Supplementary Material:
CopyRNeRF: Protecting the CopyRight of Neural Radiance Fields

Ziyuan Luo[1,2] Qing Guo[3] Ka Chun Cheung[2,4] Simon See[2] Renjie Wan[1]*
[1]Department of Computer Science, Hong Kong Baptist University
[2]NVIDIA AI Technology Center, NVIDIA
[3]IHPC and CFAR, Agency for Science, Technology and Research, Singapore
[4]Department of Mathematics, Hong Kong Baptist University
ziyuanluo@life.hkbu.edu.hk, guo_qing@cfar.a-star.edu.sg, {chcheung, ssee}@nvidia.com,
renjiewan@hkbu.edu.hk

A. Overview

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

- Section B explains the uniqueness of our method by discussing another message embedding setting.
- Section C introduces the workflow of our CopyRNeRF from watermarked color representation building, to customized rendering, and finally to copyright verification.
- Section D presents the implementation details of our method, including the network architecture and the training process.
- Section E provides additional results, including additional visual results, qualitative results for Table 4 of the main paper, and quantitative results for more lengths of raw bits.

B. Uniqueness of our method

As well as our proposed CopyRNeRF, we have discussed several strategies for protecting the copyright of implicit scene representation constructed by NeRF in our main paper: 1) directly build an implicit representation using watermarked 2D images, and 2) watermark the representation by using the copyright message as a part of the input. Besides their limitations discussed in our main paper, for 1), if the copyright message is to be changed, the whole representation needs to be trained again, which is time-consuming.

We additionally discuss another setting in this document: *why not directly watermark the synthesized 2D images with novel viewpoints*? As outlined in Figure S1, such setting does not protect the model itself. When the model is stolen by malicious users, the unwatermarked rendering images can be easily generated from the stolen model.

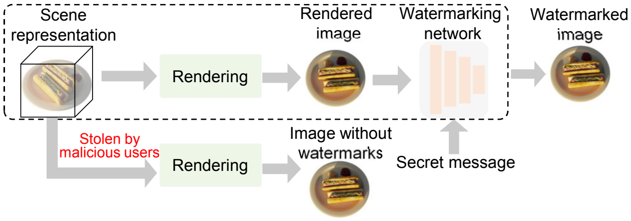


Figure S1: When directly watermarking the synthesized 2D images with novel views, the model weights are not protected. Anyone who stoles the 3D representation may generate 2D images without watermarks by skipping the watermarking network.

Instead, with our watermarked color representation and distortion-resistance rendering, the model weights are protected. If malicious users produce images by different rendering strategies, the copyright of our model can still be protected.

C. Workflow of CopyRNeRF

We would like to introduce more details about the workflow of our CopyRNeRF. A more concise diagram is illustrated in Figure S2 of this supplementary. The representation creator can create the implicit representation based on our descriptions in the main paper. Then, as outlined in Figure S2, the copyright of core model is protected by watermarking messages. Although malicious users can synthesize images with novel viewpoints by applying different rendering approaches to the core model, our method can

[*]Corresponding author.

