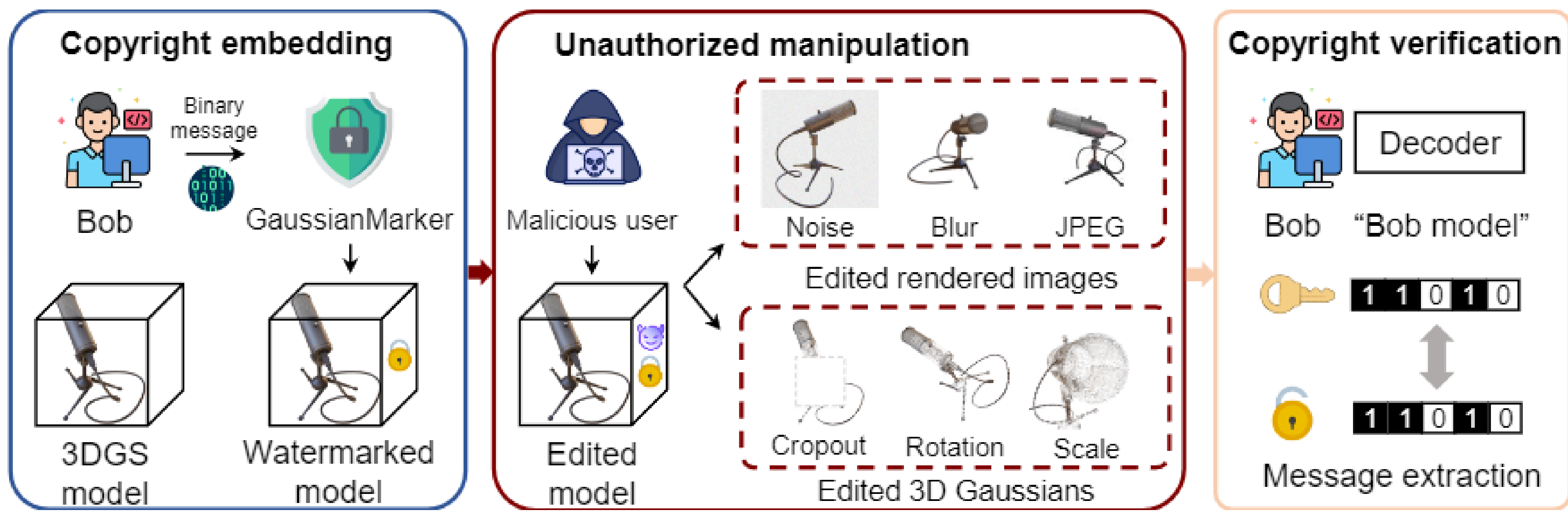
GaussianMarker: Uncertainty-Aware Copyright Protection of 3D Gaussian Splatting

Xiufeng Huang[1, 2], Ruiqi Li[1], Yiu-ming Cheung[1], Ka Chun Cheung[2], Simon See[1], Renjie Wan[1 *]

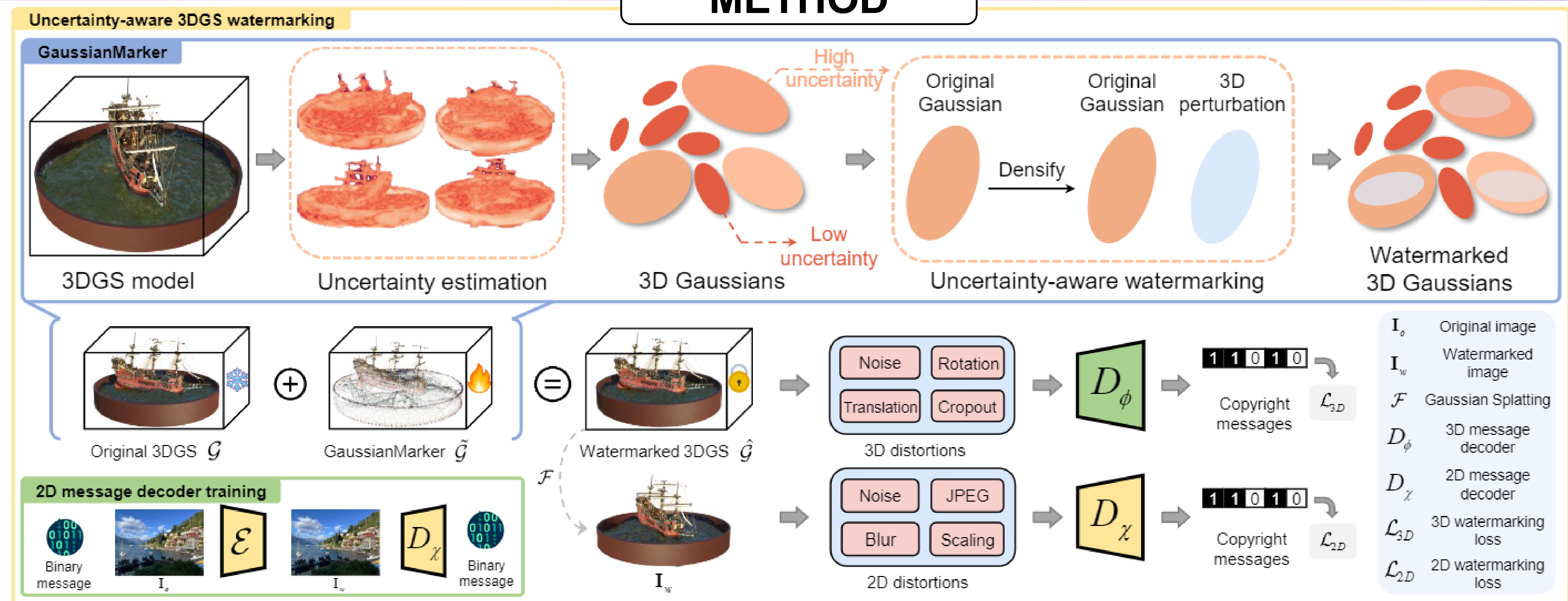
1. Department of Computer Science, Hong Kong Baptist University
2. NVIDIA AI Technology Center

