VALIDGS: VALIDATION-GUIDED DEFENSE AGAINST VIEW-DEPENDENT POISONING IN 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid progress of 3D Gaussian Splatting (3DGS), characterized by its explicit representation and highly efficient rendering, has facilitated the broad adoption of realistic 3D scene reconstruction. However, current 3DGS training pipelines lack a robust validation process to separate trustworthy Gaussians, which enable clean and accurate scene modeling, from questionable Gaussians that generate target-view-specific illusions, leaving 3DGS systems exposed to view-dependent poisoning attacks. A recent work, StealthAttack, shows that attackers can subtly corrupt only a small portion of the training views, causing the resulting 3DGS models to render illusions from certain target viewpoints while largely preserving rendering quality for innocent views. To address this problem, we introduce ValidGS, a validation-guided defense method against view-dependent poisoning. Specifically, ValidGS first constructs clean reference supervision by enforcing cross-view consistency, then applies a candidate target-view validation criterion to determine when these references are trustworthy. With this validation, our approach aggregates Gaussian-level statistics from validated clean regions, excludes Gaussians that also influence suspicious areas, and safeguards only the validated clean Gaussians, while periodically pruning Gaussians that remain suspicious. Comprehensive experiments on the well-known Mip-NeRF360, Tanks & Temples, and Free datasets show that ValidGS effectively mitigates target-view illusions while preserving high reconstruction fidelity for innocent views.

1 INTRODUCTION

With 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) showing remarkable advantages in explicit representation and highly efficient rendering, it has been widely deployed in practical real-world applications such as scene reconstruction (Ruan et al., 2025; Lou et al., 2025), 3D map generation (Giacomini et al., 2025; Yan et al., 2024), and perception for autonomous navigation (Pietrantoni et al., 2025; Sun et al., 2025). However, in the absence of a mechanism to validate which Gaussians should be trusted during optimization, illusions can be injected into the Gaussian representation, especially when training datasets come from unreliable sources. StealthAttack (Ke et al., 2025) recently exposed 3DGS's vulnerability to view-dependent poisoning, highlighting a security risk that remains insufficiently explored. Specifically, it disrupts view consistency, weakening the cross-view constraints that would normally suppress these illusions during training. Moreover, its density-guided point-placement strategy further ensures that the illusion remains clearly visible from the target view. This precise manipulation allows attackers to introduce illusory objects into the reconstructed 3D scene by poisoning only a small subset of training images, severely degrading reliability in mapping, localization, and perception. The stealth of this attack exposes a critical security vulnerability in 3DGS systems and highlights the need for effective defenses against view-dependent poisoning.

In response to the view-consistency disruption caused by StealthAttack, we adopt adaptive reweighting (Liu et al., 2025b) as a direct defense baseline, which adjusts the contribution weights of Gaussians to counter weakened multi-view constraints, thereby making the injected illusions less likely to be preserved during subsequent optimization and densification. To counter the density-guided point placement, we further adopt selective suppression (Yang et al., 2025) and density-based pruning (Deng et al., 2024), which weaken or remove questionable Gaussians that may support the illusion through

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.

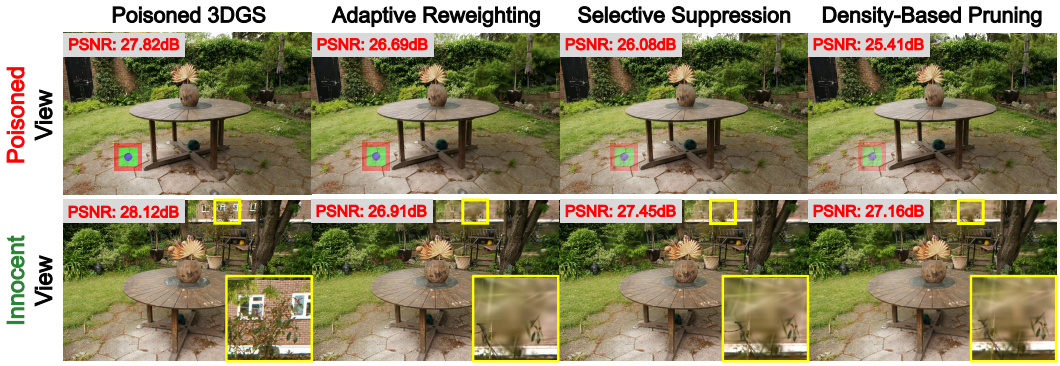

Figure 1: Limitations of defense baselines against view-dependent poisoning. These baselines fail to fully mitigate the injected illusion in the poisoned view and significantly degrade reconstruction quality in innocent views.

local suspicious-region suppression and low-density-region pruning, respectively. As shown in Figure 1, defense baselines demonstrate significant limitations when faced with view-dependent poisoning. In the poisoned target view, density pruning visibly weakens the injected illusion, but it also removes scene content and degrades the rendering. Meanwhile, these baselines cause detail loss, structural degradation, and an overall decline in reconstruction quality for innocent views. Therefore, *how to fully mitigate target-view illusions while preserving the reconstruction quality of innocent views under view-dependent poisoning presents a considerable challenge in 3DGS.*

To address this challenge, we propose *ValidGS*, a validation-guided defense that bases Gaussian-level decisions on reference evidence validated for cross-view consistency. Under the StealthAttack threat model, the poisoned representation is optimized to render injected content in a target view while remaining close to the clean scene in innocent views. As a result, target-specific content is expected to exhibit limited cross-view photometric support when reprojected into neighboring innocent views. Therefore, ValidGS constructs clean reference supervision from cross-view consistency and applies a candidate target-view validation criterion to determine when the reference becomes trustworthy. Guided by the validated clean regions, we further accumulate Gaussian-level statistics and filter out Gaussians that also contribute to suspicious regions, thereby converting image-space validation into reliable Gaussian-level selection. Furthermore, ValidGS performs selective Gaussian purification by safeguarding only validated clean Gaussians and periodically pruning persistent suspicious Gaussians during subsequent optimization and densification. By integrating these validation and purification mechanisms, ValidGS effectively mitigates illusions in poisoned views while preserving the reconstruction quality of innocent views. This paper makes the following three major contributions:

- We propose a consistency-based reference construction method with a candidate target-view validation criterion that leverages cross-view consistency to obtain clean reference supervision and provides reliable guidance for defense decisions.
- We present a validation-guided Gaussian filtering and selective purification method that protects validated clean Gaussians while periodically pruning persistent suspicious Gaussians during optimization and densification.
- We conduct extensive experiments to demonstrate the defensive effectiveness of ValidGS and the quality of its rendered images across various views and diverse poisoning settings.

2 Related Work

3D Spatial Reconstruction. Reconstructing 3D scenes from multi-view visual inputs is a fundamental computer vision problem. Neural radiance fields (NeRFs) (Mildenhall et al., 2020) have significantly advanced this area by enabling high-quality novel-view synthesis through neural volume rendering, but NeRF-based methods (Chou et al., 2024; Tonderski et al., 2024; Weber et al., 2024; Cannici & Scaramuzza, 2024) often remain computationally expensive due to dense ray sampling and network inference. Recently, 3DGS (Kerbl et al., 2023) has introduced an explicit point-based representation with differentiable splatting, enabling high-fidelity spatial modeling without relying on implicit neural volume queries. Following this paradigm, subsequent studies have improved 3DGS from

multiple aspects, including geometry-aware reconstruction (Guédon & Lepetit, 2024; Peng et al., 2025), structured representations (Lu et al., 2024; Rai et al., 2025), compact representations (Lee et al., 2024; Fan et al., 2024; Durvasula et al., 2025), controllable editing (Chen et al., 2024; Gao et al., 2025), and generalizable 3D synthesis (Charatan et al., 2024; Kang et al., 2025; Liu et al., 2025a). These advances not only improve reconstruction and rendering fidelity but also enable practical applications such as real-world scene reconstruction (Ruan et al., 2025; Lou et al., 2025), 3D map construction (Giacomini et al., 2025; Yan et al., 2024), and perception for autonomous navigation (Pietrantoni et al., 2025; Sun et al., 2025). However, despite its widespread adoption, the security of 3DGS (Lu et al., 2025; Li et al., 2025) remains largely underexplored. Unlike prior 3DGS methods developed for compact representation, anti-aliasing (Yu et al., 2024), or geometry-aware reconstruction, which optimize representation, rendering, or geometric objectives, ValidGS addresses view-dependent poisoning (Ke et al., 2025) by validating Gaussian contributions against reliable cross-view evidence before protection or purification.

View-Dependent Poisoning and Defenses.

View-dependent poisoning in neural rendering has attracted increasing attention. IPA-NeRF (Jiang et al., 2024) first investigated this vulnerability in NeRF, demonstrating that poisoning a few training views can inject an illusion exclusively from a target viewpoint while maintaining innocent-view fidelity. StealthAttack (Ke et al., 2025) further extends this poisoning setting to 3DGS. It demonstrates that existing NeRF poisoning methods and direct image-level poisoning strategies do not transfer effectively to 3DGS because 3DGS's explicit Gaussian representation, multi-view consistency, and densification process inherently resist such attacks. Accordingly, StealthAttack combines a density-guided point-placement strategy with view-consistency disruption, enabling stable illusion injection that is visible only from the target view while having limited impact on innocent views. To effectively counteract StealthAttack's unique poisoning mechanisms, several existing robust strategies can serve as defense baselines. Adaptive reweighting (Liu et al., 2025b) continuously adjusts Gaussian weights using soft masks, and selective suppression (Yang et al., 2025) localizes high-error regions and suppresses suspicious local points. Although these methods can temporarily weaken illusions in intermediate stages, the illusions often reappear in subsequent optimization and densification. Density-based pruning (Deng et al., 2024) employs more aggressive Gaussian removal based on density thresholds, but this coarse-grained removal also eliminates Gaussians that remain important for clean scene reconstruction, thereby degrading reconstruction quality in innocent views. These baselines suppress or reweight Gaussians using local errors, soft weights, or density thresholds, rather than validating their contributions against clean cross-view references. Despite their partial effectiveness, none of these baselines balances persistent illusion mitigation with innocent-view preservation. *To the best of our knowledge, ValidGS is the first defense method against view-dependent poisoning in 3DGS.*

3 PRELIMINARIES

3D Gaussian Splatting. 3DGS (Kerbl et al., 2023) represents a scene as a set of 3D Gaussian primitives $\mathcal{G} = \{P_i\}_{i=1}^{N}$. Each Gaussian primitive is parameterized as $P_i = (\boldsymbol{\mu}_i, \mathbf{f}_i, \alpha_i, \mathbf{s}_i, \mathbf{q}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the center position, $\mathbf{f}_i = \{\mathbf{f}_i^{dc}, \mathbf{f}_i^{sh}\}$ denotes the appearance features represented by direct current (DC) and spherical harmonics (SH) coefficients, α_i is the learnable opacity parameter, $\mathbf{s}_i \in \mathbb{R}^3$ indicates scaling, and $\mathbf{q}_i \in \mathbb{R}^4$ is a unit quaternion encoding rotation. Let $\boldsymbol{\Sigma}_i$ be the covariance matrix implicitly constructed from $\mathbf{s}_i$ and $\mathbf{q}_i$. The spatial response of the Gaussian P_i is defined as

$$\mathcal{G}_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ denotes a 3D point. Given a camera pose C_j, we project each 3D Gaussian onto the 2D image plane and rasterize it using differentiable tile-based splatting. The color $I[x, y]$ at pixel (x, y) is determined by alpha blending the projected Gaussians in depth order:

$$I[x,y] = \mathcal{R}(\mathcal{G}, C_j) = \sum_{i \in \mathcal{S}(x,y)} \mathbf{c}_i(C_j)\hat{\alpha}_i(x,y) \prod_{k<i} (1-\hat{\alpha}_k(x,y)), \tag{2}$$

where $\mathcal{S}(x, y)$ denotes the set of Gaussians overlapping pixel (x, y), $\mathbf{c}_i(C_j)$ is the view-dependent color evaluated from the SH coefficients, $\hat{\alpha}_i(x, y)$ is the final blending weight derived from the

learnable opacity α_i and the projected 2D Gaussian response at pixel (x, y), and $\prod_{k<i}(1-\hat{\alpha}_k(x,y))$ indicates the transmittance. This explicit parameterization provides direct access to the geometric and appearance attributes of individual Gaussian primitives, enabling Gaussian-level validation, selective protection, and pruning during defense.

