MESHGUARD: HIDDEN 3D WATERMARKING IN SPECTRAL SPACE THROUGH LAPLACIAN BELTRAMI OPERATOR AND GEOMETRY FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in 3D generation and reconstruction enable rapid creation of high-fidelity assets such as point clouds and meshes, increasing the need for reliable intellectual property protection. However, unlike well-studied 2D watermarking, 3D methods remain limited and often embed signals directly into surface geometry, making them vulnerable to distortion and adversarial attacks. We propose a robust mesh watermarking framework based on spectral geometry. Our method leverages the Laplace–Beltrami operator to extract intrinsic spectral representations and embeds watermark signals into mid-band components via eigendecomposition. The watermark is applied in the spatial domain through predicted vertex-level offsets regressed from updated eigenvectors, achieving imperceptible yet stable embedding. To enhance robustness, we incorporate adversarial training with diverse mesh perturbations, enabling reliable decoding under severe distortions. We further introduce a flexible encoding scheme that converts text or image inputs into binary watermark sequences for fusion and evaluation. Extensive experiments demonstrate improved robustness against a wide range of geometric attacks while preserving high geometric fidelity, outperforming existing approaches by a large margin.

1 INTRODUCTION

Watermarking is a fundamental technique for protecting digital assets, especially in multimedia. Most research targets 2D images, embedding signals explicitly Min et al. (2024) or implicitly Liu et al. (2024), often via generative models such as watermark-conditioned diffusion Tan et al. (2024), with parallel efforts on diffusion-based watermark removal Liu et al. (2024). As large-scale generative models advance, watermarking has also become central to protecting the intellectual property (IP) of trained models Nagai et al. (2018). GuardSplat Chen et al. (2025b) argues that an effective scheme should satisfy four properties—security, invisibility, embedding, and robustness—yet existing methods typically address only a subset, focusing on either embedding Min et al. (2024) or robustness under adversarial attacks Szyller et al. (2021).

IP protection for 3D assets remains underexplored, particularly under learning-based settings. Recent work targets neural 3D representations—Neural Radiance Fields via DreaMark Zhu et al. (2025) and spherical-harmonic coefficients of 3D Gaussians via GuardSplat Chen et al. (2025b)—but is tightly coupled to specific renderers and requires per-scene optimization, limiting transfer to common geometric formats such as meshes and point clouds. Point-cloud watermarking has received some attention Zaman et al. (2025); Yang et al. (2021), whereas mesh watermarking is still scarce: Zhu et al. (2024b) proposes an encoder–decoder framework with adversarial robustness, but its adversarial evaluation is limited. Existing mesh-based approaches embed directly into geometry or texture, scaling poorly across diverse structures and remaining vulnerable to attacks due to locally concentrated, easily detectable patterns.

We therefore propose a complete, efficient, and robust mesh watermarking framework that jointly handles embedding, adversarial robustness, and reliable decoding. Watermark inputs—text or logo images—are first converted into a binary message sequence by a BLIP-2-based encoder Li et al. (2023), supporting cross-modal inputs and efficient verification. The message is then embedded

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

in the spectral domain: we compute intrinsic spectral features via the Laplace–Beltrami operator Bobenko & Springborn (2007); Belkin et al. (2008); Andreux et al. (2014), eigen-decompose them into ordered components, and fuse the watermark with selected eigenvectors. The resulting spectral perturbation drives vertex-level offset prediction, distributing watermark information adaptively across the mesh while keeping perturbations on the order of 10^{-4}–10^{-3} to preserve geometric fidelity and visual imperceptibility.

To harden the protocol, a comprehensive adversarial attack pipeline simulates diverse geometric corruptions—vertex noise, smoothing, partial face removal, translation, rotation, and their combinations—and a dedicated MeshCNN-based decoder Hanocka et al. (2019) recovers the binary message from corrupted meshes. This encoder–attack–decoder design enables efficient end-to-end training and systematic evaluation. Extensive experiments show that our method preserves high mesh fidelity while achieving reliable, high-accuracy decoding under challenging attacks, providing a practical and scalable solution for 3D IP protection.

Contributions. 1) We introduce a flexible watermark encoding module based on BLIP-2 that converts multimodal inputs, including text and logo images, into a unified binary message representation for mesh watermarking and evaluation. **2)** We propose a novel spectral-domain watermark embedding framework based on the Laplacian-Beltrami operator and spectral fusion in eigen space, enabling robust and imperceptible watermark injection (vertex offsets at magnitude 10^{-4} to 10^{-3}) across the global mesh geometry adaptively. **3)** We design a comprehensive adversarial attack and robust decoding framework that evaluates watermark decoding accuracy under diverse mesh corruptions and reliably recovers the embedded watermark message from attacked meshes.

2 Related Work

Image watermarking and removal. Classical image watermarking spans signal-processing schemes Zhang et al. (2020); Zheng et al. (2009) and medical imaging Singh et al. (2017). Recent diffusion-based methods embed watermarks for IP protection, either directly (Waterdiff Tan et al. (2024)) or via conditioned generation Min et al. (2024). Conversely, diffusion has been exploited to attack Zhang et al. (2024b) or erase watermarks—explicitly with DERO Fang et al. (2024) or implicitly Liu et al. (2024)—to benchmark robustness. Plugmark Chen et al. (2025a) couples a boundary classifier with a diffusion knowledge extractor to produce zero-distortion watermarked images.

Watermark for models. Model watermarking Nagai et al. (2018) protects deep-model IP, with variants targeting compression resistance Nie et al. (2024) and extraction attacks Tan et al. (2023); DAWN Szyller et al. (2021) introduces adversarial training against surrogate-model theft. Extensions cover language generation Zhao et al. (2023), scalable LLM watermarking Dathathri et al. (2024), and improved robustness Zhang et al. (2024a) and reliability Zhu et al. (2024a). Compmarkgs In et al. (2025) extends Gaussian watermarking to compressed 3D Gaussians.

Watermark for multi-view images and 3D meshes. 3D watermarking has co-evolved with neural 3D representations. For radiance fields, MantleMark Luo et al. (2025a) embeds via multi-view rendering, DreaMark Zhu et al. (2025) uses score distillation, Deep 3D-to-2D Yoo et al. (2022) encodes in geometry and decodes from 2D views, and Luo et al. (2025b) renders signatures from a watermark codebook. For Gaussian Splatting, MarkSplatter Huang et al. (2025), GuardSplat Chen et al. (2025b), and 3D-GSW Jang et al. (2024) target robustness across rendered views. Traditional mesh schemes are lightweight but limited in generalization Praun et al. (1999), with spherical parameterization as an alternative Li et al. (2004). More recently, Zhu et al. (2024b) first formalized end-to-end mesh watermarking with adversarial decoding, Yu et al. (2003) addressed triangle-soup inputs, and adversarial perturbations Chen et al. (2025b) have been studied to harden downstream recognition. A broader survey of 3D watermarking is in Wu et al. (2024).