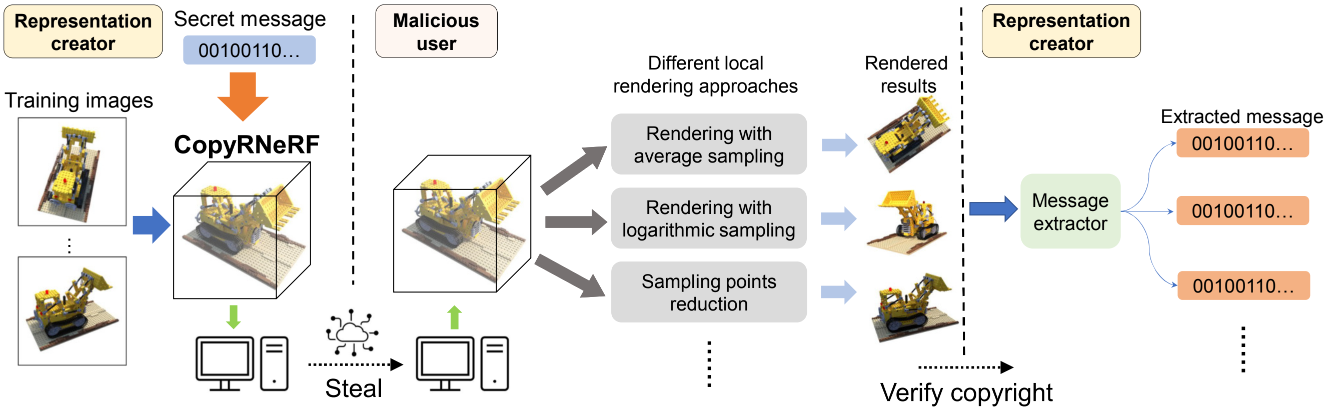


Figure S2: Illustration of workflow of our CopyRNeRF. The creator applies CopyRNeRF to generate a core model from a set of 2D images. Even if the model is stolen and different rendering approaches are applied, the model creator can still use the message extractor to reveal the message from the rendered results to verify the copyright.

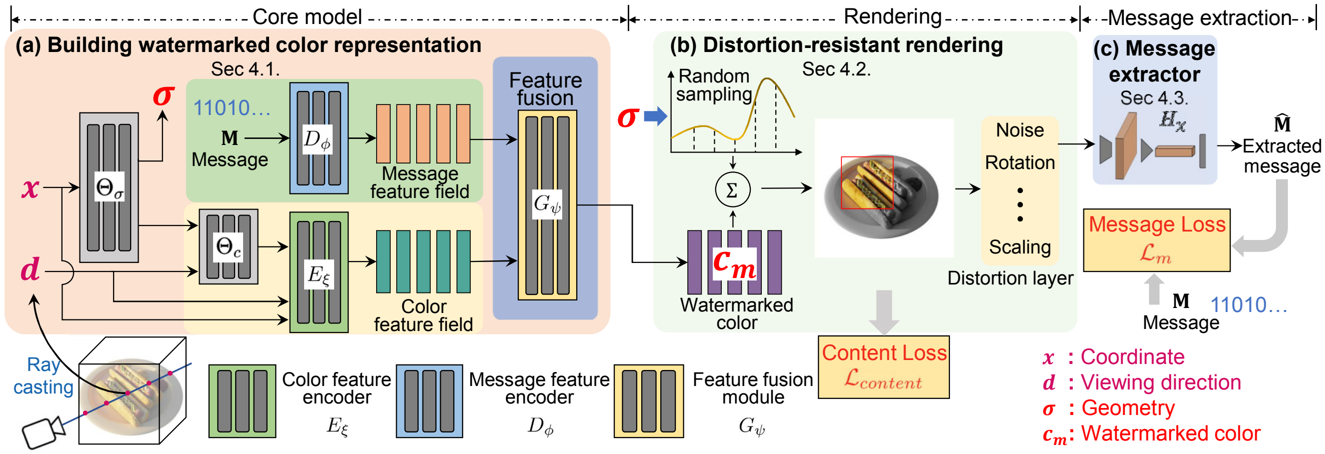


Figure S3: More illustrations of our method. Our method contains five MLPs and one CNN-based network for different purposes. The two MLPs Θ_σ and Θ_c are used to output the geometry σ and the colors $\mathbf{c}$. The watermarked color representation module uses two MLPs, E_ξ and D_ϕ to obtain the color feature field and message feature field, respectively, and then generates message representation by a feature fusion module G_ψ. A CNN-based message extractor H_χ is employed to reveal the message from 2D rendered images.

still ensure that all synthesized images with novel viewpoints are watermarked. Moreover, the trained message extractor can be directly applied to reveal the message from the synthesized images, even when different rendering strategies, distortions, and viewpoints are encountered.

