RETHINKING HIGH-FREQUENCY FORENSIC SIGNALS FOR POISON-SPLAT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Poison-Splat exploits adaptive densification in 3D Gaussian Splatting (3DGS) to increase reconstruction cost through perturbed training images. Recent defenses use *high-frequency patterns* as forensic evidence of poisoning and report strong performance on standard benchmarks. We question whether these signals provide a reliable fingerprint of Poison-Splat when legitimate scene content itself contains texture. To investigate, we introduce **TextureBench**, a benchmark of ordinary, naturally occurring scenes with legitimate fine-scale texture, paired with valid clean 3DGS reconstructions and matched poisoned views. Using this benchmark, we analyze three aspects of defense reliability: detection, preservation of clean content during purification, and their interaction in composed pipelines. The RemedyGS detector's AUROC drops from 0.980 on existing benchmarks to 0.617 on TextureBench, while several high-frequency statistics reverse their clean–poisoned ordering. Purification also causes larger reconstruction losses on clean textured scenes, and false-positive detections propagate this damage through the composed defense. Finally, under the same task-specific training supervision, a detector using frozen DINOv2 features improves TextureBench AUROC to 0.904 and reduces downstream damage with the purifier held fixed, although substantial errors remain. Our results prompt re-thinking of using high frequency patterns for Poison-Splat detection. **We include the code and benchmark in the submission.**

1 INTRODUCTION

3D Gaussian Splatting (3DGS) has become an effective representation for high-quality reconstruction and real-time rendering (Kerbl et al., 2023). Its success has enabled applications in visual SLAM (Matsuki et al., 2024; Keetha et al., 2024) and scene editing (Chen et al., 2024), with extensions to dynamic scene reconstruction (Wu et al., 2024; Li et al., 2024). Central to 3DGS is adaptive density control, which allocates additional Gaussian primitives to capture under-reconstructed scene detail. This flexibility also exposes a resource-exhaustion vulnerability. Poison-Splat exploits the densification process through bounded perturbations to training views, substantially increasing Gaussian count and the resulting memory and computation costs (Lu et al., 2025). Reliable defenses must mitigate this threat while preserving the detail needed for faithful reconstruction of legitimate scenes.

1.1 HIGH-FREQUENCY CUES IN POISON-SPLAT DETECTION

Recent work observes that Poison-Splat leaves a characteristic footprint in the training views: sharp local variation and elevated high-frequency content (Chen et al., 2026a; Li et al., 2026; Qiao et al., 2026). Existing defenses treat this high-frequency pattern as forensic evidence of poisoning, and rely on it in progressively stronger ways. Shield-Splat uses scale-normalized total variation as a proxy for poisoning, both to flag suspicious views and to set the strength of purification (Chen et al., 2026a). RemedyGS learns local texture and edge features to classify each view as clean or poisoned, and purifies only the views its detector flags (Li et al., 2026). DefenseSplat does not decide whether an input is poisoned at all; it suppresses the high-frequency wavelet components of every input during purification (Qiao et al., 2026). Spectral Defense moves the same assumption into the representation itself, pruning Gaussians by frequency and regularizing the spectrum of the reconstruction (Chen et al., 2026b). Across these defenses, high-frequency magnitude serves as the

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.

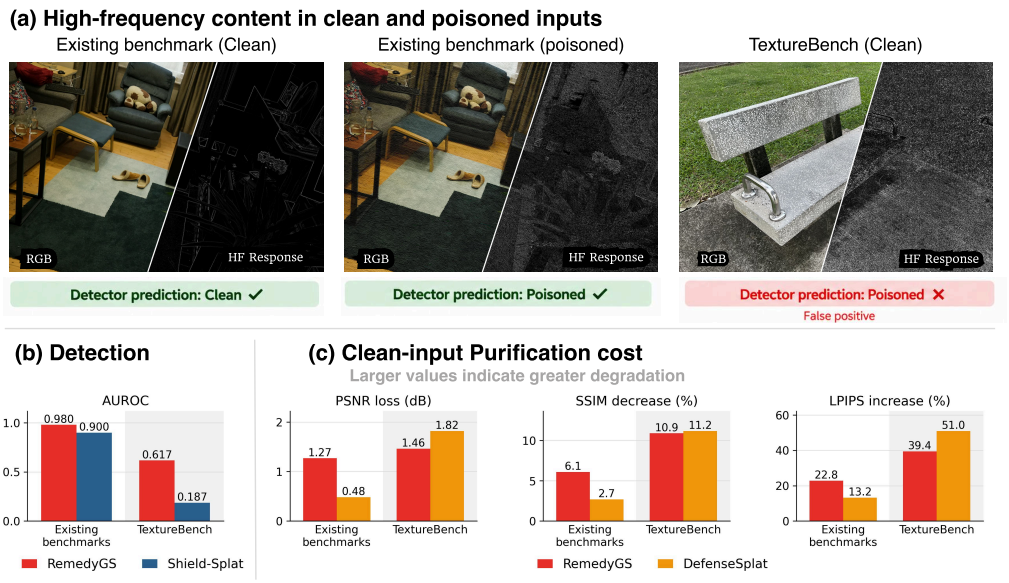


Figure 1: **Natural texture challenges Poison-Splat defenses.** (a) Clean TextureBench images exhibit high-frequency responses without adversarial perturbations, challenging their use as forensic cues. The prediction strips illustrate the resulting failure: legitimate texture is mistaken for poisoning. (b) Detectors that perform strongly on existing benchmarks degrade sharply on TextureBench. (c) Purifying clean TextureBench inputs causes greater reconstruction quality losses than on existing benchmarks, particularly in SSIM and LPIPS.

cue for detecting an attack, the target of purification, or both. Each rests on the same premise: that strong high-frequency responses are *attack-specific*, and that legitimate training views do not exhibit them to a comparable degree.

Existing evaluations report strong detection or attack-mitigation performance. Consistent with these results, the RemedyGS detector achieves 0.980 AUROC in our evaluation on pooled images from four existing benchmarks (Mildenhall et al., 2019; Wang et al., 2023; Knapitsch et al., 2017; Barron et al., 2022).

1.2 Rethinking Forensic Specificity

We question whether high frequency patterns provide a reliable forensic fingerprint of Poison-Splat. We examine their specificity in Poison-Splat detection: patterns associated with poisoning may also arise from legitimate scene content. Naturally occurring surface texture, repeated structures, and fine geometry can produce legitimate high-frequency responses without adversarial perturbations. A detector may mistake such detail for poisoning, while purification may suppress information needed for faithful reconstruction. This raises our central question: *Can high frequency patterns reliably distinguish poisoned inputs from naturally textured clean images, and how do detection errors affect reconstruction?*

We investigate this question using naturally textured clean inputs as hard negatives. Natural adversarial examples demonstrate the value of challenging real-world inputs for exposing failures overlooked by standard evaluations (Hendrycks et al., 2021). Although existing reconstruction benchmarks contain textured scenes, conventional evaluations do not specifically test clean views dominated by fine-scale detail. These views allow us to examine both whether legitimate texture triggers false detections and whether purification preserves the scene detail that reconstruction should retain.

To support this analysis, we introduce **TextureBench**, a texture-rich 3D benchmark of 28 real-world scenes. Each scene provides overlapping multi-view captures with matched clean and Poison-Splat views. Clean reconstructions enable us to examine both image-level detection and the downstream effects of defense decisions. All TextureBench scenes are excluded from detector training, model

selection, and threshold calibration. The benchmark provides a targeted stress test of defense reliability rather than an estimate of failure rates on randomly sampled scenes.

