3D-GSW: 3D Gaussian Splatting for Robust Watermarking

Youngdong Jang[1] Hyunje Park [1] Feng Yang [2] Heeju Ko [1] Euijin Choo[3] Sangpil Kim[1*]

[1] Korea University [2] Google DeepMind [3] University of Alberta

Abstract

As 3D Gaussian Splatting (3D-GS) gains significant attention and its commercial usage increases, the need for watermarking technologies to prevent unauthorized use of the 3D-GS models and rendered images has become increasingly important. In this paper, we introduce a robust watermarking method for 3D-GS that secures copyright of both the model and its rendered images. Our proposed method remains robust against distortions in rendered images and model attacks while maintaining high rendering quality. To achieve these objectives, we present Frequency-Guided Densification (FGD), which removes 3D Gaussians based on their contribution to rendering quality, enhancing real-time rendering and the robustness of the message. FGD utilizes Discrete Fourier Transform to split 3D Gaussians in high-frequency areas, improving rendering quality. Furthermore, we employ a gradient mask for 3D Gaussians and design a wavelet-subband loss to enhance rendering quality. Our experiments show that our method embeds the message in the rendered images invisibly and robustly against various attacks, including model distortion. Our method achieves superior performance in both rendering quality and watermark robustness while improving real-time rendering efficiency. Project page: `https://kuai-lab.github.io/cvpr20253dgsw/`

1. Introduction

3D representation has been at the center of computer vision and graphics. Such technology plays a pivotal role in various applications and industries, *e.g.*, movies, games, and the Metaverse industry. Since Neural Radiance Field [31] (NeRF) has shown great success in 3D representation due to photo-realistic rendering quality, it has been at the forefront of 3D content creation.

Recently, 3D Gaussian Splatting [15] (3D-GS) has gained attention for its real-time rendering performance and high rendering quality, compared to other radiance field methods [6, 9, 31, 34]. 3D-GS is an explicit representation that uses trainable 3D Gaussians. This explicit property enhances the capability of 3D-GS to generate 3D assets. Due to these properties, 3D-GS has been a transformative 3D representation.

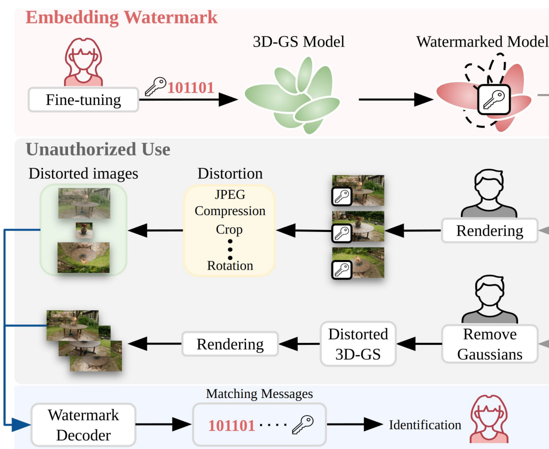

Figure 1. The unauthorized use of the 3D Gaussian Splatting model. Our method ensures that the watermark remains detectable even in distorted images and under model attacks.

While 3D-GS has advanced and practical usage has increased, it raises concerns about the unauthorized use of its 3D assets. Therefore, attempts have been made to develop digital watermarking for radiance fields to address this problem, such as WateRF [11] , which integrates watermark embedding into the rendering process. However, this method presents several challenges when applied directly to 3D-GS. First, achieving high-fidelity rendering requires redundant 3D Gaussians, which leads to substantial memory and storage overheads, especially for large-scale scenes. This also makes embedding a large amount of watermark computationally expensive, significantly increasing processing time. Secondly, there are many 3D Gaussians that have minimal impact on the rendered image. Their minimal impact on the rendered image makes it difficult to embed the watermark robustly.

To address these issues, we propose Frequency-Guided Densification (FGD) to reduce the number of 3D Gaussians to ensure both real-time rendering and robust message embedding. FGD consists of two phases. In the first phase, we

*Corresponding Author

remove 3D Gaussians based on their contribution to the rendering quality. The remaining 3D Gaussians, which significantly impact the rendered image, enable robust message embedding. In the second phase, we utilize two properties to enhance rendering quality: 1) Smaller 3D Gaussians have minimal impact on the rendered image [17]. 2) The human visual system is less sensitive to high-frequency areas [18]. To identify high-frequency areas, we apply Discrete Fourier Transform (DFT) to the rendered image in a patch-wise manner and measure the intensity of high-frequency. After that, 3D Gaussians with strong high-frequency intensity are split into smaller ones to ensure the rendering quality.

Furthermore, significant changes in the parameters of 3D-GS, optimized for high rendering quality, lead to substantial variations in the rendered output. To minimize the adjustments, we utilize a gradient mask derived from the pre-trained parameters, transmitting smaller gradients to 3D-GS during optimization. In this way, the rendering quality is not significantly decreased. To further enhance rendering quality, we design a wavelet-subband loss. Since we split 3D Gaussians in high-frequency areas, the wavelet-subband loss enhances the local structure by leveraging only high-frequency components.

Our experimental results show that our method effectively fine-tunes 3D-GS to embed the watermark into the rendered images from all viewpoints. We also evaluate the robustness of our method under various attacks, including image distortion and model attacks. We compare the performance of our method with other methods [11, 20] and demonstrate that our method outperforms other state-of-the-art radiance field watermarking methods across all metrics. Our main contributions are summarized as follows:

- We propose frequency-guided densification to effectively remove 3D Gaussians without compromising the rendering quality, to embed a robust message in the rendered image while enhancing the real-time rendering.
- We propose a gradient mask mechanism that minimizes gradients to preserve similarity to the pre-trained 3D-GS and maintain high rendering quality.
- We introduce a wavelet-subband loss to enhance rendering quality, particularly in high-frequency areas.
- The proposed method achieves superior performance and demonstrates robustness against various types of attacks, including both image and model distortions.