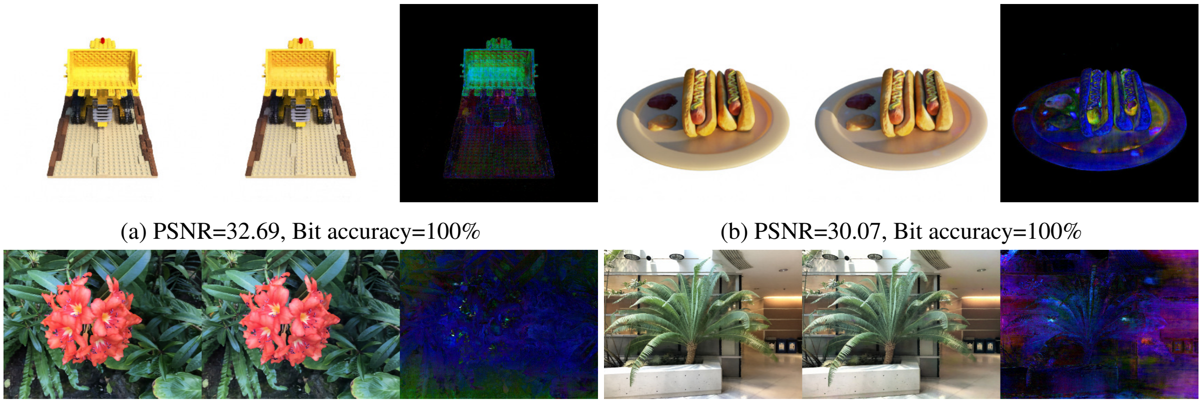
D. Implementation details

D.1. Network architecture

As outlined in Figure S3, our method contains five MLPs for different purposes. A MLP Θ_σ with 256 channels is used to map the position to geometry value and an intermediate feature the medium generation, an then a three-layer MLP Θ_c is applied to output the base colors $\mathbf{c}$.

The module for building watermarked color representation contains three MLPs. Color feature encoder E_ξ is a three-layer MLP to embed $\mathbf{c}$ queried from Θ_c, coordinates $\mathbf{x}$, and viewing directions $\mathbf{d}$ to 256-dimensional color features. Message feature encoder D_ϕ is a two-layer MLP to extract features from messages. After that, a feature fusion module G_ψ is realized by a three-layer MLP to generate the watermarked color from the color feature field and message feature field.

Our message extractor H_χ is with a CNN-based structure [4]. A convolutional layer, a normalization layer, and a ReLU activation function are combined as a base block. The message extractor contains 8 base blocks with 64 filters each and one last block with N_b filters, where N_b is



(a) PSNR=32.69, Bit accuracy=100%

(b) PSNR=30.07, Bit accuracy=100%

(c) PSNR=30.28, Bit accuracy=100%

(d) PSNR=31.68, Bit accuracy=100%

Figure S4: Additional visual results of different scenes. The message length is set to 8. We show the differences between the synthesized results and the ground truth from multi-views. From left to right: ground truth, CopyRNeRF , difference ($\times 10$).

the length of message. A pooling layer is applied to get the average of each dimension and a linear layer is used to produce the final extracted message $\hat{\mathbf{M}}$ with the dimension N_b.

D.2. Training process

The training process consists of three stages. In the first stage, we optimize Θ_σ and Θ_c to get geometry values of the scene according to $\mathcal{L}_{recon}$. The second stage aims to learn a color feature encoder E_ξ, a message feature encoder D_ϕ, and a feature fusion module G_ψ to build a watermarked color representation. Meanwhile, a message extractor H_χ is trained to extract the message from the 2D images rendered by distortion-resistant rendering module. In every training loop, a random camera pose in boundary and a random message $\mathbf{M}$ of dimension N_b are chosen. The content loss $\mathcal{L}_{content}$ is calculated by the rendered results from medium representation and message representation of the same camera pose. The message loss $\mathcal{L}_m$ is the mean squared error between embedded message $\mathbf{M}$ and the extracted message $\hat{\mathbf{M}}$. The parameters ξ, ϕ, ψ, χ are optimized with the objective functions $\mathcal{L}_{content}$ and $\mathcal{L}_m$. In the last training stage, we finetune the message extractor H_χ with E_ξ, D_ϕ, and G_ψ frozen to further improve the bit accuracy.

In every training loop, all the messages in $\{0,1\}^{N_b}$ have the same probability of being randomly selected, ensuring the consideration of all 2^{N_b} messages. When the model is prepared to be shared, a secret message $\mathbf{M}$ in $\{0,1\}^{N_b}$ should be chosen by the model creator as the invisible copyright identity. The results show that our proposed CopyRNeRF can achieve a good balance between bit accuracy and error metric values.

