Where, What, Why: Toward Explainable 3D-GS Watermarking

Supplementary Material

A. Overview

In this supplementary material, we provide further discussions, implementation details, and results as follows:

- **Section B** details the implementation of the *watermark decoder* and provides a brief comparison with alternative decoders.
- **Section C** validates the effectiveness of the *Trio-Experts* module through ablation studies.
- **Section D** presents Trio-free Oracle Hit@K Evaluation.
- **Section E** shows the performance of our decoupled fine-tuning under different settings of the two loss weights.
- **Section F** provides additional qualitative results on LLFF [7], Blender [8], and Mip-NeRF 360 [9].

B. Decoder Details

In 2D digital watermarking [1, 6, 10, 11], the primary goal is to ensure that the hidden message can be decoded only from images in which it was embedded. In 3D digital watermarking, by contrast, the goal is to ensure that watermarked renderings remain decodable from as many random viewpoints as possible. This imposes a dual challenge: it stresses both the 3D watermarking model and the 2D decoder used for extraction.

We adopt as our baseline the pretrained HiDDeN [11] decoder released with 3D-GSW [2], trained on {32, 48, 64}-bit payloads using the MS-COCO dataset [5]. In testing, we observe strong performance on the complex Mip-NeRF 360 dataset [9], but degradation on background-free Blender [8] scenes and sparsely captured LLFF [7] scenes. To address this, we fine-tune the baseline decoder on 20% of Arb-Objaverse [4]. For a fair comparison and to mitigate overfitting, we evaluate on LLFF and Blender, testing both watermarked and non-watermarked 3D-GS [3] models. Results are reported in Table 1.

Method	Bit Acc↑ (w/M)	Bit Acc↓ (w/o M)
Pretrained Decoder	95.24	52.08
Finetuned Decoder	**97.85**	**50.11**

Table 1. Comparison between the pretrained HiDDeN decoder and our fine-tuned decoder on Blender and LLFF. We report bit accuracy on watermarked data (32-bit payload) and on non-watermarked data.

For a more objective evaluation of our decoder, we additionally test CIN decoder [6] on Mip-NeRF 360, LLFF, and Blender with a 32-bit payload. For fairness, we allow part of the decoder parameters to be jointly trained during watermark embedding. Results are shown in Table. 2.

Method	PSNR↑	SSIM↑	LPIPS↓	Bit Acc↑
Ours + CIN	33.56	0.975	0.049	94.46
Ours + HiDDeN	**35.98**	**0.982**	**0.041**	**98.37**

Table 2. Comparison of our framework combined with CIN and HiDDeN. Experiments use a 32-bit payload on Blender, LLFF, and Mip-NeRF 360.

The experimental results show that fine-tuning the HiDDeN decoder is well-justified: it improves both rendering quality and bit accuracy, while also enhancing robustness on sparsely captured and background-free datasets.

C. Ablation Study on Trio-Experts

To validate the effectiveness of our Trio-Experts prior (R_1, R_2, R_3), we conduct a hit-rate analysis against three objective ground-truth attributes: **High-Frequency (HF)**, which measures whether a Gaussian lies in FFT-identified high-frequency regions; **Safe Modifiability (SAFE)**, which assesses whether it can be safely updated based on rendering contribution (gradient $\leq 8 \times 10^{-8}$); and **Multi-View Visibility (V)**, which represents its visibility ratio across training views. For each top-$K\%$ subset ranked by the Trio score (a weighted combination of R_1, R_2, and R_3), we measure the alignment rate between selected Gaussians and these ground-truth attributes. We further evaluate a one-shot corrected variant that incorporates immediate priors (HF, LC, MV) into the Trio score. As shown in Fig. 1, Trio-based selection achieves 60–92% average hit rates, demonstrating that the priors effectively capture scene-aware geometric stability, appearance sensitivity, and spatial redundancy.

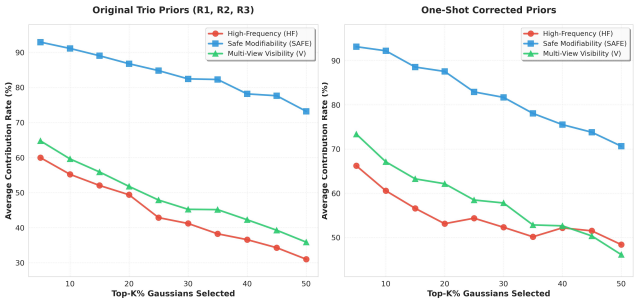


Figure 1. **Ablation study on Trio-Experts Score.** Top-K Gaussians selected by R_1, R_2, and R_3 are compared with ground-truth Gaussians in high-frequency regions, safely modifiable areas, and multi-view visible regions to validate the reliability of the Trio-Experts Score.

