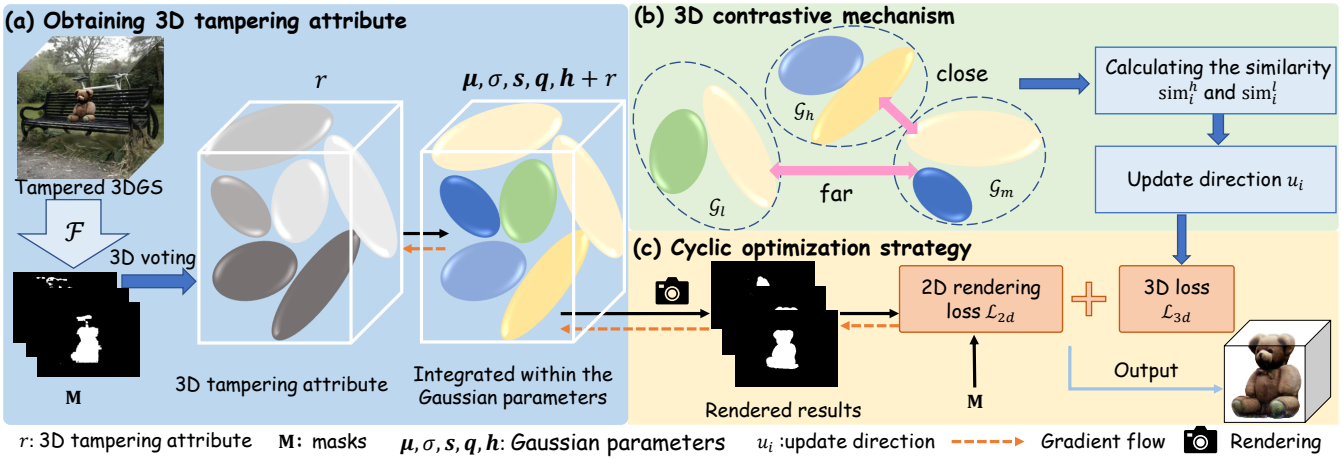

Figure 2: Illustration of our proposed method. First, the *3D tampering attribute* is integrated into the 3D Gaussian parameters and initialized via 3D voting. Next, a *3D contrastive mechanism* is introduced to seek tampering cues by comparing the similarity of key attributes between 3D Gaussians. Finally, a *cyclic optimization strategy* is employed to iteratively refine the tampering attribute by projecting it back into the 2D space, enabling joint optimization across both 2D and 3D levels.

point-based method (Kopanas et al. 2021, 2022) to calculate the pixel color C by blending N depth-ordered points:

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{2}$$

where c_i denotes the color estimated by the spherical harmonics (SH) coefficients of each Gaussian, and α_i is determined by evaluating a 2D Gaussian with covariance Σ', multiplied by a per-point opacity.

3DGS editing. Recent advancements have elevated 2D diffusion processes to 3D, extensively applying these methods in 3D editing. One way (Mikaeili et al. 2023; Poole et al. 2022; Sella et al. 2023) involves feeding the noised rendering of the current 3D model and other conditions into a 2D diffusion model (Rombach et al. 2022), using the generated scores to guide model updates. The other way (Raj et al. 2023; Shao et al. 2023; Chen et al. 2024b) focuses on 2D editing based on given prompts for multiview rendering of a 3D model, creating a multi-view 2D image dataset to guide the 3D model. GaussianEditor (Chen et al. 2024a) leverages the exemplary properties of 3DGS's explicit representation to enhance 3D editing, and employs the aforementioned guidance methods. It can be formulated as:

$$\mathcal{L}_{\text{edit}} = G(\Theta; p, e), \tag{3}$$

where G denotes the guidance, p represents the rendered camera pose, e represents the prompt, and Θ represents the 3DGS model parameters.

Our scenario. As shown in Figure 1, when a 3DGS model is created by its owner and publicly distributed over the web, malicious users can manipulate it by using advanced 3D editing tools and utilize it for some illegal applications. To prevent 3DGS model from being used for malicious purposes, the public need a method to detect such manipulation. Upon release of a tampered 3DGS model, the public can employ some approaches to locate which areas have been tampered, thus protecting the integrity and rightful ownership of the 3DGS model. To achieve this, we propose ***GS-Checker***, a method for locating tampered areas in the 3DGS model.

4 Proposed GS-Checker

In this section, we present GS-Checker in a comprehensive manner. As illustrated in Figure 2, our GS-Checker framework incorporates several key mechanisms to locate tampered areas in the 3DGS model. First, the tampering attribute is integrated into the parameters of each 3D Gaussian and initialized using a 3D voting method. Next, a 3D contrastive mechanism is introduced to assess the similarity of key attributes between 3D Gaussians, seeking tampering clues in 3D space. Furthermore, a cyclic optimization strategy is implemented to update the tampering attribute iteratively by rendering it into 2D. Our method can perform optimization at 2D and 3D levels jointly, resulting in accurate outcomes.

