# MSVS: MULTI-SHELL VIEWPOINT SAMPLING FOR COMPREHENSIVE EVALUATION OF 3D WATERMARKING

*Tomoya Matsubara*[1], *Lingfeng Yao*[2], *Chenpei Huang*[2], *Miao Pan*[2], *Hideo Saito*[1]

[1] Keio University, [2] University of Houston
tomoya.matsubara@hvrl.ics.keio.ac.jp

## ABSTRACT

Reliable copyright protection for 3D assets requires watermark verification across arbitrary viewpoints. However, existing evaluations rely on dataset splits or ad-hoc camera samplings that overlook failure cases. We introduce Multi-Shell Viewpoint Sampling (MSVS), which ensures uniform, distance-aware coverage via concentric, visibility-bounded shells and spherical sampling. MSVS reveals substantially lower bit accuracy even on high-quality renderings that an adversary would prefer. Motivated by the failure cases exposed by MSVS, we further propose a greedy subsampling strategy that selects training views guided by a locality-aware kernel. For 3D-GSW, greedy subsampling improves MSVS bit accuracy by $+0.020$ on the Blender dataset and $+0.056$ on the Stanford-ORB dataset over random selection, and the gains persist under common image attacks. MSVS establishes a comprehensive benchmark for 3D watermark evaluation, while greedy subsampling provides an efficient strategy to enhance watermark protection. [1]

***Index Terms***— Gaussian Splatting, Digital Watermarking, Copyright Protection, Novel View Synthesis, Spherical Sampling

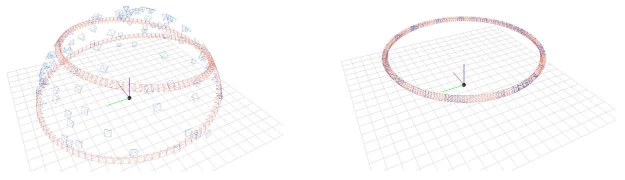
## 1. INTRODUCTION

Creating high-quality 3D assets is essential across applications and industries, including games, film, autonomous driving, and the Metaverse. However, the process has historically required substantial time and specialized expertise. Recent advances in novel view synthesis, such as Neural Radiance Fields (NeRF) [1], 3D Gaussian Splatting (3DGS) [2], and their extensions [3, 4, 5], have begun to lower this barrier. As these techniques mature, sharing user-generated 3D assets online may soon be as effortless as sharing videos, images, or audio. As sharing scales, mechanisms are needed to ensure attribution and preserve creators' incentives, since 3D assets can be re-rendered and repurposed across scenes, making uncredited reuse easy and obscuring provenance.

To detect unauthorized use of 3D assets, recent work combines neural-rendering-based generation with digital watermarking [6, 7, 8, 9, 10, 11, 12, 13]. These methods embed a payload with minimal impact on rendering quality while enabling robust recovery under common attacks (compression, noise, cropping, etc.). Payloads range from images [11] and bit strings [6, 7, 9, 10] to videos [12]; performance for bit strings is typically reported as bit accuracy with task-specific decoders such as HiDDeN [14] and MBRS [15].

However, there is no standardized protocol for selecting camera viewpoints when evaluating bit accuracy. For example, 3D-GSW [6], WateRF [8], and GaussianMarker [9] use the official train/test split on the Blender dataset [1], whereas on mip-NeRF360 [16] they

[1]Project page: https://tomoya-matsubara.github.io/we3/



(a) Blender [1] (b) GuardSplat [7]

**Fig. 1**. Camera viewpoints in the training set (blue) and test set (orange). GuardSplat cameras are generated using the official implementation's default parameters: $r = 4.031128874, \theta \in [-180°, 180°], \phi = -30°$ (spherical coordinates; $r$ radius, $\theta$ azimuth, $\phi$ elevation).