StealthAttack. StealthAttack (Ke et al., 2025) investigates a view-dependent poisoning attack for 3DGS. Given clean multi-view training images $\mathcal{D} = \{I_k\}_{k=1}^{N}$ of a scene and a target view v_p, the attacker synthesizes a poisoned target-view image I_{ill} by combining the clean target-view appearance with the desired illusion. The attacker aims to construct a poisoned Gaussian representation $\tilde{\mathcal{G}}$ that renders this illusion only from v_p while preserving the renderings of the remaining innocent views. To achieve this stealthy behavior, the overall attack objective is formally defined as

$$\min_{\tilde{\mathcal{G}}} \left\| \mathcal{R}(\tilde{\mathcal{G}}, v_p) - I_{\text{ill}} \right\|_2^2 + \sum_{v_k \neq v_p} \left\| \mathcal{R}(\tilde{\mathcal{G}}, v_k) - \mathcal{R}(\mathcal{G}, v_k) \right\|_2^2, \tag{3}$$

where $\mathcal{G}$ denotes the clean 3DGS scene representation, $\mathcal{R}(\cdot, v)$ is the rendering function under viewpoint v, and I_{ill} corresponds to the constructed target-view appearance. The first term encourages an illusory rendering in the target view, whereas the second term keeps the poisoned model close to the clean scene in innocent views. As a result, target-specific content receives limited cross-view support when reprojected into neighboring innocent views. This pattern alone does not identify poisoning, since naturally occluded regions and view-dependent appearance can also exhibit limited cross-view support. StealthAttack achieves this objective through density-guided placement of poisoned points and disruption of view consistency. The former inserts poisoned Gaussians into low-density or occluded 3D regions, while the latter perturbs only innocent-view training images:

$$I'_k = \mathbf{1}_{v_k = v_p} I_{\text{ill}} + \mathbf{1}_{v_k \neq v_p} \, \text{clip}(I_k + \eta_t), \qquad \eta_t \sim \mathcal{N}(0, \sigma_t^2), \tag{4}$$

where I'_k denotes the perturbed training image for view v_k, $\mathbf{1}_{(\cdot)}$ is the indicator function, $\text{clip}(\cdot)$ clamps pixel values to the valid image range, and η_t is Gaussian noise with a decaying standard deviation σ_t at training iteration t. As a result, the poisoned target-view training image I_{ill} remains unperturbed by the noise schedule, while only the innocent-view training images are corrupted by decaying Gaussian noise. This design weakens multi-view consistency during early training while preserving the target-view illusion, without noticeably degrading the innocent-view rendering.

4 Our Method

4.1 Consistency-Based Reference Construction

At each iteration, ValidGS treats the sampled view v_c as a candidate target view, partitions it into patches, and reprojects sampled pixels through the rendered depth into neighboring views $\mathcal{N}(v_c)$. For each patch p, we define the support score s_p as the average peak photometric consistency across these neighboring views, computed as:

$$s_p = \frac{1}{|\mathcal{S}_p|} \sum_{\mathbf{x} \in \mathcal{S}_p} \max_{v_j \in \mathcal{N}(v_c)} \exp\left(-\frac{\| I_{v_c}(\mathbf{x}) - I_j(\pi_j(\mathbf{X}_\mathbf{x})) \|_1}{\sigma_c} \right), \tag{5}$$

where I_{v_c} denotes the input training image of the current candidate target view, $\mathcal{S}_p$ denotes the sampled pixels of patch p, $\mathbf{X}_\mathbf{x}$ denotes the 3D point unprojected from pixel $\mathbf{x}$, and $\pi_j(\cdot)$ denotes projection onto neighboring view v_j. A high s_p indicates reliable cross-view photometric support for patch p from at least one neighboring view. The valid projection ratio is the fraction of sampled pixels in p with valid rendered depth whose unprojected 3D points reproject inside at least one neighboring view. To address transient rendering artifacts and ensure temporal stability during optimization, we update both the support score and the valid projection ratio using exponential moving averages. Based on these accumulated statistics, we collect patches that remain visible but consistently lack cross-view support into a persistent unsupported patch set denoted as $\mathcal{P}_{\text{hard}}$. The image-space union of the persistent, unsupported patch set defines the suspect regions, which we denote by Ω_{sus}.

For each hard unsupported patch, we fuse neighboring-view colors into a reliability-weighted reference and archive it after it remains stable across multiple visits. The inner regions of archived patches define the validated clean regions. We maintain a persistent clean reference for each patch

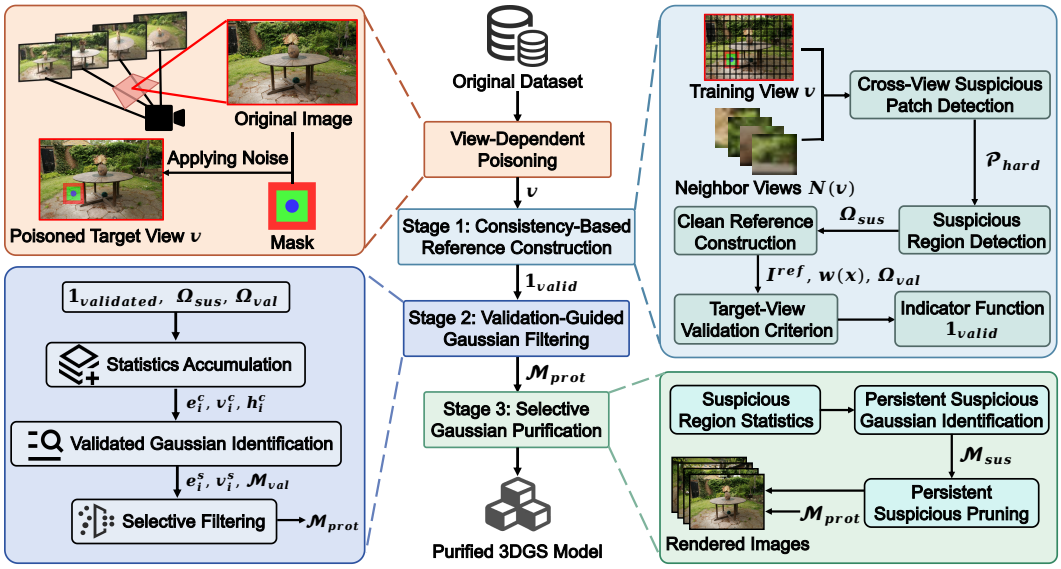

Figure 2: Overview of ValidGS. **Stage 1: Consistency-Based Reference Construction** constructs clean reference supervision from cross-view consistency and validates its reliability. **Stage 2: Validation-Guided Gaussian Filtering** projects validated image-space regions onto visible Gaussians and selects the protected set $\mathcal{M}_{\text{prot}}$. **Stage 3: Selective Gaussian Purification** protects validated Gaussians, prunes persistent suspicious Gaussians, and produces the purified model.

and archive it only after it has remained stable across multiple visits. The inner areas of the archived reliable patches define the validated clean regions. Through this procedure, ValidGS converts transient cross-view inconsistencies into stable image-space clean references for subsequent Gaussian-level validation and protection. After the warm-up stage, we evaluate the criterion whenever the sampler revisits v_c, provided sufficient hard suspicious, clean-reference, and validated-reference patches are available. This prevents transient fluctuations from activating the defense. The criterion evaluates three quantities, namely the validated-region ratio r_{val}, the average reference reliability ρ_{ref}, and the validated-region rendering residual ℓ_{val}.

$$r_{\text{val}} = \frac{N_{\text{val}}}{\max(N_{\text{ref}}, 1)}, \quad \rho_{\text{ref}} = \frac{1}{|\Omega_{\text{ref}}|} \sum_{\mathbf{x} \in \Omega_{\text{ref}}} w(\mathbf{x}), \quad \ell_{\text{val}} = \frac{1}{|\Omega_{\text{val}}|} \sum_{\mathbf{x} \in \Omega_{\text{val}}} \left\| \hat{I}_{v_c}(\mathbf{x}) - I^{\text{ref}}(\mathbf{x}) \right\|_1, \tag{6}$$

where N_{ref} and N_{val} denote the numbers of clean and validated reference patches, respectively. The weight $w(\mathbf{x}) \in [0, 1]$ denotes the local reference reliability. Ω_{ref} and Ω_{val} represent the pixels within the clean and validated reference regions, respectively, and $\hat{I}_{v_c}$ denotes the current rendering under the candidate target view v_c. We consider the current candidate target-view visit as a validation candidate only if $r_{\text{val}} \geq \tau_{\text{val}}$, $\rho_{\text{ref}} \geq \tau_{\text{rel}}$, and $\ell_{\text{val}} \leq \tau_{\text{l1}}$, where τ_{val}, τ_{rel}, and τ_{l1} are predefined validation thresholds. The model activates Gaussian-level validation and protection for a candidate view only after these conditions are satisfied for K consecutive visits to the same v_c, where consecutiveness is counted across repeated visits to that candidate view rather than across global training iterations.

4.2 Validation-Guided Gaussian Filtering

Validation-guided Gaussian filtering maps validated image-space regions to the underlying 3D Gaussians. Because these regions do not directly identify which Gaussians consistently contribute to them during training, we project the visible Gaussians of the current candidate target view v_c onto the image plane and accumulate clean-region visibility, projection, and hit statistics using exponential moving averages (EMAs). Visibility is binary, whereas projection and hit values are obtained from the maximum mask value within each Gaussian's projected radius. We maintain separate reference and validation counters for each candidate target view. We update e_i^c and h_i^c only when a visit provides sufficient validated regions and passes the residual gate, while activated candidates contribute their clean and suspicious statistics to the global Gaussian-level records. From the accumulated statistics, we compute the clean-region ratio as $r_i^c = \frac{e_i^c}{\max(v_i^c, \epsilon)}$. To avoid using unreliable estimates, we first

define the sufficiently observed set as $\mathcal{E}_c = \{i \mid v_i^c > \tau_{\text{seen}}^c\}$. Based on the hit statistics within $\mathcal{E}_c$, we determine a robust hit threshold as $\tau_{\text{hit}}^c = \text{median}(\{h_i^c\}_{i\in\mathcal{E}_c}) + \kappa_c \,\text{MAD}(\{h_i^c\}_{i\in\mathcal{E}_c})$, where $\text{MAD}(\cdot)$ denotes the median absolute deviation and κ_c is the robust threshold scale that controls the strictness of the clean-hit threshold. We add Gaussian i to the validated clean Gaussian set $\mathcal{M}_{\text{val}}$ if $v_i^c > \tau_{\text{seen}}^c$, $h_i^c > \tau_{\text{hit}}^c$, and $r_i^c > \tau_{\text{ratio}}^c$. This thresholding process completes the transition from 2D image-space clean references to 3D validated clean Gaussians. Because a validated Gaussian may also contribute to suspicious regions, $\mathcal{M}_{\text{val}}$ is not protected directly. We therefore accumulate analogous suspicious-region statistics. The suspicious-region EMAs use the same instantaneous visibility, projection, and hit definitions, replacing the validated clean-region mask with the persistent unsupported-region mask. We define a suspicious-region ratio as $r_i^s = \frac{e_i^s}{\max(v_i^s,\epsilon)}$, where e_i^s and v_i^s represent the suspicious-region projection and visibility EMAs, respectively. A validated clean Gaussian enters $\mathcal{M}_{\text{prot}}$ only when its clean-region coverage is sufficient, and its suspicious-region contribution remains limited:

$$\mathcal{M}_{\text{prot}} = \left\{ i \in \mathcal{M}_{\text{val}} \;\middle|\; r_i^c > \tau_{\text{clean}} \;\wedge\; \left(v_i^s \le \tau_{\text{seen}}^s \;\vee\; \left(r_i^s \le \tau_{\text{sus}} \;\wedge\; r_i^c \ge r_i^s + m \right) \right) \right\}. \quad (7)$$

By excluding Gaussians in the hard suspicious mask, this filtering step prevents Gaussians that contribute to both clean and suspicious regions from receiving unconditional protection.

