This is the peer reviewd version of the followng article:

Fake3DGS: A Benchmark for 3D Manipulation Detection in Neural Rendering / Nucci, D.D., Catalini, R., Borghi, G., Vezzani, R.. - 16816:(2026), pp. 1-15. (28th International Conference on Pattern Recognition, ICPR 2026 Lyon, France 2026) [10.1007/978-3-032-31666-0_1].

Springer Science and Business Media Deutschland GmbH
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

31/10/2026 23:40

(Article begins on next page)

Fake3DGS: A Benchmark for 3D Manipulation Detection in Neural Rendering

Davide Di Nucci, Riccardo Catalini, Guido Borghi, and Roberto Vezzani

University of Modena and Reggio Emilia, Via Pietro Vivarelli, 41121 Modena, Italy
name.surname@unimore.it

Abstract. Recent advances in 3D reconstruction and neural rendering, particularly 3D Gaussian Splatting, make it feasible and simple to edit 3D scenes and re-render them as highly realistic images. Therefore, security concerns arise regarding the authenticity of 3D content. Despite this threat, 3D fake detection remains largely unexplored in the literature, and most existing work is limited to 2D space. Therefore, in this paper, we formalize the concept of 3D fake detection and introduce Fake3DGS, a dataset of 3D Gaussian splatting scenes and corresponding rendered views, where fake images are produced by controlled manipulations of geometry, appearance, and spatial layout, while preserving high visual realism. Using this benchmark, we demonstrate that current state-of-the-art 2D detectors struggle to distinguish between original and 3D manipulated images. To bridge this gap, we introduce a 3D-aware detection method that leverages multi-view coherence and features derived from the Gaussian splatting representation. Experimental results demonstrate a substantial improvement in recognizing modified 3D content, underscoring the validity of the new dataset and the necessity for authenticity assessment techniques that extend beyond 2D evidence. Code and data are publicly released[1] for future investigations.

Keywords: Gaussian Splatting · 3D Fake Detection · 3D reconstruction.

1 Introduction

With the widespread adoption of Generative AI, 2D deep fake detection has become a well-established task in computer vision, with numerous datasets and benchmarks available [56,51,2,14]. However, generative methods are rapidly evolving beyond the two-dimensional domain. Recent developments integrate models such as Variational Autoencoders (VAE) [30], Generative Adversarial Networks (GANs) [20], and Diffusion Models [44] to improve 3D reconstruction fidelity and reduce inference time. For instance, modern text-to-3D pipelines rely on text-to-image generation followed by 3D synthesis to ensure both semantic consistency and geometric accuracy [54,35]. Other approaches enable complex edits on real 3D scenes [22,52].

[1] https://github.com/iot-unimore/Fake3DGS

Unlike 2D manipulations [3], edits performed directly in the 3D scene representation can produce an unlimited number of photorealistic, view-consistent images, making traditional artifact-based detection strategies limited in efficacy. This shift introduces a new class of deep fakes where geometric consistency becomes an advantage for the attacker rather than a vulnerability. The ability to edit 3D scenes is increasingly deployed in Augmented Reality or Virtual Reality environments, digital twins, and simulation pipelines for autonomous systems [28,60,61]. These elements make the authenticity of 3D content not only a scientific challenge but a security concern. Indeed, despite these progresses and risks, 3D fake detection remains largely unexplored in the literature. Also, while 2D deep fake detection benefits from mature benchmarks and shared evaluation protocols, the 3D domain still lacks definitions, datasets, and standardized procedures for assessing scene authenticity.

Motivated by these reasons, we formalize 3D fake detection as the task of determining whether a 3D scene representation has been altered in its geometry, spatial layout, or appearance parameters, regardless of how realistic its rendered images appear. We establish the first benchmark for authenticity assessment in 3D Gaussian Splatting [29], laying the foundations for future research on multi-view consistency and 3D-aware forensics.

In summary, the novel contributions of this work are the following:

- The 3D deep fake detection task is formalized and defined.
- A new dataset, called Fake3DGS has been generated, annotated and published to allow fair comparisons and to benchmark current and future techniques; the dataset contains original and modified 3D scenes directly stored as 3D Gaussian Splatting representations.
- A specific detector is proposed and compared with common solutions for fake image detection, showing promising performances and, at the same time, demonstrating the need of specific techniques to solve the task.

2 Related work

2.1 3D Novel View Synthesis

Novel view synthesis focuses on producing photorealistic images of a 3D scene from previously unseen camera viewpoints. While early methods based on Neural Radiance Fields (NeRF) [37] demonstrated that continuous radiance fields with differentiable volume rendering can achieve high-quality results, subsequent works mainly improved efficiency, *e.g.*, via multiresolution hash encodings [39] or explicit voxel grids storing density and spherical harmonics [19].