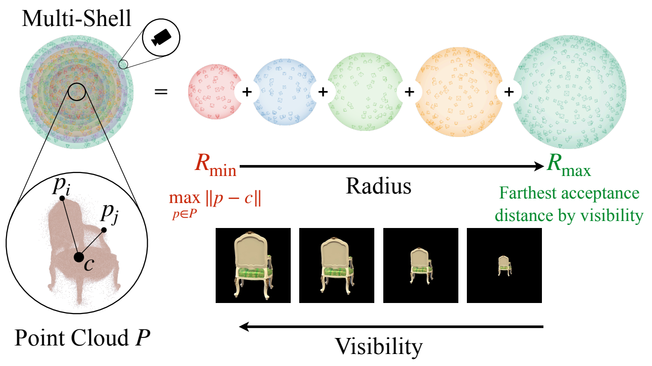


**Fig. 2**. Overview of the Multi-Shell Viewpoint Sampling (MSVS).

form the test set by subsampling every eighth image. Both protocols stem from dataset conventions intended to assess novel-view rendering quality rather than to comprehensively cover the viewpoint space. By contrast, GuardSplat [7] places cameras at spherical coordinates where radius and elevation are drawn at random from specified ranges, and azimuth is evenly spaced. Neither strategy covers the viewpoint space comprehensively, as shown in Figure 1. However, this is not a valid evaluation of 3D watermarking methods, which in principle requires high bit accuracy across all viewpoints. Since attackers can render from arbitrary viewpoints, such blind spots directly undermine copyright protection.

To reduce such blind spots in performance evaluation, we introduce **Multi-Shell Viewpoint Sampling (MSVS)**, a protocol for bit accuracy evaluation that addresses the lack of comprehensive viewpoint coverage. MSVS implements this by placing concentric, visibility-bounded shells around the object and sampling viewpoints on each shell via spherical sampling [17], as shown in Figure 2. By construction, MSVS provides near-uniform, distance-aware coverage and avoids overlooking viewpoint-dependent failures. In our experiments, bit accuracy on MSVS-sampled viewpoints is consistently and substantially lower than on the dataset-provided test views common in prior work, underscoring the importance of comprehensive viewpoint coverage. Motivated by this finding, we also propose a simple yet effective greedy subsampling strategy that augments the training set to improve watermark coverage. We model the gain from adding a training viewpoint with a nondecreasing kernel over evaluation views, and greedily select viewpoints that maximize an objective defined based on the sum of bit accuracies across evaluation cameras. Empirically, this method yields higher bit accuracy on MSVS generated test sets than a random subsampling baseline. Our main contributions are summarized as follows:

- **MSVS: a comprehensive evaluation protocol.** We introduce Multi-Shell Viewpoint Sampling (MSVS), which uses concentric spherical shells with visibility-bounded radii and stratified spherical sampling to ensure uniform, distance-aware coverage of viewpoints for evaluating 3D watermarking on 3D assets.
- **Greedy training-view selection.** We propose a simple greedy subsampling strategy that models the benefit of adding a training viewpoint with a nondecreasing kernel over evaluation cameras, efficiently selecting views that maximize the summed bit accuracy.
- **Comprehensive empirical analysis.** Across diverse assets and watermarking methods, MSVS reveals substantially lower bit accuracy than dataset-provided test viewpoints, surfacing failure modes that standard protocols miss; our greedy subsampling consistently improves MSVS test accuracy over a random baseline.

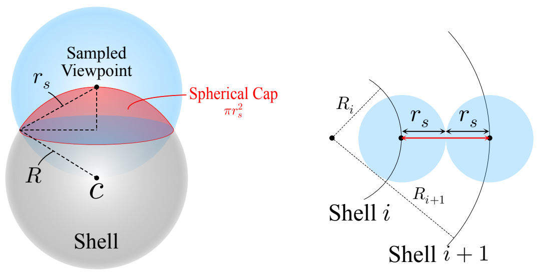
## 2. MULTI-SHELL VIEWPOINT SAMPLING

In this section we present an evaluation protocol for measuring bit accuracy in watermarked scenes that contain a single 3D object centered at the origin. Ideally, bit accuracy would be evaluated over the entire space of viewpoints. This is infeasible because the space of viewpoints is continuous. A sampling strategy is therefore required to provide sufficient spatial coverage with a practical number of evaluations. We introduce Multi-Shell Viewpoint Sampling (MSVS), which respects object visibility constraints and samples viewpoints on a set of concentric spheres centered at the object.