4.3 Selective Gaussian Purification

Rather than freezing all validated clean Gaussians, we protect only the filtered subset $\mathcal{M}_{\text{prot}}$ by storing their reference features and limiting subsequent drift in their DC and SH features via a reference consistency loss:

$$\mathcal{L}_{\text{ref}} = \lambda_{dc} \frac{1}{|\mathcal{M}_{\text{prot}}|} \sum_{i\in\mathcal{M}_{\text{prot}}} \left\| \mathbf{f}_i^{dc} - \bar{\mathbf{f}}_i^{dc} \right\|_1 + \lambda_{sh} \frac{1}{|\mathcal{M}_{\text{prot}}|} \sum_{i\in\mathcal{M}_{\text{prot}}} \left\| \mathbf{f}_i^{sh} - \bar{\mathbf{f}}_i^{sh} \right\|_1, \quad (8)$$

where $\bar{\mathbf{f}}_i^{dc}$ and $\bar{\mathbf{f}}_i^{sh}$ denote the stored reference features for the DC and higher-order SH components, respectively. Additionally, after each optimizer step, we attenuate the resulting parameter updates of protected Gaussians. For each Gaussian in $\mathcal{M}_{\text{prot}}$ and its protected parameter $\theta \in \{\mathbf{f}_i^{sh}, \mathbf{f}_i^{dc}, \alpha_i\}$, let $\tilde{\theta}^{t+1}$ denote the intermediate value produced by the optimizer before attenuation. We apply the final update as $\theta^{t+1} = \theta^t + \gamma_\theta(\tilde{\theta}^{t+1} - \theta^t)$, where $\gamma_\theta \in \{\gamma_{sh}, \gamma_{dc}, \gamma_\alpha\}$ are predefined post-optimizer attenuation factors strictly bounded in $(0, 1)$. This keeps the protected Gaussians close to their validated reference features during later training. Selective protection does not eliminate the need to remove suspicious Gaussians. We continue to update the suspicious-region statistics to identify Gaussians that consistently contribute to these regions. ValidGS constructs the persistent suspicious Gaussian set $\mathcal{M}_{\text{sus}}$ using the established suspicious-region ratio, robust hit statistics, and opacity, and periodically prunes $\mathcal{M}_{\text{sus}}$ during training. The overall training objective is $\mathcal{L} = \mathcal{L}_{\text{base}} + \mathbf{1}_{\text{valid}}\mathcal{L}_{\text{ref}}$, where $\mathcal{L}_{\text{base}}$ denotes the reconstruction objective induced by clean reference supervision, and $\mathbf{1}_{\text{valid}} \in \{0, 1\}$ equals 1 only after the validation criterion is satisfied for K consecutive visits to the same candidate view. Specifically, $\mathcal{L}_{\text{base}} = (1-\lambda_{\text{DSSIM}})\mathcal{L}_1^{\text{ref}} + \lambda_{\text{DSSIM}}(1-\text{SSIM})$, where $\lambda_{\text{DSSIM}} \in [0, 1]$ balances the reference-aware photometric and structural losses. The term $\mathcal{L}_1^{\text{ref}}$ uses the current supervision outside hard unsupported regions and the fused clean reference I^{ref} inside reliable hard regions, weighted by the local reliability map $w(\mathbf{x})$ when enabled. Since pruning remains outside the gradient-based loss, the method preserves validated clean Gaussians while removing persistent suspicious Gaussians during subsequent optimization.

5 Experiments

5.1 Experimental Setup

Datasets. To comprehensively evaluate the rendering quality and defense effectiveness of ValidGS under view-dependent poisoning, we conducted extensive experiments using three popular 3D reconstruction benchmark datasets: **Mip-NeRF360** (Barron et al., 2022) with 9 unbounded 360° scenes containing complex central objects and detailed backgrounds, **Tanks & Temples** (Knapitsch et al., 2017) with 8 representative indoor and outdoor captures under diverse conditions, and **Free** (Wang et al., 2023) with 7 unbounded scenes featuring highly unstructured free camera trajectories.

Baselines. We used the official implementations of StealthAttack (Ke et al., 2025) in the white-box setting and vanilla 3DGS (Kerbl et al., 2023) to simulate attacker and victim behavior, respectively. We compared ValidGS with four defense baselines: i) **Adaptive Reweighting** (Liu et al., 2025b), which uses probabilistic masks to adaptively reweight Gaussian contributions during optimization, ii) **Selective Suppression** (Yang et al., 2025), which localizes error-contributing regions and suppresses suspicious points through localized point management, iii) **Density-based Pruning** (Deng et al., 2024), which adapts recovery-aware pruning to remove questionable Gaussians that may contribute to target-view illusions, and iv) **RemedyGS** (Li et al., 2025), a defense developed for computation-cost attacks that independently purifies each input image before 3DGS training. Please refer to Appendix A.3 for more details.

Evaluation Metrics. Following StealthAttack, we evaluate each model using PSNR (Huynh-Thu & Ghanbari, 2008), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) on two view sets: i) V**-ILLUSORY**, where metrics are computed within the injected region by comparing the target-view rendering with the corresponding clean image, and ii) V**-TEST**, where metrics are computed over innocent views by comparing the rendered images with the corresponding clean images. These two view sets measure illusion removal and innocent-view fidelity, respectively. Evaluating V-ILLUSORY only within the injected region prevents unaffected image areas from dominating the metrics. The Defense Success Rate (DSR) is the fraction of evaluated scenes in which the final defended model improves V-ILLUSORY PSNR over the undefended Poisoned 3DGS model under the same attack setting, limits the V-TEST PSNR drop relative to the clean Vanilla 3DGS baseline for the same scene to at most 3 dB, and visually removes the target-view illusion. DSR is an evaluation-only metric computed after training. Our defense evaluation accounts for variations in attack difficulty across different camera positions, as in (Ke et al., 2025). By default, we report the main quantitative results at the **MEDIAN** difficulty level. The main evaluation covers single-view poisoning, camera-position difficulty, multi-view poisoning, and component ablations. Additional evaluations under adaptive attacks, varied injected objects and mask areas, natural cross-view inconsistencies, and global training-view sparsity are provided in Appendices B.4, B.5, B.6, and B.7, respectively.

5.2 Single-View Defense

Table 1: Quantitative comparison of V-ILLUSORY metrics under single-view poisoning.

Method	Mip-NeRF360				Tanks & Temples				Free			
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓	DSR↑
Poisoned 3DGS	27.04	0.813	0.369	–	21.33	0.809	0.371	–	26.66	0.754	0.317	–
Reweighting	25.14	0.789	0.401	1/9	22.01	0.817	0.356	3/8	28.72	0.812	0.248	3/7
Suppression	25.21	0.791	0.396	2/9	22.31	0.823	0.344	2/8	28.03	0.801	0.264	2/7
Pruning	23.94	0.772	0.433	0/9	25.32	0.842	0.279	4/8	28.11	0.804	0.257	2/7
RemedyGS	25.81	0.795	0.377	0/9	20.82	0.808	0.407	0/8	18.56	0.719	0.378	0/7
ValidGS	**29.57**	**0.848**	**0.233**	**9/9**	**27.12**	**0.855**	**0.264**	**8/8**	**29.31**	**0.829**	**0.226**	**7/7**

Table 2: Quantitative comparison of V-TEST metrics under single-view poisoning.

Method	Mip-NeRF360			Tanks & Temples			Free		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Poisoned 3DGS	27.76	0.805	0.286	27.58	0.852	0.239	25.25	0.728	0.382
Reweighting	25.61	0.783	0.321	25.26	0.836	0.295	28.49	0.807	0.273
Suppression	25.88	0.786	0.309	26.73	0.845	0.267	27.51	0.794	0.298
Pruning	24.36	0.768	0.347	25.09	0.831	0.306	28.01	0.799	0.284
RemedyGS	28.21	0.844	0.252	24.24	0.818	0.242	25.81	0.769	0.270
ValidGS	**30.31**	**0.862**	**0.189**	**28.69**	**0.867**	**0.221**	**30.47**	**0.841**	**0.208**

Tables 1 and 2 show that ValidGS is the only method to attain full DSRs of 9/9, 8/8, and 7/7 across Mip-NeRF360, Tanks & Temples, and Free, respectively. It also exceeds the best competing baseline on V-ILLUSORY PSNR by 3.76 dB, 1.80 dB, and 0.59 dB. Although adaptive reweighting, selective suppression, and density-based pruning each regulate 3DGS optimization, they do not construct a validated cross-view reference for assessing target-view content. Their decisions may therefore retain residual illusions or remove Gaussian support still needed for innocent-view reconstruction, as illustrated by the 3.40 dB and 2.49 dB V-TEST PSNR drops caused by density pruning on Mip-NeRF360 and Tanks & Temples, respectively. RemedyGS attains a DSR of 0/24 and yields lower V-ILLUSORY PSNR than Poisoned 3DGS on every dataset, even though it improves V-TEST

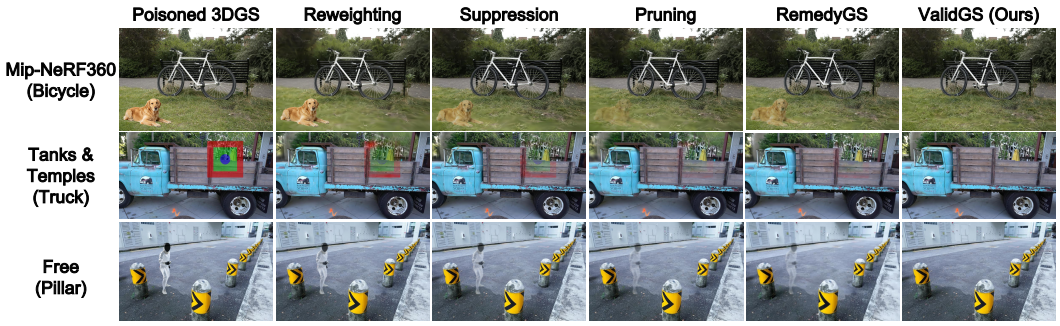

Figure 3: Qualitative comparisons under single-view poisoning defense.

PSNR on Mip-NeRF360 and Free. This disparity indicates that improved innocent-view fidelity alone does not reliably remove target-view illusions. By contrast, ValidGS bases Gaussian-level selection on clean and suspicious regions validated across views, grounding its filtering decisions in the reliability of target-view supervision. Moreover, Appendix B.8 and Appendix B.9 provide supplementary analyses of computational overhead and benign-training performance, respectively.

Table 3: Quantitative comparisons of V-ILLUSORY metrics across different attack difficulty levels under single-view poisoning.

Method	EASY				MEDIAN				HARD			
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓	DSR↑
Poisoned 3DGS	27.85	0.821	0.312	–	25.03	0.794	0.355	–	17.43	0.552	0.578	–
Reweighting	27.91	0.823	0.306	11/24	25.14	0.805	0.341	7/24	17.65	0.561	0.569	1/24
Suppression	27.84	0.825	0.301	9/24	25.17	0.804	0.340	6/24	17.52	0.558	0.572	0/24
Pruning	28.13	0.828	0.295	10/24	25.61	0.805	0.330	6/24	16.88	0.524	0.596	0/24
RemedyGS	24.52	0.796	0.352	2/24	22.03	0.777	0.387	0/24	15.91	0.501	0.615	0/24
ValidGS	**30.82**	**0.871**	**0.198**	**24/24**	**28.67**	**0.844**	**0.241**	**24/24**	**20.15**	**0.612**	**0.485**	**24/24**

Table 3 compares ValidGS with various defense baselines across different levels of attack difficulty on the three datasets. While all approaches decline as attack difficulty increases, ValidGS consistently outperforms the baselines, demonstrating superior defensive robustness.