2. Related work

2.1. 3D Gaussian Splatting

Recently, 3D Gaussian Splatting (3D-GS) [15] has brought a paradigm shift in the radiance field by introducing an explicit representation and differentiable point-based splatting methods, allowing for real-time rendering. 3D-GS has been applied to various research areas, including 3D reconstruction [14, 23, 27, 47], dynamic scenes [10, 28, 45, 48], avatar [19, 21, 33, 40] and generation [7, 22, 24, 25]. Its capability and efficiency have made 3D-GS widely used, positioning it at the forefront of 3D asset generation. As adoption grows across various applications, ensuring the integrity and reliability of generated content has become increasingly important. Therefore, the copyright protection of 3D-GS-generated assets has become essential aspect.

2.2. Frequency Transform

Discrete Fourier Transform (DFT) has played a crucial role in signal processing and image processing. Recent research [12, 13, 37, 49] has applied DFT to images and leveraged frequency signals to improve model performance and analyze images. Baig [1] utilizes DFT to estimate the quality of blurred images globally. Rao [36] leverages this ability of DFT to acquire global information about images. According to these studies, DFT can efficiently analyze global information in images. Since we need to analyze global frequency signal strength across image patches, we use DFT to transform the rendered images in a patch-wise manner.

Discrete Wavelet Transform (DWT) analyzes signals or images by decomposing them into components with different frequencies and resolutions. DWT is particularly effective at capturing local information. In the previous works [32, 35, 44], DWT has been applied to images for denoising. Tian [43] utilizes DWT as it provides both spatial and frequency information through multi-resolution analysis, enabling effective noise suppression and detailed image restoration. For the radiance field, previous works [11, 26, 39, 46] show the compatibility between the radiance field and DWT. Leveraging these advantages, we utilize DWT to compute loss functions between high-frequency local information, thereby enhancing rendering quality.

2.3. Steganography and Digital Watermarking

Steganography is employed to maintain the confidentiality of information by embedding it invisibly within digital assets. Recently, there has been growing interest in applying steganography to the radiance field [4, 8, 20]. StegaNeRF [20] fine-tunes the pre-trained radiance fields model to invisibly embed images into the rendered image. For 3D-GS, GS-hider [51] invisibly embeds 3D scenes and images into point clouds.

Digital watermarking protects digital assets by identifying the copyrights. The main difference lies in the priority of data embedding. The primary goal of digital watermarking is robustness, ensuring that embedded data can be detected even after distortions, whereas steganography prioritizes invisibility. To achieve robustness, the traditional watermarking methods [2, 38, 41, 42] have utilized DWT, embedding into the subbands of DWT. HiDDeN [52] is the end-to-end

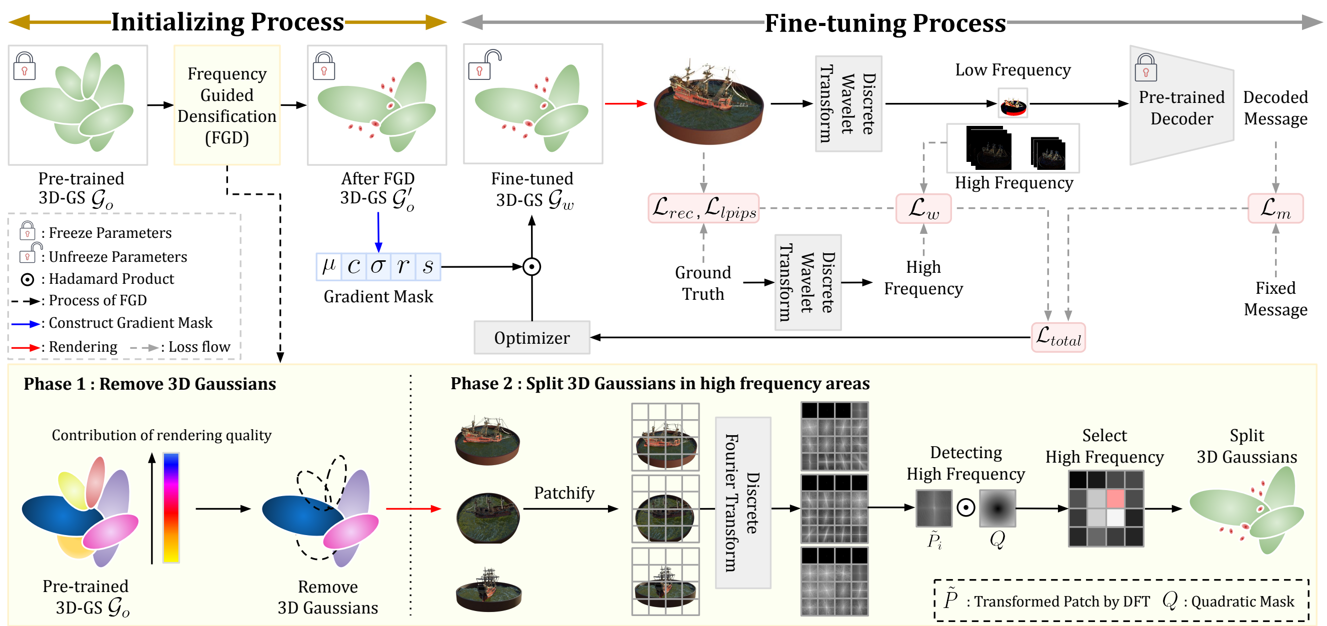

Figure 2. **3D-GSW Overview**. Before fine-tuning 3D-GS, Frequency-Guided Densification (FGD) removes 3D Gaussians based on their contribution to the rendering quality and splits 3D Gaussians in high-frequency areas into smaller ones. We also construct a gradient mask based on the parameters of an FGD-processed 3D-GS. During the fine-tuning, we apply the Discrete Wavelet Transform (DWT) to the rendered image for robustness, using the low frequency as input to a pre-trained message decoder. For rendering quality, we design a wavelet-subbands loss that utilizes only high-frequency subbands. Finally, 3D-GS is optimized through the $\mathcal{L}_{total}$.

deep learning watermarking method, which embeds the robust message by adding a noise layer. For radiance fields watermarking, CopyRNeRF [29] explores embedding the message into the rendered image from implicit NeRF. WateRF [11] enhances both high rendering quality and robustness of watermarks through DWT. In this paper, we introduce a robust digital watermarking method for 3D-GS.

