GuardSplat: Efficient and Robust Watermarking for 3D Gaussian Splatting

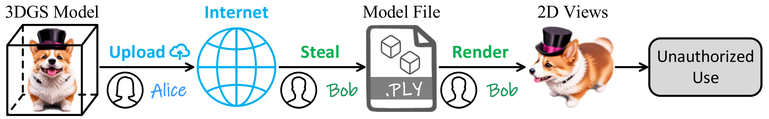
Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie*

Homepage

Project Page

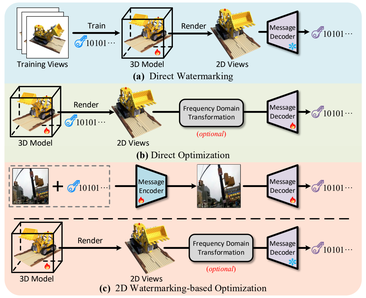
CVPR Nashville JUNE 11-15, 2025

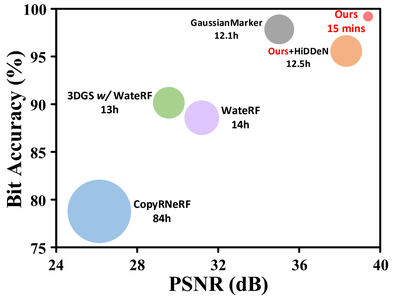
Background



3DGS creates impressive 3D assets, but the risk of unauthorized use threatens creators, requiring to protect the copyright.

Motivation





Existing methods (**left**) are inadequate for protecting the copyright of 3DGS assets considering the demands of *security*, *capacity*, *invisibility*, and optimization *efficiency* (**right**).

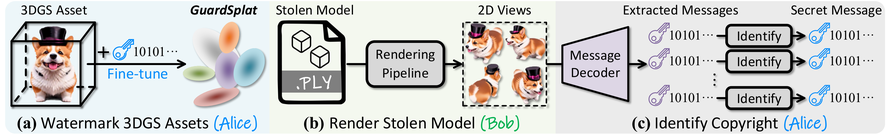
How to design an efficient method to protect and identify the copyright of 3DGS assets?

Contributions

- We present **GuardSplat**, a new and efficient watermarking framework to protect the copyright of 3DGS assets.
- We decouple the optimization of 2D watermarking and tailor a message embedding method for 3DGS rendering pipeline.
- Experiments show that we achieve superior performance and efficiency (decoder: **5mins**, watermarking: **10mins**).

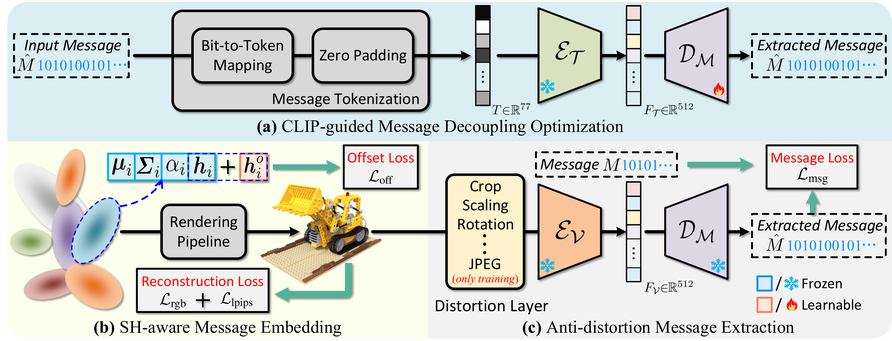
Code is at https://github.com/NarcissusEx/GuardSplat

Overview



Application scenarios. **(a)** The owners (Alice) watermark 3DGS models using **GuardSplat**. **(b)** If Thieves (Bob) render views for unauthorized use, **(c)** Alice can extract messages (purple key) to identify the copyright.