5.3 Multi-View Defense

Table 4: Quantitative comparisons under multi-view poisoning.

# of Views	Method	V-ILLUSORY				V-TEST		
		PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
2	Poisoned 3DGS	27.38	0.839	0.304	–	27.65	0.801	0.291
	Reweighting	24.72	0.778	0.419	3/24	25.38	0.789	0.336
	Suppression	24.95	0.785	0.411	4/24	25.76	0.795	0.324
	Pruning	23.82	0.771	0.435	2/24	24.85	0.782	0.352
	RemedyGS	24.25	0.774	0.428	0/24	25.84	0.798	0.245
	ValidGS	**29.05**	**0.844**	**0.243**	**24/24**	**29.75**	**0.856**	**0.198**
3	Poisoned 3DGS	27.12	0.831	0.315	–	27.51	0.798	0.295
	Reweighting	24.36	0.772	0.428	1/24	24.81	0.784	0.345
	Suppression	24.55	0.779	0.417	1/24	25.24	0.791	0.332
	Pruning	23.25	0.763	0.448	0/24	24.32	0.775	0.359
	RemedyGS	21.68	0.697	0.482	0/24	25.73	0.779	0.315
	ValidGS	**28.68**	**0.835**	**0.256**	**24/24**	**29.35**	**0.849**	**0.211**
4	Poisoned 3DGS	26.85	0.824	0.322	–	27.42	0.795	0.299
	Reweighting	23.94	0.765	0.442	0/24	24.45	0.776	0.354
	Suppression	24.08	0.771	0.429	0/24	24.88	0.784	0.341
	Pruning	22.81	0.756	0.461	0/24	23.86	0.769	0.372
	RemedyGS	20.24	0.648	0.490	0/24	24.15	0.771	0.358
	ValidGS	**28.27**	**0.829**	**0.268**	**24/24**	**28.96**	**0.842**	**0.224**

In realistic scenarios, an attacker may inject illusions into multiple viewpoints simultaneously. Following (Ke et al., 2025), we place the poisoned views at 0°, 90°, 180°, and 270°. Table 4 reports results averaged across the three datasets. ValidGS achieves a DSR of 24/24 throughout the evaluated multi-view poisoning settings and records the best performance on both V-ILLUSORY and V-TEST. As the number of poisoned views increases, all baselines remain below the undefended Poisoned 3DGS model on both view sets, and their DSR never exceeds 4/24. RemedyGS achieves 0/24

DSR at every poisoning level, with its V-ILLUSORY quality degrading further as more views are poisoned. In contrast, ValidGS uses cross-view validation to identify reliable clean support and performs Gaussian-level filtering and purification, allowing it to preserve its defense performance as the number of poisoned views increases.

5.4 Ablation Studies

We performed ablation studies to evaluate each defense component's contribution in ValidGS. Due to space limitations, implementation details of the ablation variants and supplementary hyperparameter sensitivity results are provided in Appendix A.4 and Appendix B.10, respectively. The quantitative results are reported in Table 5. Relying solely on consistency-based reference construction yields limited defensive effectiveness, indicating that image-space clean-reference supervision alone cannot stably protect against persistent view-dependent poisoning. The $(A) + (B)$ variant improves V-ILLUSORY quality over (A) alone, increasing PSNR by 4.54 dB and SSIM by 0.031 while reducing LPIPS by 0.053. It also raises the defense success rate from 0/8 to 5/8, highlighting the importance of transforming validated clean regions into Gaussian-level statistics for selective filtering.

Table 5: Comparison of different defense component combinations on the Tanks & Temples dataset. We evaluate three defense components: (A) Consistency-Based Reference Construction, (B) Validation-Guided Gaussian Filtering, and (C) Selective Gaussian Purification.

Defense Component			V-ILLUSORY			V-TEST			DSR↑
(A)	(B)	(C)	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
✓			21.64	0.811	0.356	27.43	0.797	0.298	0/8
✓	✓		26.18	0.842	0.303	27.71	0.804	0.287	5/8
✓		✓	23.47	0.834	0.328	27.58	0.801	0.291	1/8
✓	✓	✓	27.12	0.855	0.264	28.69	0.867	0.221	8/8

As illustrated in Figure 4, without validation-guided Gaussian filtering, the injected illusion tends to reappear during later training iterations. This result suggests that Gaussian-level validation is necessary for selective Gaussian purification to work effectively. The $(A) + (C)$ variant combines consistency-based reference construction with suspicious-Gaussian pruning based on suspicious-region statistics, but omits both validation-guided Gaussian filtering and clean-Gaussian protection. It yields only limited gains over consistency-based reference construction alone, improving V-ILLUSORY PSNR by 1.83 dB and SSIM by 0.023 while reducing LPIPS by 0.028, but achieving only 1/8 DSR. In contrast, the complete model combines all three components and achieves 8/8 DSR while further improving both V-ILLUSORY and V-TEST quality. These results indicate that suspicious-Gaussian pruning alone is insufficient, and that validation-guided filtering together with clean-Gaussian protection is important for reliable defense.

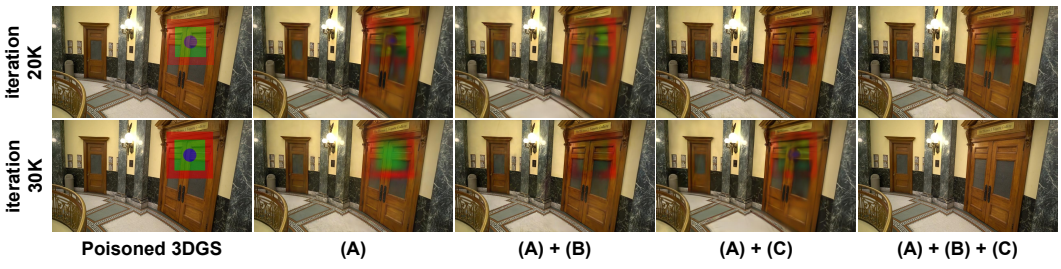

Figure 4: Qualitative analysis of defense component combinations.

6 Conclusion

In this paper, we present ValidGS, the first defense method against view-dependent poisoning in 3DGS. Our approach prevents poisoned views from persistently corrupting the reconstructed 3D scene. Building on our proposed clean reference construction and candidate target-view validation criterion, ValidGS employs a selective Gaussian purification strategy to fully mitigate target-view illusions while preserving the high reconstruction quality of innocent views. Comprehensive experimental results demonstrate ValidGS's effectiveness under diverse view-dependent poisoning settings.

AI USE STATEMENT

We used generative AI tools for code development, data analysis, and manuscript preparation, including language polishing and grammar correction. All AI-assisted outputs were reviewed, verified, and edited by the authors. We take responsibility for the content of this work, including all claims, experimental results, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or user studies. We use publicly available 3D scene datasets and controlled synthetic poisoning procedures to evaluate the security of view-dependent 3D reconstruction. The study is intended to improve the robustness of reconstruction systems and does not involve additional data collection from individuals.

REPRODUCIBILITY STATEMENT

Implementation details, attack settings, defense baselines, hyperparameters, dataset composition, and additional evaluation results are described in the main paper and appendix. The compared methods use matched attack inputs, camera selections, training schedules, and evaluation protocols. The five-seed analysis reports run-to-run variation, and the appendix records the implementation environment, including PyTorch 2.8.0+cu128 and the hardware used for the experiments, together with the configuration details needed to reproduce the reported results.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5479, 2022.

Marco Cannici and Davide Scaramuzza. Mitigating motion blur in neural radiance fields with events and frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9286–9296, 2024.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21476–21485, 2024.

Zi-Ting Chou, Sheng-Yu Huang, I-Jieh Liu, and Yu-Chiang Frank Wang. GSNeRF: Generalizable semantic neural radiance fields with enhanced 3D scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20806–20815, 2024.

Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Efficient density control for 3D Gaussian Splatting. *arXiv preprint arXiv:2411.10133*, 2024.

Sankeerth Durvasula, Sharanshangar Muhunthan, Zain Moustafa, Richard Chen, Ruofan Liang, Yushi Guan, Nilesh Ahuja, Nilesh Jain, Selvakumar Panneer, and Nandita Vijaykumar. ContraGS: Codebook-condensed and trainable Gaussian Splatting for fast, memory-efficient reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28935–28945, 2025.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. LightGaussian: Unbounded 3D gaussian compression with 15x reduction and 200+ FPS. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 140138–140158, 2024.

Xiangjun Gao, Xiaoyu Li, Yiyu Zhuang, Qi Zhang, Wenbo Hu, Chaopeng Zhang, Yao Yao, Ying Shan, and Long Quan. Mani-GS: Gaussian Splatting manipulation with triangular mesh. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21392–21402, 2025.

Emanuele Giacomini, Luca Di Giammarino, Lorenzo De Rebotti, Giorgio Grisetti, and Martin R. Oswald. Splat-LOAM: Gaussian Splatting LiDAR odometry and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27630–27639, 2025.

Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned Gaussian Splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5354–5363, 2024.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

Wenxiang Jiang, Hanwei Zhang, Shuo Zhao, Zhongwen Guo, and Hao Wang. IPA-NeRF: Illusory poisoning attack against neural radiance fields. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, volume 392, pp. 513–520, 2024.

Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. SelfSplat: Pose-free and 3D prior-free generalizable 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22012–22022, 2025.

Bo-Hsu Ke, You-Zhe Xie, Yu-Lun Liu, and Wei-Chen Chiu. StealthAttack: Robust 3D Gaussian Splatting poisoning via density-guided illusions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27400–27411, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139, 2023.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017.

Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21719–21728, 2024.

Yanping Li, Zhening Liu, Zijian Li, Zehong Lin, and Jun Zhang. RemedyGS: Defend 3D Gaussian Splatting against computation cost attacks. *arXiv preprint arXiv:2511.22147*, 2025. doi: 10.48550/arXiv.2511.22147. URL `https://arxiv.org/abs/2511.22147`.

Yifan Liu, Keyu Fan, Weihao Yu, Chenxin Li, Hao Lu, and Yixuan Yuan. MonoSplat: Generalizable 3D Gaussian Splatting from monocular depth foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21570–21579, 2025a.

Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. MaskGaussian: Adaptive 3D gaussian representation from probabilistic masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 681–690, 2025b.

Yaopeng Lou, Liao Shen, Tianqi Liu, Jiaqi Li, Zihao Huang, Huiqiang Sun, and Zhiguo Cao. MuGS: Multi-baseline generalizable Gaussian Splatting reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25583–25593, 2025.

Jiahao Lu, Yifan Zhang, Qiuhong Shen, Xinchao Wang, and Shuicheng Yan. Poison-splat: Computation cost attack on 3D Gaussian Splatting. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12346, pp. 405–421, 2020.

Jizong Peng, Tze Ho Elden Tse, Kai Xu, Wenchao Gao, and Angela Yao. A constrained optimization approach for Gaussian Splatting from coarsely-posed images and noisy LiDAR point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2961–2970, 2025.

Maxime Pietrantoni, Gabriela Csurka, and Torsten Sattler. Gaussian Splatting feature fields for (privacy-preserving) visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1082–1092, 2025.

Aashish Rai, Dilin Wang, Mihir Jain, Nikolaos Sarafianos, Kefan Chen, Srinath Sridhar, and Aayush Prakash. UVGS: Reimagining unstructured 3D Gaussian Splatting using UV mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5927–5937, 2025.

Cong Ruan, Yuesong Wang, Tao Guan, Bin Zhang, and Lili Ju. IndoorGS: Geometric cues guided Gaussian Splatting for indoor scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 844–853, 2025.

Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J. Fleet, and Andrea Tagliasacchi. RobustNeRF: Ignoring distractors with robust losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20626–20636, 2023. doi: 10.1109/CVPR52729.2023.01976.

Su Sun, Cheng Zhao, Zhuoyang Sun, Yingjie Victor Chen, and Mei Chen. SplatFlow: Self-supervised dynamic Gaussian Splatting in neural motion flow field for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27487–27496, 2025.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904, 2024.

Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-NeRF: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4150–4159, 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.

Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. NeRFiller: Completing scenes via generative 3D inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20731–20741, 2024.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense visual SLAM with 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604, 2024.

Haosen Yang, Chenhao Zhang, Wenqing Wang, Marco Volino, Adrian Hilton, Li Zhang, and Xiatian Zhu. Improving Gaussian Splatting with localized points management. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21696–21705, 2025.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

A Detailed Experimental Setup

A.1 Experiments Compute Resources

To evaluate the effectiveness of our approach, we implemented our method in PyTorch (version 2.8.0+cu128). All experiments were carried out on an Ubuntu workstation equipped with one Intel i7-13700K CPU, 64GB memory, and one NVIDIA GeForce RTX 4090 GPU.

A.2 Attack Settings

We use StealthAttack (Ke et al., 2025) only as the poisoned-data generator and keep all attack-side choices fixed before running any defense. For each scene, we select the target image according to the camera-position protocol used in the main experiments, then overlay the trigger object to produce the poisoned target image and mask. We then run the density-guided point cloud attack with the official configuration. Unless otherwise specified, we set the KDE bandwidth to 7.5, point size to 0.005, normal search radius to 0.1, and the maximum number of nearest neighbors to 30, following the released implementation and the bandwidth analysis in StealthAttack.

The density-guided point cloud attack follows the procedure outlined in StealthAttack. We begin by constructing an Axis-Aligned Bounding Box (AABB) for the scene and voxelizing the scene space. Next, we estimate the density at each voxel using Gaussian opacity and smooth the density field with a Gaussian Kernel Density Estimate (KDE). For each pixel within the illusory-object mask, the attack casts a ray from the target camera and searches for the point along the ray with the minimum density. We set the near bound to $t_{\text{min}} = 0.3$ to avoid issues with near-camera floaters, using the original scene depth at that pixel as $t_{\max}$. The inserted Gaussian points adopt colors from the corresponding illusory-object pixels, making the injected content visible in the target view while reducing its visibility from other perspectives.

We train the poisoned 3DGS model using the official StealthAttack pipeline to disrupt view consistency. We inject Gaussian noise only into non-target training views, while the poisoned target view remains unaffected. Following the noise-scheduling analysis outlined in StealthAttack, we apply linear noise decay, starting with an initial strength of 100 over 30,000 iterations. This configuration maintains the illusion of the target view while reducing cross-view constraints imposed by the innocent views.

We reuse the same poisoned scene files for poisoned 3DGS, all defense baselines, and ValidGS. For free scenes with irregular camera trajectories, we select the EASY, MEDIAN, and HARD target views using the KDE-based camera-density protocol from StealthAttack. In this protocol, we compute camera viewpoint densities within each field of view using a 10% sampling radius. For datasets with more uniform camera coverage, we also follow StealthAttack's approach and use the median-indexed frame as the default target view. This controlled setting ensures that any performance differences arise from the defense methods rather than from variations in attack generation.

A.3 Defense Baseline Settings

All defense baselines begin with the same poisoned scene files generated by StealthAttack and utilize the same 3DGS (Kerbl et al., 2023) backbone, optimizer, number of training iterations, and evaluation protocol. We adjust only the defense-specific switches and thresholds for each baseline, keeping the attack configuration and dataset split unchanged.

Adaptive Reweighting. We implement adaptive reweighting based on MaskGaussian (Liu et al., 2025b). In our baseline approach, all Gaussians remain trainable, and we adjust their contribution weights during optimization. Specifically, we enable view-weight estimation after the warm-up stage, update the view weights every 500 iterations, sample 8 neighboring views for each update, set the weighting balance at 0.5, and clamp the minimum view weight to 0.3. This configuration provides a soft defense baseline that reduces the influence of views or regions with unstable cross-view support without removing any Gaussians, preserving the scene's overall structural integrity.

Selective Suppression. We implement selective suppression based on localized point management, as described in (Yang et al., 2025). Our baseline approach divides each view into 32×32 patches and maintains per-view patch statistics using an exponential moving average (EMA) with a decay rate of 0.95. Suppression begins after a 2000-iteration warm-up period. We use a robust median-plus-MAD threshold with a scale of 4.0 to identify suspicious patches and assign a reduced pixel weight of 0.15 to these selected areas. This setting serves as a local suppression baseline that minimizes the influence of suspicious image-space supervision while still preserving the standard 3DGS densification process.

Density-Based Pruning. We implement density-based pruning by following an efficient density control method (Deng et al., 2024). In our baseline approach, we load the scene's voxel grid and smooth the density field using a KDE bandwidth of 7.5. During training, we query the density value for each Gaussian. After a warm-up period of 2000 iterations, we calculate the density threshold from the lowest 5% of density values and determine the support threshold from the lowest 10% of established visibility support values. During pruning, we remove Gaussians with low density, low support, and opacity above the pruning threshold, and we categorize them as questionable Gaussians.

RemedyGS. We implement RemedyGS (Li et al., 2025) as a cross-task defense baseline using its official pipeline, which purifies each training image with an encoder-decoder before 3DGS optimization. We keep the released purification network and apply it to the same poisoned training images used by all other methods. We then use the purified images to train the victim 3DGS model with the same backbone, optimizer, iteration budget, and evaluation protocol as the other baselines.

ValidGS. In implementation, ValidGS maintains view-specific reference and validation statistics for each candidate target view v_c. Once a candidate view satisfies the validation criterion for K consecutive visits, its validated clean and suspicious regions update the global Gaussian-level statistics for both, which are then used for subsequent protection and pruning. For multi-view poisoning, multiple activated candidate views can jointly contribute to these global Gaussian-level statistics. ValidGS employs the same poisoned inputs and training schedule as the defense baselines, with all models trained for 30000 iterations and $\lambda_{\text{DSSIM}} = 0.2$. To construct clean references, we use 32×32 patches and select the six nearest neighboring training views by camera-center distance. The color bandwidth σ_c in Equation 5 is set to 0.10. After selecting the neighboring views, we mark a patch as unsupported when its EMA support score is below 0.55, and its EMA valid projection ratio is at least 0.30. For each sampled pixel, we fuse the valid color observations from neighboring views using their mean when fewer than three are available and their median otherwise. The local reference reliability map is computed as $w(\mathbf{x}) = \exp(-\bar{d}(\mathbf{x})/0.15)$, where $\bar{d}(\mathbf{x})$ is the mean color difference between the valid neighboring colors and the fused reference color. Patch support and hard patch statistics use an EMA decay of 0.95, while Gaussian projection and hit statistics for both clean and suspicious regions use an EMA decay of 0.99. Candidate target-view validation starts after a 15,000-iteration warm-up and is evaluated whenever the standard training sampler visits a training view, which is treated as the current candidate target view v_c. We set $\tau_{\text{val}} = 0.80$, $\tau_{\text{rel}} = 0.75$, $\tau_{\text{l1}} = 0.08$, and require $K = 3$ consecutive valid visits to the same candidate target view. For Gaussian-level filtering, the parameters are set as $\tau^c_{\text{seen}} = 0.02$, $\tau^s_{\text{seen}} = 0.02$, $\tau^c_{\text{ratio}} = 0.50$, $\tau_{\text{clean}} = 0.50$, $\tau_{\text{sus}} = 0.30$, $m = 0.15$, and $\kappa_c = 3.0$. For selective Gaussian purification, we set $\lambda_{dc} = 0.02$, $\lambda_{sh} = 0.01$, $\gamma_{sh} = 0.1$, $\gamma_{dc} = 0.25$, and $\gamma_{\alpha} = 0.25$, applying a late-pruning interval of $\Delta_{\text{prune}} = 3000$ iterations with an opacity threshold of 0.10.

A.4 Ablation Variant Implementation

For the ablation study in Table 5, we train every variant with the same poisoned inputs, 3DGS backbone, optimizer, training schedule, and evaluation setup as the full ValidGS model. All enabled modules use the identical hyperparameters specified in Appendix A.3. Thus, the ablation exclusively alters which defense modules are activated during training.

The (A) variant activates only the consistency-based reference construction. It derives clean image-space reference supervision from neighboring views and applies the corresponding reference-guided image-space loss during optimization. It tracks neither clean nor suspicious Gaussian-level statistics, and it does not build $\mathcal{M}_{\text{val}}$, $\mathcal{M}_{\text{prot}}$, or $\mathcal{M}_{\text{sus}}$, thereby completely bypassing all Gaussian-level protection and pruning operations.

The $(A) + (B)$ variant further enables validation-guided Gaussian filtering. Once a candidate target view meets the validation criterion, this variant aggregates Gaussian-level statistics from the validated clean and suspicious regions. It then constructs $\mathcal{M}_{\text{val}}$ and $\mathcal{M}_{\text{prot}}$ using the same filtering rules as the full model. However, because component C remains disabled, the resulting protected set is not used for reference-feature regularization, gradient attenuation, opacity attenuation, or late-stage pruning. This setup isolates whether lifting image-space validation to Gaussian-level filtering alone is adequate without subsequent purification.

The $(A) + (C)$ variant adopts a purification-only configuration to assess suspicious-Gaussian pruning without validation-guided Gaussian filtering or clean-Gaussian protection. Without component B, this variant does not build $\mathcal{M}_{\text{val}}$ or $\mathcal{M}_{\text{prot}}$ and does not apply clean-Gaussian protection. Instead, it projects the suspicious image-space regions identified by component A onto the visible Gaussians, accumulates projection and hit statistics for these regions, and periodically prunes the resulting $\mathcal{M}_{\text{sus}}$ using the same pruning interval of $\Delta_{\text{prune}} = 3000$ iterations and opacity threshold of 0.10 as the full model. This design assesses whether suspicious-Gaussian pruning, when added to consistency-based reference construction without validation-guided filtering or clean-Gaussian protection, can effectively mitigate the target-view illusion.

The $(A) + (B) + (C)$ variant enables the full ValidGS pipeline. It first employs component A to form clean reference supervision, then utilizes component B to identify validated clean Gaussians and filter out those that also support suspicious regions. Following validation, component C protects these filtered clean Gaussians via reference-feature regularization and gradient attenuation, while also periodically pruning persistent suspicious Gaussians during later optimization and densification.

B SUPPLEMENTARY EXPERIMENTAL RESULTS

B.1 STATISTICAL EVALUATION ACROSS RANDOM SEEDS

We repeat all single-view defense experiments on the three datasets with five independent training seeds, keeping all other experimental settings fixed. Within each seed, all methods share the same initialization to enable paired comparisons. We first average each metric over the 24 scenes for each seed and then report the mean and standard deviation across the five seed-level averages.

Table 6: Five-seed statistical comparison under single-view poisoning across the three datasets. Results are averaged over 24 scenes and reported as mean ± standard deviation.

Method	V-ILLUSORY			V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Poisoned 3DGS	25.01 ± 0.16	0.792 ± 0.004	0.352 ± 0.007	26.86 ± 0.12	0.795 ± 0.003	0.302 ± 0.006
Reweighting	25.30 ± 0.21	0.806 ± 0.006	0.335 ± 0.009	26.45 ± 0.18	0.809 ± 0.005	0.296 ± 0.007
Suppression	25.18 ± 0.17	0.805 ± 0.005	0.335 ± 0.008	26.71 ± 0.14	0.808 ± 0.004	0.291 ± 0.006
Pruning	25.77 ± 0.19	0.806 ± 0.007	0.323 ± 0.010	25.81 ± 0.22	0.799 ± 0.006	0.312 ± 0.009
RemedyGS	21.73 ± 0.12	0.774 ± 0.004	0.387 ± 0.007	26.08 ± 0.15	0.810 ± 0.007	0.254 ± 0.006
ValidGS	$\mathbf{28.68 \pm 0.11}$	$\mathbf{0.845 \pm 0.003}$	$\mathbf{0.241 \pm 0.004}$	$\mathbf{29.82 \pm 0.09}$	$\mathbf{0.858 \pm 0.002}$	$\mathbf{0.205 \pm 0.005}$

Table 6 shows that ValidGS achieves the best mean performance on all six metrics across the five training seeds. For each metric, its margin over the strongest competing method exceeds the reported variation across seeds, indicating that the gains are consistent across the evaluated initializations.