INTRODUCTION



Users can apply GaussianMarker to protect their 3DGS models. If unauthorized users maliciously edit the watermarked 3DGS model, the model owners can reliably retrieve the copyright message from the altered 3D Gaussian parameters or rendered 2D images to verify ownership.

METHOD



Our method preserves 3D Gaussians with low uncertainty, maintaining the geometric structure to ensure imperceptible perturbations and high message extraction accuracy.

FORMULATION

3D Gaussian parameters with high uncertainty are more tolerant to external perturbations, we select parameters with high uncertainty to incorporate ownership messages.

The uncertainty of the 3DGS parameters can be estimated by the Hessian matrix as the approximated Fisher information.

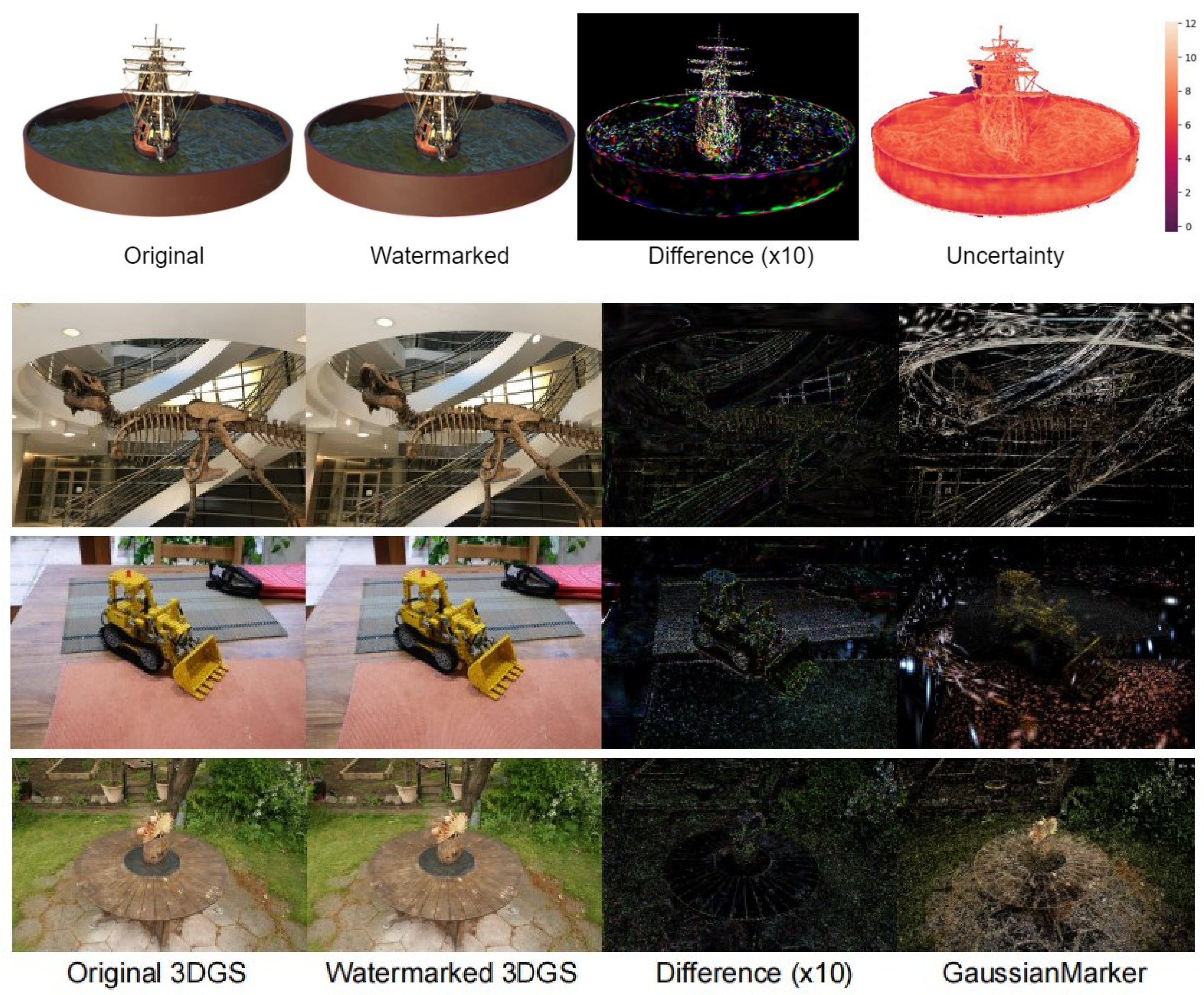
As Fisher Information is additive, we compute the model uncertainty U by summing the Hessians ofmodel parameters across all different views in the training dataset D:

$$\mathbf{U} = \sum_{i=1}^{N} \mathbf{H}[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}^*]$$

where i is the index and N is the total samples in D, and all Gaussian parameters are used to calculatethe uncertainty of the 3DGS model:

$$\mathbf{H}[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}^*] = \mathbf{H}\left[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}_\mu^*\right] + \mathbf{H}\left[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}_\mathrm{R}^*\right] + \mathbf{H}\left[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}_\mathrm{S}^*\right] + \mathbf{H}\left[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}_\mathrm{c}^*\right] + \mathbf{H}\left[\mathbf{I} \mid \mathbf{V}, \boldsymbol{\theta}_\alpha^*\right].$$

VISUALIZATION



RESULTS

Dataset	Method	PSNR/SSIM↑	LPIPS↓	Bit accuracy ↑ (%) None	Noise ($\nu = 0.1$)	JPEG ($Q = 50$)	Scaling ($s \leq 25\%$)	Blur ($\xi = 0.1$)
Blender	CopyRNeRF [4]	30.29/0.8878	0.0813	60.83	59.92	58.52	57.44	60.22
	HiDDeN [11] + 3DGS [1]	28.96/0.8812	0.0829	50.19	49.84	50.12	50.09	50.16
	3DGS [1] w/ messages	22.65/0.8066	0.1584	80.22	78.66	75.80	78.08	79.64
	3DGS [1] w/ fine-tuning	28.17/0.9047	0.0878	67.13	67.06	63.43	64.04	66.38
	Ours	**31.53/0.9082**	**0.0759**	**97.91**	**96.93**	**91.66**	**96.17**	**97.43**
LLFF	CopyRNeRF [4]	24.03/0.7747	0.2575	60.77	60.23	58.06	58.89	60.35
	HiDDeN [11] + 3DGS [1]	27.17/0.8543	0.1210	48.26	48.14	46.26	46.89	48.12
	3DGS [1] w/ messages	24.82/0.8452	0.1310	83.33	82.39	79.17	81.04	83.18
	3DGS [1] w/ fine-tuning	26.62/0.8566	0.1117	60.61	59.99	55.49	57.52	60.40
	Ours	**28.61/0.8930**	**0.0999**	**98.33**	**97.83**	**91.45**	**95.89**	**98.23**
MipNeRF360	CopyRNeRF [4]	22.47/0.8053	0.4825	58.55	57.22	55.26	55.80	57.59
	HiDDeN [11] + 3DGS [1]	27.20/0.8151	0.2143	48.75	48.03	45.93	47.75	48.56
	3DGS [1] w/ messages	24.84/0.7992	0.1705	77.08	76.75	74.26	75.54	77.00
	3DGS [1] w/ fine-tuning	27.04/0.8452	0.1357	61.67	61.45	59.94	60.56	61.51
	Ours	**29.16/0.8808**	**0.1197**	**97.32**	**97.01**	**90.77**	**95.32**	**97.18**

Table 1: Reconstruction qualities and bit accuracy compared with different baselines. PSNR/SSIM and LPIPS are computed between the original and watermarked rendered images. The results are computed on the average of all examples.

Method	Geometry difference $\mathcal{L}_1 Diff \downarrow$	SNR↑	Bit accuracy ↑ (%) None	Noise ($\sigma = 0.1$)	Translation ($t = [0, 1000]^3$)	Rotation ($r = \pm\pi/6$)	Cropout ($cr = 0.1$)
HiDDeN [11] + 3DGS [1]	0.00912	40.90	68.20	67.65	67.32	66.67	64.24
3DGS [1] w/ messages	0.10513	32.93	85.41	84.91	85.35	81.52	79.57
3DGS [1] w/ fine-tuning	0.01829	37.24	69.79	69.70	68.78	65.88	64.84
Ours w/ 2D decoder	**0.00003**	**43.23**	97.85	57.05	59.07	53.88	48.23
Ours w/ 3D decoder	**0.00003**	**43.23**	**100**	**99.91**	**98.95**	**95.83**	**92.70**

Table 2: Geometry difference and bit accuracy compared with different baselines. $\mathcal{L}_1$ distance and SNR are computed between the original and watermarked 3D Gaussians. The results are computed on the average of all examples from Blender, LLFF, and MipNeRF360.