1.3 Our Analysis and Contributions

Using TextureBench, we examine defense reliability through three complementary evaluations. **First, detection:** we test whether the detectors distinguish poisoned inputs from naturally textured clean views. **Second, purification fidelity:** we apply purifiers to clean inputs and measure the resulting reconstruction distortion. **Third, composed defense:** we examine how detector decisions affect reconstruction when only flagged views undergo purification. Together, these evaluations connect the reliability of forensic cues to the consequences of using them in a defense pipeline.

Our results expose substantial limitations (Figure 1). The RemedyGS detector achieves 0.980 AUROC on pooled existing benchmarks but only 0.617 on TextureBench. Its false-positive rate at 95% poisoning recall increases from 9.9% to 89.0%. Statistical detectors also perform poorly under score directions fixed across datasets. The consequences extend beyond detection: both evaluated purifiers cause greater reconstruction quality losses on clean TextureBench scenes, with particularly pronounced differences in SSIM and LPIPS. When detection and purification are composed, false positives trigger unnecessary processing and damage reconstruction despite the absence of an attack.

We further investigate how detection generalization depends on the image representation. Motivated by the transferability of pretrained visual features in image forensics (Ojha et al., 2023), we train classification heads on frozen MAE and DINOv2 encoders (He et al., 2022; Oquab et al., 2023) using a shared DL3DV clean/Poison-Splat training split (Ling et al., 2024). MAE and DINOv2 improve TextureBench AUROC to 0.751 and 0.904, respectively. With the purifier held fixed, DINOv2-based gating also reduces damage to clean reconstructions. These results show that detector choice affects both generalization and downstream fidelity. They do not identify the features responsible for the improvement, and improved clean-input preservation alone does not establish attack mitigation effectiveness. Our main contributions are:

- **Rethinking high-frequency forensic signals.** We empirically reassess the reliability of high frequency cues. Naturally textured clean scenes expose failures in statistical and learned detectors, challenging the use of these cues as reliable forensic fingerprints for Poison-Splat detection.
- **A systematic analysis of defense reliability.** We introduce TextureBench, a benchmark containing ordinary, naturally occurring scenes with legitimate fine-scale texture. The benchmark includes matched clean and poisoned views to support an analysis of detection, purification fidelity, and their interaction. We show how false positives damage clean reconstruction and examine how pretrained visual representations mitigate these failures.

2 Related Work

Computation-cost attacks on 3DGS. Adversarial attacks can compromise computational efficiency as well as prediction accuracy. Prior work demonstrates such threats in adaptive neural networks (Haque et al., 2020; Hong et al., 2020), image caption generation (Chen et al., 2022), and LiDAR detection (Liu et al., 2023). Training-data poisoning can also introduce backdoors that increase inference cost (Chen et al., 2023). In 3DGS, Poison-Splat exploits adaptive densification: perturbed training views induce excessive Gaussian growth, increasing memory consumption and reconstruction time (Lu et al., 2025). The attack is generated using a proxy model and can transfer to black-box victims with different 3DGS representations and configurations, without requiring access to the victim's internal implementation. Its resource-exhaustion objective differs from StealthAttack and GaussTrap, which manipulate rendered content at selected viewpoints (Ke et al., 2025; Hong et al., 2026). Our work examines the reliability of defenses against Poison-Splat and their effects on legitimate scene content.

Forensic signals and Poison-Splat defenses. Existing defenses exploit local variation, texture, and spectral patterns associated with poisoned inputs. Shield-Splat uses scale-normalized total variation to identify suspicious views and select purification strength (Chen et al., 2026a). RemedyGS learns texture and edge features for detection and applies a learned purifier to flagged inputs (Li et al.,

2026). DefenseSplat suppresses high-frequency wavelet components during purification (Qiao et al., 2026), while Spectral Defense combines frequency-aware Gaussian pruning with spectral regularization (Chen et al., 2026b). In this work, we empirically show that naturally textured clean scenes challenge the reliability of these forensic cues, exposing detection failures and reconstruction damage from unnecessary purification.

Natural counterexamples and benchmark design. Existing Poison-Splat evaluations rely on reconstruction benchmarks such as Mip-NeRF 360, Tanks and Temples, and LLFF (Barron et al., 2022; Knapitsch et al., 2017; Mildenhall et al., 2019). Although these datasets include textured scenes, their use does not establish detection reliability on clean views dominated by fine-scale detail. Studies of newly collected test sets (Recht et al., 2019), natural distribution shifts (Koh et al., 2021), and natural adversarial examples (Hendrycks et al., 2021) demonstrate the value of challenging benchmark generalization. We bring this perspective to Poison-Splat defense through TextureBench, a texture-rich 3D benchmark with matched clean and poisoned views. These hard negatives support an analysis that extends from detection reliability to purification fidelity and the effects of detector decisions on reconstruction.

3 PRELIMINARIES

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) represents a scene using a set of anisotropic Gaussian primitives Kerbl et al. (2023). Each primitive stores a 3D center $\boldsymbol{\mu}$, covariance $\boldsymbol{\Sigma}$, opacity, and view-dependent color. Given the Gaussian parameters θ and a camera pose π_k, a differentiable renderer produces an image $R(\theta, \pi_k)$. Reconstruction from K calibrated views minimizes

$$\theta^* = \arg\min_{\theta} \sum_{k=1}^{K} \mathcal{L}_{\text{rec}}(R(\theta, \pi_k), I_k)\,, \tag{1}$$

where I_k is the target image and $\mathcal{L}_{\text{rec}}$ combines photometric and structural reconstruction terms.

A key component of 3DGS is adaptive density control. During optimization, primitives in under-reconstructed regions are cloned or split, while primitives with limited contribution are pruned. This mechanism enables 3DGS to represent detailed scenes without fixing the number of Gaussians in advance. However, it also makes the resulting Gaussian count, and hence reconstruction cost, sensitive to the visual complexity of the training views.

Poison-Splat attack. Poison-Splat (Lu et al., 2025) exploits this dependence in a 3DGS-as-a-service setting: an attacker uploads poisoned images, or tampers with another user's, so that the victim's reconstruction consumes excessive resources. Given a clean capture $\mathcal{I} = \{I_k\}_{k=1}^{K}$, the attacker seeks poisoned views $\mathcal{I}_{\text{poi}} = \{I_k^{\text{poi}}\}_{k=1}^{K}$ that maximize the reconstruction cost $\mathcal{C}$ under a per-pixel budget:

$$\max_{\mathcal{I}_{\text{poi}}} \mathcal{C}(\theta^*(\mathcal{I}_{\text{poi}})) \quad \text{s.t.} \quad \theta^*(\mathcal{I}_{\text{poi}}) = \arg\min_{\theta} \mathcal{L}_{\text{rec}}(\theta; \mathcal{I}_{\text{poi}}), \qquad \|I_k^{\text{poi}} - I_k\|_\infty \le \epsilon \;\; \forall k. \tag{2}$$

Resource cost and discrete changes in Gaussian count are not differentiable, so the attack works through the densification rule: non-smooth textures raise the positional gradient past β, and Gaussian count grows with the total variation of the views. Poison-Splat therefore sets $\mathcal{C}$ to the total variation $S_{\text{TV}}(I) = \sum_{i,j} \sqrt{\|I_{i+1,j} - I_{i,j}\|_2^2 + \|I_{i,j+1} - I_{i,j}\|_2^2}$ of a proxy 3DGS model's rendered views, maximizes it while keeping the inputs within the ϵ-ball, and emits the clipped renderings as the poisoned images. The footprint this leaves is sharpened edges and elevated high-frequency content, and existing defenses treat that footprint as a signature of poisoning; whether it is one is the question this paper examines.