3. Method

3.1. Preliminary

3D-GS [15] represents the 3D world with a set of 3D Gaussian primitives, each defined as:

$$G(\mathbf{x}; \mu, \Sigma) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)} \tag{1}$$

, where the parameters mean μ and covariance Σ determine spatial distribution. To render these primitives onto an image plane, each 3D Gaussian is projected into 2D-pixel space and forms a 2D Gaussian primitive $\hat{G}$ by projective transform and its Jacobian evaluation μ. The 2D Gaussian primitives are depth-ordered, rasterized, and alpha-blended using transmittance T_i as a weight to form an image:

$$I_\pi[x, y] = \sum_{i \in N_{\mathcal{G}}} c_i \alpha_i T_i, \text{ where } T_i = \prod_{j=1}^{i-1}(1 - \alpha_j) \tag{2}$$

$$\alpha_i = \sigma_i \hat{G}_i^\pi([x, y]; \hat{\mu}, \hat{\Sigma}) \tag{3}$$

, where π, c_i, σ_i, and α_i are the viewpoint, color, opacity, and density of each Gaussian primitive evaluated at each pixel. $N_{\mathcal{G}}$ denotes the set of depth-ordered 2D Gaussian primitives that are present in the selected viewpoint.

3.2. Fine-tuning 3D Gaussian Splatting

As shown in Fig. 2, we fine-tune the pre-trained 3D Gaussian Splatting (3D-GS) $\mathcal{G}_o$ into $\mathcal{G}_w$ to ensure the rendered images from all viewpoints contain a binary message $M = (m_1, ..., m_N) \in \{0, 1\}^N$. To achieve this, we utilize a pre-trained message decoder, HiDDeN [52], denoted as D_m. Before fine-tuning, to enhance robustness, we employ Frequency-Guided Densification (FGD) to remove 3D Gaussians with minimal impact on the rendered image and split 3D Gaussians in high-frequency areas (see Sec. 3.3 and Sec. 3.4). After that, we construct a gradient mask based on the FGD-processed 3D-GS $\mathcal{G}_o'$ (see Sec. 3.5) to ensure high rendering quality. In the fine-tuning process, $\mathcal{G}_w$ renders an image $I_w \in \mathbb{R}^{3 \times H \times W}$. I_w is transformed into the wavelet subbands $\{LL_l, LH_l, HL_l, HH_l\}$, where l denotes the level of DWT. L and H are respectively denoted as low and high. Following the previous work [11], we choose the LL_2 subband as input D_m and decode the message $M' = D_m(LL_2)$, ensuring efficient and robust message embedding. Additionally, we employ high-frequency subbands for the proposed wavelet-subband loss. Further details are provided in the following sections.

3.3. Measure Contribution of Rendering Quality

The pre-trained 3D-GS includes redundant 3D Gaussians to ensure high-quality rendering. Because 3D Gaussians with minimal impact on rendering quality can also carry the message, it tends to be weakly embedded in the rendered image. To address this limitation, we remove 3D Gaussians with minimal impact on the rendered image before the fine-tuning process. Inspired by error-based densification [5], we measure the contribution of each 3D Gaussian to the rendering quality using the auxiliary loss function L_π^{aux} with a new color parameter set C' for the viewpoint π:

$$L_\pi^{aux} := \frac{\sum_{x,y\in Pix} \mathcal{E}_\pi[x,y] I_\pi^{c'}[x,y]}{H \times W} \tag{4}$$

$$\mathcal{E}_\pi = |I_\pi^{c'} - I_\pi^{gt}| \tag{5}$$

, where $I_\pi^{c'} \in \mathbb{R}^{3\times H\times W}$ and $I_\pi^{gt} \in \mathbb{R}^{3\times H\times W}$ are respectively denoted as a rendered image with C' and ground truth. We replace the parameters C with C' only when $\mathcal{G}_o$ renders $I_\pi^{c'}$, and set all of its values to zeros. During the backward process, the gradients of the auxiliary loss with respect to C' are derived as follows:

$$V_\pi = \frac{\partial L_\pi^{aux}}{\partial C'} = \sum_{x,y\in Pix} \mathcal{E}_\pi[x,y] w_\pi \tag{6}$$

$$w_\pi = \sum_{i\in N_\mathcal{G}} c_i \alpha_i T_i \tag{7}$$

, where c_i, α_i and T_i are respectively denoted as the color, the density, and the transmittance of each 3D Gaussian. We utilize this $V_\pi \in \mathbb{R}^{N_\mathcal{G}\times 3}$, as the contribution for a rendered quality at π, as it reflects each 3D Gaussian's contribution to the color of the rendered image.

3.4. Frequency-Guided Densification (FGD)

Our method aims to embed the message M robustly into the rendered image to ensure fast embedding and real-time rendering speed without a decrease in rendering quality. To achieve these objectives, we propose Frequency-Guided Densification (FGD), which removes 3D Gaussians, which have minimal impact on the rendered image, and splits 3D Gaussians in the high-frequency areas into smaller ones.

