Supplementary Material: GeometrySticker: Enabling Ownership Claim of Recolorized Neural Radiance Fields

Xiufeng Huang[1,2], Ka Chun Cheung[2], Simon See[2], and Renjie Wan[1⋆]

[1] Department of Computer Science, Hong Kong Baptist University
[2] NVIDIA AI Technology Center, NVIDIA
xiufenghuang@life.hkbu.edu.hk, {chcheung, ssee}@nvidia.com, renjiewan@hkbu.edu.hk

S1 Overview

This supplementary document provides more discussions, implementation details, and further results that accompany the paper:

- Section S2 explains the uniqueness of our method by comparing with the current NeRF ownership claiming methods under NeRF recolorizations.
- Section S3 explains the effectiveness of applying the Laplace Cumulative Distribution Function (CDF) with learnable parameters.
- Section S4 introduces the details of our reference colors and visualizes their corresponding recolorization results for NeRF. These recolorization methods are applied to different NeRF architectures to validate ownership for the recolorized NeRF.
- Section S5 presents the implementation details of our method, including the network architectures and the training process.
- Section S6 provides additional results, including additional qualitative results of the main paper.

S2 Uniquesness

As shown in Fig. S1, we demonstrate the uniqueness of using our GeometrySticker to claim ownership of a recolorized NeRF model. The current ownership protection methods such as CopyRNeRF [5] and StegaNeRF [3] can only claim the ownership when recolorization is not conducted. However, since the recent developments of NeRF recolorization methods [1, 2, 7], if a model owner *Bob* creates a NeRF model and watermark the model with CopyRNeRF [5] or StegaNeRF [3], the hidden ownership information could be vulnerable when a malicious user applies unauthorized recolorization on the NeRF model. Our GeometrySticker can be robust under different recolorizations. A model owner *Alice* can watermark her NeRF model by GeometrySticker, which can keep the hidden information consistent under different recolorizations and reliably extract the binary message from the recolorized NeRF renderings.

⋆ Corresponding author.

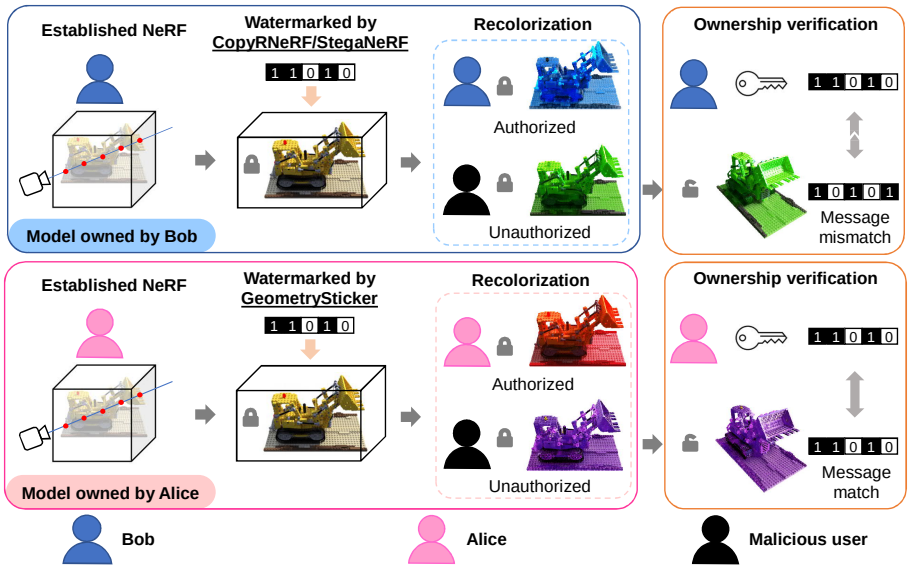


Fig. S1: Illustration of the uniqueness of our method. The first row illustrates the NeRF model owner *Bob* claims the ownership by using CopyRNeRF [5] or StegaNeRF [3]. However, when a malicious user applies unauthorized recolorization on *Bob*'s model, the hidden ownership information can be corrupted and mismatch the original secret messages. The second row illustrates the NeRF model owner *Alice* claims the ownership by GeometrySticker. The NeRF model watermarked by GeometryStricker can be robust to different recolorizations. Even if a malicious user applies unauthorized recolorization on *Alice*'s model, the hidden ownership information can still be reliably extracted and match the original secret messages.