### 2.1. Single Shell Viewpoint Sampling

We begin with sampling on the surface of a single shell, that is, a sphere with fixed radius $R$. Intuitively, viewpoints that are very close to each other should yield similar bit accuracy. Accordingly, we assume bit accuracy is approximately constant within distance $r_s$ of a viewpoint. We use $r_s$ as the minimum pairwise separation during sampling and refer to $r_s$ as the *separation radius*. This hyperparameter controls sampling density: a smaller $r_s$ produces denser coverage and stricter evaluation, at the cost of increased computation.

Given the separation radius $r_s$, each sampled viewpoint induces a spherical cap consisting of points on the sphere that lie within Euclidean distance $r_s$, as illustrated in Figure 3 (a). Since the surface area of this spherical cap equals $\pi r_s^2$ and the sphere has area $4\pi R^2$, the minimum number of points $N$ required to cover the sphere with radius $R$ satisfies $N \geq \lceil \frac{4\pi R^2}{\pi r_s^2} \rceil = \lceil 4(\frac{R}{r_s})^2 \rceil$. Equality would require a partition with no overlap or gaps, which is not achievable in general. In practice, we therefore set $N = \lceil \alpha \cdot 4(R/r_s)^2 \rceil$, where $\alpha \geq 1$ is an efficiency factor that accounts for overlap and boundary effects. After fixing $N$, we distribute the viewpoints on the sphere using the spherical Fibonacci lattice [17].



(a) Spherical cap (b) Radial step

**Fig. 3**. Illustration of (a) the spherical cap created by a single viewpoint and (b) the radial step between two shells.

### 2.2. Extension to Multi-Shell

By repeating single shell viewpoint sampling over multiple radii, the 3D viewing domain can be covered more comprehensively. However, if the admissible radius range is left unbounded, the number of sampled viewpoints can become prohibitively large. We therefore constrain the radius range using the target object's visibility in rendered images.

First, we define the lower bound on the radius (the radius of the innermost shell) as $R_{\min} = \max_{p \in P} \|p - c\|$, where $P$ denotes the point cloud of the object and $c$ is its center. This is the radius of the smallest sphere centered at $c$ that contains the entire point cloud. Since NeRF and 3DGS can render estimated depth maps once trained, 3D watermarking methods built on them can construct $P$ by unprojecting depth pixels into 3D using known camera parameters.

Second, we define the upper bound on the radius by a dataset specific visibility criterion. We first examine all dataset cameras and identify the one whose rendering yields the largest projected object area, measured by the object mask. We then keep this camera's intrinsics and orientation fixed and translate the camera along its viewing ray away from the object center. For a placement at distance $d$, let $\mathrm{vis}(d)$ be the fraction of pixels that belong to the object mask. Given a threshold $\tau$, we set $R_{\max} = \max\{d \in \mathbb{R}_+ \mid \mathrm{vis}(d) \geq \tau\}$. In practice, a binary search over $d$ yields $R_{\max}$, since $\mathrm{vis}(d)$ is nonincreasing in $d$ for fixed intrinsics and orientation.

We then sample shell radii uniformly over $[R_{\min}, R_{\max}]$. Given a separation radius $r_s$, we set the radial step to $\Delta R = 2r_s$ as illustrated in Figure 3 (b), which provides radial coverage commensurate with the assumed spatial smoothness at scale $r_s$. Thus, $R_k = R_{\min} + k\Delta R$ for $k = 0, \ldots, K$, where $K = \lfloor (R_{\max} - R_{\min})/\Delta R \rfloor$, which yields $K + 1$ radii.

## 3. GREEDY SUBSAMPLING

Our goal is to increase the number of protected viewpoints, where the embedded watermark can be reliably recovered. To this end, we start from an objective that maximizes the average bit accuracy across the evaluation viewpoints (equivalently the sum, since the set is fixed) and then use it to guide a greedy subsampling strategy.