FGD consists of two phases to achieve these goals. First, the pre-trained $\mathcal{G}_o$ renders the image $I_\pi^{c'}$ from all viewpoints, and we derive V_π from the rendered images. Based on V_π, we remove 3D Gaussians that have negligible impact on the rendering quality. Second, since large scenes require substantial memory, images rendered by 3D-GS with 3D Gaussians removed are divided into patches $P \in \mathbb{R}^{3\times M\times N}$ to improve memory efficiency. Since FGD identifies patches with strong high-frequency signals, we utilize the Discrete Fourier Transform (DFT) for the global frequency analysis. The DFT is defined as follows:

$$F[u,v] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n] e^{-j2\pi\left(\frac{u}{M}m + \frac{v}{N}n\right)} \tag{8}$$

, where f and F are respectively denoted as spatial-domain pixel value at spatial-domain image coordinate (m,n) and frequency-domain pixel value at the frequency-domain image coordinate (u,v). We transform the spatial-domain patch P into the frequency domain using DFT, revealing a complete spectrum of frequency components, i.e., $\tilde{P} = \mathbb{R}(F(P)) \in \mathbb{R}^{3\times U\times V}$. The transformed patch $\tilde{P}$ undergoes Hadamard product $\odot$ with a mask $Q \in \mathbb{R}^{3\times U\times V}$, designed to emphasize high-frequency signals, and the intensity of high-frequency E is computed as follows:

$$Q[u,v] = (\frac{2u-U}{U})^2 + (\frac{2v-V}{V})^2 \tag{9}$$

$$E = \frac{\sum_{u,v}(\tilde{P}\odot Q)_{uv}}{U\times V} \tag{10}$$

, where $(u,v) \in \mathbb{R}^{U\times V}$. We select the top $K\%$ patch $\tilde{P}$ based on E and track 3D Gaussians from the chosen patches. Based on V_π, we choose the 3D Gaussians that have less impact on the image and split them into smaller ones to enhance rendering quality. Therefore, we effectively reduce the number of 3D Gaussians to enhance rendering speed and maintain high rendering quality. With intensive optimization of 3D Gaussians that significantly impact rendering quality, a robust message can be embedded.

3.5. Gradient Mask for 3D Gaussian Splatting

Since 3D-GS $\mathcal{G}_o'$ passed through FGD renders high-quality images, we must embed the message without compromising rendering quality. To achieve this, we further reduce the gradient magnitude during fine-tuning to minimize changes in the parameters θ of $\mathcal{G}_o'$. The parameters θ consist of position μ, color c, opacity σ, rotation r, and scale s.

While StegaNeRF [20] uses a gradient mask to modify the gradient, applying this method to 3D-GS is challenging due to the zero values in its parameters. To avoid dividing by zero and further reduce the magnitude of the gradient to minimize changes in parameters, we incorporate an exponential function into the mask calculation. To reduce the gradient size of the parameter θ for each 3D Gaussian, the gradient mask $z \in \mathbb{R}^{N_{\mathcal{G}_o'}}$ is calculated as follows :

$$w = \frac{1}{e^{|\theta|^\beta}}, \quad z = \frac{w}{\sum_{i=1}^{N_{\mathcal{G}_o'}} w_i} \tag{11}$$

, where i and $\beta > 0$ are respectively denoted as the index of 3D Gaussians and the strength of gradient manipulation. We calculate the mask z for each parameter, c, σ, r and s. The gradient of the positions parameter, in particular μ, remains close to zero. Therefore, we apply a gradient mask to the parameters, except for μ. During the fine-tuning, the

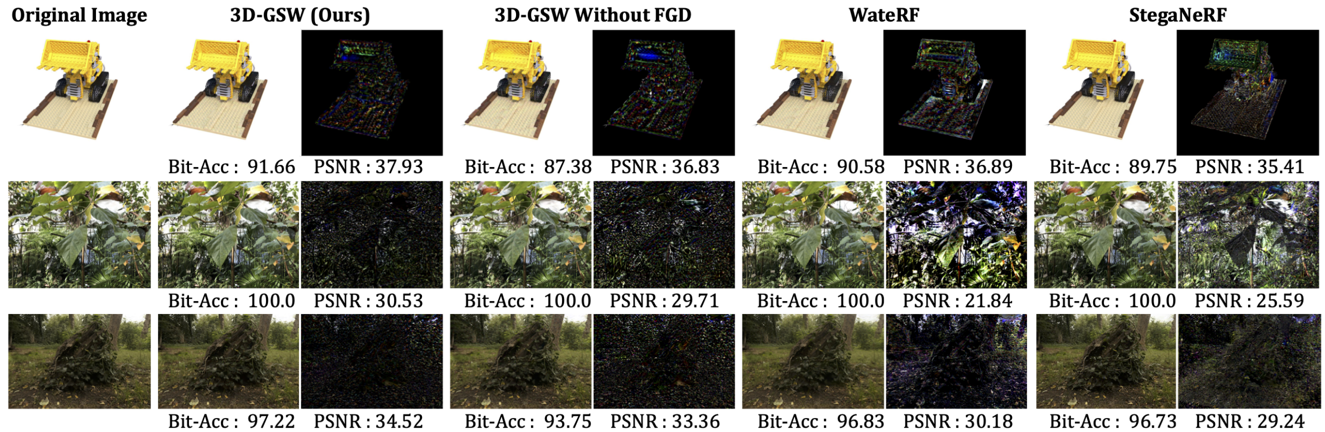

Figure 3. Rendering quality comparison of each baseline with our method. We doubled the scale of the difference map. Our method outperforms others in bit accuracy and rendering quality, using 32-bit messages for the qualitative results.

gradient is masked as $\frac{\partial \mathcal{L}_{total}}{\partial \theta} \odot z$, where $\mathcal{L}_{total}$ is Eq. 16 and $\odot$ denotes Hadamard product. Since small gradients are transmitted to 3D-GS, our gradient mask enables message embedding while preserving high rendering quality.

3.6. Losses

We model the objective of 3D-GS watermarking by optimizing: 1) the reconstruction loss, 2) the LPIPS loss [50] 3) the wavelet-subband loss, and 3) the message loss. For the reconstruction loss, $\mathcal{L}_{rec}$, we measure the difference between the original image I_o and the watermarked image I_w. We employ the loss function $\mathcal{L}_1$:

$$\mathcal{L}_{rec} = \mathcal{L}_1(I_w, I_o) \tag{12}$$

For the LPIPS loss, $\mathcal{L}_{lpips}$, we evaluate the perceptual similarity between the feature maps of I_o and I_w. This loss is typically computed by extracting feature maps from a pre-trained network $f(x)$:

$$\mathcal{L}_{lpips} = \sum_{l} \omega_l \cdot \mathbb{E}\left[(f(I_w) - f(I_o))^2\right] \tag{13}$$

, where l and ω_l are respectively denoted as the layer index of the pre-trained network and the learned scaling factor.