B.2 PURIFICATION OF TRAINED MODELS

To determine whether ValidGS can be applied after a poisoned 3DGS model has already been trained, we initialize two continuation branches from the same official 30K StealthAttack checkpoint on Mip-NeRF360. The standard-continuation branch follows the original 3DGS training procedure, whereas the ValidGS branch activates the proposed defense. Both branches use the same training images, camera poses, and continuation schedule, and we evaluate them after 1K, 3K, 5K, and 7K additional iterations. This setting evaluates checkpoint-based purification rather than data-free model sanitization because it requires access to the training data and further optimization.

Table 7: Purification from fully trained poisoned checkpoints.

Additional Iterations	Method	V-ILLUSORY				V-TEST		
		PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
1K	Standard 3DGS	26.95	0.809	0.374	0/24	27.75	0.805	0.285
	ValidGS	**27.56**	**0.820**	**0.341**	**8/24**	**28.26**	**0.815**	**0.271**
3K	Standard 3DGS	26.88	0.806	0.379	0/24	27.81	0.808	0.281
	ValidGS	**28.58**	**0.833**	**0.285**	**17/24**	**28.87**	**0.831**	**0.245**
5K	Standard 3DGS	26.82	0.803	0.383	0/24	27.84	0.810	0.278
	ValidGS	**29.17**	**0.842**	**0.251**	**24/24**	**29.51**	**0.849**	**0.214**
7K	Standard 3DGS	26.78	0.801	0.387	0/24	27.89	0.811	0.275
	ValidGS	**29.43**	**0.846**	**0.239**	**24/24**	**29.98**	**0.857**	**0.198**

Table 7 shows that standard continuation remains at 0/24 DSR throughout the additional 7K iterations, whereas ValidGS reaches full DSR after 5K iterations and maintains it at 7K. The corresponding gains over standard continuation grow from 0.61 to 2.65 dB on V-ILLUSORY PSNR and from 0.51 to 2.09 dB on V-TEST PSNR as the purification proceeds. At 7K iterations, ValidGS also reduces LPIPS relative to standard continuation by 0.148 on V-ILLUSORY and 0.077 on V-TEST. Because both methods start from the same poisoned checkpoint and receive the same additional optimization budget, the persistent 0/24 DSR of standard continuation isolates the contribution of validation-guided filtering and purification. These results show that ValidGS can purify a fully trained poisoned representation without restarting from the initial point cloud, provided that the associated training images and camera poses remain available.

B.3 FRAMEWORK COMPATIBILITY

To examine whether ValidGS is tied to the Vanilla 3DGS implementation, we integrate its validation and purification pipeline into Mip-Splatting (Yu et al., 2024) as an alternative 3DGS framework. All

Table 8: Framework compatibility of ValidGS under single-view poisoning across multiple datasets.

Method	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
Vanilla 3DGS undefended	25.03	0.794	0.355	0/24	26.97	0.798	0.298
Mip-Splatting undefended	25.35	0.803	0.341	0/24	27.28	0.814	0.282
Vanilla 3DGS + ValidGS	**28.68**	**0.845**	**0.241**	**24/24**	**29.82**	**0.857**	**0.205**
Mip-Splatting + ValidGS	**29.86**	**0.858**	**0.201**	**24/24**	**31.05**	**0.892**	**0.174**

models are trained from scratch on Mip-NeRF360 under single-view poisoning and evaluated using the same protocol. Table 8 shows that both undefended frameworks obtain a DSR of 0/24, while Mip-Splatting provides moderately better continuous reconstruction metrics than Vanilla 3DGS. After incorporating ValidGS, both frameworks attain a DSR of 24/24. The consistent transition from 0/24 to 24/24 across the two evaluated frameworks indicates that ValidGS is not restricted to the Vanilla 3DGS implementation. Moreover, the Mip-Splatting backbone retains its reconstruction-quality advantage after integrating ValidGS, outperforming Vanilla 3DGS with ValidGS on both view sets.

B.4 Performance against Adaptive Attacks

We evaluate ValidGS under two white-box adaptive variants derived from StealthAttack. The attacker knows the candidate target-view validation and Gaussian-level filtering criteria. The first variant strengthens the partial cross-view support of the injected region, whereas the second delays the emergence of suspicious signals during training.

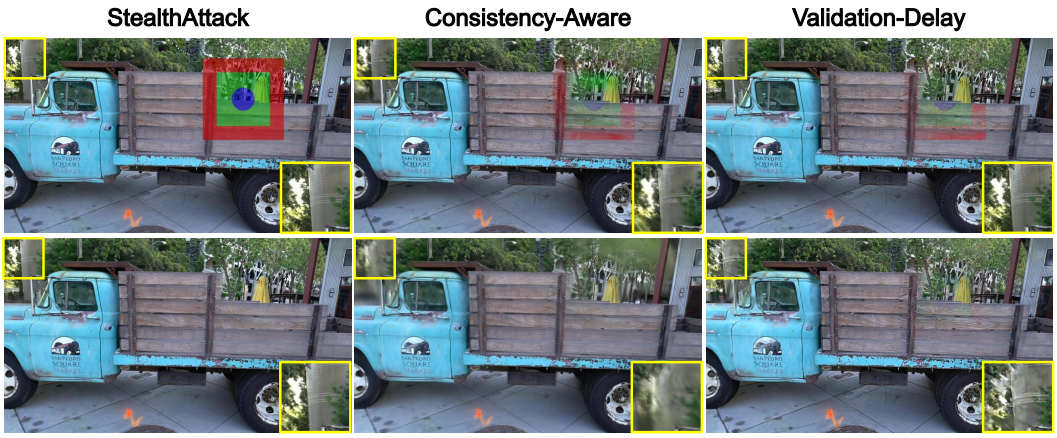

Figure 5: Qualitative comparison under adaptive attacks. The top row shows rendered images without our defense, while the bottom row shows the corresponding results with our defense applied.

Consistency-Aware poisoning targets the cross-view support used for reference construction. Matching the default defense configuration, the attacker selects the six nearest neighboring training views for each target camera by camera-center distance and modifies each selected view. For each neighboring view v_n, let I_{v_n} denote the original neighboring image, let T_{v_n} denote the target trigger patch resized to the corresponding neighboring region, and let $M_{v_n} \in [0,1]$ denote the resized soft mask obtained by applying a Gaussian blur with radius 3.0 to the target-view mask. We construct the modified neighboring image as

$$I_{v_n}^{\mathrm{ca}} = (1 - \alpha M_{v_n}) \odot I_{v_n} + \alpha M_{v_n} \odot T_{v_n}, \tag{9}$$

where $I_{v_n}^{\mathrm{ca}}$ denotes the modified neighboring image and α controls the blending strength. We evaluate $\alpha \in \{0.06, 0.12, 0.18, 0.24\}$, progressively increasing the contribution of the injected content in six selected neighboring views. This construction strengthens partial cross-view support but does not enforce geometric and photometric consistency of the injected content across relevant training views.

Validation-Delay poisoning exploits the repeated-validation mechanism of ValidGS by attempting to postpone the activation of Gaussian-level protection. We keep the poisoned scene unchanged and replace the original innocent-view noise schedule with a delayed schedule over T training iterations.

With $\sigma_0 = 300$, the noise standard deviation is defined as

$$\sigma_t^{\text{delay}} = \begin{cases} 0, & t < 0.4T, \\ \sigma_0 \frac{t-0.4T}{0.2T}, & 0.4T \le t < 0.6T, \\ \sigma_0 \left(1 - \frac{t-0.6T}{0.4T}\right), & 0.6T \le t \le T, \end{cases} \tag{10}$$

and we set $T = 30000$ in our experiments. This schedule removes the disruption during the first 40% of training, ramps it up between 40% and 60%, and then decays it linearly afterward. As a result, the poisoned region appears less suspicious in the early stages, thereby bypassing the validation criterion and re-establishing the illusion during subsequent optimization.

Table 9: Quantitative comparison of defense methods under adaptive attacks.

Adaptive Attack		Method	V-ILLUSORY				V-TEST		
			PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
Consistency-Aware	$\alpha = 0.06$	Vanilla 3DGS	24.38	0.831	0.334	0/24	26.12	0.849	0.235
		Reweighting	24.83	0.834	0.321	4/24	26.57	0.852	0.226
		Suppression	24.91	0.836	0.317	5/24	26.63	0.854	0.221
		Pruning	23.67	0.822	0.358	2/24	25.84	0.841	0.253
		RemedyGS	24.52	0.829	0.325	1/24	25.79	0.845	0.247
		ValidGS	**26.45**	**0.842**	**0.279**	**24/24**	**27.68**	**0.861**	**0.217**
	$\alpha = 0.12$	Vanilla 3DGS	23.91	0.823	0.341	0/24	25.55	0.843	0.244
		Reweighting	24.16	0.826	0.332	2/24	25.74	0.846	0.237
		Suppression	24.28	0.828	0.327	3/24	25.81	0.848	0.232
		Pruning	23.49	0.817	0.362	1/24	25.18	0.835	0.259
		RemedyGS	23.82	0.821	0.348	0/24	25.07	0.839	0.261
		ValidGS	**24.96**	**0.832**	**0.303**	**23/24**	**25.67**	**0.851**	**0.229**
	$\alpha = 0.18$	Vanilla 3DGS	23.42	0.814	0.353	0/24	24.89	0.834	0.258
		Reweighting	23.61	0.817	0.345	1/24	24.96	0.837	0.252
		Suppression	23.78	0.819	0.340	1/24	25.12	0.839	0.248
		Pruning	22.93	0.805	0.374	0/24	24.48	0.826	0.271
		RemedyGS	23.15	0.811	0.361	0/24	24.53	0.831	0.265
		ValidGS	**24.13**	**0.824**	**0.324**	**19/24**	**25.29**	**0.844**	**0.243**
	$\alpha = 0.24$	Vanilla 3DGS	22.87	0.801	0.368	0/24	24.31	0.821	0.272
		Reweighting	23.09	0.805	0.361	0/24	24.38	0.824	0.267
		Suppression	23.14	0.807	0.357	0/24	24.45	0.827	0.264
		Pruning	22.51	0.793	0.384	0/24	23.92	0.813	0.286
		RemedyGS	22.73	0.798	0.375	0/24	24.11	0.818	0.279
		ValidGS	**23.58**	**0.812**	**0.341**	**14/24**	**24.63**	**0.832**	**0.256**
Validation-Delay		Vanilla 3DGS	24.87	0.829	0.291	0/24	27.05	0.818	0.237
		Reweighting	24.93	0.832	0.288	1/24	27.14	0.821	0.233
		Suppression	25.11	0.835	0.283	2/24	27.22	0.824	0.229
		Pruning	24.08	0.816	0.312	0/24	26.67	0.809	0.254
		RemedyGS	24.65	0.825	0.301	0/24	26.83	0.815	0.246
		ValidGS	**26.09**	**0.846**	**0.274**	**24/24**	**27.42**	**0.835**	**0.218**

Table 9 highlights the difference between validation-guided defense and baselines that operate without constructing a verified clean reference. Under Consistency-Aware poisoning, the baseline defenses provide only limited improvements, and their effectiveness decreases as the injected content gains support across the neighboring views. Without explicitly validating the reference signal before making defense decisions, these methods remain vulnerable to the cross-view support the adaptive attack deliberately introduces. In contrast, ValidGS achieves the strongest attack-region recovery and substantially higher DSR throughout the sweep by jointly validating clean references and separating clean and suspicious Gaussian contributions. The decline at larger α localizes the remaining limitation to settings in which poisoning of the reference views reduces the contrast between genuine and attack-induced cross-view support. Validation-Delay poisoning instead conceals suspicious evidence during early optimization. The baselines achieve only isolated successes under this setting. ValidGS accumulates reference and Gaussian statistics across repeated visits, then activates filtering and purification when the delayed content becomes distinguishable from neighboring-view observations. Its full DSR under this setting shows that temporal delay alone is insufficient to evade the persistent validation mechanism.