3D Gaussian Splatting [29] has emerged as a compelling alternative to NeRF, offering significant gains in both training and rendering speed while preserving high image quality [15]. To mitigate its remaining limitations, a rapidly growing body of work explores complementary research directions. One line of work targets compactness and deployment on resource-constrained hardware by

compressing the Gaussian representation to reduce storage and memory overhead [38,40,58]. Another focuses on text-to-3D scene generation has quickly emerged as a prominent application, combining 3D Gaussian Splatting with powerful 2D diffusion priors [59,9,33,32,48]. In parallel, 3D scene decomposition and segmentation to enable fine-grained editing operations [7,23,8,32].

We further observe that realistically edited 3D Gaussian scenes are scarce, and there is currently no standardized way to evaluate editing methods in this setting. To fill this gap, we introduce a dedicated benchmark for 3D Gaussian scene editing, enabling systematic and reproducible comparison across existing and future approaches.

2.2 Security and manipulation for 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [29] introduces a new security threat model: an adversary can directly edit Gaussian parameters to produce view-consistent yet deceptive renderings, enabling malicious scene manipulations that may not leave obvious 2D artifacts. Recent work has analyzed this attack surface and demonstrated both training-time and post-training manipulations of 3DGS models [25].

A complementary defense direction focuses on tamper-evidence and provenance via watermarking of 3DGS assets, embedding signals in the model parameters and/or rendered outputs to enable later verification under distortions and model transformations [10,27,24,26]. In contrast to watermark-based verification (active protection), our work addresses *passive* 3D manipulation detection under realistic editing pipelines by introducing a large-scale benchmark and a scene-level detector operating on Gaussian-derived features.

2.3 Fake detection

Fake image detection has rapidly evolved into a key research topic in recent years. Early efforts [45,49,4] concentrated mainly on identifying synthetic human faces and reverse-engineering the fingerprints of specific GAN generators. A complementary line of work explicitly targets perceptible artifacts, such as facial asymmetries, geometric inconsistencies, or implausible shadows and lighting [16,17,36]. However, as image synthesis models have advanced, these obvious defects are quickly reduced and now rarely appear in state-of-the-art generative pipelines, which limits the effectiveness of methods that rely solely on visible cues. Building on the GAN-based setting, subsequent studies [13,21,50] have examined the zero-shot generalization ability of detectors when confronted with previously unseen generators, showing that different GAN architectures often share exploitable common artifacts. In parallel, Frank et al. [18] investigated the frequency domain, revealing systematic spectral discrepancies between real and generated images.

The advent of Diffusion Models [44] has shifted the focus of the community. Although diffusion-based generators exhibit their own characteristic generation signatures [11], recent work [12] has demonstrated that detectors trained exclusively on GAN outputs transfer poorly to these newer models. This observation

has spurred a wave of approaches designed specifically for diffusion-generated content [41,46], many of which build on CLIP [43] as the underlying feature space. A distinct strategy is proposed by Wang et al. [51], who detect forgeries by analyzing the difference between an input image and its reconstruction obtained from a pre-trained diffusion model.

However, all the above methods operate purely in 2D space, implicitly assuming that fake content can be revealed by pixel- or texture-level artifacts alone. This assumption breaks down in the presence of advanced 3D pipelines such as 3D Gaussian Splatting, which enable realistic edits to geometry, appearance, and layout. As we show in our experiments, state-of-the-art deepfake detectors struggle to distinguish original from 3D-edited views in this setting. To address this gap, we introduce Fake3DGS, a specific dataset of edited 3D Gaussian scenes for benchmarking image authenticity in a 3D manipulation context.

3 Fake3DGS Dataset

To evaluate 3D fake detection in a controlled yet realistic setting, we introduce Fake3DGS dataset, a large-scale benchmark of original and edited 3D Gaussian Splatting scenes. The dataset comprises more than 41k reconstructed scenes, evenly balanced between real and manipulated samples.

We export all reconstructions as `nerfstudio`[2] checkpoints, which provides a widely adopted and well-documented format for neural rendering research, making the dataset easier to use and reproduce with standard tooling. We then post-process each 3DGS model using the Gaussian splatting compression strategy of Self-Organizing Gaussian Grids (SOGS) [38], leveraging the `gsplat`[3] implementation [55] that is integrated in the `nerfstudio` ecosystem. Concretely, we independently quantize Gaussian parameters with uniform scalar quantization (8-bit per parameter in our implementation). The quantized tensors are then reorganized into locally coherent 2D grids (via sorting) and encoded with lossless PNG compression, enabling exact recovery of the quantized values at load time. This post-processing substantially reduces storage and I/O overhead, shrinking the dataset footprint from approximately 7 TB to about 200 GB, which makes large-scale training and evaluation feasible on standard research hardware while preserving faithful reconstruction of the compressed representation.