Since we modify 3D Gaussians in the high-frequency areas, we design a wavelet-subband loss $\mathcal{L}_w$ to further enhance the rendering quality of high-frequency areas. Since DWT effectively analyzes local details using several subbands, we only employ high-frequency subbands $\{LH_l, HL_l, HH_l\}$ to improve the rendering quality during embedding of the message. To utilize $\mathcal{L}_w$, I_o is transformed into wavelet subbands $\{LL_l^{gt}, LH_l^{gt}, HL_l^{gt}, HH_l^{gt}\}$. We employ the loss function $\mathcal{L}_1$:

$$\mathcal{L}_w = \sum_{l} \sum_{S} \mathcal{L}_1(S_l, S_l^{gt}), \text{ where } S \in \{LH, HL, HH\} \tag{14}$$

For the message loss $\mathcal{L}_m$, we employ a sigmoid function to confine the extracted message M' within the range of [0, 1]. The message loss is a Binary Cross Entropy between the fixed message M and the sigmoid $sg(M')$:

$$\mathcal{L}_m = -\sum_{i=1}^{N}(M_i \cdot \log sg(M_i') + (1 - M_i) \cdot \log(1 - sg(M_i'))) \tag{15}$$

Finally, 3D-GS is optimized with the total loss, which is the weighted sum of all losses:

$$\mathcal{L}_{total} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_w \mathcal{L}_w + \lambda_m \mathcal{L}_m \tag{16}$$

4. Experiments

4.1. Experimental Setting

Dataset & Pre-trained 3D-GS. We use Blender [31], LLFF [30] and Mip-NeRF 360 [3], which are considered standard in NeRF [31] and 3D-GS [15] literature. We follow the conventional NeRF [31] and 3D-GS [15], wherein we compare the results using 25 scenes from the full Blender, LLFF, and Mip-NeRF 360 datasets.

Baseline. We compare our method (3D-GSW) with three strategies for fairness: 1) StegaNeRF [11]: the steganography method for NeRF models, which embed an image into the rendered image. We add three linear layers to the watermark decoder to enable message decoding. Additionally, to apply the mask of StegaNeRF [11], we set the parameters of 3D-GS to a small value of zero to a small value 10^{-4}. 2) WateRF [11] + 3D-GS [15]: currently the state-of-the-art watermarking method for NeRF models. 3) 3D-GSW without FGD: changing our method by removing FGD.

Implementation Details. Our method is trained on a single A100 GPU. The training is completed with epochs ranging from 2 to 10. The iteration per epoch is the number of train viewpoints in the datasets. We use Adam [16] to optimize

Figure 4. We present a rendering quality comparison for 32-bit, 48-bit, and 64-bit messages. The differences (×2) between the watermarked image and the original image. Since manipulated areas are high-frequency areas where the people's eyes are less sensitive, the rendered image with our method looks more realistic and natural.

	32 bits				48 bits				64 bits			
Methods	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
StegaNeRF [20]+3D-GS [15]	93.15	32.68	0.953	0.049	89.43	32.72	0.954	0.048	85.27	30.66	0.925	0.092
WateRF [11]+3D-GS [15]	93.42	30.49	0.956	0.050	84.16	29.92	0.951	0.053	75.10	25.81	0.883	0.108
3D-GSW without FGD	94.60	34.27	0.975	0.047	86.69	30.46	0.896	0.074	82.49	28.22	0.893	0.077
3D-GSW (Ours)	**97.37**	**35.08**	**0.978**	**0.043**	**93.72**	**33.31**	**0.970**	**0.045**	**90.45**	**32.47**	**0.967**	**0.049**

Table 1. Bit accuracy and quantitative comparison of rendering quality with baselines. We show the results in 32, 48, and 64 bits. The results are the average of Blender, LLFF, and Mip-NeRF 360 datasets. The best performances are highlighted in **bold**.

3D-GS. For the decoder, we pre-train HiDDeN [52] decoder for bits = $\{32, 48, 64\}$ and freeze the parameters during our fine-tuning process. We set $\lambda_{rec} = 1$, $\lambda_{lpips} = 0.2$, $\lambda_w = 0.3$, and $\lambda_m = 0.4$ in our experiments. We remove 3D Gaussians under $V_\pi = 10^{-8}$. Also, we set the patch size $|P| = 16$, the K = 1%, and $\beta = 4$. Our experiments are conducted on five different seeds.

Evaluation. We consider three important aspects of watermarks: 1) **Invisibility**: We evaluate invisibility by using the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [50]. 2) **Robustness**: We investigate robustness by measuring bit accuracy under various distortions. The following distortions for message extraction are considered: Gaussian noise (σ = 0.1), Rotation (random select between $+\pi/6$ and $-\pi/6$), Scaling (75 % of the original), Gaussian blur (σ = 0.1), Crop (40 % of the original), JPEG compression (50 % of the original), a combination of Gaussian Noise, Crop, JPEG Compression. Furthermore, we consider a distortion of the core model, such as removing 3D Gaussians (20 %), cloning 3D Gaussians (20 %) and adding Gaussian noise (σ = 0.1) to the parameters of 3D-GS. 3) **Capacity**: We explore the bit accuracy across various message lengths, which are denoted as $M_b \in \{32, 48, 64\}$.

4.2. Experimental results

Rendering Quality and Bit Accuracy. In this section, we compare the rendering quality and bit accuracy with other methods. As shown in Fig. 3, our method is most similar to the original and achieves high bit accuracy and rendering quality. In particular, since real-world scenes have complex structures, it is difficult to render them similarly to the original. From Fig. 3, while other methods have difficulty balancing the rendering quality and bit accuracy, our method achieves a good balance. Tab. 1 shows that our method ensures rendering quality and bit accuracy across all datasets compared to other methods.