3 Method

We propose a two-stage spectral watermarking method that embeds and robustly decodes binary messages on 3D meshes under adversarial attack, as illustrated in Fig. 1. Given an input mesh $\mathbf{G}(\mathbf{v}, \mathbf{e})$ with vertices $\mathbf{v} \in \mathbb{R}^{n\times 3}$ and edges $\mathbf{e} \in \mathbb{R}^{m\times 2}$, we uniformly sample n^* vertices and build the Laplace–Beltrami operator Δ, whose eigen-decomposition yields eigenvalues $\{\lambda_i\}$ and eigenvectors $\{\mathbf{U}_i\}$. Mid-band components $\mathbf{U}_{\text{mid}}$ are then selected to balance robustness against ad-

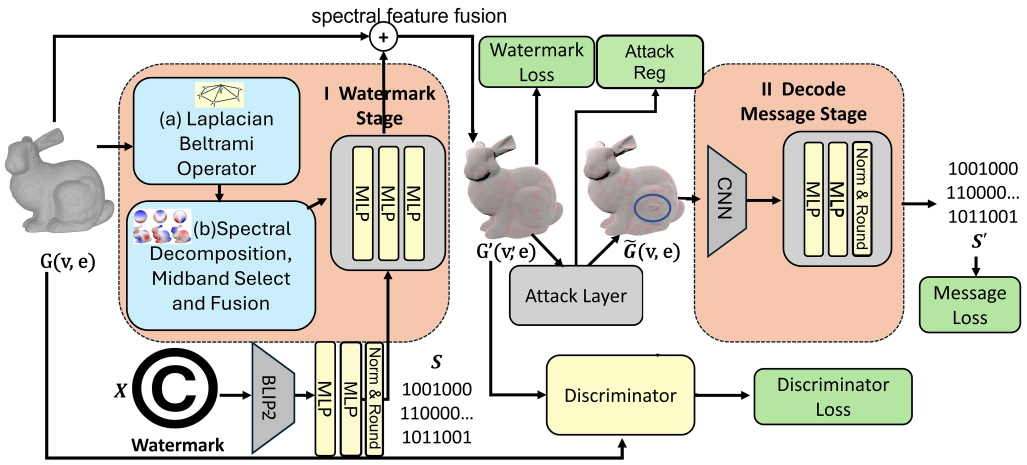


Figure 1: **System overview.** The system comprises two stages: watermark (Stage **I**) and decode message (Stage **II**) against adversarial attack, both enclosed in the light-orange dashed box. Given an input mesh $\mathbf{G}$ featuring vertices $\mathbf{v}$ and edges $\mathbf{e}$, the spectral features are extracted via the Laplace-Beltrami operator Δ. After eigen-decomposition, mid-band eigenvectors $\{\{\mathbf{u}_i\}_{i=i_{\min}+\lfloor 1/4\times n\rfloor}^{i_{\min}+\lfloor 3/4\times n\rfloor}$ are selected according to their ordered eigenvalues $\{\lambda_i\}$ and used to regress offsets $\delta\mathbf{v}$ for each sampled vertex. Concurrently, a binary watermark sequence $\mathbf{S} \in \{0,1\}^l$ is generated from an input $\mathbf{X}$ using a BLIP-2 encoder followed by linear MLP layers with a $\tanh$ activation and normalization, producing the embedded bit string. The selected eigenvectors are modulated by $\mathbf{S}$ to inject the watermark in the spectral domain, generating the watermarked mesh $\mathbf{G}'(\mathbf{v}, \mathbf{e})$. This mesh is then passed through an attack layer that produces the attacked mesh $\tilde{G}(\mathbf{v}, \mathbf{e})$, simulating geometric perturbations. In Stage **II**, a 3D CNN followed by MLP layers decodes $\tilde{\mathbf{G}}(\mathbf{v}, \mathbf{e})$ to recover the predicted binary sequence $\mathbf{S}'$. Training is jointly supervised by three losses plus regularizer: (i) watermark reconstruction loss on $\mathbf{G}'(\mathbf{v}, \mathbf{e})$, (ii) discriminator loss distinguishing $\mathbf{G}'$ from $\mathbf{G}$, (iii) message decoding loss between $\mathbf{S}'$ and $\mathbf{S}$ and (iv) a regularizer for the attack module.

versarial attacks with spectral expressiveness. In Stage **I** (watermark embedding), a binary sequence $\mathbf{S} \in \{0,1\}^l$ with $l = \frac{k}{2}$ is generated from an input $\mathbf{X}$ through a BLIP-2 encoder followed by linear layers. $\mathbf{S}$ modulates $\mathbf{U}_{\text{mid}}$, and the fused representation regresses per-vertex offsets $\delta\mathbf{v} \in \mathbb{R}^{n^*\times 3}$ that are applied to the sampled vertices to produce the watermarked mesh $\mathbf{G}'(\mathbf{v}', \mathbf{e})$. A discriminator ϕ_{disc} is jointly trained to distinguish $\mathbf{G}'$ and $\tilde{\mathbf{G}}$ from $\mathbf{G}$, enforcing imperceptibility, while an attack simulation module $\mathcal{A}(\cdot)$ corrupts $\mathbf{G}'$ into $\tilde{\mathbf{G}}(\mathbf{v}', \mathbf{e})$ via global transformations and local geometric distortions to improve robustness. In Stage **II** (message decoding), a MeshCNN-based encoder ϕ_{CNN} extracts features from $\tilde{\mathbf{G}}$, and a decoder ϕ_{dec} recovers the binary watermark $\mathbf{S}'$. The framework is trained end-to-end with reconstruction, discriminator, decoding, and regularization losses, jointly optimizing geometric fidelity, imperceptibility, attack robustness, and decoding accuracy.

