

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/188812/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Liu, Jiucui , Xiong, Guanyu, Wang, Dalei, Chai, Chengzhang , Pan, Yue and Li, Haijiang 2026. Large-scale Gaussian splatting with semantic graph modeling for fine-grained bridge surface damage management. Automation in Construction 191 , 107128. 10.1016/j.autcon.2026.107128

Publishers page: <https://doi.org/10.1016/j.autcon.2026.107128>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Large-Scale Gaussian Splatting with Semantic Graph Modeling for Fine-Grained Bridge Surface Damage Management

Jiucan Liu^a, Guanyu Xiong^a, Dalei Wang^b, Chengzhang Chai^a, Yue Pan^{b,*}, Haijiang Li^{a,*}

a. BIM for Smart Engineering Centre, School of Engineering, Cardiff University, Cardiff, UK.

b. College of Civil Engineering, Tongji University, Shanghai, China.

* Corresponding author.

Abstract:

High-fidelity 3D reconstruction with precise defect representation is crucial for bridge inspection but remains challenging. Image-based methods often lack spatial awareness, while point clouds fail to capture fine surface textures. Standard Gaussian Splatting also struggles with large-scale reconstruction and detailed defect modeling due to global optimization and densification limitations. This paper presents a Gaussian Splatting framework for photorealistic and defect-sensitive reconstruction of large bridge structures with semantic embedding and hierarchical modeling. Panoramic imaging is adopted for initialization to ensure comprehensive coverage. A block-wise defect-aware densification strategy is introduced to selectively enhance Gaussian density in texture-rich regions while maintaining computational efficiency at scale. In addition, component and corrosion segmentation are incorporated into the Gaussian representation and organized through a graph-based structural model, enabling semantic querying and structured analysis of defect instances. Experiments on real-world bridge datasets demonstrate improved texture fidelity, more accurate defect representation, and enhanced semantic accessibility for inspection and maintenance tasks.

Keywords:

3D Gaussian Splatting; Drone; Image Segmentation; Point Cloud Segmentation; Panoramic Image; Corrosion Assessment; Semantic Graph.

1. Introduction

Efficient and accurate assessment of bridge surface damage is essential for ensuring the long-term performance and safety of transportation infrastructure [1]. Bridges, as vital connecting component within transportation system, are continuously subjected to environmental degradation and traffic-induced stresses, necessitating rapid inspection of surface conditions, high-fidelity reconstruction and reliable decision-making. Traditional visual inspections, while dominant, are labor-intensive, subjective, and difficult to integrate into digital maintenance workflows. These limitations have motivated the development of automated inspection methods like unmanned aerial vehicles (UAVs) [2,3] and wall-climbing robots [4].

Existing methods for bridge surface damage detection from acquired inspection data primarily rely on three paradigms: image-based methods leveraging 2D imagery and deep learning [5], point cloud-based techniques utilizing LiDAR [6,7], and multimodal fusion strategies combining both modalities [8]. Although each paradigm offers unique strengths, they suffer from inherent limitations: image-based approaches lack 3D spatial context for accurate localization [8], point-cloud-based methods fail to capture fine-scale surface deterioration such as cracks or early-stage corrosion [9], and existing fusion pipelines typically combine modalities only at post-processing stages [10], resulting in limited cross-modal interaction and inconsistent alignment.

Recent advances in neural rendering, particularly Neural Radiance Fields (NeRF) [11] and 3D Gaussian Splatting (3DGS) [12], provide a promising alternative for photorealistic 3D reconstruction from multi-view imagery. Compared to NeRF, which suffers from long training times and low rendering efficiency, 3DGS enables fast training (often below 30 minutes) and real-time rendering at 30+ FPS [34], making it particularly attractive for large-scale infrastructure reconstruction. However, directly applying standard 3DGS to structural damage assessment poses three challenges:

(1) Achieving dense view coverage for complex structural geometries. Bridge superstructures contain occluded, narrow, or internally confined regions that are difficult to capture using conventional UAV imagery. Insufficient viewpoint diversity leads to incomplete and degraded GS reconstruction, especially in shadowed or hard-to-access areas.

(2) Preserving fine-scale defect textures under large-scale reconstruction. When applied to extensive bridge spans, standard 3DGS must distribute limited computational and memory resources across a large spatial domain. This forces Gaussians to grow in size and lead to over-smoothed textures, reducing their ability to represent detailed surface texture such as corrosion or small-scale coating loss.

(3) Structuring and retrieving defect information for engineering workflows. While 3DGS offers photorealistic rendering, it lacks explicit semantic organization. Without structured representations of components, defect regions, and their spatial relationships, the reconstructed model cannot directly support engineering tasks like component-wise defect assessment or maintenance decision-making.

To address these limitations, this paper presents BridgeGS, an enhanced 3DGS-based framework tailored for large-scale bridge surface damage assessment. The framework integrates three key components:

- Panoramic imaging-driven GS initialization for comprehensive structural coverage. To overcome the difficulty of achieving coverage of complex bridge geometries, the framework employs panoramic cameras mounted on UAVs and handheld platforms to acquire dense omnidirectional imagery of both exterior and interior regions. A panoramic-to-perspective preprocessing strategy produces perspective sub-views compatible with GS training, enabling stable camera pose estimation and reliable GS initialization.
- Block-wise and texture-oriented GS densification. To address the scalability limitations of applying standard GS to large-scale bridges, the reconstructed scene is partitioned into spatial blocks and each block undergoes texture-oriented densification supported by two mechanisms: (i) block-specific ROI extraction to minimize the optimization bias introduced by background regions, and (ii) a local-texture-oriented densification mechanism that selectively increases Gaussian density in areas exhibiting high-frequency surface variations. This enables high-fidelity rendering of corrosion-relevant textures while maintaining computational efficiency.
- Semantic graph for structured damage management and retrieval. Beyond geometric reconstruction, the framework introduces a graph-based semantic layer that encodes components, corrosion regions, and their spatial relationships as attributed nodes and edges. This structure supports semantic querying, component- or defect-specific view synthesis, and integration with inspection workflows. By organizing corrosion information into a scalable and queryable form, the semantic graph facilitates fine-grained damage retrieval and long-term condition monitoring.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 details the proposed methodology. Section 4 presents a real-world bridge case study demonstrating the effectiveness of the framework. Section 5 concludes with key findings and future research directions.

2. Literature review

This section reviews existing approaches to bridge surface damage assessment and situates the proposed framework within the current literature. The discussion is organized into four parts according to the underlying data representation. Section 2.1 reviews image-based methods that rely on 2D imagery and multi-view reconstruction. Section 2.2 examines point cloud-based techniques that exploit direct geometric measurements. Section 2.3 surveys multimodal fusion strategies that combine appearance and geometry. Section 2.4 then focuses on neural rendering, with particular emphasis on 3D Gaussian Splatting and the open challenges it faces in large-scale, defect-sensitive reconstruction.

2.1. Image-based damage assessment

Image-based methods utilizing 2D imagery have gained significant traction in structural damage assessment, particularly with the rapid advancement of deep learning techniques. Recent developments have substantially improved damage detection in terms of segmentation accuracy [13], robustness under varying lighting conditions [14], incorporation of domain knowledge [15], real-time detection capabilities [16], and simultaneous recognition of multiple damage types [5,15]. When integrated with automated platforms such as drones, these technologies have considerably enhanced the automation level of structural damage quantification [17,18].