Capacity of Message. Since bit accuracy, rendering quality, and capacity have a trade-off relationship. We explore this with message bit lengths $\{32, 48, 64\}$. As shown in Tab. 1, the bit accuracy, and rendering quality show a consistent decline as the message length increases. However, our method maintains a good balance between the invisibility and capacity of the message and outperforms the other methods as the message length becomes longer. Additionally, there is a further difference in performance compared to without FGD, depending on the message length. This shows that FGD is effective for large message embedding. From Fig. 4, our method guarantees a good balance between bit accuracy and rendering quality.

Robustness for the image distortion. This section assesses the robustness of our method in situations where the rendered images are subjected to post-processing, which potentially modifies the embedded message within the rendered image. We evaluate the bit accuracy of the rendered images containing the message under various distortions. Tab. 2 shows that other methods cannot guarantee robustness. In particular, the steganography method is weak to all attacks. Additionally, 3D-GSW without FGD, which does not remove 3D Gaussians, does not fully address robustness when embedding messages into the rendered image. In contrast, our method ensures robustness against all distortions by removing 3D Gaussians that interfere with robustness.

Robustness for the 3D-GS distortion. Since the purpose of our method is to protect both the rendered image and the core model, it is essential to consider the potential scenario of direct manipulation of the core model in cases of unau-

Methods	Bit Accuracy(%) ↑							
	No Distortion	Gaussian Noise ($\sigma = 0.1$)	Rotation ($\pm\pi/6$)	Scaling (75%)	Gaussian Blur ($\sigma = 0.1$)	Crop (40%)	JPEG Compression (50% quality)	Combined (Crop, Gaussian blur, JPEG)
StegaNeRF [20]+3D-GS [15]	93.15	54.48	67.22	73.98	73.84	75.87	73.28	76.71
WateRF [11]+3D-GS [15]	93.42	77.99	81.64	84.50	87.21	84.49	81.88	64.87
3D-GSW without FGD	92.64	80.42	68.66	84.81	78.91	76.97	82.71	84.67
3D-GSW (Ours)	**97.37**	**83.84**	**87.94**	**94.64**	**96.01**	**92.86**	**92.65**	**90.84**

Table 2. Quantitative evaluation of robustness under various attacks compared to baseline methods. The results are the average of Blender, LLFF, and Mip-NeRF 360 datasets. We conduct experiments using 32-bit messages. The best performances are highlighted in **bold**.

Methods	Bit Accuracy(%) ↑			
	No Distortion	Adding Gaussian Noise ($\sigma = 0.1$)	Removing 3D Gaussians (20 %)	Cloning 3D Gaussians (20 %)
StegaNeRF [20]+3D-GS [15]	93.15	61.82	60.24	75.56
WateRF [11]+3D-GS [15]	93.42	73.85	80.58	82.32
3D-GSW without FGD	92.64	73.20	87.99	87.27
3D-GSW (Ours)	**97.37**	**89.11**	**96.87**	**95.99**

Table 3. Robustness under model distortion. We show the results on 32-bits. The best performances are highlighted in **bold**.

thorized model usage. To manipulate 3D-GS, we select to directly add Gaussian noise to the 3D-GS parameters. Additionally, we randomly remove and clone 3D Gaussians. As shown in Tab. 3, our method is robust against 3D-GS distortion, outperforming the other methods. Furthermore, FGD robustly embeds the message into the rendered image, even if there is distortion in the model.

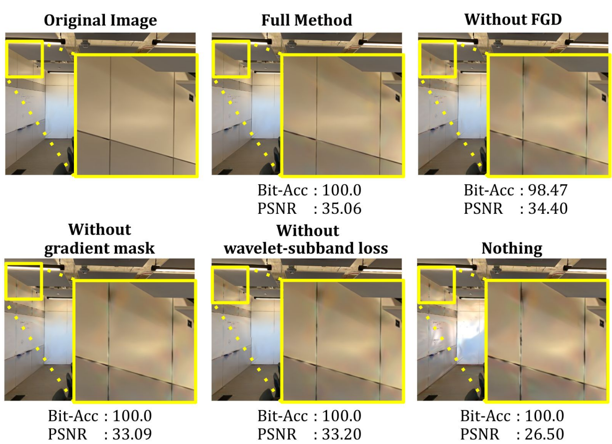

Figure 5. Rendering quality comparisons with full method(ours), without FGD, without gradient mask, without wavelet-subband loss, and base model. All images have 32-bits embedded.

Methods			Ours (3D-GSW)			
FGD	Mask	$\mathcal{L}_{wavelet}$	Bit Acc(%)↑	PSNR ↑	SSIM ↑	LPIPS ↓
–	–	–	96.50	29.96	0.951	0.072
✓	✓	–	96.16	33.56	0.971	0.052
✓	–	✓	96.37	33.26	0.967	0.054
–	✓	✓	94.60	34.27	0.975	0.047
✓	✓	✓	**97.37**	**35.08**	**0.978**	**0.043**

Table 4. Quantitative ablation study of 3D-GSW shows that the best results are achieved when all components are combined. Results are shown for 32-bit messages.

Figure 6. Qualitative result of applying FGD. We analyze the effect of FGD on the rendered image. Through FGD, we effectively control 3D Gaussians in the high-frequency area.

4.3. Ablation study

FGD, Gradient mask, and Wavelet-subband loss. In this section, we evaluate the effectiveness of FGD, gradient mask, and wavelet-subband loss. We remove each component in our method and compare the rendering quality with the bit accuracy. Fig. 5 and Tab. 4 show the results when each component is removed. First, we remove the FGD module in our method. In this case, our method tends to slightly decrease bit accuracy. Fig 6 shows that FGD effectively adjusts 3D Gaussians in high-frequency areas, resulting in a quality that is nearly identical to the original. Second, without the gradient mask and wavelet-subband loss, our method performs poorly in preserving rendering quality. When all components are absent, our method fails to maintain an appropriate trade-off between bit accuracy and rendering quality, leading to a significant decrease in rendering quality. These results show the importance of each component in achieving a good balance between the rendering quality and bit accuracy.