### 3.1. Problem Formulation

In standard machine learning, a strict separation between training and test data is used to estimate performance on unseen samples. In 3D watermarking, the goal is different; the system must guarantee high bit accuracy for every possible camera viewpoint in a continuous set $V$. Practical evaluation uses a finite discretization $V'$ of $V$, which is, in our case, the viewpoints sampled by MSVS. Training on $V'$ is justified because $V'$ is a concrete approximation of the

**Table 1**. Per-method results, averaged over scenes: bit accuracy on Test (dataset-provided) and MSVS (multi-shell).

| Dataset | Blender [1] | | | | | Stanford-ORB [18] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Bit Accuracy ↑ | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | Bit Accuracy ↑ | |
| Methods | | | | Test | MSVS | | | | Test | MSVS |
| 3D-GSW [6] | **34.77** | **0.981** | **0.023** | 0.951 | 0.764 | **35.75** | **0.976** | 0.027 | 0.996 | 0.828 |
| + Random | 32.59 | 0.971 | 0.032 | **0.975** | 0.832 | 35.51 | 0.974 | **0.026** | **1.000** | 0.851 |
| + Greedy (Ours) | 32.37 | 0.970 | 0.033 | **0.975** | **0.852** | 34.99 | 0.972 | 0.029 | **1.000** | **0.907** |
| GuardSplat [7] | **27.15** | **0.931** | **0.054** | 0.653 | 0.557 | **36.19** | **0.978** | **0.024** | 0.665 | 0.578 |
| + Random | 26.91 | 0.927 | 0.058 | **0.706** | 0.599 | 35.31 | 0.976 | 0.027 | **0.711** | 0.647 |
| + Greedy (Ours) | 26.92 | 0.927 | 0.058 | 0.705 | **0.601** | 35.34 | 0.976 | 0.027 | **0.711** | **0.650** |
| GaussianMarker [9] | **27.43** | **0.922** | **0.070** | 0.845 | 0.652 | **35.68** | **0.970** | **0.038** | 0.803 | 0.623 |
| + Random | 27.19 | 0.917 | 0.077 | **0.884** | 0.686 | 35.02 | 0.965 | 0.043 | **0.848** | 0.668 |
| + Greedy (Ours) | 27.20 | 0.918 | 0.077 | 0.882 | **0.691** | 35.11 | 0.966 | 0.042 | 0.844 | **0.669** |

full viewpoint domain, and evaluation verifies compliance with the requirement on that domain. This does not constitute data leakage, since the aim is not to infer behavior on an unknown distribution but to certify that the requirement holds across the intended operating domain. That said, training on $V'$ is typically impractical because $|V'|$ is large. Gradient based optimization makes training cost grow with $|V'|$, whereas evaluation on $V'$ only requires forward computation, which is often affordable. We therefore seek a subset of $m$ viewpoints from $V'$ with which to extend the training set, maximizing overall bit accuracy on $V'$. Formally, we pose

$$\underset{S' \subseteq V'}{\text{Maximize}} \sum_{x \in V'} b_{S_0 \cup S'}(x) \text{ s.t. } |S'| \le m, \tag{1}$$

where $S_0$ is the original training set, and $b_{S_0 \cup S'}(x)$ denotes the bit accuracy at viewpoint $x$ after training on $S := S_0 \cup S'$.

## 3.2. Approximation of Fine-Tuning

A naïve approach would be to repeatedly select the viewpoint in $V'$ with the lowest bit accuracy, add it to the training set, and fine-tune the model, iterating until the update budget $m$ is exhausted. However, this requires $m$ separate fine-tuning runs, each already highly expensive (e.g., 84 hours per CopyRNeRF [19] run [7]). Given a model pre-trained (i.e., watermarked) on the original set, fine-tuning on the set augmented by a viewpoint $s^*$ primarily increases $b_{S_0}(x)$ for viewpoints $x$ in a neighborhood of $s^*$. To avoid fine-tuning at each subsampling step, we approximate the effect of adding a viewpoint $s^*$ by an improvement factor $k$ that depends only on the distance to the nearest training viewpoint. Concretely, for any $x \in V'$, we model this effect as

$$b_{S_0 \cup \{s^*\}}(x) := b_{S_0}(x) k \left( \min_{s \in S_0 \cup \{s^*\}} d(x, s) \right) \tag{2}$$

where $d(x, s)$ is the distance between viewpoints $x$ and $s$, and $k \colon \mathbb{R}_+ \to [1, \infty)$ is nonincreasing in distance. This surrogate states that adding $s^*$ provides a locality based gain that depends only on the distance to the nearest training viewpoint. Viewpoints near $s^*$ receive a larger multiplicative improvement, distant views receive little change, and views already well covered by $S_0$ see negligible additional gain. This lets us estimate the benefit of adding $s^*$ without performing an actual fine-tuning run.