4 TEXTUREBENCH: EVALUATING DEFENSES UNDER NATURAL TEXTURE

Effective Poison-Splat defenses must distinguish malicious perturbations from legitimate scene detail. This distinction becomes challenging when clean inputs contain strong high-frequency responses of the kind used as forensic cues for poisoning. We introduce TextureBench, a benchmark of 28 texture-rich scenes, to evaluate defenses in this setting. It supports two complementary tests: whether detectors distinguish clean textured views from poisoned views, and whether purification preserves the detail needed for faithful clean reconstruction.

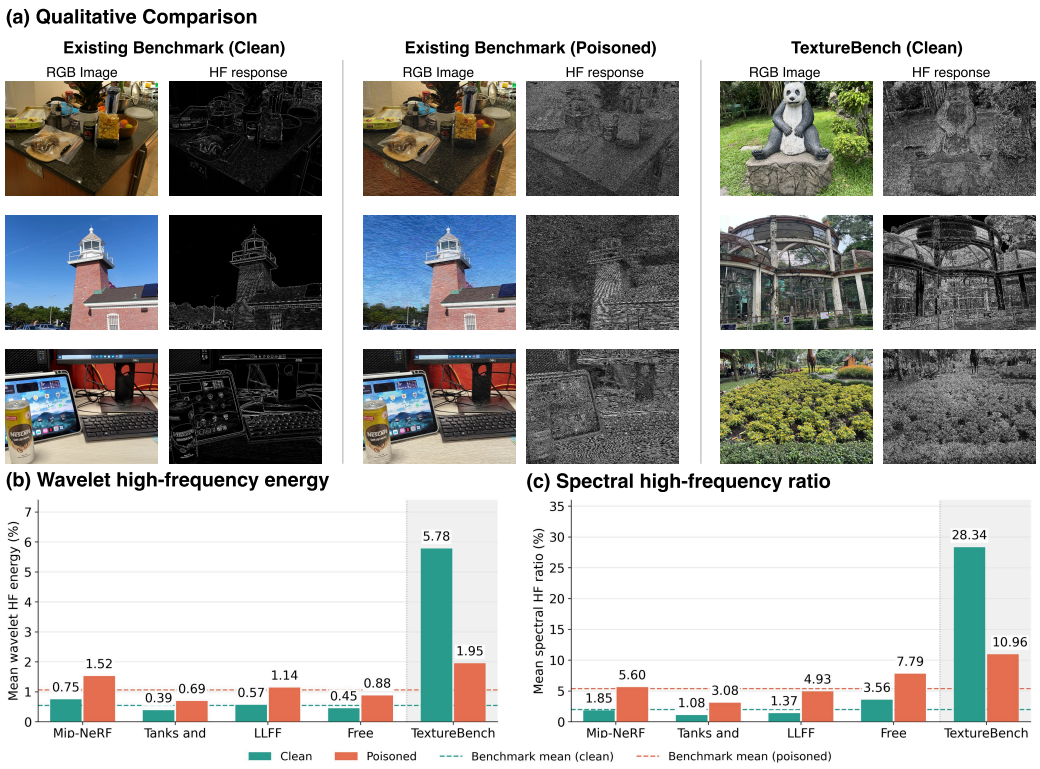


Figure 2: **High-frequency content is not exclusive to poisoning.** (a) Clean TextureBench views exhibit high-frequency responses from legitimate scene detail. (b) Mean wavelet high-frequency energy and (c) mean spectral high-frequency ratio for clean and Poison-Splat inputs ($\epsilon = 16/255$). Poisoned inputs have higher means on existing benchmarks, whereas clean inputs have higher means on TextureBench, exceeding even the poisoned means of all existing benchmarks shown. Natural texture thus challenges the use of high-frequency magnitude as a reliable forensic cue.

4.1 Benchmark Construction

Design principles. TextureBench contains ordinary real-world scenes with legitimate fine-scale texture, including vegetation, ground surfaces, architectural patterns, repeated structures, and everyday objects. We use *natural texture* to describe this content present before poisoning, including both organic and manufactured surfaces. Scene selection prioritizes diverse texture structures, detail visible across multiple views, and valid multi-view reconstruction. By emphasizing such content, TextureBench provides a targeted stress test of whether defenses fail on plausible clean inputs encountered in practice, rather than an estimate of failure rates on randomly sampled scenes.

Collection and diversity. TextureBench comprises 28 real-world scenes spanning architectural surfaces (5), repeated structures (7), vegetation and natural textures (6), everyday objects (3), and ground surfaces (7). Each scene centers on an object or a localized region with dense texture, captured through overlapping smartphone photographs, typically targeting approximately 40 views from multiple viewpoints. The captures include granular surfaces, periodic patterns, and irregular organic structures, providing diverse sources of legitimate high-frequency content. Each clean capture is paired with corresponding Poison-Splat views, enabling evaluation of both classes within the same scene. *All TextureBench scenes are excluded from detector training, model selection, and threshold calibration.*

Reconstruction validation. We validate each scene for multi-view 3D reconstruction by registering its images with COLMAP Schonberger & Frahm (2016) and training vanilla 3DGS for 30,000 iterations on the undistorted clean views. Every scene supports camera registration and a clean 3DGS

reconstruction, ensuring that its rich texture is part of reconstructable scene content. These clean reconstructions provide references for measuring quality changes introduced by defenses.

4.2 Natural Texture Challenges Forensic Specificity

High-frequency responses in clean scenes. Figure 2(a) compares clean and poisoned views from existing benchmarks with clean TextureBench views. Poisoning introduces strong high-frequency responses in the conventional examples. However, clean TextureBench scenes also exhibit strong responses from surface texture, repeated structures, and foliage. These examples illustrate that high-frequency responses can also arise from *legitimate scene content*.

Different trends across datasets. Figures 2(b,c) report mean wavelet high-frequency energy and spectral high-frequency ratio. On Mip-NeRF 360 Barron et al. (2022), Tanks and Temples Knapitsch et al. (2017), LLFF Mildenhall et al. (2019), and Free Wang et al. (2023), both means are higher for poisoned inputs than for clean inputs. TextureBench reverses this ordering: its clean views have higher mean wavelet energy (5.78% versus 1.95%) and spectral ratio (28.34% versus 10.96%) than its poisoned views. The clean TextureBench means also exceed the poisoned means of every conventional benchmark shown. Thus, the association between larger high-frequency scores and poisoning does not hold consistently across these datasets.

Implications for detection and purification. These observations show that high-frequency patterns are not specific to poisoning. On existing benchmarks, poisoned views have stronger high-frequency responses than clean views; on TextureBench, the ordering reverses. A strong response alone therefore does not establish an attack. Aggregate statistics do not determine image-level detectability, but they raise two questions that Section 5 answers: whether detectors separate natural texture from attack artifacts, and whether purification, applied to all clean views or only to those a detector flags, preserves reconstruction quality.

5 Experiments

5.1 Experimental Setup

Datasets and attack setting. Following the RemedyGS Li et al. (2026) training protocol, all learned detectors use the same dataset split from DL3DV and the same clean–Poison-Splat training pairs. We evaluate detection on Mip-NeRF 360, Tanks and Temples, LLFF, Free, and TextureBench. Reconstruction experiments use Mip-NeRF 360, Tanks and Temples, and TextureBench. *TextureBench is excluded from training, model selection, and threshold calibration.* For evaluation, we similarly follow the main benchmark in RemedyGS Li et al. (2026).