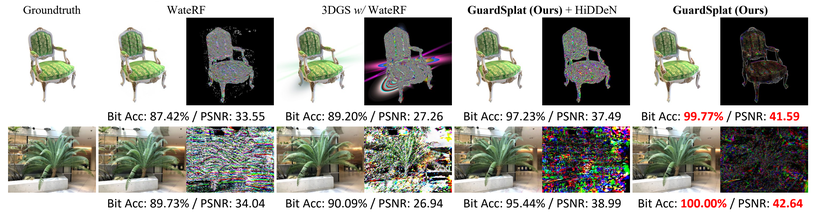
Framework



(a) We build an autoencoder pipeline of binary message M to train the message decoder $\mathcal{D}_\mathcal{M}$ using CLIP. **(b)** We employ a set of learnable SH offset $\boldsymbol{h}_i^o$ to render the watermarked views. **(c)** A differentiable distortion layer is used to simulate various visual distortions during optimization.

Experiments & Analysis

Visual Comparison



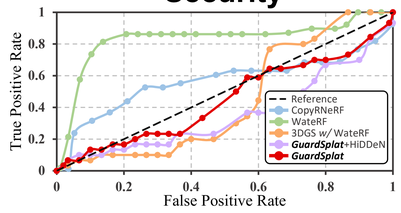
Experiments & Analysis

Capacity & Invisibility

Table 1. **Comparisons of the start-of-the-art methods** on Blender [32] and LLFF [31] datasets for bit accuracy and reconstruction qualities w.r.t various message lengths. **Bold** text indicates the best performance in this table.

Methods	16 bits				32 bits				48 bits			
	Bit Acc	PSNR	SSIM	LPIPS	Bit Acc	PSNR	SSIM	LPIPS	Bit Acc	PSNR	SSIM	LPIPS
NeRF-based Watermarking Methods												
CopyRNeRF [28]	91.16	26.29	0.9100	0.0380	78.08	26.13	0.8960	0.0410	60.06	27.56	0.8950	0.0660
WateRF [14]	95.67	32.79	0.9480	0.0330	88.58	31.19	0.9360	0.0400	85.82	30.86	0.9300	0.0400
3DGS Built on Watermarked Images												
3DGS [19] + CIN [29]	56.73	31.95	0.9194	0.1027	53.13	31.74	0.9279	0.0944	55.78	30.25	0.9139	0.1120
3DGS [19] + SSL [9]	58.94	36.51	0.9737	0.0094	61.85	35.24	0.9706	0.0179	58.79	35.88	0.9710	0.0123
3DGS [19] + HiDDeN [69]	63.07	31.59	0.9790	0.0171	52.46	34.51	0.9682	0.0205	53.08	33.42	0.9687	0.0299
3DGS [19] + DwtDctSvd [34]	55.44	34.78	0.9582	0.0399	53.15	32.32	0.9477	0.0547	51.83	31.09	0.9302	0.0704
3DGS [19] + StegaStamp [52]	79.72	35.52	0.9697	0.0181	82.36	34.04	0.9601	0.0295	83.97	32.54	0.9523	0.0406
3DGS optimized by adverting all attributes (Offset$_{all}$)												
GaussianMarker [13]	99.36	34.42	0.9822	0.0124	98.85	33.98	0.9788	0.0163	98.25	32.12	0.9723	0.0234
3DGS [19] w/ WateRF [14]	92.89	31.01	0.9678	0.0475	90.15	29.56	0.9611	0.0492	87.30	29.13	0.9562	0.0534
GuardSplat (Ours) + 2D Watermarking Decoders												
GuardSplat (Ours) + CIN [29]	95.75	37.88	0.9762	0.0092	93.35	37.42	0.9726	0.0109	92.77	37.10	0.9689	0.0124
GuardSplat (Ours) + SSL [9]	99.50	40.92	0.9935	0.0020	98.60	38.95	0.9920	0.0028	98.14	38.51	0.9909	0.0030
GuardSplat (Ours) + HiDDeN [69]	98.75	40.48	0.9909	0.0025	95.58	38.32	0.9897	0.0025	93.29	38.56	0.9886	0.0032
GuardSplat (Ours) + StegaStamp [52]	99.00	38.55	0.9903	0.0035	98.28	38.63	0.9914	0.0030	97.23	38.27	0.9892	0.0037
GuardSplat (Ours)	**99.64**	**41.55**	**0.9957**	**0.0017**	**99.04**	**39.40**	**0.9939**	**0.0022**	**98.29**	**38.90**	**0.9923**	**0.0028**