S3 Learnable Laplace CDF

We provide more ablation studies for our learnable Laplace CDF used for the selection of cover medium. As shown in Fig. S2, we calculate the mean μ and deviation β of the geometry values and use the Laplace distribution to model the geometry values distribution of a selected scene. As shown in Fig. S2 **(a)**, attaching messages to all NeRF geometry values can cause obvious distortion since the low geometry values take up the majority of the entire NeRF geometry. We apply the Laplace CDF with the fixed parameters μ and β and the CDF value $\psi = 0.99$ as the threshold to filter large geometry values for messages attachment. As shown in Fig. S2 **(b)**, applying Laplace CDF with calculated parameters can reduce perturbation but still show noticeable distortion. As shown in Fig. S2 **(c)**, our learnable Laplace CDF can adaptatively find an optimized deviation parameter β to adjust the CDF threshold ($\psi = 0.99$) for the selection of cover medium and finally make the perturbation caused by the attached messages imperceivable.

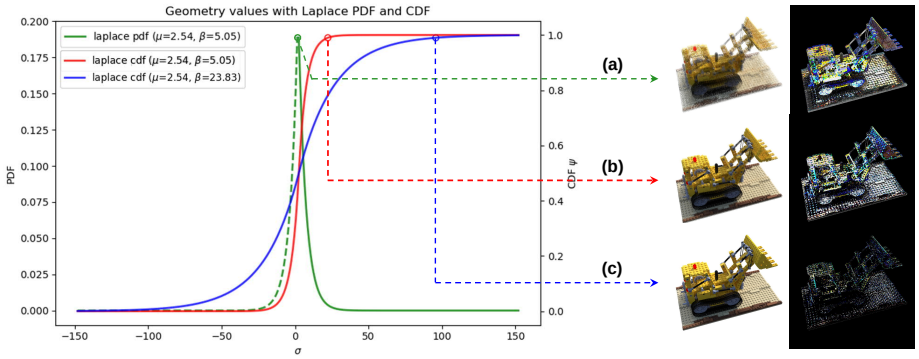
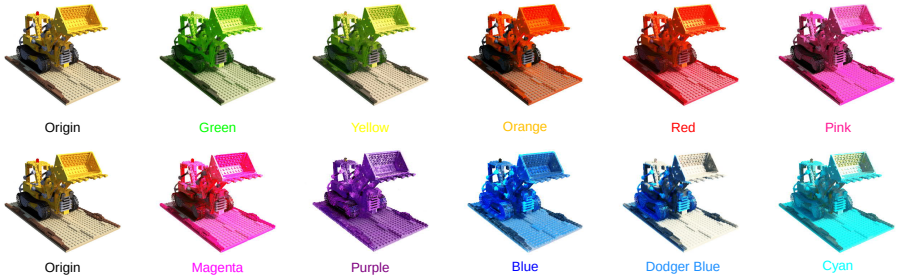


Fig. S2: Message attachment into NeRF geometry values by applying Laplace CDF with different deviation parameters. The geometry values distribution is modeled by a Laplace distribution with the mean μ and deviation β. **(a)** indicates directly attaching messages on all geometry values can cause obvious distortion. **(b)** indicates applying Laplace CDF with fixed μ and β can reduce perturbation but still show noticeable distortion. **(c)** indicates applying Laplace CDF with a learnable deviation parameter β can find an optimized threshold for filtering 3D points and make the distortion imperceivable.