Detectors and purifiers. We evaluate two defense components: a *detector* identifies suspicious views, while a *purifier* processes views before reconstruction. For RemedyGS Li et al. (2026), we use the officially released pretrained weights and evaluate its detector and purifier separately and together. Other detection baselines include Shield-Splat Chen et al. (2026a) and HH-MAD Donoho & Johnstone (1994), which measures the median absolute deviation of diagonal high-frequency wavelet coefficients. We also construct two statistics-based detectors motivated by the frequency observations in prior work: wavelet high-frequency energy based on DefenseSplat Qiao et al. (2026) and spectral high-frequency ratio based on Spectral Defense Chen et al. (2026b). For purification, we evaluate the RemedyGS purifier and the frequency-aware DefenseSplat purifier.

We further investigate a *foundation-model-based detection approach*: we extract image embeddings from frozen MAE and DINOv2 encoders He et al. (2022); Oquab et al. (2023) and train a classification head to distinguish clean from poisoned views. Both heads use the same DL3DV training split and supervision as the RemedyGS detector, allowing us to examine whether pretrained representations improve detection on texture-rich clean inputs.

Evaluation protocols and metrics. We consider three complementary protocols:

- **Detection.** We formulate Poison-Splat detection as an *anomaly detection* task Hendrycks & Gimpel (2016); Liang et al. (2017), with poisoned inputs treated as anomalies. Following RemedyGS, we use an image-level protocol: each view is scored independently. We report anomaly detection metrics, AUROC and the false-positive rate at 95% true-positive rate

Table 1: **Poison-Splat detectability across datasets.** Poisoned views are the positive class. We report AUROC (AUC)↑ and FPR@95 (FPR)↓ from pooled image scores, with score directions fixed across datasets. Existing pooled combines Mip-NeRF 360, Tanks and Temples, LLFF and Free. TextureBench is excluded from training and calibration. Bold denotes the best result in each column.

	Mip-NeRF 360		Tanks and Temples		LLFF		Free		Existing pooled		TextureBench (Ours)	
Detector	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓
Statistics-based detectors												
Shield-Splat	0.932	0.147	0.919	0.370	0.997	0.020	0.947	0.147	0.900	0.366	0.187	0.887
DefenseSplat†	0.841	0.360	0.783	0.739	0.824	0.574	0.856	0.474	0.792	0.692	0.157	0.960
Spectral Defense†	0.896	0.281	0.872	0.536	0.991	0.030	0.774	0.516	0.849	0.462	0.172	0.935
HH-MAD	0.944	0.160	0.911	0.252	0.975	0.131	0.787	0.514	0.864	0.309	0.199	0.896
Model-based detectors												
RemedyGS detector	0.998	**0.000**	0.969	0.156	**1.000**	**0.000**	0.985	0.049	0.980	0.099	0.617	0.890
MAE-based detector	**1.000**	0.001	0.988	0.031	**1.000**	**0.000**	**1.000**	**0.000**	0.986	0.056	0.751	0.726
DINOv2-based detector	**1.000**	**0.000**	**0.999**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**	**0.999**	**0.000**	**0.904**	**0.334**

†Our implementations of wavelet high-frequency energy (DefenseSplat) and spectral high-frequency ratio (Spectral Defense), following the descriptions in the respective original papers.

(FPR@95). Metrics are computed from pooled image-level scores, with score directions fixed across datasets. The existing-benchmark pool combines all four datasets.

- **Purifier-only fidelity.** We measure how much purification distorts legitimate clean content. Each purifier processes every view of a clean scene, independently of detector decisions. We reconstruct the scene from the purified views and evaluate the rendered images against the original views, measuring the effect of purification on reconstruction fidelity.
- **Composed detection–purification.** We evaluate each detector–purifier pipeline on clean and poisoned inputs. Only flagged views undergo purification before reconstruction. We measure reconstruction quality and Gaussian count to assess clean-input preservation and attack mitigation. Following RemedyGS, we use a fixed detection threshold of $p \geq 0.5$ for this evaluation.

For both reconstruction protocols, vanilla 3DGS Kerbl et al. (2023) is trained for 30,000 iterations on the resulting views. We report Gaussian count, PSNR, SSIM Wang et al. (2004), and LPIPS Zhang et al. (2018) relative to an unmodified clean reconstruction, averaging equally over scenes. The existing-benchmark reconstruction pool combines Mip-NeRF 360 and Tanks and Temples.

5.2 Poison-Splat Detectability

Strong performance on existing benchmarks does not transfer. Table 1 shows that the RemedyGS detector achieves 0.980 AUROC on the existing-benchmark pool but falls to 0.617 on TextureBench. Its FPR@95 increases from 0.099 to 0.890: detecting 95% of poisoned views requires flagging 89.0% of clean TextureBench views. Statistics-based detectors also degrade sharply, with TextureBench AUROCs ranging from 0.157 to 0.199 under score directions fixed across datasets. Strong performance on existing benchmarks therefore does not ensure reliable detection on naturally textured clean inputs.

Pretrained representations help distinguish natural texture. The statistical detectors' failures are consistent with Section 4.2: legitimate texture can produce stronger high-frequency responses than poisoning, undermining scores that associate larger responses with an attack. Under the same DL3DV supervision, MAE-based and DINOv2-based detectors improve TextureBench AUROC to 0.751 and 0.904, respectively. DINOv2 also reduces FPR@95 to 0.334, although substantial false positives remain. These results suggest that large-scale pretraining on diverse natural images provides features that help separate legitimate scene detail from attack artifacts, allowing the classification heads to use cues beyond high-frequency magnitude.

Finding 1: High-frequency patterns are not specific to poisoning; strong detection performance on existing benchmarks does not reliably transfer to naturally textured scenes.

Table 2: **Purifier-only fidelity on clean inputs.** Clean uses images without purification. Each purifier processes all views before 30k-step 3DGS training. PSNR, SSIM, and LPIPS measure reconstruction distortion against the original clean images. Results are averaged equally over scenes. Red values show percentage changes from the clean baseline. Bold and underline mark the largest and second-largest relative Gaussian-count reductions and quality losses, respectively.

Purifier	#G (M)		PSNR↑		SSIM↑		LPIPS↓	
	Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%
Mip-NeRF 360								
Clean (no purification)	2.69	–	29.48	–	0.878	–	0.229	–
DefenseSplat purifier	1.83	−32.0	28.74	−2.5	0.844	−3.9	0.271	+18.3
RemedyGS purifier	2.16	−19.7	27.77	−5.8	0.811	−7.6	0.295	+28.8
Tanks and Temples								
Clean (no purification)	1.48	–	27.17	–	0.869	–	0.277	–
DefenseSplat purifier	1.08	−27.0	27.01	−0.6	0.856	−1.5	0.298	+7.6
RemedyGS purifier	1.20	−18.9	26.48	−2.5	0.835	−3.9	0.324	+17.0
Existing benchmarks (pooled)								
Clean (no purification)	2.16	–	28.47	–	0.874	–	0.250	–
DefenseSplat purifier	1.50	−30.6	27.99	−1.7	0.850	−2.7	0.283	+13.2
RemedyGS purifier	1.74	−19.4	27.20	−4.5	0.821	−6.1	0.307	+22.8
TextureBench (ours)								
Clean (no purification)	4.09	–	24.92	–	0.819	–	0.208	–
DefenseSplat purifier	2.87	−29.8	23.10	**−7.3**	0.727	**−11.2**	0.314	**+51.0**
RemedyGS purifier	2.77	**−32.3**	23.46	−5.9	0.730	−10.9	0.290	+39.4

5.3 Purification Fidelity on Clean Inputs

Purification causes larger losses on TextureBench. Table 2 evaluates each purifier on clean inputs independently of detector decisions. On existing benchmarks, DefenseSplat and RemedyGS reduce PSNR by 0.48 and 1.27 dB, respectively. On TextureBench, these losses increase to 1.82 and 1.46 dB. The difference is more pronounced in SSIM and LPIPS. DefenseSplat reduces SSIM by 11.2% and increases LPIPS by 51.0% on TextureBench, compared with 2.7% and 13.2% on the pooled existing benchmarks. RemedyGS shows the same trend: SSIM decreases by 10.9% and LPIPS increases by 39.4%, compared with 6.1% and 22.8%, respectively. Both purifiers therefore cause greater aggregate quality losses on naturally textured clean scenes.