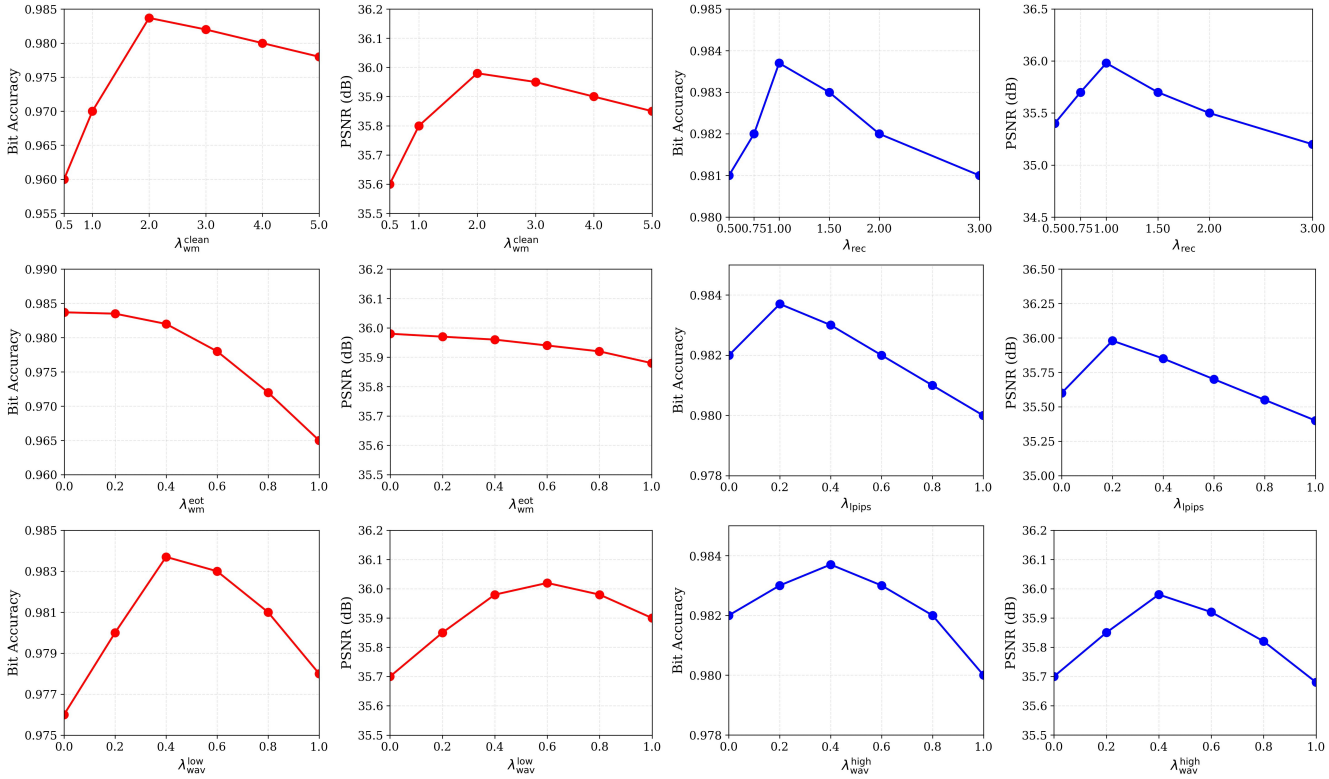


Figure 2. **Ablation on loss weights.** Evaluated on Mip-NeRF 360, LLFF, and Blender with a 32-bit payload. Red curves denote watermark-side weights ($\lambda_{\text{wm}}^{\text{clean}}, \lambda_{\text{wm}}^{\text{eot}}, \lambda_{\text{wav}}^{\text{low}}$); blue curves denote visual-side weights ($\lambda_{\text{rec}}, \lambda_{\text{lpips}}, \lambda_{\text{wav}}^{\text{high}}$).

D. Trio-free Oracle Hit@K Evaluation

To verify whether our Trio-based carrier selection aligns with an oracle derived from *measured* watermarking effects, we conduct a Trio-free Oracle Hit@K evaluation. Specifically, we build a pseudo ground-truth carrier set $\mathcal{C}^*$ by probing each Gaussian's impact and ranking candidates by $\uparrow \Delta\text{BA}$ and $\downarrow \Delta\text{PSNR}$, while retaining only those satisfying $\text{BA} \geq 0.95$ and $\text{PSNR} \geq 32$. We then report $\text{Hit@}K = |\hat{\mathcal{C}} \cap \mathcal{C}^*|/K$ over 5 Mip-NeRF 360 scenes $\times$ 3 random seeds with a 32-bit payload. As shown in Tab. 3, the full model achieves the highest Hit@K, substantially outperforming single-cue variants and the ablation without uncertainty U, indicating that the Trio evidence with uncertainty attenuation more reliably identifies effective and safe watermark carriers.

Table 3. **Oracle carrier Hit@K.** Evaluated on 5 Mip-NeRF 360 scenes $\times$ 3 seeds with a 32-bit payload (mean $\pm$ std).