4.1 Obtaining 3D tampering attribute

In our approach, the attribute is initially obtained at 2D level and then project back to the 3D space to affiliate with each 3D Gaussian. Specifically, we directly feed rendered images from the 3DGS model into a pretrained 2D tampering localization network to obtain the masks as follows:

$$\mathbf{M}_j = \mathcal{F}(\mathcal{R}_j), \tag{4}$$

where $\mathcal{R}_j$ represents the rendered image of the 3DGS model in the j-th viewpoint, $\mathcal{F}$ represents the 2D tampering localization backbone, and $\mathbf{M}_j$ is the corresponding masks generated by the 2D model in the j-th viewpoint. With the masks, we project them back to the 3D space to initially get whether a specific 3D Gaussian is manipulated or not. We define that when the center point's projection falls within the tampered region indicated by the masks, the 3D vote is 1 in that viewpoint. If its projection is within the authentic region, the 3D vote is 0. If the projection lies outside the masks region, the

3D vote is -1, which can be considered as an abstention in that viewpoint. Specifically, it can be expressed as follows:

$$\mathbf{V}_{ij} = \mathbb{I}\left[\mu_i \mathbf{P}_j \in \mathbf{M}_j^+\right] + 0 \cdot \mathbb{I}\left[\mu_i \mathbf{P}_j \in \mathbf{M}_j^-\right] - \mathbb{I}\left[\mu_i \mathbf{P}_j \notin \mathbf{M}_j\right], \tag{5}$$

where $\mathbf{P}_j$ represents the projection matrix in the j-th viewpoint, μ_i represents the coordinates of the i-th 3D Gaussian center point. $\mathbf{M}_j+$ is the tampered region indicated by the masks, and $\mathbf{M}_j-$ is the authentic region. $\mathbf{V}_{ij}$ represents the value of the i-th row and j-th column element in the voting matrix $\mathbf{V}$, *i.e.*, it indicates whether the i-th 3D Gaussian center point belongs to the manipulated region in the j-th viewpoint.

Then, we determine whether each 3D Gaussian belongs to the tampered 3GDS area by calculating its total number of votes in different viewpoints, and the higher number of votes represents the higher probability of its being manipulated. Note that we do not count abstentions in this process. Specifically, it can be calculated as follows:

$$\mathbf{T}_i = \sum_{j=0}^{N-1} \mathbf{V}_{ij}, \text{ if } \mu_i \mathbf{P}_j \in \mathbf{M}_j, \tag{6}$$

where N represents the number of viewpoints, $\mathbf{T}_i$ represents the total number of votes for the i-th 3D Gaussian. Then, we treat the 3D Gaussians that receive the majority of votes as being tampered, which implies a consensus in the majority of viewpoints. This means, the number of viewpoints which consider the 3D Gaussian as tampered regions exceeds that of authentic regions or abstentions. Since the 2D tampering localization model does not consider the structure in 3D space, it may give inconsistent prediction results in different viewpoints. Therefore, with this 3D voting method, we can utilize the consistency of the 3D structure among different viewpoints to remove some incorrect results.

Finally, the 3D tampering attribute uses the total number of votes $\mathbf{T}_i$ after reaching consensus as the initial value. We integrate this tampering attribute into the parameters of the 3DGS model. Then, for the i-th 3D Gaussian, in addition to the mean position $\boldsymbol{\mu}_i$, opacity σ_i, scaling factor $\boldsymbol{s}_i$, rotation factor $\boldsymbol{q}_i$ and SH coefficients $\boldsymbol{h}_i$, the 3D tampering attribute $r_i \in \mathbb{R}^1$ is used to indicate whether this 3D Gaussian has been tampered.

4.2 3D contrastive mechanism

As shown in Figure 3, we compare the distributions of attribute values (*e.g.,* rotation factor) between tampered and authentic 3D Gaussians. Considering that tampered 3D Gaussians exhibit certain differences in parameter distributions compared with authentic Gaussians, we further leverage the properties of 3D Gaussian to facilitate the performance of tampering localization.