Natural detail is vulnerable to purification. The high-frequency patterns in these clean inputs are legitimate scene content that reconstruction should preserve. Purification also reduces the TextureBench Gaussian count by 29.8% with DefenseSplat and 32.3% with RemedyGS. Fewer Gaussians alone do not establish detail loss, but the accompanying fidelity degradation is consistent with suppression of natural scene detail. Because no attack is present, these losses represent damage to clean reconstruction. Modest degradation on conventional benchmarks therefore does not establish that a purifier preserves texture-rich scene content.

Finding 2. The evaluated purifiers cause greater reconstruction quality losses on clean, naturally textured scenes than on existing benchmarks.

5.4 Composed Detection–Purification

We evaluate detection–purification pipelines on clean and poisoned inputs (Table 3). These settings test whether a defense preserves legitimate inputs while mitigating attacks.

Better gating reduces damage to clean inputs. The evaluated pipelines largely preserve clean reconstruction quality on existing benchmarks, but cause substantial damage on TextureBench. Among the evaluated defense pipelines, DINOv2 with RemedyGS purification best preserves clean TextureBench reconstructions across all four metrics: Gaussian count remains closest to the clean baseline, while PSNR and SSIM are highest and LPIPS is lowest. With the RemedyGS purifier fixed,

Table 3: **Detection–purification on clean and poisoned inputs.** Only flagged views undergo purification before 30k-step 3DGS training. No defense uses the corresponding inputs without purification. Quality metrics are computed on training views against the original clean images and averaged equally over scenes. $\Delta\%$ *denotes percentage change from the unmodified clean baseline.* Bold and underline mark the best and second-best results, respectively, among defense pipelines on TextureBench. For clean inputs, results are ranked by closeness to the clean baseline.

Detector	Purifier	Clean inputs								Poisoned inputs							
		#G (M)		PSNR↑		SSIM↑		LPIPS↓		#G (M)		PSNR↑		SSIM↑		LPIPS↓	
		Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%	Value	Δ%
Mip-NeRF 360																	
No defense		2.69	–	29.48	–	0.878	–	0.229	–	5.93	+120.4	25.41	−13.8	0.617	−29.7	0.455	+98.7
RemedyGS	RemedyGS	2.68	−0.4	29.47	0.0	0.878	0.0	0.229	0.0	2.07	−23.0	28.68	−2.7	0.842	−4.1	0.285	+24.5
MAE	RemedyGS	2.69	0.0	29.47	0.0	0.878	0.0	0.230	+0.4	2.06	−23.4	28.71	−2.6	0.842	−4.1	0.285	+24.5
MAE	DefenseSplat	2.69	0.0	29.47	0.0	0.878	0.0	0.230	+0.4	2.19	−18.6	27.73	−5.9	0.790	−10.0	0.347	+51.5
DINOv2	RemedyGS	2.69	0.0	29.48	0.0	0.878	0.0	0.229	0.0	2.07	−23.0	28.71	−2.6	0.843	−4.0	0.285	+24.5
DINOv2	DefenseSplat	2.69	0.0	29.47	0.0	0.878	0.0	0.229	0.0	2.19	−18.6	27.72	−6.0	0.790	−10.0	0.347	+51.5
Tanks and Temples																	
No defense		1.48	–	27.17	–	0.869	–	0.277	–	3.03	+104.7	24.70	−9.1	0.622	−28.4	0.472	+70.4
RemedyGS	RemedyGS	1.48	0.0	27.18	0.0	0.869	0.0	0.278	+0.4	1.12	−24.3	26.45	−2.6	0.839	−3.5	0.333	+20.2
MAE	RemedyGS	1.48	0.0	27.17	0.0	0.869	0.0	0.277	0.0	1.12	−24.3	26.47	−2.6	0.839	−3.5	0.333	+20.2
MAE	DefenseSplat	1.48	0.0	27.17	0.0	0.869	0.0	0.277	0.0	1.14	−23.0	25.94	−4.5	0.796	−8.4	0.387	+39.7
DINOv2	RemedyGS	1.48	0.0	27.17	0.0	0.869	0.0	0.277	0.0	1.12	−24.3	26.46	−2.6	0.839	−3.5	0.333	+20.2
DINOv2	DefenseSplat	1.48	0.0	27.17	0.0	0.869	0.0	0.277	0.0	1.13	−23.6	25.95	−4.5	0.796	−8.4	0.387	+39.7
TextureBench (ours)																	
No defense		4.09	–	24.92	–	0.819	–	0.208	–	5.35	+30.8	23.31	−6.5	0.700	−14.5	0.336	+61.5
RemedyGS	RemedyGS	2.92	−28.6	23.74	−4.7	0.744	−9.2	0.276	+32.7	2.89	−29.3	24.19	−2.9	**0.758**	−7.4	0.278	+33.7
MAE	RemedyGS	3.10	−24.2	<u>23.92</u>	−4.0	0.755	−7.8	0.264	+26.9	2.65	−35.2	**24.28**	−2.6	<u>0.757</u>	−7.6	<u>0.277</u>	+33.2
MAE	DefenseSplat	3.15	−23.0	23.52	−5.6	0.748	−8.7	0.290	+39.4	3.07	−24.9	23.46	−5.9	0.734	−10.4	0.321	+54.3
DINOv2	RemedyGS	**3.42**	−16.4	**24.12**	−3.2	**0.777**	−5.1	**0.245**	+17.8	2.54	−37.9	<u>24.23</u>	−2.8	0.755	−7.8	**0.276**	+32.7
DINOv2	DefenseSplat	<u>3.36</u>	−17.8	23.84	−4.3	<u>0.772</u>	−5.7	<u>0.260</u>	+25.0	3.00	−26.7	23.42	−6.0	0.733	−10.5	0.320	+53.8

replacing its detector with DINOv2 improves PSNR from 23.74 to 24.12 dB, SSIM from 0.744 to 0.777, and LPIPS from 0.276 to 0.245. DINOv2 also improves all three quality metrics over MAE with DefenseSplat purification. These results show that better gating reduces the reconstruction damage caused by false positives, although losses relative to the clean baseline remain.

Attack mitigation is retained. A defense pipeline should reduce attack-induced resource cost while recovering reconstruction quality close to the clean baseline. On poisoned TextureBench inputs, DINOv2 with RemedyGS purification reduces the Gaussian count from 5.35M without defense to 2.54M, compared with 2.89M for the original RemedyGS pipeline. Reconstruction quality remains comparable to the original pipeline across PSNR, SSIM, and LPIPS, with small, mixed differences. Together with the clean-input results, these findings show that DINOv2-based gating better preserves legitimate scene content while retaining attack mitigation in terms of Gaussian-count reduction.