3.1 Spectral Feature Extraction

Given an input 3D mesh $\mathbf{G}$ with vertices $\mathbf{v} \in \mathbb{R}^{n\times 3}$ and edges $\mathbf{e} \in \mathbb{R}^{m\times 2}$, we construct a resampled graph $\mathbf{G}^*$ before applying the Laplace–Beltrami operator, which mitigates the non-uniform triangle density typical of arbitrary inputs. Specifically, we **(i)** subsample existing mesh vertices via farthest-point sampling together with their incident edges to preserve the original connectivity, and **(ii)** draw additional points uniformly from the mesh surface, reconstructing local connectivity through k-nearest-neighbor edges followed by tangent-plane Delaunay triangulation. The union of the two vertex sets and their combined edge set forms $\mathbf{G}^*(\mathbf{v}^*, \mathbf{e}^*)$ with $\mathbf{v}^* \in \mathbb{R}^{n^*\times 3}$, on which spectral features are extracted via the Laplace–Beltrami operator Δ discretized with cotangent weights:

$$\Delta_{ij}\mathbf{G} = \begin{cases} \frac{1}{2A_i}\sum_{j\in\mathcal{N}(i)}(\cot\alpha_{ij} + \cot\beta_{ij}) & \text{if } i = j, \\ -\frac{1}{2A_i}(\cot\alpha_{ij} + \cot\beta_{ij}) & \text{if } j \in \mathcal{N}(i), \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where α_{ij} and β_{ij} are the angles opposite edge (i, j), $\mathcal{N}(i)$ is the one-ring neighborhood of vertex i, and A_i is the barycentric cell area from adjacent edge cross products Floater et al. (2006). This

operator captures intrinsic surface geometry and is isometry-invariant, making it well-suited for robust watermark embedding.

3.2 Binary Watermark Sequence Generation

The binary watermark $\mathbf{S} \in \{0,1\}^l$ with length $l = \frac{k}{2}$ is derived from input $\mathbf{X}$ (text prompt or logo) via a BLIP-2 encoder ϕ_{enc} matching half the eigenvector count k:

$$\mathbf{z} = \phi_{\text{enc}}(\mathbf{X}), \quad \mathbf{z} \in \mathbb{R}^d. \tag{2}$$

The embedding is projected through MLP layers ϕ_{msg} and a linear layer g_θ followed by a sigmoid activation, mapping to per-bit probabilities in $(0,1)$, then discretized via thresholding at 0.5 to recover the binary message:

$$\hat{\mathbf{S}} = \sigma\big(g_\theta(\phi_{\text{msg}}(\mathbf{z}))\big) \in (0,1)^l, \tag{3}$$

$\mathbf{S}$ is then injected into $\mathbf{G}$ through spectral eigenvector modulation (Sec. 3.3), yielding the watermarked mesh $\mathbf{G}'$.

3.3 Spectral Decomposition and Eigen-Fusion

We eigen-decompose the Laplacian as

$$\Delta_{ij}\mathbf{G} = \mathbf{U}^T\mathbf{\Lambda}\mathbf{U}, \tag{4}$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_k)$ holds the ordered eigenvalues $\lambda_1 \leq \cdots \leq \lambda_k$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$ the corresponding eigenvectors with feature dimension $d_{\mathbf{u}}$. We retain the mid-band eigenvectors,

$$\mathbf{U}_{\text{mid}} = \{\mathbf{u}_i\}_{i=i_{\min}+\lfloor k/4 \rfloor}^{i_{\min}+\lfloor 3k/4 \rfloor}, \tag{5}$$

to avoid low-frequency flat regions and high-frequency sharp curvatures, balancing expressiveness and robustness; $\lfloor \cdot \rfloor$ denotes integer rounding for the indexing.

To embed the watermark, $\mathbf{S} \in \{0,1\}^l$ is partitioned into groups of $\lfloor \frac{2l}{k} \rfloor$ bits and averaged into a shorter sequence $\mathbf{S}' \in \{0,1\}^{k/2}$ matching $\mathbf{U}_{\text{mid}} \in \mathbb{R}^{\frac{k}{2} \times d_{\mathbf{u}}}$. Modulation is performed by eigenvalue-weighted element-wise scaling:

$$\tilde{\mathbf{U}}_{\text{mid}} = \mathbf{U}_{\text{mid}} \odot (\boldsymbol{\omega} \odot \mathbf{S}'^{\top}), \tag{6}$$

with $\boldsymbol{\omega} = \left\{ \frac{1}{\lambda_i + \epsilon} \right\}_{i=i_{\min}+\lfloor k/4 \rfloor}^{i_{\min}+\lfloor 3k/4 \rfloor}$ and $\epsilon = 10^{-5}$ for numerical stability. Replacing $\mathbf{U}_{\text{mid}} \leftarrow \tilde{\mathbf{U}}_{\text{mid}}$ yields the fused basis $\tilde{\mathbf{U}}$, which regresses offsets $\delta\mathbf{v} \in \mathbb{R}^{k \times 3}$ added to the sampled vertex coordinates:

$$\mathbf{G}'(\mathbf{v}', \mathbf{e}) = \mathbf{G}(\mathbf{v} + \delta\mathbf{v}, \mathbf{e}). \tag{7}$$

3.4 Adversarial Attack Simulation

The watermarked mesh $\mathbf{G}'$ is passed through an attack layer $\mathcal{A}(\cdot)$ with randomly initialized but range-constrained parameters, simulating five learnable attack types. Global transformations comprise anisotropic scaling $\mathbf{s} \in \mathbb{R}^{1\times 3}$, translation $\mathbf{t} \in \mathbb{R}^{1\times 3}$, and rotation $\mathbf{R} \in \mathbb{R}^{3\times 3}$; local perturbations comprise additive Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^{n'\times 3}$ on sampled vertices and random masking out C vertices, leaving $n' = n - C$. The attacked mesh is

$$\tilde{\mathbf{G}}(\mathbf{v}', \mathbf{e}) = \mathcal{A}(\mathbf{G}'(\mathbf{v}', \mathbf{e})). \tag{8}$$

Prior to the attack, a linear discriminator ϕ_{disc} distinguishes $\mathbf{G}'$ and $\tilde{\mathbf{G}}$ from $\mathbf{G}$, encouraging imperceptibility. For other non-learnable attacks, such as remeshing, the watermark embedding and message decoding modules are trained separately.

3.5 WATERMARK MESSAGE DECODING

A MeshCNN-based 3D CNN Hanocka et al. (2019) ϕ_{CNN} extracts features from $\tilde{\mathbf{G}}$:

$$\mathbf{f} = \phi_{\text{CNN}}(\tilde{\mathbf{G}}(\mathbf{v}', \mathbf{e})) \in \mathbb{R}^{n' \times d}. \tag{9}$$

A decoder ϕ_{dec} then predicts the binary message:

$$\mathbf{S}' = \text{norm}\left(\frac{1 + \tanh(\phi_{\text{dec}}(\mathbf{f}))}{2}\right) \in \{0, 1\}^l. \tag{10}$$

The decoder takes a watermarked (potentially attacked) mesh: each point with neighbours are mapped into a 6-dim feature, processed by a PointNet-style backbone (Conv1D: $64 \to 128 \to 256 \to 512$, BN+ReLU, global max-pool) and an MLP ($512 \to 256 \to L$, Dropout 0.3/0.2, Tanh) to recover the L-bit message.