Despite these advances, image-based approaches face inherent limitations stemming from their reliance on single-view 2D imagery, which fundamentally constrains spatial reasoning and makes accurate 3D localization of damage difficult. An alternative paradigm within image-based methods reconstructs three-dimensional geometry directly from multi-view imagery using Structure-from-Motion (SfM) [19,20] and Multi-View Stereo (MVS) [21,22]. Owing to their low cost and minimal hardware requirements, SfM/MVS-based pipelines have gained traction in infrastructure inspection. Kassotakis et al. [19] employed SfM photogrammetry to analyze geometric uncertainties in arch structures, while Morgenthal et al. [21] developed a UAV-based inspection workflow integrating flight planning, surface reconstruction, and defect detection. Xu et al. [23] presented a distributed photogrammetric pipeline that partitions large-scale bridge reconstruction into sub-models, substantially improving computational efficiency. However, these purely image-based techniques remain sensitive to texture deficiency, repetitive patterns, and challenging lighting conditions, thereby limiting reconstruction stability and damage detectability in real-world scenarios. Moreover, the resulting outputs often manifest as sparse point clouds or dense point clouds severely contaminated by noise [24], limiting their suitability for high-precision surface condition assessment. Collectively, these limitations hinder the effective application of purely image-based methods in detailed bridge damage evaluation and decision-support workflows.

2.2. Point cloud-based damage assessment

Point cloud-based approaches have emerged as an alternative paradigm for structural damage assessment, with LiDAR and other 3D scanning technologies [8]. enabling direct acquisition of spatial data while inherently preserving geometric context [25]. Specialized algorithms have been developed for detecting and quantifying various types of structural damage from point clouds. For crack detection, researchers have proposed methods exploiting geometric discontinuities and local surface variations [7]. Volumetric assessment techniques have been developed to quantify material loss and surface degradation [28].

The primary advantage of point cloud-based methods lies in their direct representation of 3D geometry, which facilitates accurate spatial measurements and structural analysis without the

ambiguities inherent in 2D projections. This geometric fidelity makes them particularly suitable for tasks requiring precise dimensional information, such as deformation monitoring.

However, the absence of visual texture in raw point cloud data fundamentally limits the ability to detect fine-scale surface deterioration. Phenomena such as hairline cracks, surface discoloration, or early-stage corrosion, which are often visible in color imagery, cannot be reliably identified from geometry alone [26]. This limitation highlights the need for richer surface representations that integrate both geometric and appearance information.

2.3. Multimodal information fusion

Recognizing the complementary strengths of image-based and point cloud-based modalities, recent research has explored multimodal fusion strategies for structural damage analysis [27]. These approaches aim to leverage the geometric precision of point clouds alongside the rich appearance information captured in imagery.

One category of fusion methods integrates image-based damage detection with auxiliary spatial information through registration with 3D models. Zhao et al. [28] proposed registering UAV photogrammetric point clouds with Building Information Models (BIM) for 3D localization in concrete dams. Zeibak-Shini et al. [29] utilized terrestrial laser scanning (TLS) to capture damaged building facades and integrate them into BIM models. Isailović et al. [30] presented point cloud-based spalling detection with semantic enrichment of as-built Industry Foundation Class (IFC) models. However, these registration-based strategies remain critically dependent on stable GPS signals or high-quality BIM models, resources frequently unavailable in large-scale, real-world bridge environments.

Another category directly combines co-captured images and point clouds for integrated analysis. Lin et al. [9] proposed a workflow merging images and point clouds through 3D-to-2D and 2D-to-3D projections to support damage detection and mapping. Vetrivel et al. [31] developed a point-cloud-based method for post-disaster building assessment that delineates buildings from oblique-image point clouds and classifies surface gaps to distinguish damage from acquisition artefacts. Xiao et al. [32] introduced a UAV-based bridge inspection pipeline that combines 3D semantic segmentation of point clouds with 3D-to-2D projection to extract image regions of interest, followed by a deep-learning model for crack detection, significantly improving accuracy by filtering background distractions.

While these multimodal pipelines demonstrate the benefits of combining geometric and visual information, they typically fuse modalities only at the post-processing stage, limiting the depth of cross-modal interaction [8]. More critically, point clouds, being inherently sparse and irregular samples of the scene, are ill-suited for producing high-fidelity appearance representations. In regions with limited image overlap or weak texture, the reconstructed points become sparse and their RGB attributes noisy or incomplete, making it difficult to support photorealistic rendering or preserve fine-scale surface appearance. Projection-based fusion further amplifies these issues: occlusion, incomplete coverage, and acquisition noise introduce misalignment between 3D geometry and 2D imagery, leading to inaccurate

damage projections and degraded texture quality. As a result, the reconstructed 3D surfaces lack the visual fidelity needed for detailed inspection, reducing the robustness of downstream tasks such as damage retrieval [28], severity quantification [6,28], and maintenance planning [33].

2.4. Neural rendering for infrastructure assessment

Driven by the need for high-fidelity reconstruction that preserves surface appearance, neural rendering has recently emerged as a promising alternative for reconstructing 3D structures from multi-view imagery [34,35]. Two influential paradigms have dominated recent developments: Neural Radiance Fields (NeRF) [11] and 3D Gaussian Splatting (3DGS) [12].

NeRF represents scenes via neural networks conditioned on spatial coordinates and viewing directions, enabling photorealistic novel view synthesis. Kim et al. [36] explored NeRF for reconstructing small-scale bridge models in laboratory environments, enabling damage localization and segmentation. Luo et al. [35] developed a lightweight NeRF-based framework for post-earthquake building assessment, integrating automatic semantic segmentation with UAV imagery. While these studies underscore the potential of neural rendering for structural inspection, NeRF suffers from lengthy training times (often exceeding 12 hours) and limited rendering efficiency (1–5 FPS), hindering its scalability to large infrastructure projects.

In contrast, 3DGS models scenes as collections of view-adaptive Gaussian ellipsoids that can be directly rasterized, enabling training in under 30 minutes and real-time rendering at 30+ FPS [12,37]. These substantial performance advantages make 3DGS particularly attractive for large-scale and appearance-focused inspection scenarios. Beyond infrastructure applications, 3DGS has demonstrated versatility in geometry reconstruction [38,39], 4D rendering [40], and scene understanding [41], underscoring its potential as a general-purpose 3D representation. Recent work has begun exploring 3DGS for built environment applications. Liang et al. [34] employed 3DGS to digitize church buildings, embedding semantic features from multimodal large models to enable realistic rendering and semantic retrieval. However, their work primarily targets general digitalization and BIM-related applications without specific optimization for damage detection tasks.

Despite these promising developments, applying 3DGS to large-scale bridge damage assessment reveals several critical challenges that current methods fail to adequately address. At the foundational level, standard 3DGS encounters three inherent limitations when applied to structural damage representation. First, bridges feature intricate geometries, truss structures, arches, complex deck systems, posing significant challenges for comprehensive data capture. Inadequate coverage or poorly planned imaging paths lead to incomplete reconstructions. Second, scaling to large infrastructure introduces substantial computational demands while reducing the model's capacity to represent high-frequency signals, resulting in aliasing artifacts that diminish effectiveness in capturing fine-grained textures critical for damage assessment [42]. Third, existing 3DGS methods lack semantic information

management mechanisms aligned with AEC workflows, offering no structured representation of damage-related metadata such as component labels, damage categories, and spatial relationships.