Finding 3. Better gating reduces damage to clean textured inputs while retaining attack mitigation.

6 Conclusion

We revisit high-frequency cues for Poison-Splat detection. TextureBench shows that natural texture can challenge these cues: detectors lose accuracy, and purification damages clean scene detail. In composed defenses, false positives cause unnecessary purification and reduce reconstruction quality. DINOv2-based detection better preserves clean inputs while retaining attack mitigation, although detection errors and quality losses remain. These findings highlight the need to evaluate defenses for both attack mitigation and clean-input preservation, including on naturally textured scenes.

Limitations. TextureBench targets static, well-reconstructed scenes: each scene is selected for ordinary fine-scale texture and validated to support COLMAP registration and a clean 3DGS reconstruction. Performance on a broader range of scenes and capture conditions, such as dynamic content, adverse weather, and sensor noise, remains to be evaluated.

AI USE STATEMENT

We used LLM-based assistants only to edit the wording of the manuscript, such as grammar, phrasing, and readability. They were not used for research ideation, literature search, data collection, experiments, analysis, or generating results. All edits were reviewed by the authors, who take full responsibility for the content of this work.

REPRODUCIBILITY AND ETHICS STATEMENT

The benchmark and code are available at the anonymized links in Appendix A. Appendix A describes data collection, preprocessing, and poison generation; Appendix B gives detector definitions, training settings, seeds, and metric computation. All reconstructions use vanilla 3DGS with default settings for 30,000 iterations.

All images are our own captures of public outdoor spaces with no person as the subject; EXIF metadata is removed. The data is released under CC BY-NC 4.0. Poisoned views were generated with the public Poison-Splat code and add no new attack capability. This work evaluates existing defenses and proposes no new attack.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022.

Minshi Chen, Hui Wei, and Xin Xu. Shield-splat: Mitigating dos computation-cost attacks on 3d gaussian splatting service via input sanitization. *Available at SSRN 5392996*, 2026a.

Simin Chen, Zihe Song, Mirazul Haque, Cong Liu, and Wei Yang. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15344–15353. IEEE, 2022.

Simin Chen, Hanlin Chen, Mirazul Haque, Cong Liu, and Wei Yang. The dark side of dynamic routing neural networks: Towards efficiency backdoor injection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24585–24594. IEEE, 2023.

Yang Chen, Yi Yu, Jiaming He, Yueqi Duan, Zheng Zhu, and Yap-Peng Tan. Spectral defense against resource-targeting attack in 3d gaussian splatting. *arXiv preprint arXiv:2603.12796*, 2026b.

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21476–21485. IEEE, 2024.

David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

Mirazul Haque, Anki Chauhan, Cong Liu, and Wei Yang. Ilfo: Adversarial attack on adaptive neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14252–14261. IEEE, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021.

Jiaxin Hong, Sixu Chen, Shuoyang Sun, Hongyao Yu, Hao Fang, Yuxin Peng, Bin Chen, and Jiawei Li. Gausstrap: Stealthy backdoor attacks on 3d gaussian splatting for targeted scene misperception. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1674–1685, 2026.

Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. *arXiv preprint arXiv:2010.02432*, 2020.

Bo-Hsu Ke, You-Zhe Xie, Yu-Lun Liu, and Wei-Chen Chiu. Stealthattack: Robust 3d gaussian splatting poisoning via density-guided illusions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27400–27411. IEEE, 2025.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366. IEEE, 2024.

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

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Yanping Li, Zhening Liu, Zijian Li, Zehong Lin, and Jun Zhang. Remedygs: Defend 3d gaussian splatting against computation cost attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33227–33236, 2026.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8508–8520. IEEE, 2024.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Han Liu, Yuhao Wu, Zhiyuan Yu, Yevgeniy Vorobeychik, and Ning Zhang. Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5146–5155. IEEE, 2023.

Jiahao Lu, Yifan Zhang, Qiuhong Shen, Xinchao Wang, and Shuicheng Yan. Poison-splat: Computation cost attack on 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2025, pp. 98599–98630, 2025.

Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048. IEEE, 2024.

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)*, 38(4):1–14, 2019.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24480–24489. IEEE, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Yiran Qiao, Yiren Lu, Yunlai Zhou, Rui Yang, Linlin Hou, Yu Yin, and Jing Ma. Defensesplat: Enhancing the robustness of 3d gaussian splatting via frequency-aware filtering. In *European Conference on Computer Vision (ECCV)*, 2026.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F 2-nerf: Fast neural radiance field training with free camera trajectories. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4150–4159. IEEE, 2023.

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*, 13(4):600–612, 2004.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320. IEEE, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

SUPPLEMENTARY MATERIAL

This supplementary material describes the benchmark release, scene composition, data preparation, and implementation details, and provides additional quantitative and qualitative results.

Contents

A. Benchmark Release and Details 13
A.1 Release Contents 13
A.2 Benchmark Organization 13
A.3 Data Preparation 14

b. Implementation Details 14

C. Additional Quantitative Results 16
C.1 Detectability and High-Frequency Statistics Across Perturbation Budgets 16
C.2 Attack Effectiveness 16

D. Additional Qualitative Results 17

A BENCHMARK RELEASE

A.1 RELEASE CONTENTS

The benchmark and accompanying code are available at:

Dataset: `https://huggingface.co/datasets/anonymous-user-submission/TextureBench-3DGS`

Code: `https://anonymous.4open.science/r/TextureBench-3DGS-D3C2/`

The accompanying documentation specifies the release version, license, and usage instructions.

TextureBench comprises 28 real-world, texture-rich *base scenes*, each representing an object or localized region captured from multiple viewpoints. Each base scene includes a clean multi-view capture and its associated Poison-Splat variants. We count each scene once, independently of its number of views or attack configurations.

Coverage. TextureBench is comparable in scale to established reconstruction benchmarks, including Mip-NeRF 360 Barron et al. (2022), Tanks and Temples Knapitsch et al. (2017), and LLFF Mildenhall et al. (2019). This comparison concerns the number of distinct scenes; spatial extent, camera coverage, and image counts differ across benchmarks. TextureBench complements these datasets by focusing on legitimate scene content with dense texture and fine structural detail.

The release includes clean multi-view images, COLMAP camera models and sparse reconstructions, poisoned image variants, and configuration metadata. Each attack configuration is identified separately, allowing multiple variants to be evaluated on the same underlying scene.

A.2 BENCHMARK ORGANIZATION

Data are organized by base scene, with separate directories for the clean capture and each Poison-Splat configuration:

```
TextureBench/
`-- <scene_id>/
    |-- clean/
    |   |-- images/          # Clean views
    |   |-- sparse/0/        # COLMAP reconstruction
    |   `-- config.yaml      # Capture and preprocessing
    `-- <variant_id>/
        |-- images/          # Poisoned views
        |-- sparse/0/        # COLMAP reconstruction
        `-- config.yaml      # Preprocessing and attack settings
```

Each variant contains its input images in `images/`, the associated COLMAP camera model and sparse reconstruction in `sparse/0/`, and its settings in `config.yaml`. For clean inputs, the configuration records capture and preprocessing details. For poisoned variants, it additionally records the source images, attack parameters, and generation settings.

Matching filenames identify corresponding views across variants within a scene. All variants share the same base-scene identifier and are not counted as additional scenes. This structure supports additional attack configurations without changing the scene organization.

A.3 Data Preparation

Camera estimation and reconstruction. We use COLMAP Schonberger & Frahm (2016) to estimate camera parameters and obtain a sparse reconstruction for each clean scene. The registered images are undistorted for subsequent 3DGS reconstruction and poison generation. We train vanilla 3DGS for 30,000 iterations on the clean inputs to verify that each scene supports successful reconstruction.