3.6 LOSS FUNCTIONS

The total objective combines a reconstruction loss, a discriminator loss, an adversary loss, a message-decoding loss, and an attack regularizer with balance weights, λ_{disc}, λ_{adv}, λ_{reg}:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{disc}}\mathcal{L}_{\text{disc}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{msg}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{11}$$

Reconstruction loss enforces geometric fidelity via the ℓ_2 distance between original and watermarked vertices:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{v}'_i - \mathbf{v}_i\|_2^2. \tag{12}$$

Discriminator loss. A binary cross-entropy term trains the discriminator ϕ_{disc} to distinguish the watermarked mesh $\mathbf{G}'$ (label 1) from the clean mesh $\mathbf{G}$ (label 0):

$$\mathcal{L}_{\text{disc}} = \mathbb{E}[-\log \sigma(\phi_{\text{disc}}(\mathbf{G}'))] + \mathbb{E}[-\log(1 - \sigma(\phi_{\text{disc}}(\mathbf{G})))]. \tag{13}$$

Adversarial loss. The watermarking model is trained against a frozen copy of ϕ_{disc} using the non-saturating objective,

$$\mathcal{L}_{\text{adv}} = \mathbb{E}\big[-\log \sigma(\phi_{\text{disc}}(\mathbf{G}'))\big], \tag{14}$$

encouraging the discriminator's prediction on $\mathbf{G}'$ to approach 1, *i.e.,*, making $\mathbf{G}'$ progressively indistinguishable from $\mathbf{G}$.

Message decoding loss encourages correct recovery of $\mathbf{S}$ from the watermarked mesh under attacks:

$$\mathcal{L}_{\text{msg}} = \mathbb{E}\Big[-\tfrac{1}{K}\sum_{k=1}^{K}\big(S_k \log \hat{S}_{\tilde{\mathbf{G}},k} + (1 - S_k)\log(1 - \hat{S}_{\tilde{\mathbf{G}},k})\big)\Big], \tag{15}$$

where $\mathbf{S} = \{S_k\}_{k=1}^K \in \{0, 1\}^K$ is the ground-truth K-bit message, and $\hat{\mathbf{S}}_{\tilde{\mathbf{G}}} = \{\hat{S}_{\tilde{\mathbf{G}},k}\}_{k=1}^K \in (0, 1)^K$ is the per-bit decoding probability recovered from the attacked mesh $\tilde{\mathbf{G}}$.

Regularization loss penalizes the ℓ_2 norm of attack-layer parameters $\boldsymbol{\theta}_{\mathcal{A}} = \{\mathbf{p}_k\}$ to prevent excessive perturbations:

$$\mathcal{L}_{\text{reg}} = \|\boldsymbol{\theta}_{\mathcal{A}}\|_2^2 = \sum_k \|\mathbf{p}_k\|_2^2. \tag{16}$$

4 EXPERIMENTS

Implementation details. The Laplace–Beltrami operator is built on $n^* = 4096$ sampled points with $k = 256$ eigenvectors, and the binary watermark length is $l = 128$. Attack strength is matched to the watermark perturbation scale: for vertex offsets of order 10^{-3}, attacks are drawn at comparable magnitudes ($\{0.1, 0.5\}$ global; $\{10^{-2}, 5 \times 10^{-2}, 10^{-3}\}$ local) with randomized type, location, region, and intensity. Each iteration samples one of 200 logo images and embeds it across all meshes in the batch, keeping the binary encoding fixed so training focuses on robust fusion and decoding. The BLIP-2 encoder is frozen; only the subsequent linear layers are trained. We train for three days on a dual NVIDIA RTX A6000 workstation (96 GB total VRAM) with batch size 64 and learning rate 10^{-4}; the loss weights λ_{disc}, λ_{adv}, λ_{reg} in Eq. 11 are 0.8, 0.6, and 0.3.

Datasets and baselines. We use ShapeNet Chang et al. (2015) and Objaverse++ Lin et al. (2025). From ShapeNet, we take the geometrically complex *car*, *train*, *motorcycle*, and *airplane* categories rather than the simpler *chair/table* classes of prior work Zhu et al. (2024b); from Objaverse++, we curate ∼10,000 objects, mostly animated characters whose richer geometry and wider spectral feature distribution better reflect IP-protection scenarios. Each ShapeNet category and the Objaverse++ subset are split 9,000/1,000 for training/testing. Watermarks are 200 Logo-2K+ logos spanning 10 business categories, binarized (for text or symbolic content), and then fed to BLIP-2.

We compare against the learning-based Wang et al. (2022) and Deep3DMark Zhu et al. (2024b), and four recent non-learning algorithms Peng et al. (2021; 2022); Tsai & Liu (2022); Hou et al. (2023). For fairness, methods without an integrated decoder are decoded with Deep3DMark's unified bin-voting non-learning decoder to remove learning-based decoder bias, while end-to-end methods (e.g., Wang et al. (2022)) keep their original decoders; long semantic messages are truncated to fit Deep3DMark's limited maximum message length. We further implement two baselines. *Explicit Blending* embeds watermark patterns into manually selected flat regions by directly modifying vertices and faces with configurable scale and edge smoothing, either inlaying a 2D watermark at fixed thickness or converting it into a 3D mesh and blending it with the target geometry Liu et al. (2005). *Implicit Non-learning* retains our BLIP-2 encoder Li et al. (2023), Laplace–Beltrami operator, eigendecomposition, and decoder, but replaces the learned embedding module with hand-crafted geometric operations (e.g., multiplicative feature fusion) to inject the watermark into the $\frac{1}{2}$-band eigenvectors. Detailed implementation parameters are provided in the supplementary material.

Metrics. Watermark quality is measured by Chamfer Distance (CD) and Hausdorff Distance (HD) against the ground-truth mesh, and decoding robustness by Bit Error Rate (BER) and Bit Accuracy (BitAcc). All watermarked meshes are evaluated through the unified decoder of Fig. 1. Runtime is averaged over all test meshes and covers both watermark embedding and message decoding.