Therefore, we propose a 3D contrastive mechanism that increases or reduces the corresponding 3D tampering attribute value by comparing the similarity between Gaussian attributes. Specifically, we consider Gaussians with high 3D tampering attribute values $\mathcal{G}_h$ as tampered and Gaussians with very low values $\mathcal{G}_l$ as authentic. Then the main focus is

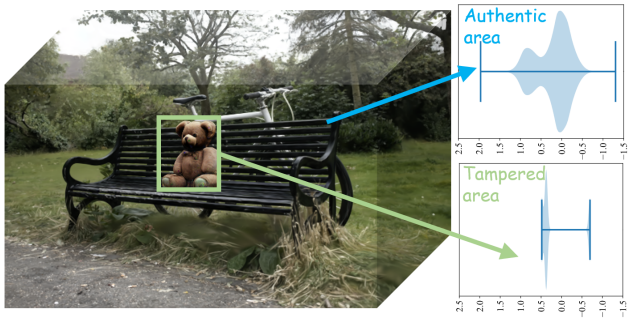

Figure 3: Statistical distribution of tampered and authentic 3D Gaussian properties. Noticeable anomalies are observed in the tampered regions, where parameter distributions deviate from those of the authentic parts. These discrepancies serve as key indicators of tampering, motivating our approach to seek tampering traces at the 3D level.

on those Gaussians with intermediate values $\mathcal{G}_m$. The similarity of Gaussian attributes between $\mathcal{G}_m$ and $\mathcal{G}_h$, $\mathcal{G}_l$ is compared to determine whether they are more likely to be tampered or authentic. If $\mathcal{G}_m$ is more similar to $\mathcal{G}_h$, the corresponding 3D tampering attribute value is increased, and on the contrary, its 3D tampering attribute value is decreased. The process of comparing Gaussian attributes can be represented as:

$$\begin{aligned} \text{sim}_i^h &= \|\mathbf{F}_i^m - \mathbf{F}^h\|_2^2, \\ \text{sim}_i^l &= \|\mathbf{F}_i^m - \mathbf{F}^l\|_2^2, \end{aligned} \tag{7}$$

where sim_i^h represents the similarity between i-th 3D Gaussian in $\mathcal{G}_m$ and $\mathcal{G}_h$, sim_i^l represents the corresponding similarity with $\mathcal{G}_l$. $\mathbf{F}_i^m$ represents i-th 3D Gaussian's attributes in $\mathcal{G}_m$. $\mathbf{F}^h$ and $\mathbf{F}^l$ indicate the average value of Gaussian attributes of $\mathcal{G}_h$ and $\mathcal{G}_l$, respectively. Here, we leverage the original attributes of each 3D Gaussian, including $\boldsymbol{\mu}_i$, σ_i, $\boldsymbol{s}_i$, $\boldsymbol{q}_i$ and $\boldsymbol{h}_i$. Then, the update direction of 3D tampering attributes can be expressed as:

$$u_i = \text{sign}\left(\Delta \text{sim}_i\right), \text{ where } \Delta \text{sim}_i = \text{sim}_i^l - \text{sim}_i^h, \tag{8}$$

where u_i represents the update direction of i-th 3D Gaussian's tampering attribute in $\mathcal{G}_m$, $\text{sign}(\cdot)$ indicates the sign function. Our 3D contrastive mechanism can be implemented through a loss function, and the detailed description of the formulation is presented in the next section.

4.3 Cyclic optimization strategy

After obtaining 3D tampering attributes, we introduce a cyclic optimization strategy to update them iteratively. Since the 3D tampering attribute has been integrated as a unique attribute into the 3DGS model parameters, we can render it into 2D by using the differentiable rasterization algorithm. Particularly, we optimize the 3D tampering attribute by calculating the difference between the rendered results and the masks generated by the 2D tampering localization backbone. The process of 3D tampering attribute rendering can

be expressed as follows:

$$\mathbf{M}^R(\mathbf{r}) = \sum_{i=1}^{G} r_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{9}$$

where r_i represents the 3D tampering attribute of the i-th 3D Gaussian, G represents the number of 3D Gaussians overlapping the ray $\mathbf{r}$, and $\mathbf{M}^R(\mathbf{r})$ represents the corresponding rendered results, and α_i is determined by evaluating a 2D Gaussian with covariance Σ', multiplied by a per-point opacity.

After obtaining the rendered results, we construct a loss function consisting of the rendered results and masks generated by the 2D model to perform the optimization. This allows us to update the 3D tampering attributes by using the gradient descent algorithm. Specifically, our goal is to increase the 3D tampering attribute value of the manipulated area and decrease the value of the authentic area. Thus, considering that the manipulated area is 1 and authentic area is 0 in the masks, the 2D rendering loss function can be calculated as follows:

$$\begin{aligned}\mathcal{L}_{2d} = &-\lambda_1 \sum_{\mathbf{r}\in \mathrm{R}(\mathbf{I}_j)} \mathbf{M}_j(\mathbf{r})\mathbf{M}^R(\mathbf{r}) \\ &+ \lambda_2 \sum_{\mathbf{r}\in \mathrm{R}(\mathbf{I}_j)} \big(1-\mathbf{M}_j(\mathbf{r})\big)\mathbf{M}^R(\mathbf{r}),\end{aligned} \tag{10}$$

where λ_1 and λ_2 are hyperparameters that balance different loss terms, $\mathrm{R}(\mathbf{I}_j)$ indicates the set of rays for image $\mathbf{I}_j$ in the j-th viewpoint. In this process, we only optimize the 3D tampering attributes without affecting other parameters of the 3D Gaussian model.