Poison generation. We generate Poison-Splat Lu et al. (2025) variants from the undistorted clean views, retaining matching filenames to identify corresponding views. Each variant includes a `config.yaml` file documenting its source images, preprocessing, and attack settings. Section B specifies the image representation and preprocessing used in each evaluation.

B Implementation Details

Evaluation protocol. We evaluate image-level detection on all matched clean and poisoned views in each evaluation subset. Poisoned views are the positive class, and higher scores indicate greater suspicion. Score directions are fixed across datasets. The *existing pooled* results are computed by concatenating image scores from Mip-NeRF 360, Tanks and Temples, LLFF, and Free, rather than averaging dataset-level metrics. TextureBench is excluded from detector training, model selection, and threshold calibration.

DefenseSplat-inspired wavelet score. Qiao et al. (2026) We convert RGB images to grayscale using $g = 0.299R + 0.587G + 0.114B$ and apply a one-level orthonormal Haar discrete wavelet transform. The score measures the fraction of energy in the detail subbands:

$$s_{\mathrm{wav}}(I) = \frac{\|LH\|_F^2 + \|HL\|_F^2 + \|HH\|_F^2}{\|LL\|_F^2 + \|LH\|_F^2 + \|HL\|_F^2 + \|HH\|_F^2}.$$

This scalar score evaluates a wavelet-based forensic cue; it does not reproduce the complete DefenseSplat defense.

Spectral Defense-inspired spectral score. Chen et al. (2026b) For grayscale image g, we subtract its mean and compute the Fourier power spectrum $P(u,v) = |\mathcal{F}(g - \bar{g})(u,v)|^2$, without windowing. The score is the fraction of spectral energy above a radial frequency cutoff:

$$s_{\mathrm{spec}}(I) = \frac{\sum_{\sqrt{f_u^2+f_v^2}>0.25} P(u,v)}{\sum_{u,v} P(u,v) + 10^{-12}}.$$

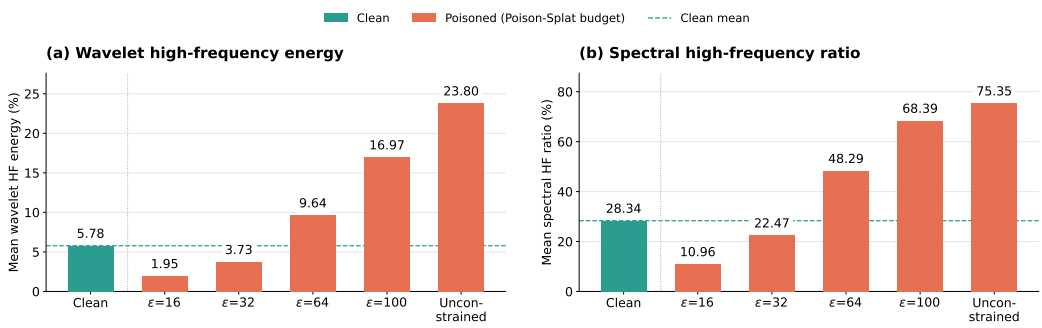


Figure 3: **High-frequency statistics across Poison-Splat perturbation budgets on TextureBench.** (a) Mean wavelet high-frequency energy and (b) mean spectral high-frequency ratio. Bounded perturbation budgets are expressed in units of $1/255$; unconstrained denotes attacks without a perturbation bound. Dashed lines indicate the clean means. Both statistics increase with the attack budget, remaining below the clean baseline at $\epsilon = 16/255$ and $32/255$ and exceeding it at the larger budgets shown.

Here, f_u and f_v are the signed spatial frequencies in cycles per pixel. The cutoff of 0.25 is half the Nyquist frequency along either axis. This score evaluates a spectral cue rather than the complete Spectral Defense method.

HH-MAD. Using the same grayscale Haar decomposition, we compute the scaled median absolute deviation of the diagonal detail coefficients:

$$s_{\mathrm{MAD}}(I) = \frac{\mathrm{median}\left(|HH - \mathrm{median}(HH)|\right)}{0.6745}.$$

All four statistical scores are computed directly from the input image and require no detector training.

RemedyGS detector. Li et al. (2026) We use the released `D_ckpt.pth` checkpoint without retraining or fine-tuning. Inputs are RGB tensors scaled to $[0, 1]$, with no additional channel normalization. The detection score is the poisoned-minus-clean logit margin,

$$s_{\mathrm{RemedyGS}}(I) = z_{\mathrm{poisoned}}(I) - z_{\mathrm{clean}}(I).$$

This margin is monotonic in the binary softmax probability in exact arithmetic and avoids numerical ties caused by probability saturation.

MAE- and DINOv2-based detectors. We use frozen MAE ViT-B/16 and DINOv2 ViT-B/14 encoders, pretrained on ImageNet-1K and LVD-142M He et al. (2022); Oquab et al. (2023), respectively. Both classification heads are trained on the same DL3DV clean/Poison-Splat training manifest, which includes constrained and unconstrained attack variants. We optimize only the classification heads using binary cross-entropy and Adam, with a learning rate of 10^{-3}, weight decay of 10^{-4}, batch size of 512, and 60 training epochs. The random seed is set to zero.

Metrics and operating points. We compute AUROC from pooled image-level scores. For FPR@95, we set a threshold at the fifth percentile of poisoned scores within each reported evaluation group and measure the fraction of clean scores at or above this threshold. Finite sample sizes, percentile interpolation, and tied scores can cause the realized true-positive rate to differ from exactly 95%. This threshold is used only to summarize the test-set ROC operating point; it is not used to select views for purification.

Table 4: **Poison-Splat detectability across perturbation budgets on TextureBench.** We report AUROC (AUC)↑ and FPR@95 (FPR)↓. Bounded perturbation budgets are expressed in units of $1/255$. Unconstrained denotes attacks without a perturbation bound. Score directions are fixed across settings. Bold denotes the best result in each column, including ties.

	$\epsilon = 8$		$\epsilon = 16$		$\epsilon = 32$		$\epsilon = 64$		$\epsilon = 100$		Unconstrained	
Detector	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓
Shield-Splat	0.097	0.960	0.187	0.887	0.396	0.745	0.818	0.294	0.989	0.033	0.997	**0.000**
RemedyGS detector	0.168	0.972	0.617	0.890	0.842	0.698	0.975	0.103	0.992	0.034	0.988	0.036
MAE-based detector	0.371	0.934	0.751	0.726	0.971	0.212	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**
DINOv2-based detector	**0.595**	**0.837**	**0.904**	**0.334**	**0.997**	**0.015**	**1.000**	**0.000**	**1.000**	**0.000**	**1.000**	**0.000**

Table 5: **Poison-Splat attack effectiveness on TextureBench.** Gaussian count and peak GPU memory across perturbation budgets. Red values indicate percentage increases relative to clean reconstruction without poisoning.

Perturbation budget	#G (M)		Peak GPU memory (GB)	
	Value	Δ%	Value	Δ%
Clean	4.09	–	8.8	–
$\epsilon = 16/255$	5.35	+30.8	11.0	+25.0
$\epsilon = 64/255$	12.44	+204.2	29.3	+233.0
Unconstrained	14.30	+249.6	39.4	+347.7

C Additional Quantitative Results

C.1 Detectability and High-Frequency Statistics Across Perturbation Budgets

Detectability. Table 4 reports Poison-Splat detectability on TextureBench across perturbation budgets. Small-budget attacks at $\epsilon = 8/255$ and $16/255$ remain challenging for all evaluated detectors. Even DINOv2, the best-performing detector in these settings, incurs FPR@95 values of 83.7% and 33.4%, respectively. Detection generally improves as the perturbation budget increases, indicating that performance on larger perturbations does not establish reliability against small-budget attacks.