Following Equation 2, each subsampling step selects the next viewpoint by maximizing

$$s^* = \arg \max_{s \in V' \setminus S} \sum_{x \in V'} w_S(x) k \left( \min_{s' \in S \cup \{s\}} d(x, s') \right) \tag{3}$$

where $w_S(x)$ is a nonnegative weight. With $w_S(x) = b_S(x)$, the objective favors viewpoints that already exhibit high bit accuracy. To emphasize poorly protected viewpoints, we instead set $w_S(x) = \max\{0, 1 - b_S(x)\}$, which downweights already protected views and shifts the budget toward weak regions.

## 3.3. Greedy Algorithm for Submodular Optimization

Let $f(S) = \sum_{x \in V'} w(x) k \left( \min_{s \in S} d(x, s) \right)$ with fixed nonnegative weights $w(x) = w_{S_0}(x)$. Then $f(S)$ is nonnegative, monotone, and submodular. Maximizing a monotone submodular function under a cardinality constraint is NP-hard [20], so we employ a greedy algorithm, which in each step is equivalent to Equation 3 with $w_S(x)$ replaced by the fixed $w(x)$. By caching $\min_{s \in S} d(x, s)$ for all $x \in V'$ and updating it incrementally when testing a candidate $s$, the greedy algorithm runs in $\mathcal{O}(m|V'|^2)$ time. The objective is a sum over viewpoints, so its evaluation is readily vectorized and can benefit from GPU acceleration.

## 4. EXPERIMENTS

### 4.1. Experimental Setting

MSVS and the greedy subsampling strategy are method-agnostic. We apply them to 3D-GSW [6], GuardSplat [7], and GaussianMarker [9], selected for their relatively short training times [7]. The experiments have two stages: (1) embed a 32-bit message into clean models pre-trained with 3DGS [2] and evaluate bit accuracy on MSVS; (2) extend the training set with $m = 10$ MSVS viewpoints and fine-tune once (baseline: random $m = 10$). For 3D-GSW and GaussianMarker we use the official default training configurations; for GuardSplat [2] we enable a distortion layer with Gaussian blur, brightness jitter, JPEG compression, cropping, rotation, resizing, and additive zero-mean Gaussian noise. During fine-tuning, ground truth for the new viewpoints is rendered by the stage-1 watermarked models. Unless otherwise noted, we set $r_s = 0.5$, $\alpha = 1.1$, and $\tau = 0.1$. We use $k(d) = 1 + e^{-(d/d')^2}$, where $d'$, set to 1.5, controls the decay rate. This kernel is designed so that when a viewpoint is added to the training set ($d = 0$), a random baseline of 0.5 becomes 1.0, consistent with the assumption that training drives bit accuracy at those viewpoints to near 1.0.

**Datasets**: Blender [1], commonly used for object-centric 3D watermarking [8, 19, 6, 7, 9]; and real-world Stanford-ORB [18],

[2]Unlike other methods, GuardSplat has explicit distortion layers.

**Table 2**. Per-method bit accuracy on MSVS viewpoints under common image attacks, averaged over all scenes in each dataset.

| Dataset | Blender [1] | | | | | Stanford-ORB [18] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | None | Noise ($\sigma = 0.1$) | JPEG (50%) | Scaling (75%) | Blur ($\sigma = 0.1$) | None | Noise ($\sigma = 0.1$) | JPEG (50%) | Scaling (75%) | Blur ($\sigma = 0.1$) |
| 3D-GSW [6] | 0.764 | 0.631 | 0.726 | 0.727 | 0.764 | 0.828 | 0.634 | 0.810 | 0.842 | 0.828 |
| + Random | 0.832 | 0.678 | 0.789 | 0.794 | 0.832 | 0.851 | 0.665 | 0.831 | 0.861 | 0.851 |
| + Greedy (Ours) | **0.852** | **0.685** | **0.806** | **0.816** | **0.852** | **0.907** | **0.688** | **0.888** | **0.914** | **0.907** |
| GuardSplat [7] | 0.557 | 0.574 | 0.562 | 0.558 | 0.557 | 0.578 | 0.585 | 0.583 | 0.578 | 0.578 |
| + Random | 0.599 | **0.598** | 0.599 | 0.601 | 0.599 | 0.647 | **0.616** | 0.636 | 0.648 | 0.647 |
| + Greedy (Ours) | **0.601** | 0.597 | **0.600** | **0.602** | **0.601** | **0.650** | **0.616** | **0.639** | **0.651** | **0.650** |
| GaussianMarker [9] | 0.652 | 0.509 | 0.555 | 0.616 | 0.652 | 0.623 | 0.499 | 0.541 | 0.625 | 0.623 |
| + Random | 0.686 | **0.511** | 0.573 | 0.644 | 0.686 | 0.668 | **0.501** | **0.568** | 0.663 | 0.668 |
| + Greedy (Ours) | **0.691** | **0.511** | **0.575** | **0.649** | **0.691** | **0.669** | **0.501** | 0.567 | **0.666** | **0.669** |