In our optimization process, we first use 2D rendering loss to optimize the 3D tampering attribute score. After a certain number of iterations, the difference of 3D tampering attribute scores between $\mathcal{G}_h$ and $\mathcal{G}_l$ becomes larger to facilitate the distinction. Then, we use both 2D rendering loss and 3D contrastive mechanism for optimization. The loss function at this point can be defined as:

$$\mathcal{L}_{cyc} = \beta\mathcal{L}_{2d} + \gamma\mathcal{L}_{3d}, \tag{11}$$

where β and γ denote hyperparameters that balance 2D rendering loss and 3D loss $\mathcal{L}_{3d}$. The loss function of 3D contrastive mechanism $\mathcal{L}_{3d}$ can be calculated as follows:

$$\mathcal{L}_{3d} = -\frac{1}{K}\sum_{i\in K} u_i r_i, \tag{12}$$

where K is the number of 3D Gaussians in $\mathcal{G}_m$. By optimizing this loss function, we can increase or decrease the corresponding 3D tampering attribute scores. After several iterations, we can obtain more accurate results for localizing the manipulated areas in the 3DGS model.

4.4 Implementation details

We implement our method using PyTorch. As our goal is to locate the tampered areas in 3DGS, we first manipulate the 3DGS models trained in standard settings by using the 3DGS editing methods (*i.e.*, GaussianEditor (Chen et al. 2024a) and Gaussian Grouping (Ye et al. 2024)). Next, the rendered images of these scenes are fed into the 2D tampering localization backbone (Kwon et al. 2025) to obtain the corresponding masks, where the confidence score threshold is set to 0.5. After obtaining the masks, we employ the 3D voting method to project them back to the 3D space and integrate within each 3D Gaussian. Then, our approach can update the 3D tampering attributes iteratively, which leverages the 2D rendering loss and 3D contrastive mechanism to perform optimization at both 2D and 3D levels. During the optimization process, the weights of the different loss functions in Equation (10) and Equation (11) are set as $\lambda_1 = 1.0$, $\lambda_2 = 10.0$, $\beta = 1.0$ and $\gamma = 10.0$, and the learning rate for optimizing the 3D tampering attribute values is set to 1.0. We choose the Adam optimizer to update the tampering attributes. The threshold of the 3D tampering attributes after normalization is set to 0.1. All of our experiments are performed on a single NVIDIA Tesla V100 GPU.

5 Experiments

5.1 Experimental settings

Dataset. In order to evaluate the effectiveness of our proposed method, we construct a 3DGS manipulation dataset containing multiple scenes and manually annotated mask labels. Specifically, we use GaussianEditor (Chen et al. 2024a) and Gaussian Grouping (Ye et al. 2024), two latest and effective 3DGS editing methods, to edit 3DGS models trained on Mip-NeRF360 (Barron et al. 2022) and InstructNeRF2NeRF (Haque et al. 2023) datasets. All these 3DGS models are trained under standard settings (Kerbl et al. 2023). The dataset includes three types of tampering: **1) Object incorporation:** incorporating objects at a certain location in 3D scenes; **2) Object modification:** editing the properties of objects in 3D scenes, such as color, shape and so on; **3) Object removal:** removing some objects from the 3D scenes and filling the holes generated at the interface. In total, our dataset comprises 11 tampered 3DGS models, strictly following the scenes number of 3DGS editing works.
Baselines. To the best of our knowledge, there is no method specifically for the 3DGS tampering localization. Therefore, we compare with three strategies to guarantee a fair comparison: **1) SAFIRE (Kwon et al. 2025)+SA3D (Cen et al. 2025):** locating 3D tampered regions from the results of 2D tampering localization model (Kwon et al. 2025) with a state-of-the-art 3DGS segmentation method SA3D (Cen et al. 2025); **2) SAFIRE (Kwon et al. 2025)+SAGD (Hu et al. 2024):** locating 3D tampered regions from the results of 2D tampering localization model (Kwon et al. 2025) with a 3DGS segmentation method SAGD (Hu et al. 2024); **3) Inverse with SAFIRE (Kwon et al. 2025):** directly project the results generated by the 2D tampering localization model (Kwon et al. 2025) back to 3D Gaussians.
Evaluation methodology. We evaluate the proposed method as well as the baseline methods using the standard manipulation localization metrics (Ma et al. 2024). Specifically, we use the average value of F_1 score and IoU between the rendered results and ground-truth masks of different viewpoints to evaluate the performance of different

Method	Object incorporation		Object modification		Object removal	
	F1↑	IoU↑	F1↑	IoU↑	F1↑	IoU↑
Inverse with SAFIRE (Kwon et al. 2025)	0.1191	0.0647	0.2743	0.1614	0.3896	0.2435
SAFIRE (Kwon et al. 2025)+SA3D (Cen et al. 2025)	0.1576	0.0915	0.8348	0.7168	0.0177	0.0090
SAFIRE (Kwon et al. 2025)+SAGD (Hu et al. 2024)	0.4671	0.3880	0.6230	0.4557	0.4866	0.3223
Proposed GS-Checker	**0.9507**	**0.9081**	**0.9017**	**0.8232**	**0.7812**	**0.6433**

Table 1: Comparison of quantitative tampering localization performance with baseline methods. The best results are in **bold**.