Security

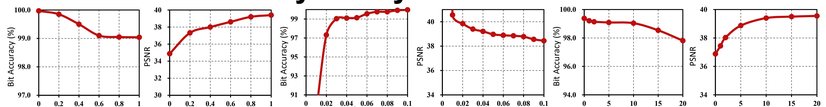


Robustness

Table 2. **Comparisons of the start-of-the-art methods** on Blender [32] and LLFF [31] datasets for bit accuracy w.r.t various distortion types. We show the results on 16-bit messages. **Bold** text indicates the best performance in this table.

Methods	None	Noise (μ=0.1)	Rotation ($\pm$90)	Scaling (1/2$\sim$2)	Blur (σ=0.1)	Crop (40%)	Brightness (0.5$\sim$1.5)	JPEG (10% quality)	VAE Attack [68] Bmshj2018	Combined
CopyRNeRF [28]	91.16	90.04	88.13	89.33	90.06	–	–	–	–	–
WateRF [14]	95.67	95.36	93.13	93.29	95.25	95.40	90.91	86.99	51.73	84.12
3DGS [19] w/ WateRF [14]	92.89	87.35	88.28	90.33	91.92	89.07	88.71	88.49	55.48	86.37
GaussianMarker [13]	99.36	99.13	70.84	97.89	94.40	98.52	95.78	86.22	52.00	83.49
GuardSplat (Ours) + CIN [29]	95.75	94.87	90.89	94.50	95.16	93.82	93.97	88.61	49.25	84.03
GuardSplat (Ours) + SSL [9]	99.50	99.57	86.78	84.53	98.79	77.54	94.31	92.99	47.42	74.85
GuardSplat (Ours) + HiDDeN [69]	98.75	96.63	90.02	95.93	94.87	97.25	94.97	90.04	53.14	88.70
GuardSplat (Ours) + StegaStamp [52]	99.00	98.38	53.21	95.17	98.17	51.34	95.48	88.81	80.12	64.75
GuardSplat (Ours)	**99.64**	**99.60**	**94.56**	**98.75**	**99.27**	**98.71**	**97.46**	**94.70**	**82.35**	**93.38**

Parameter Sensitivity Analysis



Performance across different hyper-parameter values with $N_L = 32$ bits on the Blender and LLFF datasets.

Ablation Study

Table 3. **Various embedding methods** on Blender [32] and LLFF [31] datasets with N_L=32 bits. **Bold** text denotes the best score.

	Bit Acc	PSNR	SSIM	LPIPS
Offset$_{all}$	98.79	36.56	0.9804	0.0123
Offset$_{dc}$	74.59	36.98	0.9828	0.0146
Offset$_{rest}$	98.25	38.70	0.9892	0.0077
SH-aware (Ours)	99.04	39.40	0.9939	0.0022

Table 4. **Different loss combinations** with N_L=32 bits on Blender [32] and LLFF [31] datasets. The first row denotes the original 3DGS, and $\mathcal{L}_{recon} = \mathcal{L}_{rgb} + \mathcal{L}_{lpips}$ indicates the reconstruction loss.

$\mathcal{L}_{msg}$	$\mathcal{L}_{recon}$	$\mathcal{L}_{off}$	Bit Acc	PSNR	SSIM	LPIPS
			53.41	*inf*	1.0000	0.0000
✔			100.00	31.79	0.9604	0.0379
✔	✔		99.26	36.88	0.9831	0.0101
✔	✔	✔	99.04	39.40	0.9939	0.0022