Figure 1 shows representative examples from Fake3DGS dataset. For each underlying scene we report a real rendering and its edited counterpart generated with either GaussCtrl [52] or Instruct-GS2GS [48] (see Sect. 3.2). The edits cover the three instruction families used in our benchmark background/surface changes (*e.g.*, grass → snow), object appearance changes (*e.g.*, color), and object type/material substitutions (*e.g.*, sheep → giraffe) while keeping the overall scene visually plausible. Captions under each image indicate the text prompt used to produce the corresponding edited sample, highlighting that the result-

[2] https://github.com/nerfstudio-project/nerfstudio
[3] https://github.com/nerfstudio-project/gsplat

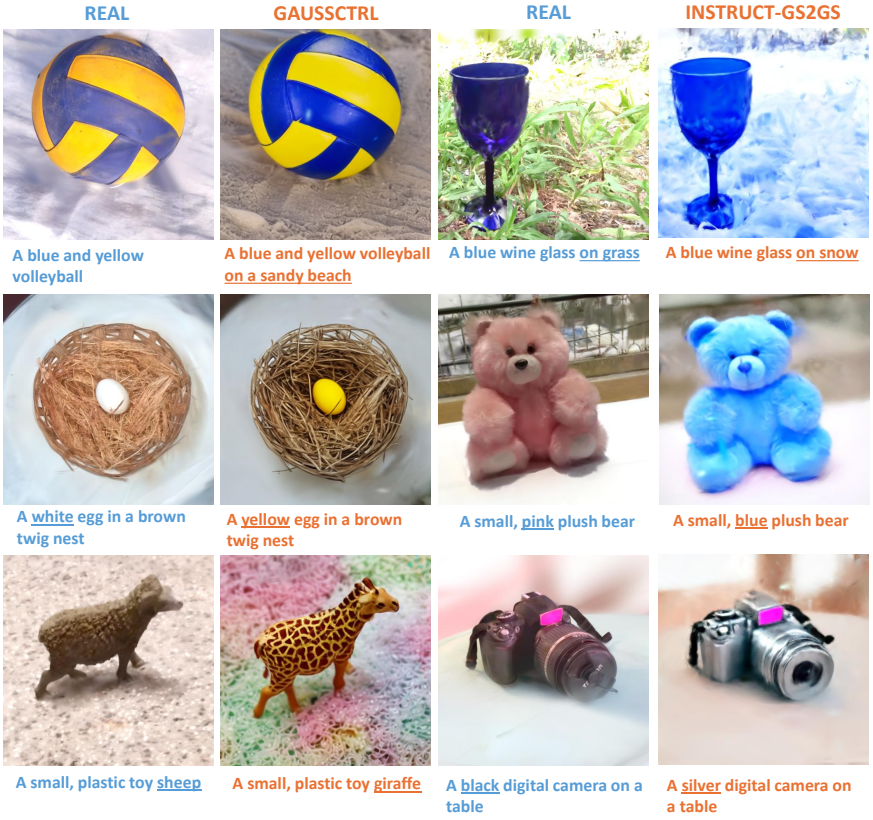

Fig. 1. Sample renderings from the Fake3DGS Dataset. Below each view of the scene, the corresponding prompt used to generate edited samples is reported.

ing manipulations are often photorealistic and view consistent, making detection based solely on 2D artifacts challenging.

3.1 Real 3D Scenes

Building a real-world benchmark for our task requires processing a large number of scenes, each with multi-view imagery, accurate camera poses, and a stable 3D scene representation for evaluation. Collecting and reconstructing such data from scratch would be prohibitively time-consuming and would introduce additional variability due to the reconstruction pipeline itself (*e.g.*, pose estimation failures, incomplete coverage, or inconsistent scene quality). For this reason, we started from UCO3D [34], an existing large-scale dataset that already provided these prerequisites in a consistent format.

UCO3D contains a wide variety of objects, spanning more than 1000 categories, captured in diverse indoor and outdoor environments, and additionally

provides pretrained 3D Gaussian Splatting reconstructions. Leveraging these ready-to-use reconstructions allows scaling the dataset construction to many scenes while keeping reconstruction quality consistent across categories, so that our evaluation primarily reflects the behavior of the editing method rather than artifacts of the reconstruction process.