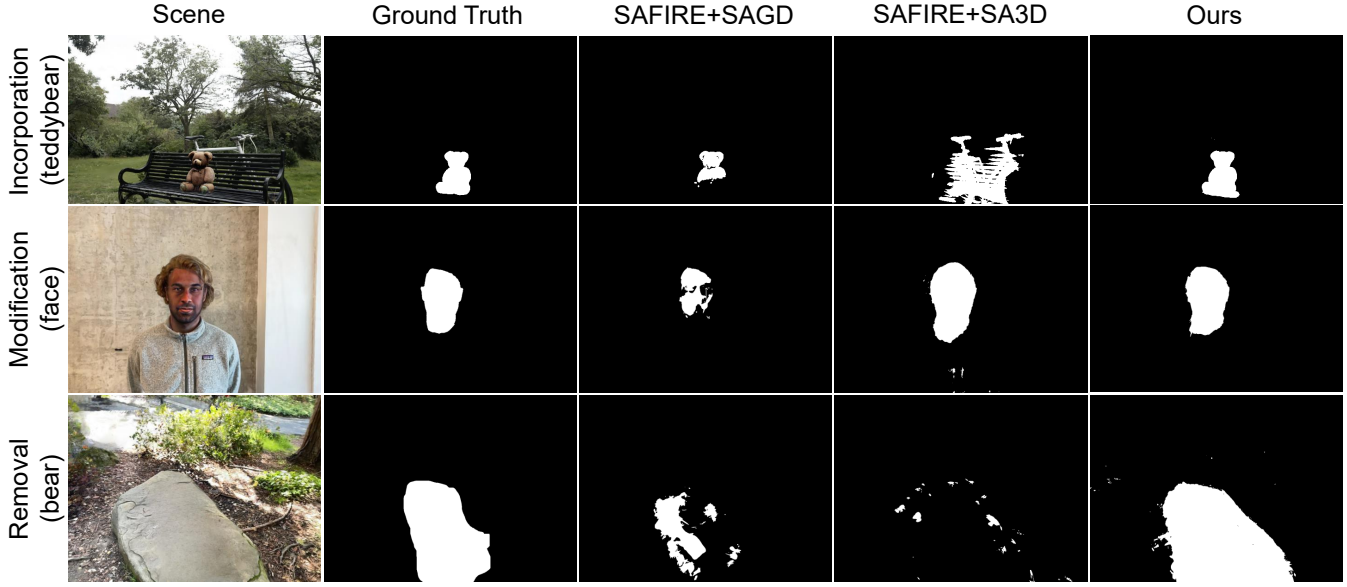

Figure 4: Qualitative results of 3DGS tampering localization in different scenes. Columns from left to right are: rendered images of tampered 3D scenes, ground-truth, SAFIRE (Kwon et al. 2025)+SAGD (Hu et al. 2024), SAFIRE (Kwon et al. 2025)+SA3D (Cen et al. 2025) and ours.

methods. In addition to the normal situations, we also verify whether the proposed method can retain the 3D tampering localization performance against different distortions, including 2D Gaussian noise, Gaussian blur, 3D scale noise and opacity noise.

5.2 Experimental results

Quantitative results. We compare the tampering localization performance of our method with all baselines on three tampering types, and the results are presented in Table 1. It can be observed that our method achieves the best tampering localization performance in different tampering types. This proves that our method can efficiently and accurately locate the tampered areas in 3D scenes. Comparatively, other methods have lower performance and often struggle to locate the tampered areas in these 3D scenes. This discrepancy underscores the challenges associated with directly applying traditional 3DGS segmentation methods or attempting to project masks back to 3D Gaussians. These approaches often fail to identify tampering traces at the 3D level. In contrast, our proposed method can be well applied to the 3DGS tampering localization task, which demonstrates the effectiveness of our method for 3DGS tampering localization.