B.5 Complex Objects under Multi-View Poisoning

We further evaluate ValidGS under multi-view poisoning with Colorful Square, Dog, and Person as the injected objects. For each object, we use mask ratios of 10% and 20% and poison two, three, or four target views, resulting in 18 attack configurations. We aggregate the DSR over the 24 scenes used in our main evaluation, and we average the image-quality metrics over the corresponding V-ILLUSORY and V-TEST views. Within each configuration, the undefended and defended models use identical poisoned inputs, camera parameters, and training schedules.

Table 10: Defense performance under multi-view poisoning with different injected objects, mask ratios, and numbers of poisoned views across the 24 evaluated scenes.

Object	Mask	Views	Method	V-ILLUSORY				V-TEST		
				PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
Colorful Square	10%	2	Poisoned 3DGS	25.18	0.804	0.348	–	26.82	0.795	0.303
			ValidGS	**28.37**	**0.847**	**0.240**	**24/24**	**29.34**	**0.856**	**0.207**
		3	Poisoned 3DGS	24.83	0.796	0.362	–	26.61	0.791	0.311
			ValidGS	**28.02**	**0.840**	**0.249**	**24/24**	**29.05**	**0.849**	**0.216**
		4	Poisoned 3DGS	24.47	0.788	0.376	–	26.39	0.783	0.322
			ValidGS	**27.66**	**0.835**	**0.261**	**24/24**	**28.73**	**0.845**	**0.227**
	20%	2	Poisoned 3DGS	22.76	0.764	0.421	–	26.12	0.774	0.337
			ValidGS	**27.18**	**0.825**	**0.286**	**24/24**	**28.41**	**0.836**	**0.254**
		3	Poisoned 3DGS	22.31	0.751	0.444	–	25.84	0.765	0.350
			ValidGS	**26.72**	**0.810**	**0.302**	**24/24**	**28.07**	**0.826**	**0.263**
		4	Poisoned 3DGS	21.79	0.736	0.469	–	25.51	0.754	0.364
			ValidGS	**26.19**	**0.797**	**0.323**	**24/24**	**27.68**	**0.815**	**0.279**
Dog	10%	2	Poisoned 3DGS	24.59	0.808	0.329	–	26.21	0.785	0.327
			ValidGS	**27.43**	**0.841**	**0.257**	**24/24**	**28.47**	**0.848**	**0.244**
		3	Poisoned 3DGS	24.16	0.799	0.344	–	25.96	0.778	0.336
			ValidGS	**27.04**	**0.829**	**0.272**	**24/24**	**28.18**	**0.839**	**0.255**
		4	Poisoned 3DGS	23.68	0.786	0.363	–	25.67	0.770	0.351
			ValidGS	**26.62**	**0.811**	**0.282**	**24/24**	**27.84**	**0.826**	**0.265**
	20%	2	Poisoned 3DGS	22.67	0.758	0.398	–	25.37	0.749	0.371
			ValidGS	**26.43**	**0.803**	**0.309**	**24/24**	**27.69**	**0.817**	**0.283**
		3	Poisoned 3DGS	22.13	0.744	0.422	–	25.03	0.738	0.386
			ValidGS	**25.91**	**0.792**	**0.330**	**24/24**	**27.32**	**0.800**	**0.299**
		4	Poisoned 3DGS	21.54	0.727	0.451	–	24.64	0.724	0.404
			ValidGS	**25.33**	**0.772**	**0.352**	**24/24**	**26.88**	**0.784**	**0.315**
Person	10%	2	Poisoned 3DGS	23.91	0.787	0.358	–	25.75	0.763	0.353
			ValidGS	**26.86**	**0.819**	**0.276**	**24/24**	**28.03**	**0.820**	**0.262**
		3	Poisoned 3DGS	23.39	0.771	0.377	–	25.43	0.752	0.366
			ValidGS	**26.41**	**0.806**	**0.293**	**24/24**	**27.70**	**0.809**	**0.273**
		4	Poisoned 3DGS	22.81	0.755	0.402	–	25.08	0.739	0.381
			ValidGS	**25.89**	**0.794**	**0.310**	**24/24**	**27.27**	**0.798**	**0.289**
	20%	2	Poisoned 3DGS	21.93	0.731	0.438	–	24.78	0.717	0.411
			ValidGS	**25.72**	**0.780**	**0.338**	**24/24**	**26.91**	**0.788**	**0.306**
		3	Poisoned 3DGS	21.29	0.714	0.467	–	24.37	0.701	0.431
			ValidGS	**25.13**	**0.762**	**0.367**	**24/24**	**26.46**	**0.773**	**0.331**
		4	Poisoned 3DGS	20.56	0.692	0.503	–	23.89	0.682	0.457
			ValidGS	**24.45**	**0.741**	**0.390**	**24/24**	**25.94**	**0.753**	**0.354**

Table 10 quantifies attack severity across object type, mask area, and the number of poisoned views. Averaged over the two-, three-, and four-view settings, increasing the mask ratio from 10% to 20% reduces the undefended V-ILLUSORY PSNR by 2.54 dB, 2.03 dB, and 2.11 dB for Colorful Square, Dog, and Person, respectively. SSIM decreases by 0.046, 0.055, and 0.059, while LPIPS increases by 0.083, 0.078, and 0.090. The consistent degradation across all three metrics confirms that the larger masks produce more disruptive target-view attacks rather than merely changing the appearance of a small trigger. Across all 18 configurations, ValidGS achieves a DSR of 24/24 and improves both V-ILLUSORY and V-TEST metrics over the corresponding undefended model. These results extend the evaluation beyond small injected patterns and show that the defense remains effective for larger, semantically structured objects under multi-view poisoning.

B.6 Natural Multi-View Inconsistencies

We further evaluate single-view poisoning on five RobustNeRF scenes (Sabour et al., 2023), whose training views contain moving or transient distractors that introduce natural cross-view inconsistencies. Using the official split, we construct poisoned training data with StealthAttack's default Colorful Square mask. Vanilla 3DGS, StealthAttack, and ValidGS use the same image splits, camera parameters, and training schedule. Table 11 shows that StealthAttack remains effective when the training views contain natural transient distractors. ValidGS restores the attacked-view metrics to slightly above the clean Vanilla 3DGS reference, achieves a DSR of 5/5, and outperforms the competing defenses on the reported measures. Its V-TEST metrics are also comparable to or better than those of Vanilla 3DGS, indicating that the validation and purification steps do not broadly remove scene content in these five scenes. These results show that natural cross-view inconsistencies do not prevent ValidGS from separating the evaluated target-view poisoning signal from useful scene support.

Table 11: Quantitative comparison under single-view poisoning on RobustNeRF.

Method	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
Vanilla 3DGS	26.43	0.865	0.071	–	26.57	0.877	0.064
StealthAttack	24.48	0.851	0.112	–	26.21	0.869	0.071
Reweighting	24.08	0.835	0.124	0/5	24.86	0.841	0.102
Suppression	24.23	0.839	0.119	1/5	25.12	0.848	0.095
Pruning	23.41	0.819	0.141	0/5	23.75	0.812	0.121
RemedyGS	23.67	0.826	0.134	0/5	24.31	0.825	0.113
ValidGS	**27.19**	**0.871**	**0.070**	**5/5**	**27.45**	**0.894**	**0.061**

B.7 Global Training-View Sparsity

We evaluate ValidGS under global training-view sparsity on Mip-NeRF360, Tanks & Temples, and Free. For each scene, we retain the poisoned target view and a fixed set of held-out test views, uniformly subsample the remaining innocent training views, and reconstruct the neighboring-view lists from the retained training cameras. We also report the average number of valid neighboring views and the valid projection ratio measured during reference construction.

Table 12: ValidGS performance and cross-view availability under global training-view sparsity on Mip-NeRF360, Tanks & Temples, and Free.

Retained Views	V-ILLUSORY				V-TEST			Cross-View Availability	
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓	Valid Neighbors	Projection Ratio
90%	28.61	0.880	0.103	24/24	26.74	0.848	0.145	5.05	1.000
70%	26.85	0.853	0.158	24/24	26.03	0.833	0.154	4.97	1.000
50%	23.86	0.812	0.199	24/24	24.89	0.803	0.173	4.54	0.859
30%	22.76	0.731	0.280	21/24	23.15	0.750	0.211	3.96	0.849
10%	15.52	0.543	0.533	17/24	18.48	0.556	0.363	2.24	0.723

Table 12 shows that ValidGS maintains a DSR of 24/24 at retention ratios of 50% or higher, although reconstruction quality declines as training views are removed. At 30% and 10%, fewer valid neighboring observations coincide with lower DSR and further degradation on both view sets. Fewer neighboring observations provide less evidence for clean-reference construction and subsequent Gaussian-level validation, while the reduced training set also limits general scene reconstruction.

Figure 6 relates these local support measurements to reconstruction quality across the same view subsampling settings. The effective neighbor count measures how many neighboring views provide valid projections, whereas the valid projection ratio measures how much of the attacked region is covered by at least one neighbor. A region can therefore retain projection coverage while losing the additional views used to construct its reference.

B.8 Performance Comparison on Computational Overhead

Table 13 reports a computational overhead measurement on the Mip-NeRF360 dataset. For ValidGS, the measured time includes the entire defense pipeline, comprising clean reference construction, candidate target-view validation,

Table 13: Quantitative comparison of computational overhead.

Method	Time	Storage	Peak VRAM
Vanilla 3DGS	41m33s	734MB	13.67GB
StealthAttack	58m57s	876MB	13.46GB
ValidGS	**35m58s**	**692MB**	**12.13GB**

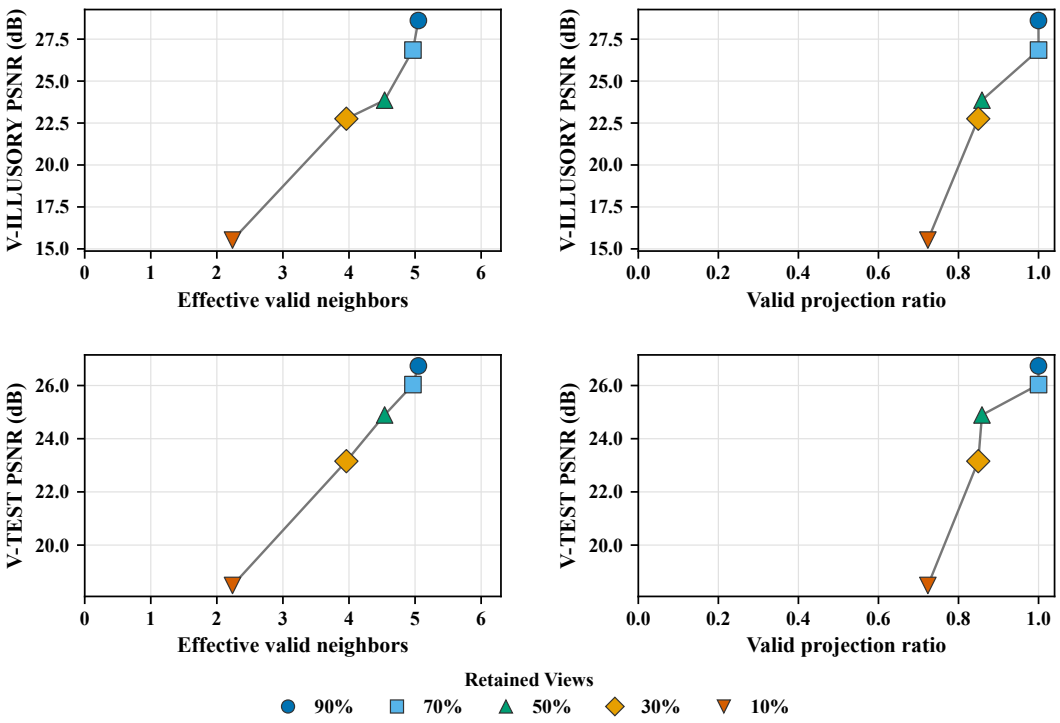

Figure 6: Local cross-view availability and rendering quality under global training-view sparsity on Mip-NeRF360, Tanks & Temples, and Free.