4.1 Baseline Comparisons

Tab. 1 reports reconstruction fidelity (left) and decoding accuracy (right) under five attack types, with all meshes normalized to a unit bounding box $[-1, 1]^3$. Reconstruction uses Chamfer (CD) and Hausdorff (HD) distances between the attacked watermarked mesh and the ground truth. Attacks applied to the reconstructed mesh are: rotation (random angle in $[0°, 30°]$), translation (random shift in $[0, 1]$), Gaussian noise (magnitude 10^{-3} on ∼15% of sampled vertices), smoothing (k-NN local averaging with $k = 30$), and crop-holes with radii $\{10^{-3}, 10^{-2}, 10^{-1}\}$; the *combination* setting randomly applies any three. These perturbations simulate realistic global and local distortions while preserving overall structure, and the CD/HD values track attack severity monotonically. Full model training shows that watermark offset grows with attack type and intensity. Qualitative examples of attacked meshes appear in Fig. 2. Tab. 1 groups our three implementations—explicit blending of target mesh and logo mesh, implicit non-learning, and ours—alongside the baselines. Our method achieves the best performance across various attack types and severities, except under translation, where the implicit non-learning method reaches the ceiling due to its inherent translation invariance. Fig. 3 demonstrates that our method (green) consistently achieves the highest bit accuracy across all attack types and severity levels, exhibiting only a marginal performance drop under translation. Among the other baseline methods, Deep3DMark Zhu et al. (2024b) shows the strongest robustness under the most severe attack settings. Fig. 4 provides qualitative comparisons among the top three methods: ours yields minimal vertex offsets (sparse red heatmap) on ShapeNet *train*, *car*, and *airplane*, whereas the explicit method introduces visible geometric distortions and the implicit non-

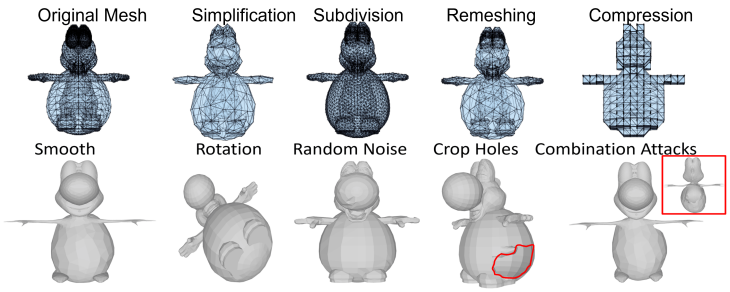


Figure 2: Visualization of nine mesh attacks: four topological attacks, smoothing, rotation, random perturbation, cropping, and combined attacks.

learning approach shows broader offsets reaching 5×10^{-2}. Beyond the standard attacks, we further evaluate topology attacks in Tab. 2. While simplification, compression, and related perturbations noticeably degrade the top two baselines, our method remains stable and consistently outperforms them.

For inference time, ranked from slowest to fastest on 1K vertices: Implicit Non-learning (≈ 12 min) $>$ Explicit Blending (≈ 3 min) $\gg$ PENG2021 Peng et al. (2021) (38.4 s) $\gg$ PENG2022 Peng et al. (2022) (5.8 s) $\approx$ TSAI2022 Tsai & Liu (2022) (5.1 s) $>$ HOU2023 Hou et al. (2023) (4.2 s) $\gg$ Ours (0.054 s) $\gg$ WANG2022 Wang et al. (2022) (0.037 s) $>$ Deep3DMark Zhu et al. (2024b) (0.012 s). More results are provided in the supplementary material.

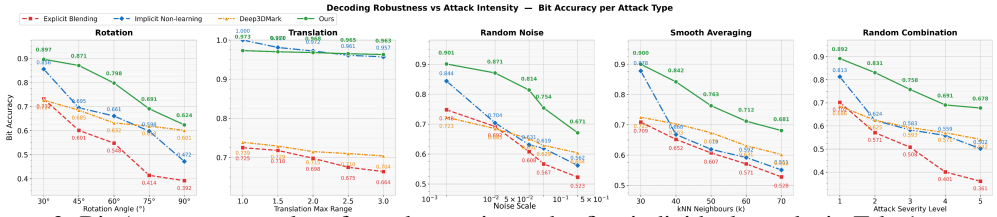


Figure 3: Bit Accuracy as a plot of attack severity under five individual attacks in Tab. 1 compared to top performance baselines.

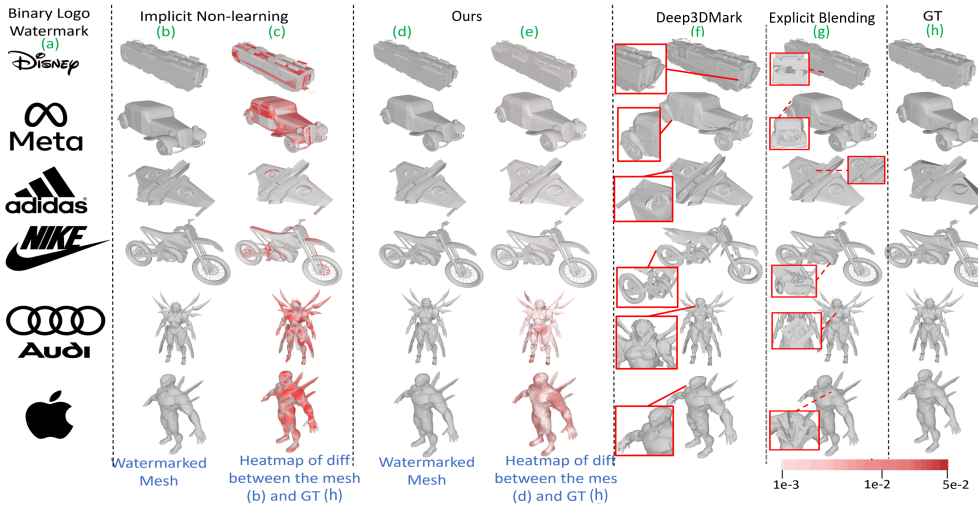


Figure 4: Qualitative comparison of watermarking methods on ShapeNet ("train", "car", "airplane", "motorcycle") and Objaverse++ classes ("hero"): implicit non-learning, our implicit learning-based approach, and explicit mesh blending. (a) Input watermark logo as a binary mask; (h) ground-truth mesh. (b,d,f,g) Watermarked meshes from the implicit non-learning method, our method, Deep3DMark, and explicit blending, respectively. (c,e) Corresponding difference heatmaps (red colormap indicates the difference norm, 10^{-3}–5×10^{-2}) relative to (h), highlighting geometric deviations. Zoom-in views emphasize the embedded logo regions, including subtle or non-visible areas from the selected viewpoint. Dashed lines separate each method and the ground-truth meshes for clearer visual comparison across results.

Table 1: Baseline comparison under attacks (attack strength is indicated next to each attack type), and tested on combined samples from ShapeNet classes and Objaverse++. We evaluate reconstruction difference (watermarked mesh vs. GT) and decoding robustness under five attacks accordingly (Rotation; Translation; Random Noise; Smooth Averaging; and Random Combination of Four Attacks). ↓ indicates lower is better, ↑ indicates higher is better.