To address scalability challenges, recent works [37,43,44] have developed large-scale 3DGS reconstruction methods, yet these solutions introduce new limitations for damage-critical applications. CityGaussian [37] introduces a divide-and-conquer training strategy combined with level-of-detail (LoD) representations to enable efficient rendering across vastly different spatial scales. CityGo [44] further adopts a hybrid representation that combines proxy geometry with residual and surrounding Gaussians, achieving lightweight and real-time rendering of urban scenes from aerial imagery. CityGaussianV2 [43] improves geometric accuracy and scalability through decomposed-gradient densification, depth regression, and parallelized training, striking a balance between visual quality and efficiency at the city scale. While these approaches demonstrate impressive scalability and rendering performance, their design objectives are primarily aligned with city-level visualization and photorealistic rendering. To achieve scalability, they rely on LoD aggregation, importance-aware downsampling, or proxy-based abstractions, which inherently smooth or absorb localized surface irregularities. As a result, small-scale defects that are critical for structural inspection, yet negligible in terms of global rendering contribution, are often underrepresented or treated as noise.

Similarly, efforts to improve deployability through model compression present fundamental conflicts with damage preservation requirements. LightGaussian [45] introduces a pruning-and-recovery strategy that identifies Gaussians with low global significance and applies knowledge distillation and vector quantization to reduce storage cost while preserving overall rendering quality. HAC [46] further advances this direction by incorporating structured hash-grid-based context modeling and entropy coding into the GS optimization process, enabling substantial compression ratios while maintaining fidelity. FlexGS [47] proposes an elastic inference framework that dynamically selects and transforms subsets of Gaussians to meet device-specific memory constraints without requiring retraining. Despite their effectiveness in reducing model size and improving rendering efficiency, these methods are fundamentally optimized for global rate-distortion trade-offs, where distortion is measured primarily by rendering or reconstruction error. As a result, Gaussian primitives corresponding to localized, spatially sparse, and visually subtle surface anomalies, such as corrosion or fine cracks, tend to have low statistical significance within the optimization objective. Even when compression is integrated into the optimization process, as in HAC [46], the learned representation remains biased toward preserving globally dominant structures and textures. Consequently, these methods do not explicitly guarantee the preservation of defect-sensitive information critical for infrastructure inspection.

Moreover, none of these frameworks provide mechanisms for embedding defect semantics or organizing damage information in a manner compatible with infrastructure lifecycle management. This limitation reduces practical utility for structural safety evaluation and digital operation workflows, as inspection outcomes must be systematically documented, queried, and integrated with existing asset management systems.

The present work addresses these interconnected limitations by developing an enhanced 3DGS framework specifically tailored for large-scale bridge damage assessment. Our approach integrates three key contributions: (1) comprehensive data acquisition strategies that ensure complete geometric coverage of complex bridge structures, (2) scalable high-fidelity reconstruction that explicitly preserves defect-sensitive features while maintaining computational efficiency, and (3) damage-oriented semantic information management aligned with AEC workflows for practical deployment in infrastructure lifecycle management.

3. Methodology

This section outlines the methodology of the proposed framework, as illustrated in Figure 1, which comprises four main stages: panoramic image-based GS initialization, block-wise and texture-oriented GS densification, semantic extraction, and component-level damage information management.

This section presents the methodology of the proposed framework, as illustrated in Figure 1, which provides an overview of the entire pipeline. The framework consists of four main stages: panoramic image-based GS initialization, block-wise and texture-oriented GS densification, semantic extraction, and component-level damage information management.

The pipeline starts with GS initialization, which serves as the data acquisition and model initialization stage. Specifically, a global Gaussian Splatting (GS) model is reconstructed from panoramic images, together with the estimation of accurate camera-GS pose relationships. This initialized GS model and the associated poses provide the geometric and photometric foundation for all subsequent stages.

The core contribution of this work lies in the block-wise and texture-oriented GS densification stage. First, the global GS representation of large-scale infrastructure is spatially partitioned into multiple blocks, and the associated images are grouped accordingly. This block-wise strategy significantly reduces the computational burden of GS training for each block, which is particularly critical when dealing with infrastructure-scale datasets containing a large number of images. By alleviating resource constraints, this design enables more detailed texture reconstruction.

Second, within each block, a texture-oriented GS densification strategy is employed. During GS optimization, image regions corresponding to background areas or regions irrelevant to the target components are filtered out to mitigate their negative impact on the optimization process. In addition, a high-frequency texture-aware loss term is incorporated into the objective function, encouraging the densification process to focus on fine-grained texture details that are essential for damage characterization.

The semantic extraction stage is designed to support damage localization and assessment by extracting two complementary types of semantic information: damage semantics and component

semantics. Damage semantics are obtained through image-based segmentation methods, while component semantics are derived using geometry-based segmentation applied to the GS representation.

Finally, to enable damage information management for as-is structures without BIM models, all extracted information is organized into a semantic graph-based representation. In this pipeline, structural components are modeled as graph nodes, and their spatial adjacency relationships are encoded as edges. Damage and semantic attributes are integrated as node properties, allowing the semantic graph to support advanced semantic querying and damage-oriented image rendering. This representation facilitates scalable and lifecycle-oriented damage management for large-scale infrastructure.

The following subsections describe each stage in detail.

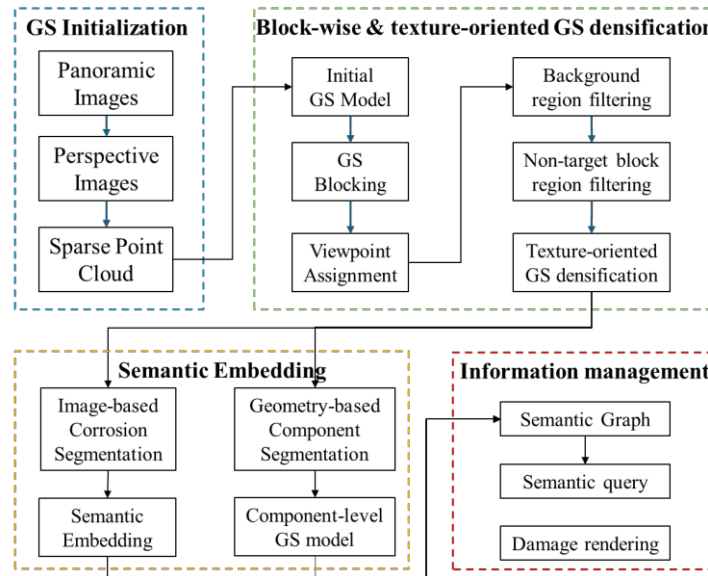


Figure 1 Proposed overall pipeline.

3.1. Panoramic image-based GS initialization

This stage establishes the geometric and photometric foundation for the subsequent reconstruction. It proceeds in two steps: Section 3.1.1 describes the collection of panoramic images and their preprocessing into perspective sub-views suitable for GS training, while Section 3.1.2 presents the reconstruction of the initial global GS model together with the estimation of the camera-GS pose relationships.