Subband	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
LL, LH, HL, HH	96.01	34.93	0.977	0.048
LH, HL, HH	**97.37**	**35.08**	**0.978**	**0.043**

Table 5. Ablation study on subband selection for wavelet-subband loss. Results represent the average score across Blender, LLFF, and Mip-NeRF 360 datasets using 32-bit messages.

Wavelet-subband loss. Increasing the performance of both bit accuracy and rendering quality is challenging. To address this challenge, we design wavelet-subband loss. Since we modify 3D Gaussians in high-frequency areas, we utilize only the high-frequency subbands $\{LH, HL, HH\}$ to further ensure the rendering quality of those areas. Tab. 4 and Fig. 5 show that wavelet-subband loss effectively en-

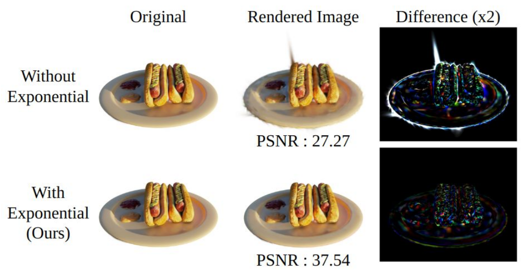

Figure 7. Qualitative comparison of the proposed gradient mask effect. For objects without a background, our method effectively adjusts 3D Gaussian parameters to prevent rendering beyond the object's boundary, preserving the original quality.

hances rendering quality. Additionally, Tab. 5 shows that using only high-frequency subbands results in higher rendering quality, with high bit accuracy.

Gradient mask for 3D-GS. Before the fine-tuning process, the pre-trained 3D-GS already has a high rendering quality. Since this property, if there is a large change in 3D-GS parameters, rendering quality can be decreased. When a gradient is propagated to a parameter, the gradient mask of our method ensures that the transmitted gradient is smaller than that of previous methods. Our gradient mask controls gradient transmission and minimizes parameter changes, thereby preserving rendering quality. Fig 7 shows that our gradient mask (with exponential) enhances rendering quality more effectively than a previous method (without exponential).

Control the number of 3D Gaussians. In this section, we present more details about the effect of controlling the number of 3D Gaussians. In the first phase of Frequency Guided Densification (FGD), we derive the contribution of rendering quality, V_π, for each 3D Gaussian. Fig. 8 shows that removing 3D Gaussians with the contribution below 10^{-8} (removing 28.13 %) has minimal impact on rendering quality and increases slightly bit accuracy. However, when FGD removal exceeds 50 %, the bit accuracy and performance of LPIPS decrease. From the experimental results, reducing approximately 28% 3D Gaussians preserves high bit accuracy and rendering quality.

Comparison of time and storage. The advantages of 3D-GS are the high rendering quality and real-time rendering. However, the pre-trained 3D-GS contains redundant 3D Gaussians to achieve high-quality results, leading to storage capacity issues and increasing the time required for message embedding. Furthermore, other methods render twice during the fine-tuning process, resulting in inefficient embedding time for 3D-GS. To address these issues, we remove 3D Gaussians without a decrease in the rendering quality. Tab. 6 shows that our method reduces storage of 3D-GS and message embedding time. In particular, our method enhances the real-time rendering. Notably, since the other methods maintain the number of 3D Gaussians, they follow the FPS and storage of pre-trained 3D-GS [15].

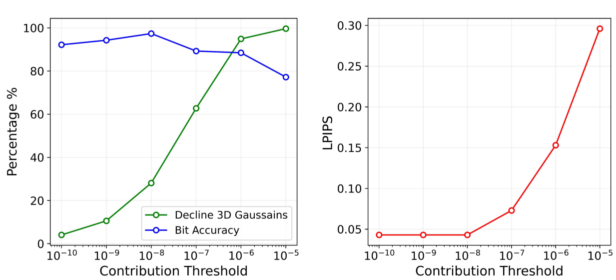

Figure 8. The impact of 3D Gaussians removal is based on the contribution of rendering quality. Declining 3D Gaussians refers to reducing the number of 3D Gaussians. The results are shown for 32-bit messages.

Methods	Fine-tune ↓	FPS ↑	Storage ↓
3D-GS [15]	-	56.65	833.89 MB
StegaNeRF [20]+3D-GS [15]	58h 56m	56.65	833.89 MB
WateRF [11]+3D-GS [15]	6h 47m	56.65	833.89 MB
3D-GSW (Ours)	**21m 03s**	**72.68**	**640.21 MB**

Table 6. Results on the large-scale Mip-NeRF 360 dataset for 64-bits. All scores are averaged across Mip-NeRF 360 scenes, with 'fine-tunes' referring to embedded messages. For a fair comparison, we utilize the pre-trained models provided by 3D-GS [15]. The first row is the performance of the pre-trained models.

5. Conclusion

We introduce the robust watermarking method for 3D Gaussian Splatting (3D-GS), developing a novel densification method, Frequency-Guided Densification (FGD), which ensures real-time rendering speed and robustness while improving rendering quality. We propose the gradient mask to ensure high rendering quality and introduce a wavelet-subband loss to enhance the rendering quality of high-frequency areas. Our experiments show that our method ensures the message and is robust against the distortion of the model compared to the other methods. Our method provides a strong foundation for exploring the broader implications and challenges of 3D-GS watermarking. It underscores the potential of advanced watermarking techniques to address ownership and security issues in the context of a rapidly evolving 3D industry. In future work, we aim to extend our approach to embed multi-modal data, further broadening its applications and enhancing its utility in diverse domains. This expansion will broaden the scope of our method's applications and enhance its adaptability and utility across a wide range of domains.