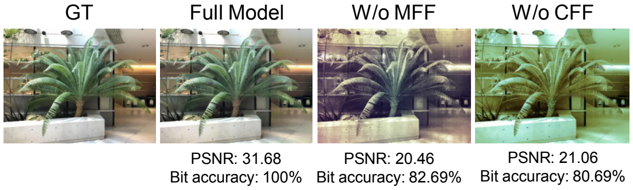


Figure S5: Visual quality comparisons for our full model, our model without Message Feature Field (MFF) and our model without Color Feature Field (CFF).

E. Additional results

E.1. Visual results for CopyRNeRF

We present more qualitative results on Blender dataset [3] and LLFF dataset [2], as shown in Figure S4. Our method clearly reaches a high bit accuracy while maintaining the high-quality novel view synthesis.

E.2. Qualitative results for Table 4

We have discussed the effectiveness of message feature field and color feature field of CopyRNeRF (Section 5.2 of our main paper). We further provide the qualitative evaluations in Figure S5 of this document. We first remove the color feature field and directly combine the color component with the message features, and then remove the message feature field and combine the message directly with the color feature field. In both cases, the models perform poorly in preserving the visual quality of the rendered results.

E.3. Quantitative results for more bit lengths

In this section, we display results for more lengths of raw bits. The results of bit accuracy and reconstruction quality for 8 bits, 16 bits, 32 bits, and 48 bits are shown in Table 1, Table 2, Table 3, and Table 4, respectively.

References

[1] Zhaoyang Jia, Han Fang, and Weiming Zhang. MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 5

[2] 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)*, 2019. 3

[3] 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. In *Proceedings of the European Conference on Computer Vision*, 2020. 3, 5

[4] 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*, 2018. 2, 5

Table 1: Bit accuracies and reconstruction qualities compared with our baselines. ↑ (↓) means higher (lower) is better. We show the results on $N_b = 8$ bits. The results are averaged on all all examples. The best performances are highlighted in **bold**.

	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
Proposed CopyRNeRF	**100%**	**30.28**	**0.934**	0.037
HiDDeN [4]+NeRF[3]	50.25%	27.75	0.926	0.034
MBRS [1]+NeRF [3]	51.38%	29.09	0.929	**0.020**
NeRF with message	63.19%	20.26	0.691	0.117
CopyRNeRF in geometry	68.00%	17.61	0.638	0.147

Table 2: Bit accuracies and reconstruction qualities compared with our baselines. ↑ (↓) means higher (lower) is better. We show the results on $N_b = 16$ bits. The results are averaged on all all examples. The best performances are highlighted in **bold**.

	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
Proposed CopyRNeRF	**91.16%**	26.29	0.910	0.038
HiDDeN [4]+NeRF[3]	50.19%	26.53	0.917	0.035
MBRS [1]+NeRF [3]	50.53%	**28.79**	**0.925**	**0.022**
NeRF with message	52.22%	22.33	0.773	0.108
CopyRNeRF in geometry	60.16%	20.24	0.771	0.095

Table 3: Bit accuracies and reconstruction qualities compared with our baselines. ↑ (↓) means higher (lower) is better. We show the results on $N_b = 32$ bits. The results are averaged on all all examples. The best performances are highlighted in **bold**.

	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
Proposed CopyRNeRF	**78.08%**	26.13	0.896	0.041
HiDDeN [4]+NeRF[3]	50.11%	26.24	0.913	0.038
MBRS [1]+NeRF [3]	49.80%	**28.38**	**0.921**	**0.025**
NeRF with message	50.00%	20.13	0.682	0.122
CopyRNeRF in geometry	54.86%	18.07	0.710	0.143

Table 4: Bit accuracies and reconstruction qualities compared with our baselines. ↑ (↓) means higher (lower) is better. We show the results on $N_b = 48$ bits. The results are averaged on all all examples. The best performances are highlighted in **bold**.

	Bit Acc↑	PSNR ↑	SSIM ↑	LPIPS ↓
Proposed CopyRNeRF	**60.06%**	27.56	0.895	0.066
HiDDeN [4]+NeRF[3]	50.04%	26.16	0.908	0.043
MBRS [1]+NeRF [3]	50.14%	**28.24**	**0.918**	**0.031**
NeRF with message	51.04%	22.12	0.837	0.125
CopyRNeRF in geometry	53.36%	23.71	0.871	0.092