Method	Type	Reconstruction Error	Decoding Accuracy					
		CD ↓ / HD ↓	BER ↓ / BitAcc ↑					
		–	W/o Attack	Rot (0-30°)	Trans (0-1)	Rand Noise ($\times 10^{-3}$)	Smooth (k=30)	Combo
PENG2021 Peng et al. (2021)	Algo	0.0271 / 0.681	0.386 / 0.614	0.479 / 0.521	0.459 / 0.541	0.498 / 0.502	0.488 / 0.512	0.512 / 0.488
PENG2022 Peng et al. (2022)		0.0239 / 0.523	0.418 / 0.582	0.382 / 0.618	0.379 / 0.621	0.416 / 0.584	0.439 / 0.561	0.466 / 0.534
TSAI2022 Tsai & Liu (2022)		0.0212 / 0.189	0.362 / 0.638	0.428 / 0.572	0.372 / 0.628	0.386 / 0.614	0.412 / 0.588	0.449 / 0.551
HOU2023 Hou et al. (2023)		0.0162 / 0.594	0.328 / 0.672	0.352 / 0.648	0.419 / 0.581	0.342 / 0.658	0.388 / 0.612	0.426 / 0.574
Implicit Non-learning		0.0010 / 0.148	**0.000 / 1.000**	0.144 / 0.856	**0.000 / 1.000**	0.156 / 0.844	0.122 / 0.878	0.187 / 0.813
Explicit Blending		0.0092 / 0.547	0.224 / 0.776	0.268 / 0.732	0.275 / 0.725	0.252 / 0.748	0.291 / 0.709	0.298 / 0.702
WANG2022 Wang et al. (2022)	DNN	0.0318 / 1.241	0.437 / 0.563	0.541 / 0.459	0.482 / 0.518	0.568 / 0.432	0.506 / 0.494	0.589 / 0.411
Deep3DMark Zhu et al. (2024b)		0.0125 / 0.572	0.235 / 0.765	0.273 / 0.727	0.261 / 0.739	0.277 / 0.723	0.275 / 0.725	0.314 / 0.686
Ours		**0.0002 / 0.024**	**0.000 / 1.000**	**0.103 / 0.897**	0.027 / 0.973	**0.099 / 0.901**	**0.100 / 0.900**	**0.108 / 0.892**

Table 2: Baseline comparison under **topology-changing attacks** at two severity levels (**E**: easy / **H**: difficult). Each cell: CD ↓ / BitAcc ↑. Shaded : full model (Ours).

Method	Severity	Simplification	Subdivision	Remesh	Compression
HOU2023	E	0.0181 / 0.534	0.0165 / 0.612	0.0168 / 0.584	0.0173 / 0.579
	H	0.0274 / 0.461	0.0193 / 0.510	0.0241 / 0.486	0.0216 / 0.531
Implicit Non-learning	E	0.0029 / 0.813	0.0021 / 0.842	0.0019 / 0.846	0.0025 / 0.835
	H	0.0058 / 0.602	0.0043 / 0.675	0.0041 / 0.678	0.0053 / 0.654
Explicit Blending	E	0.0106 / 0.657	0.0095 / 0.728	0.0098 / 0.694	0.0115 / 0.638
	H	0.0192 / 0.512	0.0143 / 0.620	0.0164 / 0.609	0.0148 / 0.613
WANG2022	E	0.0334 / 0.421	0.0322 / 0.458	0.0391 / 0.403	0.0327 / 0.435
	H	0.0461 / 0.348	0.0385 / 0.409	0.0431 / 0.384	0.0378 / 0.412
Deep3DMark 2024	E	0.0157 / 0.581	0.0124 / 0.631	0.0135 / 0.630	0.0169 / 0.602
	H	0.0252 / 0.490	0.0185 / 0.574	0.0236 / 0.492	0.218 / 0.541
Ours	E	**0.0018 / 0.868**	**0.0018 / 0.877**	**0.0026 / 0.787**	**0.0036 / 0.875**
	H	**0.0019 / 0.864**	**0.0018 / 0.871**	**0.0030 / 0.695**	**0.0039 / 0.867**

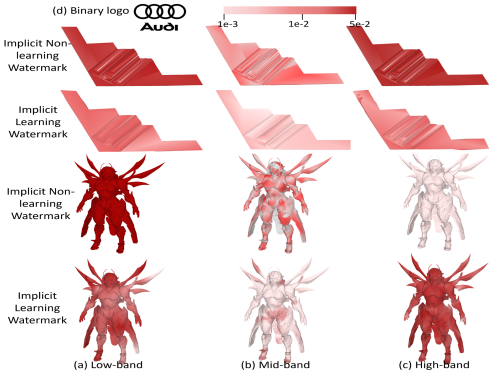


Figure 5: Ablation study of spectral-band fusion for watermark embedding, comparing low-frequency (lowest 25% eigenvalues), mid-frequency (middle 50%), and high-frequency (highest 25%) bands by difference heatmap.

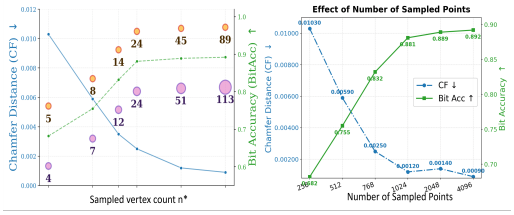


Figure 6: The left plot shows Chamfer Distance (CD, ↓) and Bit Accuracy (BitAcc, ↑). Stacked bubbles denote **GPU memory (MB)** (top) and **runtime (ms)** (bottom), with areas proportional to the corresponding values. The right plot reports the same metrics against the number of sampled input points. All results are evaluated under combined attacks.

4.2 ABLATION STUDY

We first present an ablation study in Fig. 5, comparing spectral fusion strategies for both implicit non-learning and learning methods across low-, mid-, and high-frequency bands under the same watermark logo. The results show that mid-band fusion in the implicit learning approach consistently preserves minimal vertex distortion, outperforming both inter-band variations in the non-learning method and intra-band variants of the learning approach. Tab. 4 ablates four core modules: medium-band LBO fusion gives the best geometry–decoding trade-off; BLIP-2 with MLP projection and binary quantization shows discrete message representation is critical for bit recovery; learnable attack training with $\mathcal{L}_{\text{reg}}$ yields a more transferable watermark than manual or single-type schedules; and the 3D-CNN decoder outperforms shallower alternatives via spatial feature aggregation. Tab. 3 shows reconstruction loss drives fidelity while message loss governs decoding accuracy.