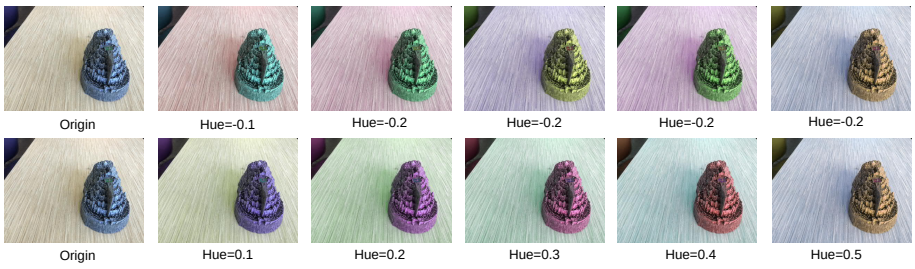
S4 More details on recolorization

We select 10 reference colors from the Standard sRGB / Rec.709 color gamut including green, yellow, orange, red, pink, megenta, purple, blue, dodger blue, cyan. We recolorize the NeRF models by using colors' name for the CLIP-based method or assigning RGB values for the palette-based method. We also convert the NeRF renderings into HSV format to recolorize the images by changing the hue channel. As shown in Fig. S3a, the palette-based method can precisely recolorize NeRF by editing the palette's colors to the reference colors. As shown in Fig. S3b, though the CLIP-based method can roughly conduct the recolorization via the text prompts, the results are uncontrollable since the recolorization under the same prompts may have some differences as shown in Fig. S3d. Thus, it is hard to get the same results for an unwatermarked NeRF model and a watermarked NeRF model. As shown in Fig. S3c, color-jittering is an image-level recolorization by converting images into HSV format and shifting the intensity of the hue channels in a scale of $[-0.5, 0.5]$. For a fair comparison across different baselines, we only use color-jittering in our reconstruction quality computation for PSNR/SSIM and LPIPS in the main manuscript Table 1, since CLIP-based recolorization is uncontrollable and palette-based recolorization is not applicable to CopyRNeRF [5] and StegaNeRF [3]. All the testing set images in the main manuscript Section 5.1 are recolorized for computing reconstruction quality or message extraction bit accuracies.

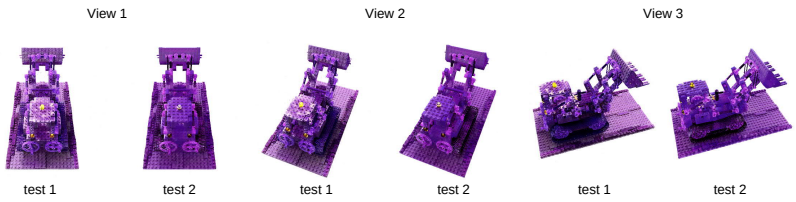
(a) Palette-based recolorization



(b) CLIP-based recolorization

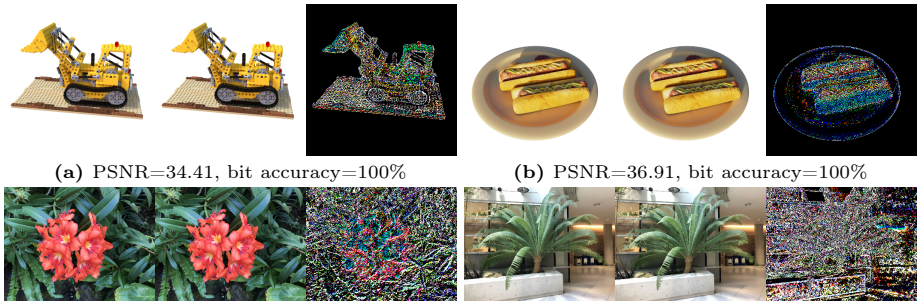


(c) Color-jittering recolorization



(d) CLIP-based recolorization with the same text prompt "purple" can have different results.

Fig. S3: Recolorization results by using different methods: **(a)** is the Palette-based recolorization. The first column is the reference image with the original color palette, and others are recolorized by assigning reference colors to different base colors in the color palette. **(b)** is the CLIP-based recolorization. The first column is the reference image, and the others are recolorized by using the colors' name as the text prompt. **(c)** is the color-jittering recolorization. The first column is the original image and others are recolorized by changing the hue of the original image by shifting the intensities with the range of $[-0.5, 0.5]$ in the hue channel. **(d)** indicates CLIP-based recolorization with the same text prompt can have different results.



(a) PSNR=34.41, bit accuracy=100% **(b)** PSNR=36.91, bit accuracy=100%

(c) PSNR=31.62, bit accuracy=100% **(d)** PSNR=30.70, bit accuracy=100%

Fig. S4: Additional results for different scenes. The message length is 48 bits. We visualize the residual maps between the unwatermarked renderings and the watermarked renderings. From left to right: unwatermarked, GeometrySticker, residual maps (×10).

S5 Implementation details

S5.1 Network achitectures

In our proposed GeometrySticker, the message sticker $\Theta_{\mathbf{m}}$ is an MLP layer. In specific, it has 80 input channels, which are a concatenation of the message $\mathbf{M}$ in 48 dimensions and positional encoding $\gamma_x(\mathbf{x})$ in 32 dimensions. The message sticker $\Theta_{\mathbf{m}}$ has two hidden layers with 64 dimensions and 1-dimensional output for the message embedding m. For the message extractor D_χ, we use the VGG16 network [6] as the backbone feature extractor. An average pooling is then performed, followed by a final linear layer with a fixed output dimension N_b to produce the continuous predicted message $\hat{\mathbf{M}}$. For the watermark classifier C_ϕ, we use a similar architecture with the message extractor D_χ with the VGG16 network [6] as the feature extractor followed by an average pooling layer and a final 1-dimensional layer for classification.