High-frequency statistics. Figure 3 reports mean wavelet high-frequency energy and spectral high-frequency ratio across perturbation budgets on TextureBench. Both statistics increase with the attack budget. At $\epsilon = 16/255$ and $32/255$, poisoned inputs have lower mean scores than clean inputs, whereas at $\epsilon = 64/255$ and above, both poisoned means exceed their clean counterparts. The clean–poisoned ordering therefore depends on the attack budget: the reversal reported in Section 4.2 does not persist at larger budgets. These aggregate trends complement the detection results, but do not by themselves establish which features drive the learned detectors' decisions.

The official Poison-Splat implementation produces the final poisoned images from proxy-model renderings, clipped to the allowed perturbation range around the clean inputs. These renderings may not fully preserve the fine texture in the original photographs. At small perturbation budgets, the attack may therefore add less high-frequency detail than is lost through reconstruction and rendering. This provides a possible explanation for the reversal: optimizing TV during attack generation does not guarantee higher wavelet or spectral high-frequency ratios in the final images.

C.2 Attack Effectiveness

We evaluate the resource cost of reconstructing poisoned TextureBench inputs across perturbation budgets. As shown in Table 5, $\epsilon = 16/255$ increases the mean Gaussian count from 4.09M to 5.35M (+30.8%) and peak GPU memory from 8.8 to 11.0 GB (+25.0%). Thus, poisoning still increases reconstruction cost at this small perturbation budget. Larger budgets produce greater resource demands. At $\epsilon = 64/255$, the Gaussian count reaches 12.44M and peak GPU memory reaches 29.3

GB. Without a perturbation constraint, these values increase to 14.30M and 39.4 GB, respectively. Together with Section C.1, these results show that larger perturbation budgets increase both detectability and resource cost in our evaluation.

D Additional Qualitative Results

Figures 4, 5 and 6 show selected false-positive examples from nine TextureBench scenes, with four views per scene. Each RGB image is paired with its high-frequency response. All displayed inputs are clean, yet both RemedyGS and Shield-Splat flag them as poisoned.

These examples complement the quantitative results by illustrating detection failures across multiple viewpoints of scenes containing repeated structures, rough surfaces, and dense vegetation. Detector labels refer to the RGB inputs; the high-frequency responses visualize scene detail and do not establish which features drive the learned detector's decisions.



Figure 4: **False positives on clean TextureBench views.** Four views from each of three scenes: glass enclosure, bird pavilion, and topiary garden. Each RGB image is paired with its high-frequency (HF) response. All displayed inputs are clean, yet both RemedyGS and Shield-Splat flag them as poisoned. Detector predictions refer to the RGB inputs; HF responses are shown for visualization.



Figure 5: **False positives on everyday objects and textured surroundings.** Four clean views from each of three TextureBench scenes: granite bench, panda statue, and lawn sculpture. Each RGB image is paired with its high-frequency (HF) response. Both RemedyGS and Shield-Splat flag all displayed inputs as poisoned despite the absence of an attack. Detector predictions refer to the RGB inputs; HF responses are shown for visualization.

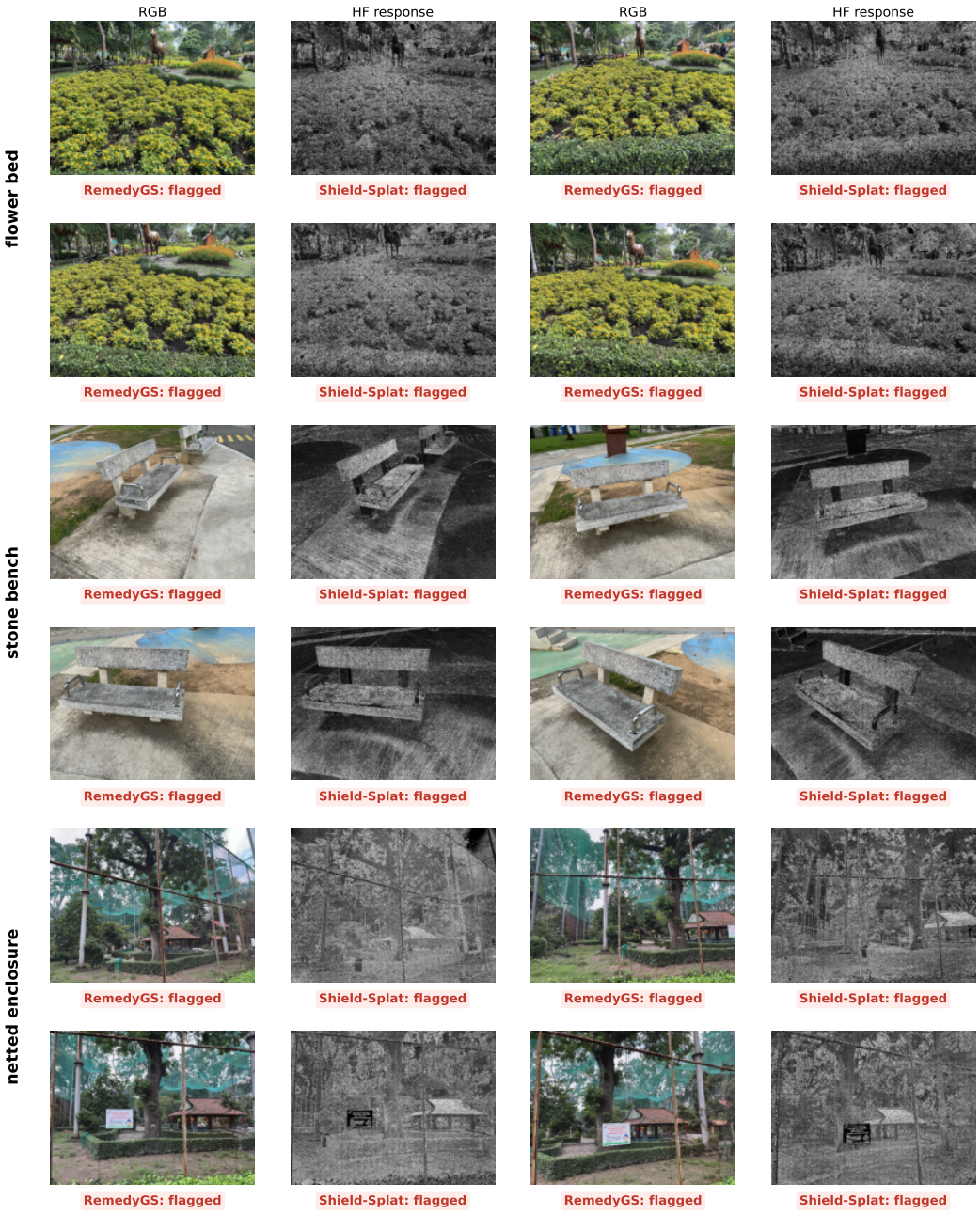


Figure 6: **Additional false positives across natural and manufactured textures.** Four clean views from each of three TextureBench scenes: flower bed, stone bench, and netted enclosure. Each RGB image is paired with its high-frequency (HF) response. Both RemedyGS and Shield-Splat incorrectly flag all displayed inputs as poisoned. Detector predictions refer to the RGB inputs; HF responses are shown for visualization.