3.1.1. Panoramic image collection and preprocessing

Bridges exhibit highly complex spatial configurations, making comprehensive visual inspection challenging, particularly when relying on automated platforms such as UAVs that require precise flight-path planning and camera pose control to avoid occlusions and ensure sufficient coverage. Recent advances in panoramic imaging technology provide an effective alternative by enabling omnidirectional visual acquisition without elaborate orientation strategies, thereby reducing operational complexity while improving inspection completeness. Motivated by these advantages, this study explores the combined use of panoramic cameras with both UAV-based and manual inspection workflows to achieve

efficient and visually comprehensive documentation of bridge structures. As illustrated in Figure 2 (a), a UAV equipped with a panoramic camera is deployed to capture the exterior geometry through encircling flight paths. By contrast, due to safety management restrictions that prevent UAV operation within confined deck spaces, interior and deck-level views are acquired using a handheld panoramic camera carried along the bridge deck. This dual-mode acquisition strategy ensures uniform coverage of critical structural components and provides high-quality multi-view imagery that supports subsequent GS initialization and semantic processing.

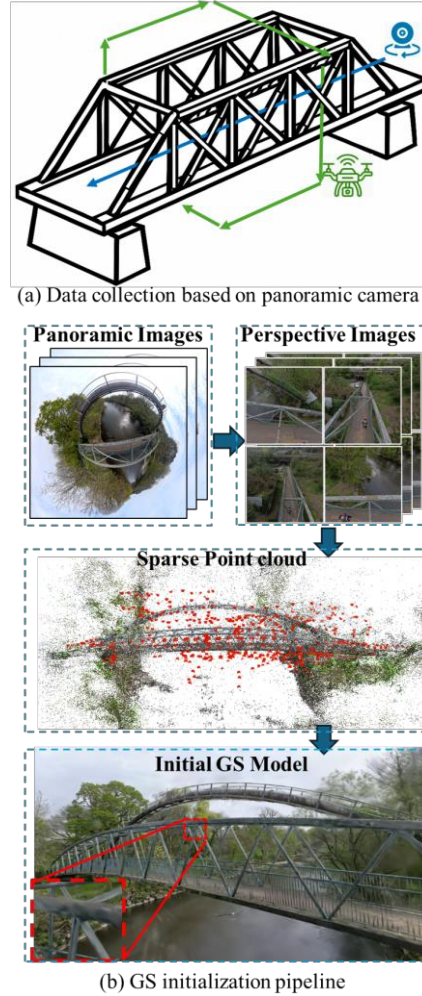


Figure 2 Pipeline of panoramic image-based GS initialization

However, panoramic imagery introduces a fundamental challenge to the standard GS reconstruction pipeline. Conventional GS generation relies on an initial Structure-from-Motion (SfM) stage, which estimates camera poses and a sparse 3D structure from overlapping views. SfM assumes a pinhole camera model to enforce consistent geometric constraints across images. In contrast, panoramic cameras employ equirectangular projection, which introduces severe non-linear distortions and deviates substantially from the pinhole assumption. As a result, feature correspondences become less reliable, camera pose estimation becomes unstable, and the overall 3D initialization deteriorates, ultimately compromising the quality of subsequent GS optimization.

To address this issue, a two-stage preprocessing strategy is adopted. In the first stage, each panoramic image $\tilde{I}$ is converted into a series of perspective images $\{I_1, I_2, \dots, I_m\}$ using fixed field-of-view (FOV) sampling over discrete view directions. These perspective views are more compatible with conventional SfM tools and yield more stable camera pose estimates.

Conceptually, as shown in Equation (1), a panoramic image can be viewed as an equirectangular projection of a spherical domain, where each pixel corresponds to a latitude–longitude pair similar to a world map unwrapped from the globe. Accordingly, each pixel (u, v) is first mapped to spherical angles (λ, φ) , which are then expressed in Cartesian coordinates through spherical-to-Cartesian projection:

$$\begin{cases} x = \cos(\varphi)\sin(\lambda) \\ y = \sin(\varphi) \\ z = \cos(\varphi)\cos(\lambda) \end{cases} \quad (1)$$

To extract a view oriented in a desired direction, the corresponding 3D ray is rotated using a composite rotation matrix R , defined by yaw and pitch rotations as shown in Equation (2)~(4):

$$R_{yaw} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2)$$

$$R_{pitch} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 1 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3)$$

$$R = R_{yaw} \cdot R_{pitch} \quad (4)$$

where θ and ϕ denote the yaw and pitch angles in radians, respectively. The rotated vector are subsequently transformed to spherical coordinates through:

$$\lambda = \arctan(x/z) \quad (5)$$

$$\varphi = \arcsin\left(\frac{y}{\sqrt{x^2 + y^2 + z^2}}\right) \quad (6)$$

Finally, the spherical coordinates are then mapped to pixel coordinates (u, v) in the original equirectangular panoramic image via:

$$u = \left(\frac{\lambda}{2} + 1\right) \frac{W}{2} \quad (7)$$

$$v = \left(\frac{2\varphi}{\pi} + 1\right) \frac{H}{2} \quad (8)$$

where W and H represent the width and height of the panoramic image. Bilinear interpolation is employed to sample the original image at these computed coordinates, yielding the desired perspective projection views.

In the second stage, the SfM process is applied to the perspective images $\{I_i\}$, recovering relative camera poses and a sparse 3D point cloud. Specifically, the SfM output includes:

- (i) camera intrinsics $\{K_i\}_{i=1}^N$ with $K_i \in \mathbb{R}^{3 \times 3}$;
- (ii) camera extrinsics $T = \{R_i | t_i\}_{i=1}^N$, where $R_i \in \mathbb{R}^{3 \times 3}$ and $t_i \in \mathbb{R}^3$; and
- (iii) a sparse 3D point set $P = \{p_j\}_{j=1}^M$, where $p_i \in \mathbb{R}^3$.

The projection of a 3D point p_j onto image I_i is defined as:

$$\lambda_{i,j} \begin{bmatrix} u_{i,j} \\ v_{i,j} \\ 1 \end{bmatrix} = K_i [R_i \quad t_i] \begin{bmatrix} p_j \\ 1 \end{bmatrix} \quad (9)$$

where $\lambda_{i,j} \in \mathbb{R}^+$ is a depth-related scalar and $(u_{i,j}, v_{i,j})$ is the 2D image coordinate of the projection of p_j in image I_i . These projection relationships define a set of 2D–3D correspondences, which will be utilized in following GS training.

3.1.2. GS initialization

Following the SfM stage, the Gaussian Splatting model is initialized using the recovered camera parameters and corresponding image frames. Following prior work[48], the scene is represented as a set of anisotropic Gaussian primitives:

$$g_j = (p_j, \Sigma_j, c_j, \alpha_j, f_j) \quad (10)$$

where $p_j \in \mathbb{R}^3$ denotes the 3D center, $\Sigma_j \in \mathbb{R}^{3 \times 3}$ is the covariance matrix controlling the shape and orientation, $c_j \in \mathbb{R}^3$ is the RGB color, $\alpha_j \in \mathbb{R}$ is the opacity and $f_j \in \mathbb{R}^d$ is a view-dependent feature vector. These parameters are jointly optimized to minimize the photometric reconstruction loss between rendered images $\{\hat{I}_i\}_{i=1}^N$ and captured images $\{I_i\}_{i=1}^N$.

$$L = \sum_{i=1}^N \|\hat{I}_i - I_i\|^2 \quad (11)$$

As conventional GS pipelines are not designed for large-scale infrastructure scenes, naïve training often leads to optimization instability and loss of fine-scale texture crucial for damage assessment. This motivates the block-wise GS refinement strategy introduced in the next section, enabling scalable training while preserving high-frequency surface details.