To avoid biasing the benchmark toward categories with many instances, we balance the subset across categories. Specifically, for each category we select

$$n = \min_{c} \texttt{num_objects}(c),$$

i.e., the minimum number of available instances across categories, and sample n objects per category.

3.2 Fake 3D Scenes Generation

As mentioned, we employed two different methods to edit real scenes.

The first one is GaussCtrl [52], which edits rendered images from 3D Gaussian Splatting and recreates the edited scene through depth-conditioned editing based on ControlNet [57] for geometry consistency and attention-based latent code alignment for improving consistency during editing.

The second one is Instruct-GS2GS [48] which uses InstructPix2Pix [5] to iteratively edit the input images while optimizing the underlying scene, resulting in an optimized 3D scene edited accordingly to the instruction.

To construct our dataset, edited captions were automatically generated using an LLM, specifically the Meta-Llama-3-8B-Instruct [1]. We chose this model because it is open-source and can be deployed locally, allowing full control over generation parameters, reproducibility of the editing process, and large-scale caption generation without reliance on external APIs [6]. For each original caption, we create one edited caption by the following procedure: (i) we randomly sample one prompt template from the three defined below (the major differences between them are highlighted in italic); (ii) we append a fixed suffix to the input caption with the goal to enforce the output format as a single caption only. (iii) The resulting full prompt, obtained by concatenating the prompt, the caption, and the suffix, is then fed to the LLM, and its output is used as the edited caption. For clarity and replicability, we report the prompts and the suffix used.

> **Prompt 1**: Modify the following sentence by changing *the material or the type of the main object*, but do not change *the color, the background, or the shape.*
>
> **Prompt 2**: Modify the following sentence by changing *the background or surface on which the main object stands*, but do not change *the color, the shape, or any attribute of the main object.*
>
> **Prompt 3**: Modify the following sentence by changing *the color of the main object*, but do not change *the shape or any other attribute of the main object.*

Suffix: The output should be a single caption without any additional explanation or text.

Three edited captions were generated for each scene, corresponding to the three different prompts. For each scene, one caption was randomly selected and assigned, ensuring a balanced distribution of edit types across the dataset.

3.3 Benchmark

We adopt different dataset splitting strategies depending on the evaluation setting. In all cases, splits are performed at the edit level rather than enforcing a strict scene-level separation, allowing different edited versions of the same underlying 3D scene to appear across partitions.

1. **Mixed split.** In the mixed setting we perform an $80-20\%$ train–test split, resulting in approximately 33k scenes for training and 8k scenes for testing. This setting evaluates overall fake detection performance in a classic machine learning setup in which all training data (in this case, the generators) are known and available at training time.
2. **Cross-editing split.** To evaluate generalization across editing methods, we define two additional splits where the model is trained on fake scenes generated with one editing approach and tested on the other (and vice versa). This setup ensures that the manipulation method used at test time is unseen during training, reflecting a more realistic scenario in which newly developed editing techniques are deployed after the detection model has already been trained. In this manner, we investigate the performance of a detector against unseen generators.

4 Experimental Results

4.1 Baselines

Since, to the best of our knowledge, no method specifically addresses fake detection for 3D Gaussian scenes, we adopt state-of-the-art 2D deepfake detectors as baselines and evaluate them on renderings of the corresponding 3DGS scenes. In addition to our method, we compare against established detectors designed to recognize synthetic images produced by modern generators.

In our experiments, all baselines are fine-tuned on our training split (starting from the authors' released weights) to ensure a fair comparison under the same data distribution. Specifically, we include the models of Wang *et al.* [50], which build on a ResNet-50 backbone [47] and investigate training strategies based on different image transformations to improve robustness and cross-generator generalization. Furthermore, we evaluate the reconstruction-based approach of Wang *et al.* [51], which detects synthetic content by comparing an input image to its reconstruction and analyzing the resulting residual, and we also include CoDE (Contrastive Deepfake Embeddings) [2], which learns a deepfake

oriented embedding space via contrastive learning while enforcing global/local agreement between full-image and cropped-view representations to capture both global cues and fine-grained artifacts. Finally, we include CLIP-based baselines following Ojha *et al.* [41]: we load pretrained CLIP encoders [43] and train a linear classifier on top of the CLIP image embedding. In this setting, we report results for two CLIP variants (ViT-B/16 and ViT-L/14), finetuned on our training split under the same protocol as the other baselines. Additionally, we evaluate self-supervised visual backbones based on DINOv2 [42], by fine-tuning the corresponding pretrained representations with an identical linear classification head.