Limitations. Since our proposed method requires the pre-trained decoder, the decoder pre-training must be done first. Fortunately, the decoder only needs to be trained once per length of bits, and after training, the pre-training process for the corresponding length is not required.

Acknowledgements

This research was supported by the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (Research on neural watermark technology for copyright protection of generative AI 3D content, RS-2024-00348469, 47%; International Collaborative Research and Global Talent Development for the Development of Copyright Management and Protection Technologies for Generative AI, RS-2024-00345025, 42%), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2025-00521602, 10%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program(Korea University), 1%), and Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City.

References

[1] Md Amir Baig, Athar A Moinuddin, Ekram Khan, and M Ghanbari. Dft-based no-reference quality assessment of blurred images. *Multimedia Tools and Applications*, 81(6): 7895–7916, 2022. 2

[2] Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE transactions on image processing*, 10(5): 783–791, 2001. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 5

[4] Monsij Biswal, Tong Shao, Kenneth Rose, Peng Yin, and Sean Mccarthy. Steganerv: Video steganography using implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 888–898, 2024. 2

[5] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting, 2024. 4

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 1

[7] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21401–21412, 2024. 2

[8] Weina Dong, Jia Liu, Lifeng Chen, Wenquan Sun, and Xiaozhong Pan. Stega4nerf: cover selection steganography for neural radiance fields. *Journal of Electronic Imaging*, 33(3): 033031–033031, 2024. 2

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 1

[10] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024. 2

[11] Youngdong Jang, Dong In Lee, MinHyuk Jang, Jong Wook Kim, Feng Yang, and Sangpil Kim. Waterf: Robust watermarks in radiance fields for protection of copyrights, 2024. 1, 2, 3, 5, 6, 7, 8

[12] Xi Jia, Joseph Bartlett, Wei Chen, Siyang Song, Tianyang Zhang, Xinxing Cheng, Wenqi Lu, Zhaowen Qiu, and Jinming Duan. Fourier-net: Fast image registration with band-limited deformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1015–1023, 2023. 2

[13] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13919–13929, 2021. 2

[14] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussian-shader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024. 2

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):1–14, 2023. 1, 2, 3, 5, 6, 7, 8

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[17] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. *arXiv preprint arXiv:2311.13681*, 2023. 2

[18] Jong-Seok Lee and Touradj Ebrahimi. Perceptual video compression: A survey. *IEEE Journal of selected topics in signal processing*, 6(6):684–697, 2012. 2

[19] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19876–19887, 2024. 2

[20] Chenxin Li, Brandon Y Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang. Steganerf: Embedding invisible information within neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 441–453, 2023. 2, 4, 6, 7, 8

[21] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 2

[22] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6517–6526, 2024. 2

[23] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2024. 2

[24] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8576–8588, 2024. 2

[25] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6657, 2024. 2

[26] Ange Lou, Benjamin Planche, Zhongpai Gao, Yamin Li, Tianyu Luan, Hao Ding, Terrence Chen, Jack Noble, and Ziyan Wu. Darenerf: Direction-aware representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5042, 2024. 2

[27] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2

[28] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8900–8910, 2024. 2

[29] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22401–22411, 2023. 3

[30] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 5

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 5

[32] S Kother Mohideen, S Arumuga Perumal, and M Mohamed Sathik. Image de-noising using discrete wavelet transform. *International Journal of Computer Science and Network Security*, 8(1):213–216, 2008. 2

[33] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 788–798, 2024. 2

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1

[35] Varad A Pimpalkhute, Rutvik Page, Ashwin Kothari, Kishor M Bhurchandi, and Vipin Milind Kamble. Digital image noise estimation using dwt coefficients. *IEEE transactions on image processing*, 30:1962–1972, 2021. 2

[36] R Radha Kumari, V Vijaya Kumar, and K Rama Naidu. Deep learning-based image watermarking technique with hybrid dwt-svd. *The Imaging Science Journal*, pages 1–17, 2023. 2

[37] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in neural information processing systems*, 34:980–993, 2021. 2

[38] MS Raval and PP Rege. Discrete wavelet transform based multiple watermarking scheme. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, pages 935–938. IEEE, 2003. 2

[39] Daniel Rho, Byeonghyeon Lee, Seungtae Nam, Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Masked wavelet representation for compact neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20680–20690, 2023. 2

[40] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. *arXiv preprint arXiv:2403.05087*, 2024. 2

[41] Mark J Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992. 2

[42] Peining Tao and Ahmet M Eskicioglu. A robust multiple watermarking scheme in the discrete wavelet transform domain. In *Internet Multimedia Management Systems V*, pages 133–144. SPIE, 2004. 2

[43] Chunwei Tian, Menghua Zheng, Wangmeng Zuo, Bob Zhang, Yanning Zhang, and David Zhang. Multi-stage image denoising with the wavelet transform. *Pattern Recognition*, 134:109050, 2023. 2

[44] Una Tuba and Dejan Zivkovic. Image denoising by discrete wavelet transform with edge preservation. In *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4. IEEE, 2021. 2

[45] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2

[46] Muyu Xu, Fangneng Zhan, Jiahui Zhang, Yingchen Yu, Xiaoqin Zhang, Christian Theobalt, Ling Shao, and Shijian Lu. Wavenerf: Wavelet-based generalizable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18195–18204, 2023. 2

[47] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20923–20931, 2024. 2

[48] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 2

[49] Hu Yu, Jie Huang, Feng Zhao, Jinwei Gu, Chen Change Loy, Deyu Meng, Chongyi Li, et al. Deep fourier up-sampling. *Advances in Neural Information Processing Systems*, 35: 22995–23008, 2022. 2

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 6

[51] Xuanyu Zhang, Jiarui Meng, Runyi Li, Zhipei Xu, Yongbing Zhang, and Jian Zhang. Gs-hider: Hiding messages into 3d gaussian splatting. *arXiv preprint arXiv:2405.15118*, 2024. 2

[52] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018. 2, 3, 6