3.2. Block-wise and texture-oriented GS densification

Large-scale bridge reconstruction typically involves thousands of panoramic-derived images and a very large number of Gaussian primitives. Directly optimizing a global GS model under these conditions is computationally expensive and often unstable, making it difficult to preserve the fine-grained textures required for damage assessment. To address these challenges, we adopt a two-stage GS densification strategy, as illustrated in Figure 3.

In the first stage, a ‘divide-and-conquer’ strategy is employed in which the global GS representation is partitioned into multiple spatial blocks, and the corresponding input images are grouped according to their block assignment. This decomposition substantially reduces the optimization scale for each sub-region, leading to improved convergence and reconstruction stability. A visual illustration of this block-wise partitioning is provided in Figure 7.

In the second stage, texture-oriented GS densification is performed independently for each block, where two key enhancements are introduced. First, region-of-interest (ROI) extraction is applied to all

images associated with a given block at two levels. At the global level, the bridge structure is separated from the background regions. At the block level, only the image regions that spatially correspond to the current block are retained, while irrelevant regions are excluded. This dual-level ROI filtering restricts the optimization to the spatial extent of the target block, enabling the GS training to focus on local geometry and appearance. An example of this process is shown in Figure 9.

Second, a texture-aware densification procedure is incorporated to increase the representational capacity around high-frequency surface details, which is essential for reliable characterization of corrosion and other fine-scale damage.

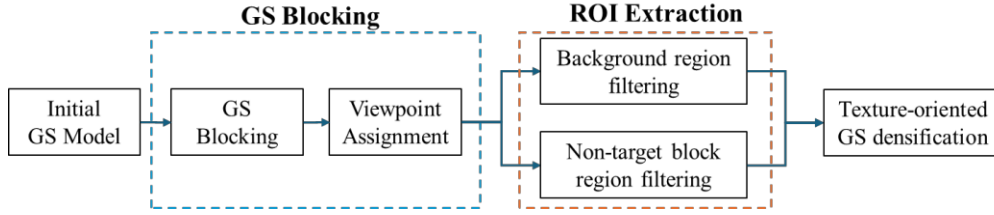


Figure 3 Process of Block-wise Training

3.2.1. GS blocking and viewpoint assignment

In this work, we employ a ‘divide-and-conquer’ approach on top of the initialized GS. This design reduces the computational burden of large-scale scene reconstruction while enhancing GS fidelity during optimization, thereby mitigating issues such as blurring and artifacts.

Formally, let the entire GS scene be represented by the set $G = \{g_i \mid i = 1, \dots, N\}$ where g_i denotes a Gaussian primitive. We first partition G into K spatial blocks $\{B_k\}_{k=1}^K$ according to their 3D coordinates using a bounding-box-based segmentation: $B_k = \{g_i \in G \mid p_i \in X_k\}$, where p_i is the center coordinate of g_i , and X_k is the axis-aligned bounding box of block k . After segmentation, the associated images and viewpoints are classified to match each block for subsequent block-based optimization. Let $V = \{v_j \mid j = 1, \dots, M\}$, denote the set of camera viewpoints. The assignment of viewpoints to blocks follows two criteria:

- Coordinate-based assignment:

Spatial proximity between the camera center and the block is used as the primary criterion. Specifically, a viewpoint v_j with camera center c_j is assigned to block B_k if c_j lies within the block’s bounding box X_k . In bridge inspection scenarios, X_k may represent the structural envelope of a segment (e.g., the deck and enclosed truss region of a truss bridge), so any camera center within this region is included in V_k for optimization.

- Projection-based assignment:

Some viewpoints lie outside the spatial range of a block B_k yet still contain important visual information about that block. To identify such viewpoints, we assess their visual relevance through an opacity-based rendering test. For a candidate viewpoint v_j , we render an image $\hat{I}_j$ in which all Gaussians belonging to B_k are set to zero opacity, while Gaussians from all other blocks remain unchanged. This ensures that regions corresponding to B_k are completely removed from the rendered image.

If suppressing B_k causes little difference between $\hat{I}_j$ and the original ground-truth image I_j , then B_k contributes minimally to that viewpoint, and v_j should not be associated with B_k . Conversely, if the removal of B_k leads to a large discrepancy between the two images, the viewpoint must have observed B_k extensively, and thus should be included in its optimization.

To quantify this effect, we measure the similarity between $\hat{I}_j$ and I_j using the Structural Similarity Index (SSIM), formulated in Equation (12):

$$SSIM(\hat{I}_j, I_j) = \frac{(2\mu_{\hat{I}_j}\mu_{I_j} + c_1)(\sigma_{\hat{I}_j I_j} + c_2)}{(\mu_{\hat{I}_j}^2 + \mu_{I_j}^2 + c_1)(\sigma_{\hat{I}_j}^2 + \sigma_{I_j}^2 + c_2)} \quad (12)$$

where μ_{I_j} and $\mu_{\hat{I}_j}$ denote the mean pixel values of images I_j and $\hat{I}_j$ respectively. The standard deviations are represented by σ_{I_j} and $\sigma_{\hat{I}_j}$, while $\sigma_{\hat{I}_j I_j}$ indicates the covariance between the two images. The constants c_1 , c_2 , and c_3 are stabilizing parameters that prevent division by near-zero values. A viewpoint is assigned to block B_k if:

$$SSIM(\hat{I}_j, I_j) \leq \tau_0 \Rightarrow v_j \in B_k \quad (13)$$

where τ_0 is a predefined similarity threshold. Viewpoints meeting the threshold are considered to substantially observe the target block and are included in its optimization, while those below the threshold are excluded.

This projection-based assignment ensures that each block incorporates not only spatially proximal viewpoints but also distant viewpoints that nonetheless observe the block prominently. In doing so, the global optimization problem is decomposed into multiple smaller local tasks that can be executed independently or parallelized, reducing computational overhead while improving the fidelity of block-wise reconstruction.

3.2.2. Block-specific ROI extraction

Although GS blocking and viewpoint grouping are completed, each block must be optimized with high fidelity, particularly with respect to surface textures of the target structure. This requires addressing two challenges. First, as shown in Figure 9 (a), although images have been grouped by block, they still contain substantial background or off-block regions. Training on these irrelevant regions dilutes optimization capacity and can introduce spurious artifacts. Second, as noted in prior works [42,49], individual Gaussians have limited ability to model high-frequency texture variations. Since superficial bridge damage often manifests as fine-scale surface changes, a targeted densification mechanism is necessary to enhance the representational capacity around local textures.

To address these challenges, we introduce two complementary strategies: (i) block-specific ROI extraction, which ensures that optimization focuses exclusively on structurally relevant areas; (ii) local-texture-oriented Gaussian refinement, which improves the ability of Gaussians to encode high-frequency damage-related features. The rest of this subsection details the ROI extraction process; the local-texture-oriented Gaussian refinement will be presented in Section 3.2.3.