Distortion	FIRE+3D		Ours	
	F1↑	IoU↑	F1↑	IoU↑
None	0.8348	0.7168	0.9017	0.8232
Gauss. noise	0.4806	0.3176	0.8903	0.8046
Gauss. blur	0.8088	0.6795	0.8568	0.7532
scale noise	0.8302	0.7100	0.8801	0.7877
opacity noise	0.8129	0.6851	0.8959	0.8140

Table 2: Quantitative tampering localization performance of our method and SAFIRE (Kwon et al. 2025)+SA3D (Cen et al. 2025) (FIRE+3D) under different distortions. "None" indicates that no distortion has been applied.

Qualitative results. We visualize the qualitative results of 3DGS tampering localization in different scenes, and the results are shown in Figure 4. It can be observed that our proposed method can locate the tampered areas effectively across different 3D scenes. This demonstrates the effectiveness of our method. In comparison, the other methods are relatively inferior in localizing the tampered areas.

Robustness evaluation. To verify the robustness of the pro-

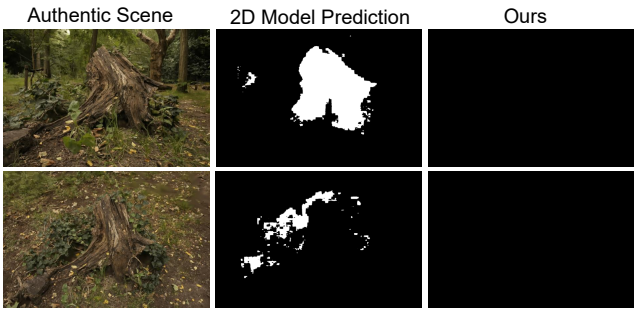

Figure 5: Results on the authentic scene *stump*.

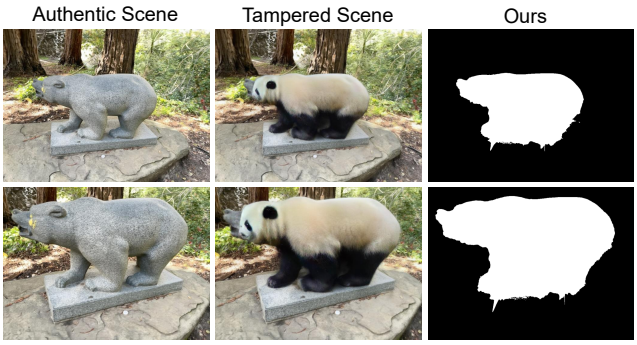

Figure 6: Qualitative tampering localization results on DGE (Chen, Laina, and Vedaldi 2024) editing method.

posed method, we perform 3DGS tampering localization experiments under different distortions, including 2D Gaussian noise and Gaussian blur. We also verify the tampering localization performance in the case of adding noise to 3D Gaussian parameters, including adding noise to the scaling parameters as well as the opacity parameters of the 3DGS model. The results of the robustness evaluation experiments are shown in Table 2. It can be observed that our method can still maintain a relatively accurate 3D tampering localization performance under different distortions, thus avoiding being misguided to produce false results.

Results on authentic scene. We conduct experiments on the authentic scene to verify whether our proposed method would incorrectly detect the authentic scene as tampered or not. The experimental results are shown in Figure 5. It can be observed that the 2D tampering localization model incorrectly predicts some areas of the authentic scene as tampered, while our method can remove these erroneously localized areas. This proves the effectiveness of our method and the credibility of its prediction results.

Results on different 3DGS editing methods. In addition to GaussianEditor (Chen et al. 2024a) and Gaussian Grouping (Ye et al. 2024), we also evaluate the performance of our method on other editing methods, such as DGE (Chen, Laina, and Vedaldi 2024). The experimental results are presented in Figure 6. It can be found that our method has excellent tampering localization performance on different 3DGS editing methods. This demonstrates the effectiveness and generalization of our method.

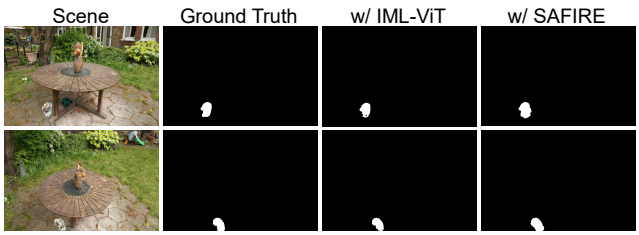

Figure 7: Qualitative tampering localization results on different 2D models. Our method is flexible and capable of integrating with different 2D tampering localization models to ensure accurate 3DGS tampering localization.

Case	Method	F1↑	IoU↑
(a)	w/o cyclic optimization	0.8485	0.7384
(b)	w/o 3D contrastive	0.8788	0.7847
(c)	Our full method	**0.8842**	**0.7934**

Table 3: Comparisons for our full method, our method without cyclic optimization strategy and our method without 3D contrastive mechanism.