Table 3: Ablation study on the effect of individual and total losses (6 types) by Chamfer Distance (CD) and Bit Accuracy (BitAcc) Metrics under combined attacks.

Loss Combination	1	2	3	4	5	6
$\mathcal{L}_{\text{recon}}$	✗	✓	✓	✓	✓	✓
$\mathcal{L}_{\text{disc}}$	✓	✗	✓	✓	✓	✓
$\mathcal{L}_{\text{adv}}$	✓	✓	✗	✓	✓	✓
$\mathcal{L}_{\text{msg}}$	✓	✓	✓	✗	✓	✓
$\mathcal{L}_{\text{reg}}$	✓	✓	✓	✓	✗	✓
CD ↓	0.0093	0.0021	0.0046	0.0011	0.0014	**0.0009**
BitAcc ↑	0.535	0.841	0.670	0.472	0.863	**0.892**

Table 4: Ablation study on model design across four modules: (*i*) spectral watermark fusion,(*ii*) multimodal message encoding, (*iii*) adversarial attack training strategy, and (*iv*) message decoding architecture. Metrics: Chamfer Distance (CD, ↓) and Bit Accuracy (BitAcc, ↑) under combined attacks. **Bold**: best per block. Green : full model (Ours).

Module	**Variant**	**CD** ↓	**BitAcc** ↑
(i) Watermark Fusion	w/o LBO Eigen-decomposition	0.0041	0.735
	Full-band spectral fusion	0.0019	0.812
	Low-band fusion only	0.0014	0.831
	High-band fusion only	0.0022	0.803
	Mid-band fusion (Ours)	**0.0009**	**0.892**
(ii) Message Encoding	CLIP encoder (replace BLIP-2)	0.0011	0.821
	w/o MLP projection head	0.0012	0.818
	w/o binary quantization (float message)	0.0012	0.723
	BLIP-2 + MLP + binary rounding (Ours)	**0.0009**	**0.892**
(iii) Attack Training	Manual random attack (non-learning)	0.0014	0.809
	Single attack type only	0.0012	0.851
	Attack + $\mathcal{L}_{\text{reg}}$ (Ours)	**0.0009**	**0.892**
(iv) Message Decoding	Deep3DMark Bin-voting decoder	0.0009	0.685
	MLP decoder (no 3D-CNN)	0.0009	0.706
	Shallow decoder (fewer layers)	0.0009	0.831
	3D-CNN decoder + MLP (Ours)	**0.0009**	**0.892**

Fig. 6 ablates the key hyperparameters: Higher vertex resolution reduces mesh distortion and improves decoding accuracy, at only a marginal cost in memory and runtime, while increasing the number of sampled points shows a similar trade-off, reducing CD and improving bit accuracy.

The *recoverability* of BLIP-2 semantic embedding enables converting the decoded bit sequence back into the original watermarks, whereas random or hash bits carry no learned semantic features and therefore cannot reconstruct the underlying watermark content. As presented in table 5, the BLIP-2-based encoder yields lower distortion and higher robustness against attacks than the simpler alternatives, which suffer significant BitAcc drops. This indicates that semantic embeddings from BLIP-2 or CLIP are more robust to attacks for watermarking than traditional SHA- or random bit.

Table 5: Ablation on various watermark message encoding methods under both clean and combined-attack conditions, CD ↓, BitAcc ↑. Shaded : Ours BLIP2.

Variant	Clean (W/o Attack)		Combo Attack	
	CD ↓	BitAcc ↑	CD ↓	BitAcc ↑
SHA-256 hash encoding (crypto, no semantics)	0.0028	0.942	0.0092	0.852
Random binary generator (input mapped to bits)	0.0016	0.971	0.0096	0.826
CLIP	0.0007	0.985	0.0036	0.867
BLIP-2-based (Ours)	**0.0002**	**1.000**	**0.0009**	**0.892**

5 Conclusion

We propose a robust mesh watermarking method based on spectral geometry fusion. It extracts intrinsic spectral features via the LB operator and embeds watermark signals in the spectral domain, achieving imperceptible yet resilient watermarking by encoding them as binary sequences.

Limitations & Outlook Despite its strong robustness, for CAD/BIM models with highly varying triangle sizes, further improvements are needed to generalize across more serious attacks. In future work, we will extend the proposed spectral watermarking to IP protection of 4D assets.

REFERENCES

Mathieu Andreux, Emanuele Rodola, Mathieu Aubry, and Daniel Cremers. Anisotropic laplace-beltrami operators for shape analysis. In *European conference on computer vision*, pp. 299–312. Springer, 2014.

Mikhail Belkin, Jian Sun, and Yusu Wang. Discrete laplace operator on meshed surfaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pp. 278–287, 2008.

Alexander I Bobenko and Boris A Springborn. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Pengzhen Chen, Yanwei Liu, Xiaoyan Gu, Enci Liu, Zhuoyi Shang, Xiangyang Ji, and Wu Liu. Plugmark: A plug-in zero-watermarking framework for diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17335–17345, Honolulu, Hawai'i, 2025a. IEEE.

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*, pp. 16325–16335, Nashville, the USA, 2025b. IEEE.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Han Fang, Kejiang Chen, Yupeng Qiu, Zehua Ma, Weiming Zhang, and Ee-Chien Chang. Dero: Diffusion-model-erasure robust watermarking. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 2973–2981, Melbourne, Australia, 2024. ACM Multimedia.

Michael S Floater, Kai Hormann, and Géza Kós. A general construction of barycentric coordinates over convex polygons. *advances in computational mathematics*, 24(1):311–331, 2006.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019.

Gangyang Hou, Bo Ou, Min Long, and Fei Peng. Separable reversible data hiding for encrypted 3d mesh models based on octree subdivision and multi-msb prediction. *IEEE Transactions on Multimedia*, 26:2395–2407, 2023.

Xiufeng Huang, Ziyuan Luo, Qi Song, Ruofei Wang, and Renjie Wan. Marksplatter: Generalizable watermarking for 3d gaussian splatting model via splatter image structure. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 12189–12198, Dublin, Ireland, 2025. ACM Multimedia.

Sumin In, Youngdong Jang, Utae Jeong, MinHyuk Jang, Hyeongcheol Park, Eunbyung Park, and Sangpil Kim. Compmarkgs: Robust watermarking for compressed 3d gaussian splatting. *arXiv preprint arXiv:2503.12836*, pp. 1–9, 2025.

Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting for robust watermarking. *arXiv preprint arXiv:2409.13222*, pp. 1–9, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742, Honolulu, Hawaii, USA, 2023. PMLR, PMLR.