The goal of block-specific ROI extraction is to ensure that each GS block is refined using only the image regions that both spatially and semantically correspond to that block. As illustrated in Figure 9, this is achieved through a two-stage filtering pipeline that explicitly addresses potential segmentation errors and block-level misalignment.

Stage 1: Semantic region extraction via SAM.

We first apply the Segment Anything Model (SAM) [50] to each input image independently. SAM is chosen because it provides robust, annotation-free segmentation across diverse viewpoints, illumination conditions, and background complexities, conditions commonly encountered in bridge inspection scenarios. Instead of producing a single mask, SAM generates a set of candidate region masks, each corresponding to a different coherent semantic or geometric region in the image. However, these masks are category-agnostic and do not explicitly indicate which regions belong to the bridge. Therefore, the resulting set of masks $\{M_1^{(l)}\}$ must undergo an explicit selection step to isolate those corresponding to bridge structures. Examples of the SAM-generated candidate masks are shown in Figure 9 (b).

Stage 2: Block-aware filtering via GS-based opacity rendering.

To identify which SAM mask corresponds to the target block, we obtain a block-specific projection mask using GS rendering. For each training viewpoint assigned to block B_k , we render the scene while retaining only the Gaussians in B_k and suppressing all others. This produces an opacity map whose high-opacity pixels represent the visible footprint of block B_k . A binary mask M_2 is then generated via thresholding:

$$M_2(x, y) = \begin{cases} 1, & \alpha(x, y) > \tau_1 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $\alpha(x, y)$ denotes the accumulated Gaussian opacity at pixel (x, y) , and τ is a predefined threshold. Figure 9 (c) and (d) illustrate examples of M_2 generated from two adjacent blocks viewed from the same camera pose.

Because of the ellipsoidal nature of Gaussians, the boundaries of M_2 are not perfectly sharp but provide an effective approximation of the block-specific footprint. Intersecting the SAM mask with the GS mask effectively removes irrelevant areas and yields a high-quality ROI containing only the image regions associated with the target block. The final ROI mask for each image is obtained by intersecting the two masks:

$$M_{Final} = M_1 \cap M_2 \quad (15)$$

Figure 9 (e) and (f) show the resulting filtered regions used for training. This two-stage process restricts optimization to the portions of the image that correspond exclusively to the target block, thereby improving both computational efficiency and the quality of block-wise refinement.

3.2.3. Local-texture-oriented GS refinement

While ROI extraction localizes the optimization spatially, it does not fully resolve the intrinsic representational limitation of individual Gaussians [42,49]. As described in Sec. 3.1.2, each Gaussian $g \in Gg$ is parameterized as a 3D ellipsoid with attributes including color c , opacity α , and view-

dependent shading modeled by spherical harmonics $H(\theta, \phi)$. Rendering is performed via α -blending across Gaussians, while optimization iteratively updates these parameters to minimize the reconstruction error [48]. However, when covering large spatial extents under memory constraints, the model inevitably relies on relatively large Gaussians to span broad regions. This is suboptimal, as large Gaussians cannot faithfully capture fine-grained textures due to the inherently low-frequency expressiveness of spherical harmonics.

To further enhance fine-detail reconstruction, a local-texture-guided densification mechanism is introduced, consisting of two complementary components: loss-driven texture awareness and progressive GS densification.

(1) Loss-driven Texture Awareness

The objective of this component is to explicitly guide the optimization toward regions containing high-frequency texture content. In GS, each Gaussian encodes only limited local detail, and reconstruction is primarily supervised by RGB-intensity differences. However, RGB fidelity alone is insufficient to recover superficial damage features because distinct texture patterns may share similar average color values, leading to ambiguity in areas with corrosion, stains, or fine surface irregularities.

To remedy this limitation, we introduce an image-gradient consistency term. Image gradients capture the magnitude of local intensity changes and therefore serve as an effective descriptor of texture variation. Regions with high-frequency patterns, such as corrosion edges, exhibit strong gradient responses, whereas smooth or uniform regions produce weak gradients. Incorporating gradient information into the loss function strengthens supervision over these texture-rich areas and encourages Gaussians to better reproduce fine structural details. For a ground-truth image $I_{gt} \in \mathbb{R}^{H \times W \times 3}$ and a rendered prediction $I_{pred} \in \mathbb{R}^{H \times W \times 3}$, the gradient field is computed via Sobel filtering:

$$\nabla I(x, y) = \sqrt{(\nabla_x I(x, y))^2 + (\nabla_y I(x, y))^2 + \epsilon} \quad (16)$$

where ϵ is a small constant for numerical stability. This yields the gradient reconstruction loss as:

$$L_{grad} = \|\nabla I^{gt} - \nabla I^{pred}\|^2 \quad (17)$$

To balance global color fidelity with perceptual similarity, the gradient term is combined with L_{SSIM} :

$$L = L_{SSIM} + \lambda_{grad} L_{grad} \quad (18)$$

Here, λ_{grad} is a weight to balance global fidelity against local texture preservation. This formulation ensures that regions with strong gradient variations, indicative of high-frequency details such as cracks or corrosion, are explicitly emphasized during optimization.

(2) Progressive GS Densification

Although the texture-awareness loss introduced in the previous section effectively guides optimization toward texture-rich regions, it does not on its own increase the representational capacity of

the model. Under realistic memory and runtime constraints, it is infeasible to indefinitely increase the number of Gaussians or prolong training. Consequently, densification must be both selective and structurally aware: additional Gaussians should be allocated primarily to regions with complex local textures, while also avoiding oversized or geometrically unstable primitives that can degrade rendering quality.

To this end, we introduce a progressive densification mechanism governed by three complementary parameters, the footprint radius r_j , the gradient-based importance γ_j , and the aspect ratio a_j . These parameters jointly determine whether a Gaussian g_j should be subdivided, balancing texture adaptivity, spatial compactness, and geometric regularity.

For footprint radius r_j : as depicted in Sec.3.1.2, each Gaussian g_j is parameterized by its covariance matrix Σ_j , which defines both its spatial extent and orientation. The effective image-space footprint radius r_j , is derived from the eigenvalues of Σ_j , reflecting the projected scale of the Gaussian ellipsoid:

$$r_j = (\det(\Sigma_j))^{1/6} = (\lambda_{1j}\lambda_{2j}\lambda_{3j})^{1/6} \quad (19)$$

where $\lambda_{1j}\lambda_{2j}\lambda_{3j}$ are the eigenvalues of Σ_j corresponding to the principal axes of the Gaussian ellipsoid. This formulation captures the overall spatial size of the Gaussian while accounting for its shape and orientation. When r_j becomes excessively large, a single Gaussian spans a wide region and fails to provide sufficient spatial resolution for fine textures. To regulate this behavior, we introduce a size threshold τ_{size} . If $r_j > \tau_{size}$, the Gaussian is considered overly coarse and is subdivided to increase local representational density.