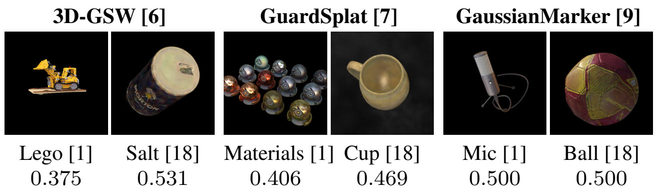


**Fig. 4**. Example of low bit accuracy despite moderate-to-high visual quality, sampled by MSVS; numbers indicate bit accuracy.

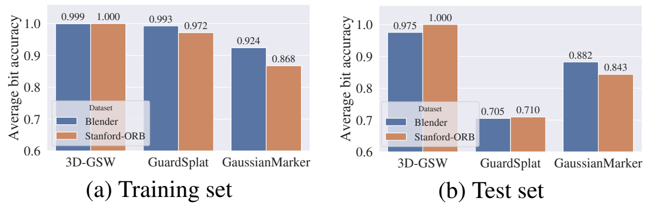


**Fig. 5**. Bit accuracy after training with the greedy algorithm.

where we use *baking, ball, chips, cup, curry, gnome, grogu, pepsi, pitcher*, and *salt* because other scenes lack an original image with $\text{vis}(\cdot) \geq \tau$. For Stanford-ORB, we apply the object mask to remove background and downsample by 2. For robustness, following [6, 7, 9], we add zero-mean Gaussian noise with standard deviation 0.1, JPEG at 50% quality, scaling to 75% of the original size, and Gaussian blur with a $3 \times 3$ kernel and 0.1-pixel standard deviation, then evaluate on the MSVS viewpoints.

## 4.2. Experimental Results

Table 1 reports per method results, including rendering quality evaluation using Structural Similarity Index (SSIM) [21], Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [22]. Embedding a watermark in additional views slightly degrades rendering quality because it introduces noise that is ideally invisible to human observers but measurable in quantitative evaluations.

**MSVS broadens evaluation coverage**. For all methods, bit accuracy on MSVS generated test sets is consistently lower than on the test sets provided by the dataset. This discrepancy is also observed in bit accuracy for attacked images as shown in Table 2. This indicates that MSVS samples viewpoints with low bit accuracy that are not covered by the original splits and therefore evaluates watermark coverage more strictly. Figure 4 presents examples of renderings with low bit accuracy. The rendering quality of low-texture objects (salt and cup) is only moderate, which suggests optimization difficulty and may partly explain the low bit accuracy. However, even well-textured objects with high rendering quality can exhibit low bit accuracy. This suggests that existing methods, when trained on splits curated for novel view synthesis, generalize well in terms of image quality for unseen views but do not achieve equally broad watermark coverage.