	None	Geo-only	App-only	Red-only	W/O U	Full
Hit@K $\uparrow$	0.09	0.64	0.73	0.56	0.68	**0.87**

E. Ablation Study on Loss Weights

Although the decoupled finetuning framework is designed to mitigate the training conflict between rendering quality and watermark accuracy, the watermark and visual losses are only weakly coupled through low-frequency constraints. As a result, the optimizer lacks a unified descent direction and cannot expose cross-step gradient conflicts between the two updates. Therefore, selecting appropriate loss weights is crucial for our task. We evaluate this on Mip-NeRF 360, LLFF, and Blender with a 32-bit payload, using three random seeds; the results are shown in Fig. 2.

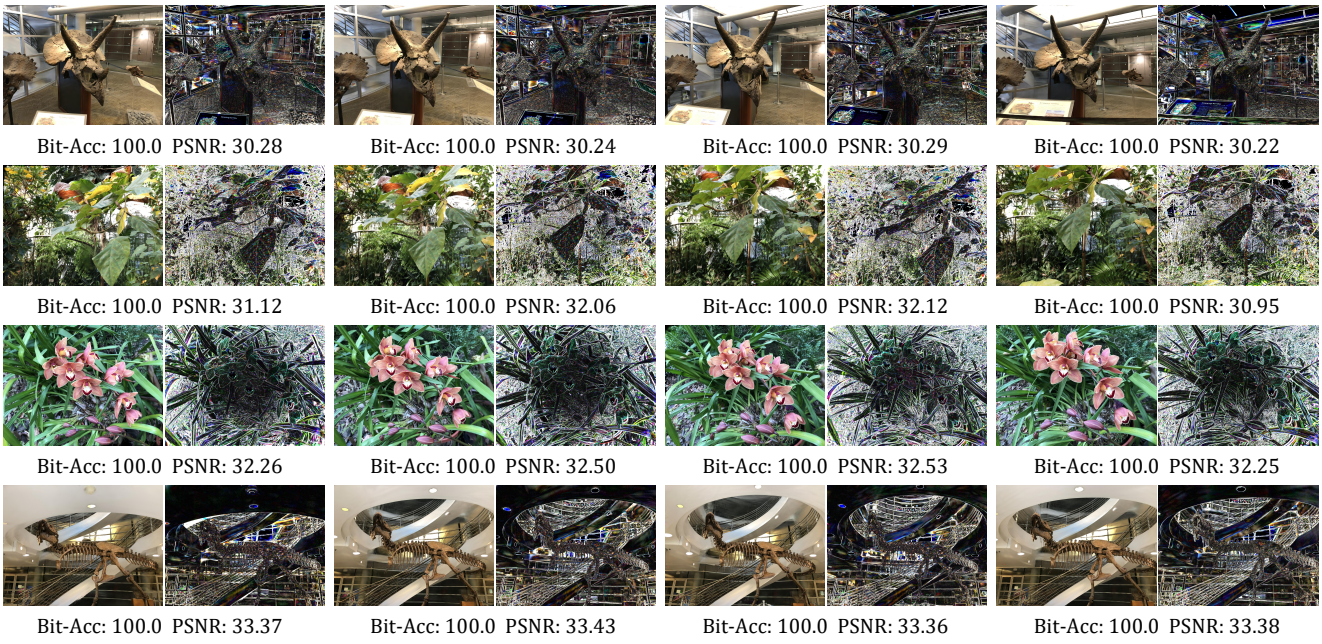
The results not only validate the experimental setup reported in the main paper (loss weights $\lambda_{\text{rec}} = 1.0$, $\lambda_{\text{lpips}} = 0.2$, $\lambda_{\text{wav}}^{\text{high}} = 0.3$, $\lambda_{\text{wm}}^{\text{clean}} = 2.0$, $\lambda_{\text{wm}}^{\text{eot}} = 0.6$, and $\lambda_{\text{wav}}^{\text{low}} = 0.4$), but also demonstrate that, under appropriately tuned weights, the decoupled framework can jointly optimize rendering quality and bit accuracy while effectively avoiding mode collapse.

F. Additional Qualitative Results

In this section, we provide additional 32-bit watermarking results from Mip-NeRF 360, LLFF, and Blender that are not shown in the main paper.

(a) **Rendering quality on Mip-NeRF 360 (32-bit payload).** Rendered outputs and difference maps (magnified $\times 8$).

(b) **Rendering quality on Blender (32-bit payload).** Rendered outputs and difference maps (magnified $\times 8$).



(c) **Rendering quality on LLFF (32-bit payload).** Rendered outputs and difference maps (magnified $\times 8$).

Figure 3. **Additional qualitative results (32-bit payload).** (a) Mip-NeRF 360, (b) Blender, (c) LLFF. Difference maps are magnified $\times 8$.

References

[1] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022. 1

[2] Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting for robust watermarking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5938–5948, 2025. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[4] Zhibing Li, Tong Wu, Jing Tan, Mengchen Zhang, Jiaqi Wang, and Dahua Lin. IDArb: Intrinsic decomposition for arbitrary number of input views and illuminations. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 1

[6] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1532–1542, 2022. 1

[7] 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. 1

[8] 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

[9] Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ricardo Martin-Brualla, and Jonathan T. Barron. MultiNeRF: A Code Release for Mip-NeRF 360, Ref-NeRF, and RawNeRF, 2022. 1

[10] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[11] 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. 1