For Gradient-Based Importance γ_j : to identify regions containing rich local textures, each Gaussian is assigned a gradient-based importance score γ_j , which quantifies the textural discrepancy between the predicted and ground-truth images over the Gaussian’s projected region Ω_j :

$$\gamma_j = \frac{1}{|\Omega_j|} \sum_{(x,y) \in \Omega_j} \|\nabla I^{gt}(x,y) - \nabla I^{pred}(x,y)\|^2 \quad (20)$$

A higher γ_j indicates a larger gradient-domain reconstruction error in the projected region, reflecting insufficient representational capacity of the current Gaussian rather than the intrinsic texture strength of the surface. Consequently, even visually smooth regions with subtle corrosion or weathered materials can exhibit elevated γ_j values if the rendered gradients deviate from the ground truth, ensuring that such regions are not overlooked during densification. To ensure that under-reconstructed regions receive higher representational capacity, we define a texture threshold τ_{grade} . When $\gamma_j > \tau_{grade}$, the Gaussian is deemed inadequate for modeling local surface detail and is subdivided to improve its expressive capability.

For Aspect Ratio a_j : In addition to texture awareness, maintaining geometric regularity is important to prevent over-stretched Gaussians that may introduce rendering artifacts. The anisotropy of each Gaussian is measured by its aspect ratio:

$$a_j = \frac{\max(\lambda_{1j}, \lambda_{2j}, \lambda_{3j})}{\min(\lambda_{1j}, \lambda_{2j}, \lambda_{3j})} \quad (21)$$

A large a_j indicates a highly anisotropic and potentially unstable primitive. When $a_j > \tau_{shape}$, the Gaussian is considered geometrically unreliable and is subdivided to improve spatial stability.

At each optimization iteration, the thresholds for the three parameters, τ_{size} , τ_{grad} , and τ_{shape} , are adaptively determined from the statistical distributions of all visible Gaussians:

$$\tau_{grad} = \mu_\gamma + \beta\sigma_\gamma, \quad \tau_{size} = \mu_r + \beta\sigma_r, \quad \tau_{shape} = \mu_a + \beta\sigma_a \quad (22)$$

where μ and σ denote the mean and standard deviation of each parameter's distribution, and β controls the sensitivity of refinement. A Gaussian g_j is marked for subdivision if it satisfies any of the following conditions:

$$r_j > \tau_{grad}, \quad \gamma_j > \tau_{size}, \quad a_j > \tau_{shape} \quad (23)$$

This statistically balanced formulation ensures that densification is selectively directed toward texture-rich, spatially large, or geometrically unstable GSs, enhancing representational precision without excessive memory growth. The subdivision process itself follows the methodology established in prior GS research [12], to which interested readers are referred for implementation details.

In summary, block-specific ROI extraction confines the optimization to structurally relevant regions within each block, preventing wasted computation on background or off-block content. Complementarily, the local-texture-oriented GS refinement adaptively increases representational capacity where fine-scale details are present, ensuring faithful reconstruction of surface textures critical for damage assessment. Together, these two mechanisms enable block-wise GS optimization to remain both targeted and texture-preserving, thereby enhancing the reliability of large-scale surface reconstruction for downstream structural inspection and damage quantification tasks.

3.3. Graph-based corrosion damage management

Although the reconstructed GS model provides a high-fidelity three-dimensional representation of the bridge superstructure, it inherently lacks semantic information, which limits its capacity to support damage assessment and maintenance decision-making. To overcome this limitation, an integrated semantic enrichment pipeline is developed to couple geometry-based component segmentation, image-based corrosion segmentation, and graph-based information management. The proposed pipeline consists of three main stages:

(1) Component segmentation, which partitions the GS reconstruction into bridge members based on geometry information;

(2) Image-based corrosion segmentation, which applies a fine-tuned image segmentation network to detect corrosion regions directly from the images and subsequently projects the identified corrosion masks onto the GS representation; and

(3) Graph-based information management, which organizes the extracted geometric and semantic information into a unified knowledge graph for efficient querying and semantic-guided rendering.

The following subsections describe each stage in detail.

3.3.1.Component segmentation

Although the GS model yields a detailed geometric representation of the bridge, structural assessment in engineering practice is inherently component-oriented. Consequently, component-level semantic decomposition is required to transform the raw GS reconstruction into a representation that aligns with established inspection and asset-management workflows. To achieve this, a geometry-driven segmentation approach is adopted to partition GS model into structural components based on spatial distribution, connectivity, and principal orientation.

The proposed segmentation strategy operates in two stages: (1) direction-constrained clustering to identify principal orientations of components, and (2) spatial refinement to delineate individual components along these orientations. This hierarchical approach leverages the inherent geometric regularity of bridge structures, where components typically align with a limited set of dominant directions.

Stage 1: Direction-Constrained Clustering

As shown in **Algorithm 1**, the first stage establishes directional constraints by identifying principal structural orientations. The Gaussian splat centers are first sampled to form a point cloud representation, which undergoes voxel-based downsampling to reduce computational complexity. Initial spatial clustering via density-based clustering algorithm (DBSCAN) [51] separates the point cloud into distinct groups, each representing potential structural components. Principal Component Analysis (PCA) is then applied to each cluster to extract its dominant orientation vector. These orientation vectors are subsequently grouped using angular similarity metrics, where clusters with principal directions within a specified threshold are consolidated. This procedure yields a compact set of principal structural directions that abstract the overall geometric configuration and provide directional constraints for subsequent refinement. The complete procedure is detailed in Algorithm 1.

Stage 2: Spatial Refinement and Component Extraction

Building upon the directional grouping, the second stage performs spatial refinement to extract individual structural components following Algorithm 2. Within each direction-constrained group, spatial subclustering identifies geometrically coherent regions. Linear approximations are then fitted to these subclusters to represent their spatial alignment. Intersection points between fitted lines define natural segmentation boundaries for curved or joint-like components, while proximity-based assignment

612 allocates remaining points to their nearest linear component. This refinement process, formalized in
 613 Algorithm 2, distinguish the GS representation into structural components, which can support
 614 component-wise damage assessment.

615 The generalizability of this pipeline allows it to be adapted to different structural systems
 616 beyond the studied case, providing a scalable approach for semantic embedding in 3D reconstruction.

Algorithm 1: Direction-Constrained Point Cloud Segmentation

Input:

$P \in \mathbb{R}^{N \times 3}$: Input point cloud,
 v_{size} : Voxel size for downsampling,
 θ_{th} : Angular threshold in degrees for direction similarity,
 $min_{samples}$: Minimum samples for DBSCAN,
 ϵ : DBSCAN clustering radius for initial segmentation.

Output:

$G = \{G_1, G_2, \dots, G_k\}$: Direction-based grouped clusters,
 $D = \{d_1, d_2, \dots, d_k\}$: Average principal directions for each group.