For all experiments involving 2D detectors, we render images from each 3D scene and fine-tune/evaluate the baselines on these renderings using the same split protocol described in Section 3.3.

4.2 Proposed method

We propose a 3D fake detector, here referred to as *Fake3DD*, that operates directly on the Gaussian splatting representation. As backbone, we select the PointTransformerV3 model [53], a transformer-based architecture for point clouds that treats a point set as an unordered collection of 3D samples and learns features by local self-attention: for each point, the model aggregates information from nearby points in space. This design is well-suited to our setting because a 3DGS scene can be naturally viewed as a set of primitives distributed in 3D, where the authenticity cues may depend on spatial context rather than on individual Gaussians in isolation.

Inspired by the work of Wu et al. [53] and Li et al. [31], we applied some modifications to the original PointTransformerV3 model to enable the processing of Gaussian splats as input primitives instead of standard point clouds. These changes mainly concern the input feature representation, which is extended to incorporate the attributes associated with each Gaussian. Beyond these adaptations, we benefit from the inherent flexibility of the adopted architecture, which naturally supports batching scenes with varying numbers of Gaussian splats. This property allows us to avoid padding or resampling strategies and enables efficient training and inference across heterogeneous scenes.

We propose to represent each Gaussian through the 3D coordinates of its mean point, and instead of the traditional color feature used for point clouds, we provide opacity, scale, quaternions, spherical harmonics s_h with the zeroth-order term s_0 representing the view-independent color component. Both the encoder and decoder stages of PointTransformerV3 are used to extract contextualized Gaussian-level features. Given that each 3D scene is represented by a variable number of Gaussian splats, these Gaussian-level features are aggregated into a single scene-level representation using a global mean pooling operation. Specifically, features belonging to the same scene are averaged based on their batch indices, producing a fixed-dimensional embedding per scene. These embeddings are then passed to a classification head for the final binary prediction (fake or real classes).

Table 1. Experimental results on Fake3DGS dataset under different train/test split protocols. We report overall accuracy together with class-wise accuracy on fake and real samples. For each backbone, we evaluate (i) cross-edit generalization by training on one editing method (GaussCtrl or Instruct-GS2GS) and testing on the other, and (ii) the mixed setting where training and testing use the combined data.

	Train		**Test**		**Accuracy** (%)		
Backbone	GaussCtrl	Instruct-GS2GS	GaussCtrl	Instruct-GS2GS	Overall	Fake	Real
CLIP ViT-B	✓			✓	80.7	70.6	95.9
CLIP ViT-B		✓	✓		74.3	41.5	98.5
CLIP ViT-B	✓	✓	✓	✓	84.0	84.2	93.8
CLIP ViT-L	✓			✓	81.5	67.1	96.6
CLIP ViT-L		✓	✓		75.4	44.5	98.2
CLIP ViT-L	✓	✓	✓	✓	84.0	83.7	94.3
DINOv2-B	✓			✓	76.8	72.7	89.2
DINOv2-B		✓	✓		73.2	45.9	93.3
DINOv2-B	✓	✓	✓	✓	87.8	89.7	85.8
CoDE [2]	✓			✓	80.2	60.5	92.4
CoDE [2]		✓	✓		79.3	54.1	95.8
CoDE [2]	✓	✓	✓	✓	92.2	91.4	92.9
DM [12]	✓			✓	83.8	74.8	95.8
DM [12]		✓	✓		82.4	80.3	96.9
DM [12]	✓	✓	✓	✓	90.4	89.3	97.2
UFD [41]	✓			✓	79.5	69.4	91.9
UFD [41]		✓	✓		78.2	59.5	95.4
UFD [41]	✓	✓	✓	✓	89.9	90.6	89.2
Fake3DD (Ours)		✓	✓		98.2	97.8	98.6
Fake3DD (Ours)	✓			✓	98.6	99.1	98.3
Fake3DD (Ours)	✓	✓	✓	✓	**98.9**	**98.1**	**99.5**

4.3 Results

Table 1 summarizes the performance of 2D baselines and our 3D Gaussian-based detector under mixed and cross-edit protocols. In the mixed setting (both editors in train and test), most 2D detectors achieve reasonably high overall accuracy, with the strongest baseline reaching 92.2% (CoDE). However, when evaluated under cross-edit generalization, all 2D methods exhibit a pronounced drop. Importantly, this degradation is largely driven by failures on the *Fake* class: *e.g.*, CLIP and DINO variants can maintain high *Real* accuracy (often above 93–98%) while their *Fake* accuracy collapses (down to 41.5-45.9% in several cases).