Li Li, David Zhang, Zhigeng Pan, Jiaoying Shi, Kun Zhou, and Kai Ye. Watermarking 3d mesh by spherical parameterization. *Computers & Graphics*, 28(6):981–989, 2004.

Chendi Lin, Heshan Liu, Qunshu Lin, Zachary Bright, Shitao Tang, Yihui He, Minghao Liu, Ling Zhu, and Cindy Le. Objaverse++: Curated 3d object dataset with quality annotations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6813–6822, 2025.

Yu-Shen Liu, Hui Zhang, Jun-Hai Yong, Pi-Qiang Yu, and Jia-Guang Sun. Mesh blending. *The Visual Computer*, 21(11):915–927, 2005.

Zhili Liu, Kai Chen, Yifan Zhang, Jianhua Han, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, and James T Kwok. Implicit concept removal of diffusion models. In *European Conference on Computer Vision*, pp. 457–473, Milan, Italy, 2024. Springer, Springer.

Ziyuan Luo, Jun Liu, Haoliang Li, Anderson Rocha, and Renjie Wan. Mantlemark: Migrating watermarks from multi-view images to radiance fields via frequency modulation. *IEEE Transactions on Information Forensics and Security*, 21:256–271, 2025a.

Ziyuan Luo, Anderson Rocha, Boxin Shi, Qing Guo, Haoliang Li, and Renjie Wan. The nerf signature: Codebook-aided watermarking for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4652–4667, 2025b.

Rui Min, Sen Li, Hongyang Chen, and Minhao Cheng. A watermark-conditioned diffusion model for ip protection. In *European Conference on Computer Vision*, pp. 104–120, Milan, Italy, 2024. Springer, Springer.

Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(1):3–16, 2018.

Hewang Nie, Songfeng Lu, Junjun Wu, and Jianxin Zhu. Deep model intellectual property protection with compression-resistant model watermarking. *IEEE Transactions on Artificial Intelligence*, 5 (7):3362–3373, 2024.

Fei Peng, Bo Long, and Min Long. A general region nesting-based semi-fragile reversible watermarking for authenticating 3d mesh models. *IEEE transactions on circuits and systems for video technology*, 31(11):4538–4553, 2021.

Fei Peng, Tongxin Liao, and Min Long. A semi-fragile reversible watermarking for authenticating 3d models in dual domains based on variable direction double modulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8394–8408, 2022.

Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 49–56, Los Angeles, CA, USA, 1999. ACM SIGGRAPH.

Amit Kumar Singh, Basant Kumar, Ghanshyam Singh, and Anand Mohan. *Medical image watermarking*. Springer, India, 2017.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N Asokan. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 4417–4425, Chengdu, China, 2021. ACM Multimedia.

Jingxuan Tan, Nan Zhong, Zhenxing Qian, Xinpeng Zhang, and Sheng Li. Deep neural network watermarking against model extraction attack. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 1588–1597, Ottawa, Canada, 2023. ACM.

Yuqi Tan, Yuang Peng, Hao Fang, Bin Chen, and Shu-Tao Xia. Waterdiff: Perceptual image watermarks via diffusion model. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3250–3254, 513 Yeongdong-daero, Gangnam District, Seoul, South Korea, 2024. IEEE, IEEE.

Yuan-Yu Tsai and Hong-Lin Liu. Integrating coordinate transformation and random sampling into high-capacity reversible data hiding in encrypted polygonal models. *IEEE Transactions on Dependable and Secure Computing*, 20(4):3508–3519, 2022.

Feng Wang, Hang Zhou, Han Fang, Weiming Zhang, and Nenghai Yu. Deep 3d mesh watermarking with self-adaptive robustness. *Cybersecurity*, 5(1):24, 2022.

Xiao Wu, Hu Guan, Ying Huang, Chunhua Song, Baoning Niu, Shuwu Zhang, and Jie Liu. A comprehensive review of three-dimensional watermarking algorithms. In *Proceedings of the 2024 10th International Conference on Communication and Information Processing*, pp. 469–475, Hainan, China, 2024. IEEE.

Jiaqi Yang, Xuequan Lu, and Wenzhi Chen. A robust scheme for 3d point cloud copy detection. *arXiv preprint arXiv:2110.00972*, pp. 1–9, 2021.

Innfarn Yoo, Huiwen Chang, Xiyang Luo, Ondrej Stava, Ce Liu, Peyman Milanfar, and Feng Yang. Deep 3d-to-2d watermarking: Embedding messages in 3d meshes and extracting them from 2d renderings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10031–10040, New Orleans, LA, USA, 2022. IEEE CVF.

Zhiqiang Yu, Horace HS Ip, and LF Kwok. A robust watermarking scheme for 3d triangular mesh models. *Pattern recognition*, 36(11):2603–2614, 2003.

Khandoker Ashik Uz Zaman, Mohammad Zahangir Alam, Mohammed NM Ali, and Mahdi H Miraz. Deep neural watermarking for robust copyright protection in 3d point clouds. *arXiv preprint arXiv:2510.27533*, pp. 1–9, 2025.

Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12805–12812, the Hilton New York Midtown, New York, the USA, 2020. AAAI.

Jie Zhang, Dongdong Chen, Jing Liao, Zehua Ma, Han Fang, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Robust model watermarking for image processing networks via structure consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6985–6992, 2024a.

Lijun Zhang, Xiao Liu, Antoni V Martin, Cindy X Bearfield, Yuriy Brun, and Hui Guan. Attack-resilient image watermarking using stable diffusion. *Advances in Neural Information Processing Systems*, 37:38480–38507, 2024b.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning*, pp. 42187–42199, 1 Expo Drive, Singapore 486150, 2023. PMLR, PMLR.

Dong Zheng, Sha Wang, and Jiying Zhao. Rst invariant image watermarking algorithm with mathematical modeling and analysis of the watermarking processes. *IEEE transactions on image processing*, 18(5):1055–1068, 2009.

Hongyu Zhu, Sichu Liang, Wentao Hu, Li Fangqi, Ju Jia, and Shi-Lin Wang. Reliable model watermarking: Defending against theft without compromising on evasion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10124–10133, Melbourne, Australia, 2024a. ACM Multimedia.

Xingyu Zhu, Guanhui Ye, Xiapu Luo, and Xuetao Wei. Rethinking mesh watermark: Towards highly robust and adaptable deep 3d mesh watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7784–7792, Vancouver, Canada, 2024b. AAAI.

Xingyu Zhu, Xiapu Luo, and Xuetao Wei. Dreamark: Rooting watermark in score distillation sampling generated neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11013–11021, Pennsylvania, USA, 2025. AAAI.