Different 2D tampering localization models. We conduct experiments on different 2D tampering localization models as well, including IML-ViT (Ma et al. 2023) and SAFIRE (Kwon et al. 2025). The experimental results are presented in Figure 7. It can be observed that our approach is adaptable and can incorporate various 2D tampering localization models to achieve precise 3DGS tampering localization. This flexibility enhances the robustness and effectiveness of our proposed method.

Ablation study. In this section, we further perform ablation studies to verify the effectiveness of 3D contrastive mechanism and cyclic optimization strategy. Table 3 shows the results of different component combinations for locating 3DGS tampered areas. It can be observed that each component has an improvement on the performance of the 3DGS tampering localization, which demonstrates the effectiveness of each component.

6 Conclusion

In this paper, we introduce ***GS-Checker***, a novel approach to locate the tampered areas of 3D Gaussian Splatting (3DGS) models. By integrating a 3D tampering attribute into 3D Gaussian parameters, we can exploit the properties of the 3DGS model for capturing tampering information effectively. Additionally, a 3D contrastive mechanism is employed to identify tampering cues by comparing the similarity of key attributes among 3D Gaussians. We further adopt a cyclic optimization strategy to iteratively refine the tampering attribute, achieving precise localization results. Extensive experimental results show that our method can achieve accurate 3DGS tampering localization performance and be robust to different distortions. Thus, it can effectively prevent 3DGS models from being used by malicious users for negative applications.

Acknowledgements

Renjie Group is supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2024A1515012822, and the Research Grant Council (RGC) of the Hong Kong SAR, under a GRF Grant 12203124 and a ECS Grant 22201125. We also thank the support of the São Paulo Research Foundation (Fapesp) through the Horus Project #2023/12865-8 and the Brazilian National Council for Scientific and Technological Research (CNPq) through complementary grants.

References

Bao, C.; Zhang, Y.; Yang, B.; Fan, T.; Yang, Z.; Bao, H.; Zhang, G.; and Cui, Z. 2023. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and 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*.

Cen, J.; Fang, J.; Zhou, Z.; Yang, C.; Xie, L.; Zhang, X.; Shen, W.; and Tian, Q. 2025. Segment anything in 3d with radiance fields. *International Journal of Computer Vision*.

Chen, M.; Laina, I.; and Vedaldi, A. 2024. DGE: Direct gaussian 3D editing by consistent multi-view editing. In *European Conference on Computer Vision*.

Chen, Y.; Chen, Z.; Zhang, C.; Wang, F.; Yang, X.; Wang, Y.; Cai, Z.; Yang, L.; Liu, H.; and Lin, G. 2024a. GaussianEditor: Swift and controllable 3D editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chen, Y.; Zhang, C.; Yang, X.; Cai, Z.; Yu, G.; Yang, L.; and Lin, G. 2024b. IT3D: Improved text-to-3D generation with explicit view synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dong, C.; Chen, X.; Hu, R.; Cao, J.; and Li, X. 2022. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gao, W.; Aigerman, N.; Groueix, T.; Kim, V.; and Hanocka, R. 2023. Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Guillaro, F.; Cozzolino, D.; Sud, A.; Dufour, N.; and Verdoliva, L. 2023. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Guo, X.; Liu, X.; Ren, Z.; Grosz, S.; Masi, I.; and Liu, X. 2023. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Haque, A.; Tancik, M.; Efros, A. A.; Holynski, A.; and Kanazawa, A. 2023. Instruct-NeRF2NeRF: Editing 3D scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Hu, X.; Wang, Y.; Fan, L.; Fan, J.; Peng, J.; Lei, Z.; Li, Q.; and Zhang, Z. 2024. SAGD: Boundary-enhanced segment anything in 3D Gaussian via Gaussian decomposition. arXiv:2401.17857.

Huang, X.; Li, R.; Cheung, Y.-m.; Cheung, K. C.; See, S.; and Wan, R. 2024. GaussianMarker: Uncertainty-aware copyright protection of 3D gaussian splatting. *Advances in Neural Information Processing Systems*.

Huang, X.; Luo, Z.; Song, Q.; Wang, R.; and Wan, R. 2025. MarkSplatter: Generalizable watermarking for 3D gaussian splatting model via splatter image structure. In *Proceedings of the 33rd ACM International Conference on Multimedia*.

Islam, A.; Long, C.; Basharat, A.; and Hoogs, A. 2020. DOA-GAN: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*.

Kopanas, G.; Leimkühler, T.; Rainer, G.; Jambon, C.; and Drettakis, G. 2022. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics*.

Kopanas, G.; Philip, J.; Leimkühler, T.; and Drettakis, G. 2021. Point-Based Neural Rendering with Per-View Optimization. In *Computer Graphics Forum*.

Kwon, M.-J.; Lee, W.; Nam, S.-H.; Son, M.; and Kim, C. 2025. SAFIRE: Segment any forged image region. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Li, H.; and Huang, J. 2019. Localization of deep inpainting using high-pass fully convolutional network. In *proceedings of the IEEE/CVF international conference on computer vision*.