**Greedy subsampling improves MSVS accuracy**. The greedy algorithm substantially improves bit accuracy on the MSVS viewpoints, especially for 3D-GSW, whereas gains are marginal for GuardSplat on Blender and for GaussianMarker on Stanford-ORB. Two factors likely explain this result. First, as Figure 5 shows, GuardSplat attains very high training set bit accuracy but much lower test set accuracy. This gap indicates limited generalization in spatial watermark coverage: adding a new training viewpoint mainly boosts accuracy at that viewpoint, with little benefit transferring to nearby views. Consequently, most MSVS viewpoints yield similar marginal gains, so the choice of subsampling strategy matters little. Second, GaussianMarker exhibits substantially lower training-set bit accuracy than the other methods, indicating that it sometimes fails to embed a watermark even at training viewpoints. In such cases, adding a viewpoint does not reliably improve its bit accuracy, which violates the monotonic-improvement assumption of our kernel model that multiplies $b_S(x)$ by $k(d) \geq 1$. This mismatch reduces the advantage of the greedy algorithm. By contrast, 3D-GSW trains reliably and generalizes better across neighboring views, so greedy subsampling yields large improvements on MSVS.

## 5. CONCLUSION

We introduced Multi-Shell Viewpoint Sampling (MSVS), a visibility-bounded, object-centric protocol that evaluates 3D watermark bit accuracy over a broad, well-covered set of viewpoints. MSVS consistently reveals failures that dataset-provided test views miss, and our greedy subsampling strategy further improves protection, especially for 3D-GSW, with gains that persist under common image attacks. MSVS provides a method-agnostic, reproducible protocol that we recommend as a standard for future 3D watermarking benchmarks. As future work, we will explore data-driven kernels $k(d)$ that replace the current handcrafted heuristic by learning or calibrating parameters from pre-trained, watermarked models, with the goal of reducing manual tuning and further improving accuracy and efficiency.

## 6. REFERENCES

[1] 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*, vol. 65, no. 1, pp. 99–106, 2021.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis, "3d gaussian splatting for real-time radiance field rendering.," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.

[4] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[5] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger, "Mip-splatting: Alias-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19447–19456.

[6] 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*, 2025, pp. 5938–5948.

[7] Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie, "Guardsplat: Efficient and robust watermarking for 3d gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16325–16335.

[8] Youngdong Jang, Dong In Lee, MinHyuk Jang, Jong Wook Kim, Feng Yang, and Sangpil Kim, "Waterf: Robust watermarks in radiance fields for protection of copyrights," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12087–12097.

[9] Xiufeng Huang, Ruiqi Li, Yiu-ming Cheung, Ka Chun Cheung, Simon See, and Renjie Wan, "Gaussianmarker: Uncertainty-aware copyright protection of 3d gaussian splatting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 33037–33060, 2024.

[10] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan, "Protecting nerfs' copyright via plug-and-play watermarking base model," in *European Conference on Computer Vision*. Springer, 2024, pp. 57–73.

[11] Chenxin Li, Brandon Y Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang, "Steganerf: Embedding invisible information within neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 441–453.

[12] Chenxin Li, Hengyu Liu, Zhiwen Fan, Wuyang Li, Yifan Liu, Panwang Pan, and Yixuan Yuan, "Instantsplamp: Fast and generalizable stenography framework for generative gaussian splatting," in *The Thirteenth International Conference on Learning Representations*, 2025.

[13] Yifeng Yang, Hengyu Liu, Chenxin Li, Yining Sun, Wuyang Li, Yifan Liu, Yiyang Lin, Yixuan Yuan, and Nanyang Ye, "Concealgs: Concealing invisible copyright information in 3d gaussian splatting," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[14] 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)*, 2018, pp. 657–672.

[15] 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, pp. 41–49.

[16] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.

[17] Ricardo Marques, Christian Bouville, Mickaël Ribardiere, Luis Paulo Santos, and Kadi Bouatouch, "Spherical fibonacci point sets for illumination integrals," in *Computer Graphics Forum*. Wiley Online Library, 2013, vol. 32, pp. 134–143.

[18] Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Elliott Wu, Jiajun Wu, et al., "Stanford-orb: a real-world 3d object inverse rendering benchmark," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46938–46957, 2023.

[19] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan, "Copyrnerf: Protecting the copyright of neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 22401–22411.

[20] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[21] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[22] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

## 7. FUNDING ACKNOWLEDGEMENTS

This work was supported by JST SPRING Grant Number JPMJSP2123 and JST ASPIRE Grant Number JPMJAP2326, Japan.

## 8. COMPLIANCE WITH ETHICAL STANDARDS

This is a numerical simulation study for which no ethical approval was required.