Gaussian-level filtering, protection, and pruning. This measurement includes the all-candidate validation procedure, where sampled training views are checked as candidate target views during optimization. Under this setting, ValidGS reduces training time, model storage, and peak VRAM by 13.4%, 5.7%, and 11.3% relative to vanilla 3DGS, and by 39.0%, 21.0%, and 9.9% relative to the StealthAttack pipeline. This efficiency improvement mainly comes from periodic pruning, which removes persistent, suspicious Gaussians and reduces the number of trainable primitives used during later optimization and densification.

B.9 Performance on Benign Training Data

To assess ValidGS's practical utility, we report its performance in a benign training scenario without poisoning attacks. In this configuration, we train all models on the clean Mip-NeRF360 dataset using the same backbone, training budget, and evaluation protocol as in the main experiments.

Table 14: Quantitative comparison on benign training data.

Method	PSNR↑	SSIM↑	LPIPS↓	Time	Storage	Peak VRAM
Vanilla 3DGS	27.21	0.815	0.214	41m33s	734MB	13.67GB
ValidGS	**26.86**	**0.804**	**0.243**	**37m27s**	**724MB**	**13.27GB**

Table 14 reports the reconstruction quality and training cost for ValidGS and vanilla 3DGS when trained on unpoisoned data. On the clean Mip-NeRF360 dataset, ValidGS closely tracks vanilla 3DGS in the benign setting, suggesting that the defense method does not materially interfere with standard scene optimization in the absence of poisoning. Relative to vanilla 3DGS, ValidGS lowers PSNR by 0.35 dB and SSIM by 0.011, and raises LPIPS by 0.029. In clean scenes, cross-view validation difficulties trigger cautious filtering and pruning, resulting in a slight drop in quality. Besides, in benign scenes, patches near occlusions, thin geometry, and view-dependent effects are harder to validate consistently across views. This validation difficulty leads to more conservative updates for some useful Gaussians, or to their earlier pruning, compared with vanilla 3DGS. Regarding efficiency, ValidGS shortens the training time by 4m06s, reduces model storage by 10MB, and lowers peak VRAM by 0.40GB. These results indicate that the added validation and candidate-specific

bookkeeping remain well contained under benign training. The efficiency improvements mainly stem from periodic pruning, which discards redundant Gaussians during optimization and reduces the cost of subsequent densification and rendering state management. The decreases in storage and peak VRAM further show that periodic pruning keeps the retained Gaussian set compact, so the defense does not inflate the final model size or peak memory usage on clean scenes.

B.10 HYPERPARAMETER SENSITIVITY

We evaluate hyperparameter sensitivity on the final evaluation scenes from Mip-NeRF360, Tanks & Temples, and Free. For each experiment, we vary one hyperparameter and keep all other settings fixed. We average each continuous metric across the three datasets and compute DSR over all 24 evaluated scenes. The bold entries indicate the fixed configuration used in the main experiments. We report these experiments to characterize its sensitivity across the tested ranges. All manually specified hyperparameters are shared across scenes and datasets. We do not tune them for individual scenes or separately for indoor and outdoor scenes. The clean hit threshold τ_{hit}^{c} is computed automatically for each candidate view using sufficiently observed Gaussians. It is defined as the median of their hit statistics plus κ_c times the corresponding median absolute deviation.

Table 15: Performance under different consecutive validation requirements, averaged across Mip-NeRF360, Tanks & Temples, and Free.

K	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
1	27.64	0.812	0.269	18/24	28.68	0.825	0.231
3	**28.67**	**0.844**	**0.241**	**24/24**	**29.82**	**0.857**	**0.206**
5	28.27	0.821	0.258	22/24	29.16	0.837	0.214

Table 15 evaluates the number of consecutive successful validations required before activating Gaussian-level operations. Reducing K from 3 to 1 lowers DSR from 24/24 to 18/24. It also decreases PSNR by 1.03 dB on V-ILLUSORY and 1.14 dB on V-TEST, with corresponding degradation in SSIM and LPIPS. This pattern is consistent with single-step validation being more susceptible to transient rendering fluctuations, which can activate the defense using an unstable reference and lead to erroneous filtering of clean Gaussians. Increasing K to 5 is more reliable than using a single validation but lowers DSR to 22/24 and remains worse than $K = 3$ on all six continuous metrics. Requiring five consecutive validations may delay intervention, allowing suspicious Gaussians to accumulate before filtering begins. Among the tested values, $K = 3$ best balances validation stability and timely defense activation.

Table 16: Performance under different validated-region thresholds, averaged across Mip-NeRF360, Tanks & Temples, and Free.

τ_{val}	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
0.70	27.82	0.827	0.254	21/24	28.92	0.832	0.220
0.80	**28.67**	**0.844**	**0.241**	**24/24**	**29.82**	**0.857**	**0.206**
0.90	28.16	0.828	0.252	23/24	29.34	0.840	0.212

Table 16 evaluates the minimum fraction of validated clean-reference patches required to activate Gaussian-level operations. Lowering τ_{val} from 0.80 to 0.70 reduces DSR from 24/24 to 21/24 and decreases PSNR by 0.85 dB on V-ILLUSORY and 0.90 dB on V-TEST. The permissive threshold can admit insufficiently supported reference patches, introducing unreliable validation signals into Gaussian-level filtering. Raising the threshold to 0.90 is less harmful, but its stricter activation condition still yields a DSR of 23/24 and slightly degrades all six continuous metrics relative to 0.80. This result is consistent with delayed activation of the protection and filtering mechanisms. Among the tested values, $\tau_{\text{val}} = 0.80$ best balances reference reliability and timely intervention.

Table 17: Performance under different clean-reference residual thresholds, averaged across Mip-NeRF360, Tanks & Temples, and Free.

τ_{l1}	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
0.06	28.29	0.839	0.245	23/24	29.52	0.849	0.211
0.08	**28.67**	**0.844**	**0.241**	**24/24**	**29.82**	**0.857**	**0.206**
0.10	27.86	0.826	0.255	20/24	28.95	0.835	0.221

Table 17 evaluates how closely the current rendering must match the clean reference within validated regions. Tightening τ_{l1} from 0.08 to 0.06 lowers DSR from 24/24 to 23/24 and produces modest degradation on both view sets. This stricter residual requirement may reject otherwise usable clean-reference regions and delay defense activation. By contrast, relaxing the threshold to 0.10 reduces DSR to 20/24, lowers PSNR by 0.81 dB on V-ILLUSORY and 0.87 dB on V-TEST, and increases LPIPS by 0.014 and 0.015, respectively. This stronger degradation is consistent with an overly permissive residual criterion that accepts noisy or inaccurate renderings as clean references and propagates unreliable signals to Gaussian-level filtering. Among the tested values, the main-experiment setting $\tau_{\text{l1}} = 0.08$ performs best across all reported metrics.

Table 18: Performance under different suspicious-ratio thresholds, averaged across Mip-NeRF360, Tanks & Temples, and Free.

τ_{sus}	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
0.20	28.05	0.824	0.254	23/24	29.12	0.835	0.218
0.30	**28.67**	**0.844**	**0.241**	**24/24**	**29.82**	**0.857**	**0.206**
0.40	27.79	0.815	0.267	21/24	29.28	0.839	0.213

Table 18 evaluates τ_{sus}, the normalized suspicious-region contribution ratio rather than an absolute photometric threshold. It determines whether a validated Gaussian has sufficiently limited suspicious-region involvement to enter the protected set and is applied unchanged across datasets and scene types. Tightening τ_{sus} from 0.30 to 0.20 lowers DSR to 23/24 and reduces V-TEST PSNR by 0.70 dB. This stricter criterion can exclude clean-support Gaussians with modest suspicious-region overlap, increasing the risk of removing useful scene content. Increasing τ_{sus} to 0.40 lowers DSR to 21/24, decreases V-ILLUSORY PSNR by 0.88 dB, and increases LPIPS by 0.026 relative to the 0.30 setting. The relaxed criterion admits Gaussians with greater suspicious-region contributions into the protected set, allowing more attack-related content to persist. Among the tested values, $\tau_{\text{sus}} = 0.30$ provides the most reliable separation between clean scene support and suspicious Gaussian contributions.

Table 19: Performance under different pruning intervals, averaged across Mip-NeRF360, Tanks & Temples, and Free.

Δ_{prune}	V-ILLUSORY				V-TEST		
	PSNR↑	SSIM↑	LPIPS↓	DSR↑	PSNR↑	SSIM↑	LPIPS↓
1500	28.18	0.826	0.251	22/24	29.03	0.834	0.219
3000	**28.67**	**0.844**	**0.241**	**24/24**	**29.82**	**0.857**	**0.206**
6000	27.84	0.818	0.263	19/24	29.31	0.840	0.212

Table 19 evaluates how often persistent suspicious Gaussians are removed during later optimization. Shortening Δ_{prune} from 3000 to 1500 iterations lowers DSR from 24/24 to 22/24 and decreases V-TEST PSNR by 0.79 dB. This behavior is consistent with pruning Gaussians before their accumulated statistics have stabilized, which can remove useful support for innocent-view reconstruction. Extending the interval to 6000 iterations preserves V-TEST quality better than the 1500-iteration setting but reduces DSR to 19/24 and produces the lowest V-ILLUSORY quality among the tested intervals. Infrequent pruning leaves suspicious Gaussians in the representation for longer and may therefore delay complete illusion removal. The main-experiment interval of 3000 iterations provides the best reported metrics and full DSR, balancing timely removal against the risk of pruning.

C Broader Impacts

This work studies the security and reliability of 3DGS under poisoned training data and introduces a defense against view-dependent poisoning. Although the injected content may be confined to a target view and remain inconspicuous in innocent views, such a localized failure can still affect downstream systems that use 3DGS for scene reconstruction, 3D mapping, localization, or autonomous navigation. ValidGS therefore offers broader benefits in robustness and trustworthiness for these reconstruction and perception pipelines. The problem is also security-sensitive. Systems trained on user-generated or weakly curated data can be manipulated to encode deceptive, view-dependent content, which may affect robotics, navigation, or digital content creation if left unaddressed. We present this work as a defensive study that characterizes this integrity risk and evaluates a concrete mitigation strategy, rather than as a method for deploying such attacks. More broadly, trustworthy 3DGS deployment requires attention to the integrity of training data and learned representations in addition to rendering

fidelity and computational efficiency. We hope this work motivates further research on secure 3D scene reconstruction, rigorous defensive evaluation, and reliable deployment practices for neural rendering systems.

D LIMITATION

During training, ValidGS maintains candidate target-view-specific reference and validation statistics and periodically applies Gaussian-level filtering and pruning. These operations do not incur substantial additional overhead in the evaluated settings, and pruning can reduce later optimization cost, but their memory and computation requirements have not been evaluated for substantially longer schedules or larger scenes. ValidGS also assumes access to the training images and camera poses needed for reference construction and continued optimization. On unpoisoned data, the conservative filtering and pruning introduce a small reconstruction-quality cost relative to vanilla 3DGS, reflecting the benign-data accuracy trade-off of the added protection. Future work should improve scalability to longer schedules and larger scenes, reduce reliance on training-view information where possible, and extend ValidGS to broader data-corruption settings while mitigating this benign-data accuracy cost.