Li, R.; and Cheung, Y.-m. 2024. Variational multi-scale representation for estimating uncertainty in 3D gaussian splatting. *Advances in Neural Information Processing Systems*.

Li, R.; and Cheung, Y.-m. 2025. Modeling and Identifying Distractors with Curriculum for Robust 3D Gaussian Splatting. In *Proceedings of the 33rd ACM International Conference on Multimedia*.

Li, Y.; and Zhou, J. 2018. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Transactions on Information Forensics and Security*.

Liu, S.; Zhang, X.; Zhang, Z.; Zhang, R.; Zhu, J.-Y.; and Russell, B. 2021. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Luo, Z.; Rocha, A.; Shi, B.; Guo, Q.; Li, H.; and Wan, R. 2025. The NeRF signature: Codebook-aided watermarking for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ma, X.; Du, B.; Jiang, Z.; Hammadi, A. Y. A.; and Zhou, J. 2023. IML-ViT: Benchmarking image manipulation localization by vision transformer. arXiv:2307.14863.

Ma, X.; Zhu, X.; Su, L.; Du, B.; Jiang, Z.; Tong, B.; Lei, Z.; Yang, X.; Pun, C.-M.; Lv, J.; et al. 2024. IMDL-BenCo: A Comprehensive Benchmark and Codebase for Image Manipulation Detection & Localization. In *Advances in Neural Information Processing Systems*.

Mikaeili, A.; Perel, O.; Safaee, M.; Cohen-Or, D.; and Mahdavi-Amiri, A. 2023. Sked: Sketch-guided text-based 3D editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3D using 2D diffusion. In *International Conference on Learning Representations*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*.

Raj, A.; Kaza, S.; Poole, B.; Niemeyer, M.; Ruiz, N.; Mildenhall, B.; Zada, S.; Aberman, K.; Rubinstein, M.; Barron, J.; et al. 2023. Dreambooth3D: Subject-driven text-to-3D generation. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Sella, E.; Fiebelman, G.; Hedman, P.; and Averbuch-Elor, H. 2023. Vox-e: Text-guided voxel editing of 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Shao, R.; Sun, J.; Peng, C.; Zheng, Z.; Zhou, B.; Zhang, H.; and Liu, Y. 2023. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. arXiv:2305.20082.

Snavely, N.; Seitz, S. M.; and Szeliski, R. 2006. Photo tourism: exploring photo collections in 3D. In *ACM Transactions on Graphics*.

Song, Q.; Luo, Z.; Cheung, K. C.; See, S.; and Wan, R. 2024a. Geometry cloak: Preventing tgs-based 3D reconstruction from copyrighted images. *Advances in Neural Information Processing Systems*.

Song, Q.; Luo, Z.; Cheung, K. C.; See, S.; and Wan, R. 2024b. Protecting NeRFs' copyright via plug-and-play watermarking base model. In *European Conference on Computer Vision*.

Song, Q.; Luo, Z.; Cheung, K. C.; See, S.; and Wan, R. 2025. Align 3D Representation and Text Embedding for 3D Content Personalization. In *Proceedings of the 33rd ACM International Conference on Multimedia*.

Wang, C.; Chai, M.; He, M.; Chen, D.; and Liao, J. 2022a. Clip-NeRF: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wang, C.; Jiang, R.; Chai, M.; He, M.; Chen, D.; and Liao, J. 2023. NeRF-Art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*.

Wang, J.; Wu, Z.; Chen, J.; Han, X.; Shrivastava, A.; Lim, S.-N.; and Jiang, Y.-G. 2022b. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wu, J.; Bian, J.-W.; Li, X.; Wang, G.; Reid, I.; Torr, P.; and Prisacariu, V. A. 2024. Gaussctrl: Multi-view consistent text-driven 3D gaussian splatting editing. In *European Conference on Computer Vision*.

Xu, T.; and Harada, T. 2022. Deforming radiance fields with cages. In *European Conference on Computer Vision*.

Ye, M.; Danelljan, M.; Yu, F.; and Ke, L. 2024. Gaussian Grouping: Segment and edit anything in 3D scenes. In *European Conference on Computer Vision*.

Zhang, X.; Li, R.; Yu, J.; Xu, Y.; Li, W.; and Zhang, J. 2024a. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhang, X.; Meng, J.; Li, R.; Xu, Z.; Zhang, Y.; and Zhang, J. 2024b. GS-Hider: Hiding messages into 3D gaussian splatting. *Advances in Neural Information Processing Systems*.

Zhang, X.; Tang, Z.; Xu, Z.; Li, R.; Xu, Y.; Chen, B.; Gao, F.; and Zhang, J. 2025. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.