Notably, the strongest cross-edit baseline is DM trained on GaussCtrl and tested on Instruct-GS2GS ($G \rightarrow I$), achieving 83.8% overall accuracy. However, the class-wise results indicate that this performance is still driven by a much higher accuracy on real samples (95.8%) than on fake samples (74.8%). This gap suggests that, under an unseen editor, the detector remains conservative and tends to misclassify a substantial portion of edited scenes as real.

In contrast, our method reaches 98.7% on the same $G \rightarrow I$ protocol and remains well balanced across classes (99.1% on fake and 98.4% on real). Compared to the best cross-edit baseline, this corresponds to a +14.9 percentage points gain in overall accuracy, largely explained by a +24.3 pp improvement on the fake class (74.8% → 99.1%), while also improving real accuracy (+2.6 pp). These

results support the hypothesis that 2D detectors tend to exploit editor-specific cues that do not transfer across manipulation pipelines, whereas operating directly on the 3DGS representation provides more editor-agnostic evidence for authenticity.

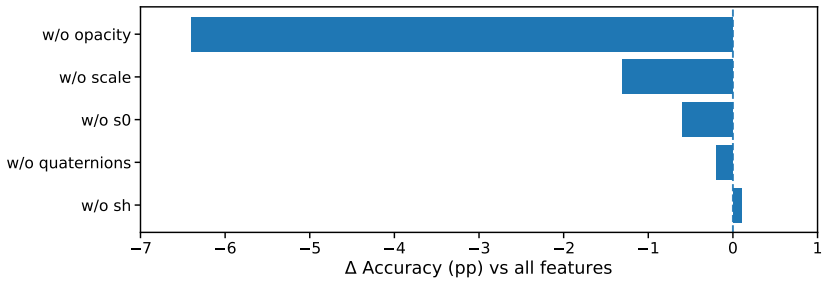

Fig. 2. Ablation over Gaussian feature groups. We report the change in accuracy (in percentage points) obtained by removing each feature group from the input, relative to the full-feature model.

4.4 Ablation Study

As shown in Figure 2, we conducted an ablation study to assess the contribution of each Gaussian attribute to the detection performance.

We remove one feature group at a time from the input and report the resulting change in accuracy with respect to the full model. Removing opacity caused the largest drop in accuracy (92.5%), indicating that transparency information is critical. Scale and the zeroth-order spherical harmonics coefficient s_0 provide a moderate contribution to detection performance. Scale captures geometric properties related to the spatial extent and distribution of Gaussian primitives, which can be affected by scene editing. The s_0 term encodes view-independent color information; however, its impact is limited due to redundancy with other attributes and the scene-level mean pooling operation. Quaternion-based orientation features exhibit minimal influence on performance, suggesting that Gaussian orientation remains largely consistent between real and fake scenes and is therefore less informative for this task and carry limited discriminative information. Higher-order SH coefficients had minimal impact suggesting that view-dependent appearance cues are largely irrelevant for this binary classification task. Despite removing s_h seems to improve the overall accuracy of 0.1, this was not the case in the cross setting, resulting in a lower accuracy w.r.t using s_h. We therefore retain s_h in all main experiments to favor robustness and generalization across editing methods.

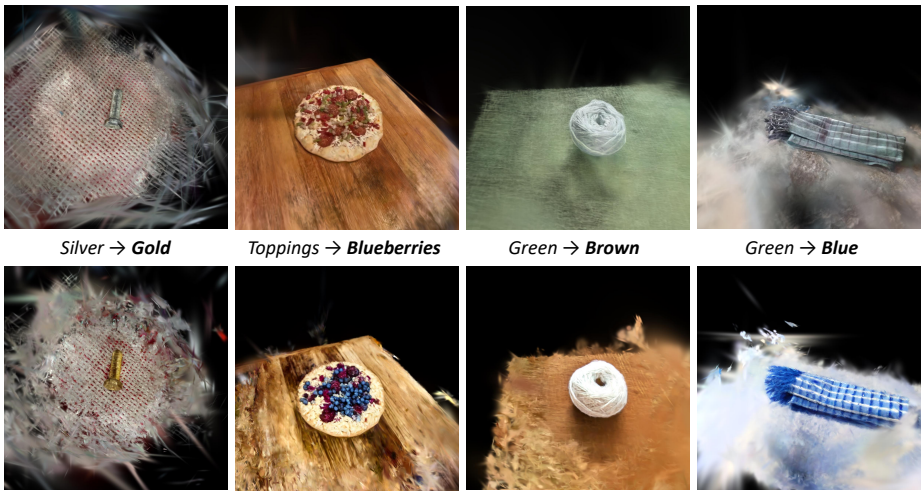

Fig. 3. Some sample renderings of the data in our dataset. Below each image, there is the correspondent prompt used to generate edited samples.