S5.2 Training process

The training process consists of two stages. In the first stage, we establish a NeRF scene by optimizing Θ_σ and Θ_c to get the geometry and color values of the scene according to $\mathcal{L}_{cont}$. In the second stage, we keep the geometry MLP Θ_σ and color MLP Θ_c unchanged and train the message sticker Θ_m and Laplace CDF with the learnable deviation parameter β for message attachment and key points selection. Meanwhile, we train a message extractor D_χ to extract the hidden message from the 2D watermarked renderings. In addition, we also train the watermarking classifier C_ϕ to classify whether the NeRF renderings contain watermarkings or not. The $\mathcal{L}_{cont}$ is measured by the mean squared error between the watermarked rendering images and the ground truth images. The $\mathcal{L}_{msg}$ is a binary cross entropy loss calculated between the embedded messages $\mathbf{M}$ and the extracted messages $\hat{\mathbf{M}}$. The $\mathcal{L}_{cls}$ is a binary cross entropy loss calculated between

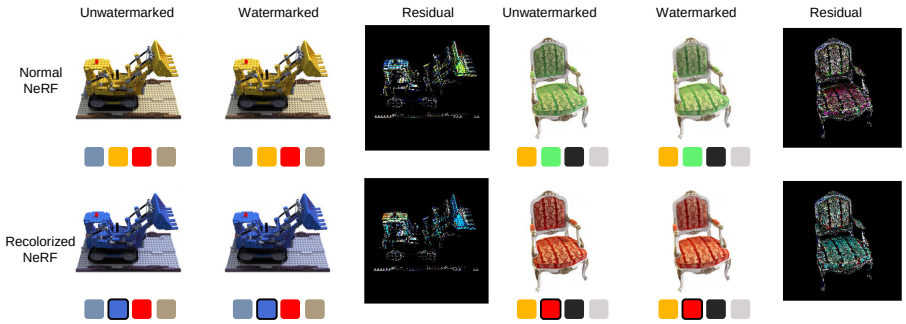


Fig. S5: Residual maps for NeRF renderings before and after palette-based recolorizations. The first row shows the residual maps before palette-based recolorizations. The second row shows the residual maps after palette-based recolorizations. Each residual map shows the differences between the unwatermarked renderings and watermarked renderings by GeometrySticker.

the watermarked rendering image $\mathbf{I}_w$ and the unwatermarked rendering images $\mathbf{I}_u$. $\mathcal{L}_{sparse}$ is the sparsity loss [4] to force the CDF value ψ to be close to either zero or one. The network Θ_m and parameters χ, ϕ and β are optimized with the objective functions $\mathcal{L}_{cont}$, $\mathcal{L}_{msg}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{sparse}$. In every training loop, we attach the message $\mathbf{M}$ with a random camera pose and apply 2D distortions on the watermarked rendering images.

S6 Additional results

We provide additional results to validate the effectiveness of our GeometrySticker. As shown in Fig. S4, we evaluate the qualitative and quantitative results of the reconstruction quality and bit accuracies of our GeometrySticker on the selected scene. The watermarked rendered images have high reconstruction quality with minimal discrepancies compared with the original rendered images. From the residual maps, we can observe that the hidden messages are sparsely embedded into the geometrical structure of the object or scene.

We further validate the consistency of our GeometrySticker under different recolorizations. As shown in Fig. S5, the message perturbation attached by GeometrySticker remains consistent from non-recolorized NeRF models to recolorized NeRF models. These results show our method successfully embeds secret messages into the geometry representation and disentangles them with the color representation, thus claiming ownership under various NeRF recolorizations.

References

1. Gong, B., Wang, Y., Han, X., Dou, Q.: RecolorNeRF: Layer Decomposed Radiance Field for Efficient Color Editing of 3D Scenes. In: Proceeding of the ACM International Conference on Multimedia (MM) (2023) 1

2. Kuang, Z., Luan, F., Bi, S., Shu, Z., Wetzstein, G., Sunkavalli, K.: PaletteNeRF: Palette-based Appearance Editing of Neural Radiance Fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognision (CVPR) (2023) 1
3. Li, C., Feng, B.Y., Fan, Z., Pan, P., Wang, Z.: StegaNeRF: Embedding Invisible Information within Neural Radiance Fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 1, 2, 3
4. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural Volumes: Learning Dynamic Renderable Volumes from Images. ACM Transactions on Graphics (ToG) (2019) 6
5. Luo, Z., Guo, Q., Cheung, K.C., See, S., Wan, R.: CopyRNeRF: Protecting the CopyRight of Neural Radiance Fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023) 1, 2, 3
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 5
7. Wang, C., Chai, M., He, M., Chen, D., Liao, J.: CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognision (CVPR) (2022) 1