5 Conclusions and Future Work

In this work, we introduced the important – but still underexplored – concept of 3D fake detection, which currently represent a security threat. To further investigate this novel task, we presented Fake3DGS, a large-scale benchmark composed of over 40k real and edited 3D scenes generated using different editing methods. We also proposed a novel 3D fake detection framework based on Gaussian splatting representations. By leveraging Gaussian-level attributes and a modified PointTransformerV3 architecture, our approach effectively captures scene-level inconsistencies introduced by editing operations.

Experimental results demonstrate that while 2D-based fake detection methods perform well when trained and tested on similar editing distributions, they struggle to generalize across unseen editing methods. In contrast, our 3D approach consistently outperforms 2D baselines, particularly in cross-edit evaluation settings, highlighting its robustness to editor-specific artefacts and its ability to generalize to novel manipulations. An extensive ablation study further shows that opacity and geometric attributes play a crucial role in detection, whereas view-dependent appearance cues contribute marginally to performance.

Future work will focus on evaluating the proposed framework using a broader range of 3D editing methods to assess its generalization capabilities further. Additionally, we plan to expand the dataset by explicitly annotating the spatial location and extent of edits within each scene, enabling more fine-grained tasks such as edit localization and region-level fake detection. We believe these directions will help advance the study of reliable and robust fake detection in 3D content generation pipelines.

References

1. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
2. Baraldi, L., Cocchi, F., Cornia, M., Baraldi, L., Nicolosi, A., Cucchiara, R.: Contrasting Deepfakes Diffusion via Contrastive Learning and Global-Local Similarities. In: Proceedings of the European Conference on Computer Vision (2024)
3. Borghi, G., Franco, A., Graffieti, G., Maltoni, D.: Automated artifact retouching in morphed images with attention maps. IEEE Access **9**, 136561–136579 (2021)
4. Borghi, G., Pancisi, E., Ferrara, M., Maltoni, D.: A double siamese framework for differential morphing attack detection. Sensors **21**(10), 3466 (2021)
5. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
6. Catalini, R., Biagi, F., Salici, G., Borghi, G., Vezzani, R., Biagiotti, L.: Llms and humanoid robot diversity: The pose generation challenge. In: International Conference on Social Robotics. pp. 531–538. Springer (2025)
7. Cen, J., Zhou, Z., Fang, J., Shen, W., Xie, L., Jiang, D., Zhang, X., Tian, Q., et al.: Segment anything in 3d with nerfs. In: NeurIPS (2023)
8. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: CVPR (2024)
9. Chen, Z., Wang, F., Wang, Y., Liu, H.: Text-to-3d using gaussian splatting. In: CVPR (2024)
10. Chen, Z., Wang, G., Zhu, J., Lai, J., Xie, X.: Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In: CVPR (2025)
11. Corvi, R., Cozzolino, D., Poggi, G., Nagano, K., Verdoliva, L.: Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In: CVPR (2023)
12. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: ICASSP (2023)
13. Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., Verdoliva, L.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv preprint arXiv:1812.02510 (2018)
14. Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition. pp. 5781–5790 (2020)
15. Di Nucci, D., Tomei, M., Borghi, G., Ciuffreda, L., Vezzani, R., Cucchiara, R.: Brum: Robust 3d vehicle reconstruction from 360 sparse images. In: 2025 IEEE Intelligent Vehicles Symposium (IV). pp. 1353–1360. IEEE (2025)
16. Farid, H.: Lighting (in) consistency of paint by text. arXiv preprint arXiv:2207.13744 (2022)
17. Farid, H.: Perspective (in) consistency of paint by text. arXiv preprint arXiv:2206.14617 (2022)
18. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: ICML (2020)
19. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
20. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)

21. Gragnaniello, D., Cozzolino, D., Marra, F., Poggi, G., Verdoliva, L.: Are gan generated images easy to detect? a critical analysis of the state-of-the-art. ICME (2021)
22. He, R., Huang, S., Nie, X., Hui, T., Liu, L., Dai, J., Han, J., Li, G., Liu, S.: Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6966–6975 (2024)
23. Hu, X., Wang, Y., Fan, L., Fan, J., Peng, J., Lei, Z., Li, Q., Zhang, Z.: Semantic anything in 3d gaussians. arXiv:2401.17857 (2024)
24. Huang, X., Luo, Z., Song, Q., Wang, R., Wan, R.: Marksplatter: Generalizable watermarking for 3d gaussian splatting model via splatter image structure (2025)
25. Hull, M., Yang, H., Mehta, P., Phute, M., Cho, A., Wang, H., Lau, M., Lee, W., Lunardi, W.T., Andreoni, M., Chau, P.: 3d gaussian splat vulnerabilities (2025)
26. In, S., Jang, Y., Jeong, U., Jang, M., Park, H., Park, E., Kim, S.: Compmarkgs: Robust watermarking for compressed 3d gaussian splatting (2025)
27. Jang, Y., Park, H., Yang, F., Ko, H., Choo, E., Kim, S.: 3d-gsw: 3d gaussian splatting for robust watermarking. In: CVPR (2025)
28. Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H.Y.K., Gao, F., Yang, Y., Jiang, C.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In: ACM TOG (2024)
29. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG (2023)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
31. Li, Y., Ma, Q., Yang, R., Li, H., Ma, M., Ren, B., Popovic, N., Sebe, N., Konukoglu, E., Gevers, T., et al.: Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
32. Li, Z., Chen, Y., Zhao, L., Liu, P.: Controllable text-to-3d generation via surface-aligned gaussian splatting (2025)
33. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In: CVPR (2024)
34. Liu, X., Tayal, P., Wang, J., Zarzar, J., Monnier, T., Tertikas, K., Duan, J., Toisoul, A., Zhang, J.Y., Neverova, N., et al.: Uncommon objects in 3d. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14102–14113 (2025)
35. Liu, Y.T., Guo, Y.C., Luo, G., Sun, H., Yin, W., Zhang, S.H.: Pi3d: Efficient text-to-3d generation with pseudo-image diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19915–19924 (2024)
36. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). IEEE (2019)
37. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM (2021)
38. Morgenstern, W., Barthel, F., Hilsmann, A., Eisert, P.: Compact 3d scene representation via self-organizing gaussian grids. In: ECCV (2024)
39. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022)
40. Niedermayr, S., Stumpfegger, J., Westermann, R.: Compressed 3d gaussian splatting for accelerated novel view synthesis. In: CVPR (2024)
41. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: CVPR (2023)

42. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (TMLR) (2024)
43. 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.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763 (2021), https://proceedings.mlr.press/v139/radford21a.html
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
45. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. In: CVPR (2019)
46. Sha, Z., Li, Z., Yu, N., Zhang, Y.: De-fake: Detection and attribution of fake images generated by text-to-image generation models. In: ACM (2023)
47. Shi, Z., zheng, h., Xu, C., Dong, C., Pan, B., xueshuo, X., He, A., Li, T., Fu, H.: Resfusion: Denoising diffusion probabilistic models for image restoration based on prior residual noise. In: NeurIPS (2024)
48. Vachha, C., Haque, A.: Instruct-gs2gs: Editing 3d gaussian splats with instructions (2024), https://instruct-gs2gs.github.io/
49. Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., Liu, Y.: Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. arXiv preprint arXiv:1909.06122 (2019)
50. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: CVPR (2020)
51. Wang, Z., Bao, J., Zhou, W., Wang, W., Hu, H., Chen, H., Li, H.: Dire for diffusion-generated image detection. In: ICCV (2023)
52. Wu, J., Bian, J.W., Li, X., Wang, G., Reid, I., Torr, P., Prisacariu, V.A.: Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In: European Conference on Computer Vision. pp. 55–71. Springer (2024)
53. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler, faster, stronger. In: CVPR (2024)
54. Xu, J., Wang, X., Cheng, W., Cao, Y.P., Shan, Y., Qie, X., Gao, S.: Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20908–20918 (2023)
55. Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M., Kanazawa, A.: gsplat: An open-source library for gaussian splatting. Journal of Machine Learning Research (2025)
56. Yuezun, L., Xin, Y., Pu, S., Honggang, Q., Siwei, L.: Celeb-df: A large-scale challenging dataset for deepfake forensics. In: IEEE Conference on Computer Vision and Patten Recognition (CVPR) (2020)
57. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023)
58. Zhang, X., Ge, X., Xu, T., He, D., Wang, Y., Qin, H., Lu, G., Geng, J., Zhang, J.: Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In: ECCV (2024)

59. Zhou, S., Fan, Z., Xu, D., Chang, H., Chari, P., Bharadwaj, T., You, S., Wang, Z., Kadambi, A.: Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In: ECCV (2024)
60. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21634–21643 (2024)
61. Zhou, Y., Simon, M., Peng, Z., Mo, S., Zhu, H., Guo, M., Zhou, B.: Simgen: Simulator-conditioned driving scene generation. arXiv preprint arXiv:2406.09386 (2024)