Procedure:

Initialize:

Apply voxel downsampling to obtain a reduced set of points:

$$P' = \text{VoxelDownSample}(P, v_{size})$$

Initial Clustering:

Apply *DBSCAN* clustering:

$$C = \{C_1, C_2, \dots, C_k\} = \text{DBSCAN}(P', \epsilon, min_{samples})$$

Principal Direction Computation:

For each $C_i \in C$:

Compute covariance matrix $Cov_i = Cov(C_i^T)$

Perform eigen-decomposition $Cov_i = V_i \Lambda_i V_i^T$, where $\Lambda_i = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ and $\lambda_1 > \lambda_2 > \lambda_3$

Identify the principal direction: $v_i = u_{max}$ such that $\lambda_{max} = \max(\lambda_1, \lambda_2, \lambda_3)$

Direction-based Grouping:

Construct angular distance matrix for principal directions $M \in \mathbb{R}^{n \times n}$ where $M_{i,j} = 1 - |v_i \cdot v_j|$

Convert threshold: $\epsilon_{angle} = 1 - \cos(\text{deg2rad}(\theta_{th}))$

Apply *DBSCAN* on C with distance matrix M and radius ϵ_{angle} to obtain direction labels $\{l_1, l_2, \dots, l_n\}$

Group Formation:

For each direction group l in $\{l_1, l_2, \dots, l_n\}$,

Create direction group $G_l = \{C_i | \text{direction}(C_i) = l\}$

Compute average direction $d_l = \frac{1}{|G_l|} \sum_{C_i \in G_l} v_i$

Return: Direction-grouped clusters G and average directions D

617

Algorithm 2 Direction-Constrained Spatial Refinement

Input:

$G = \{G_1, G_2, \dots, G_k\}$: Direction-based grouped clusters from Algorithm 1,
 $D = \{d_1, d_2, \dots, d_k\}$: Predefined normalized direction vectors from Algorithm 1,
 ϵ : Spatial clustering radius

Output:

$\{S_1, S_2, \dots, S_m\}$: Segmented linear components

Procedure:

Direction Inheritance:

For each direction group $G_j \in G$, directly use the associated direction vector $d_j \in D$ from Algorithm 1.

Spatial Clustering:

Within each group G_j , apply DBSCAN with radius ϵ to extract spatial subclusters.

Line Fitting and Merging:

For each subcluster,

Perform PCA line fitting to obtain line L_k with centroid c_k , direction v_k , and range $[p_{min}, p_{max}]$.

Merge collinear subclusters if their fitted lines are directionally aligned and partially overlapped.

Point Assignment:

Assign remaining unclustered points to the nearest line segment L_k based on perpendicular distance and inclusion within range $[p_{min}, p_{max}]$.

Return: Segmented linear components $\{S_1, S_2, \dots, S_m\}$

3.3.2. Image-based corrosion segmentation

While component segmentation establishes the geometric basis for damage assessment, evaluating the surface condition of each component further requires accurate corrosion detection. To this end, we introduce a 2D–3D integrated segmentation strategy that leverages the GS framework’s explicit mapping between image pixels and 3D Gaussian primitives, enabling seamless transfer of corrosion semantics from image space to the 3D representation. As illustrated in Figure 4, the proposed corrosion segmentation workflow operates in three stages: multi-view rendering, corrosion semantic segmentation, and 3D projection.

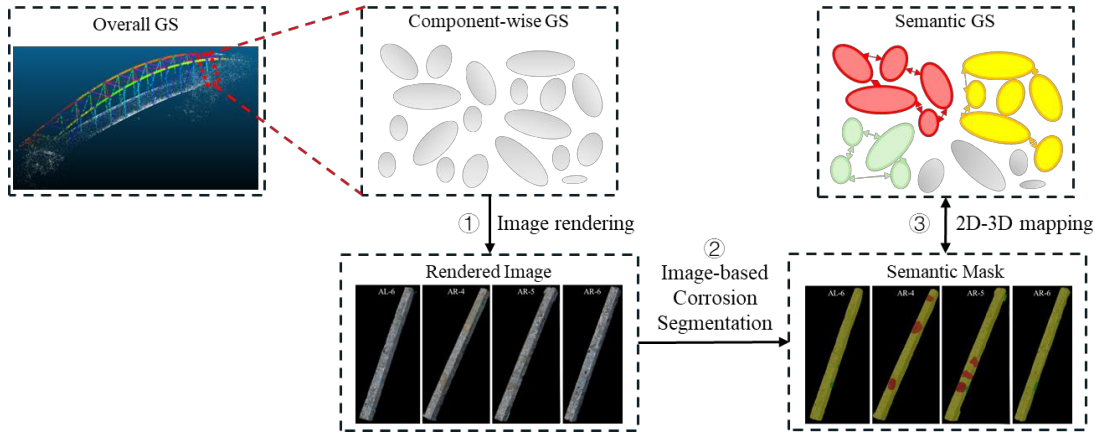


Figure 4 Pipeline of corrosion segmentation on GS model

First, for each segmented component, the corresponding GS model is rendered from multiple viewpoints to comprehensively capture surface appearance. These viewpoints are computed using an Oriented Bounding Box (OBB)-based camera placement strategy shown in Algorithm 3, which positions cameras perpendicular to the principal faces of the component’s OBB at a specified inspection distance. This ensures complete coverage while maintaining consistent viewing geometry across components.

Algorithm 3 Viewpoint Generation

Input:

P : Input point cloud,
 d : Inspection distance of the camera,
 (fov_x, fov_y) : field of view
 ϵ : Spatial clustering radius

Output:

$V = \{v_1, v_2, \dots, v_6\}$: Six-view camera configurations

Procedure:

Point Cloud Preprocessing:

Load the point cloud P from file and remove outlier to obtain a denoised point cloud P' .

OBB Estimation via Plane Extraction:

Extract multiple dominant planes from P' using RANSAC.

Denote the resulting OBB as B .

Viewpoint Generation from OBB Faces:

Generate six face-based viewpoints $\{F_1, F_2, \dots, F_6\}$, from OBB B , where each face provides:

- camera origin O_i ,
- look-at target t_i ,
- local right vector x_i ,
- local up vector y_i .

Camera Pose Construction:

For each face-based view F_i :

Compute forward vector $f_i = \text{normalize}(t_i - o_i)$

Construct camera rotation matrix $R_i = [x_i, y_i, f_i]$.

Construct camera-to-world transform $C2W_i = [R_i \mid o_i]$.

Compute world-to-camera transform $W2C_i = C2W_i^{-1}$.

Extract rotation $R_i^{(w2c)}$ and translation $T_i^{(w2c)}$.

Compute projection matrix P_i .

Form full projection transform $T_i^{proj} = W2C_i \cdot P_i$

create a camera record for GS rendering $v_i = \{R_i^{(w2c)}, T_i^{(w2c)}, fov_x, fov_y, o_i, T_i^{proj}\}$

Return: Viewpoints $V = \{v_1, v_2, \dots, v_6\}$

The rendered images are subsequently processed using a fine-tuned semantic segmentation model to generate pixel-level corrosion masks. The segmentation framework was tailored specifically for corrosion detection on bridge surfaces, addressing a significant domain adaptation challenge. Geometric patterns of surface damage in the investigated structures differ substantially from those in widely used open-source datasets [15], necessitating a transfer learning approach. The model was initialized with a DeepLabv3+ [52,53] backbone to leverage robust feature extraction capabilities, then fine-tuned on domain-specific data to capture characteristic corrosion patterns.

The framework classifies three surface condition categories: healthy areas (green), risk areas affected by biological colonization (yellow), and high-risk corroded surfaces (red), as shown in Figure 4. To address the inherently uneven distribution of damage across bridge structures, data augmentation was applied through strategic extraction of representative image patches, enhancing both diversity and coverage of training samples. Dataset configuration, training protocols, and quantitative performance evaluation are detailed in Section 3.3.1.

Following the generation of 2D semantic masks, these masks are embedded into the 3DGS representation through a semantic embedding procedure following the Gaga method [48], as exemplified in Figure 23. This procedure establishes a precise one-to-one correspondence between image pixels and their associated 3D Gaussian primitives through the pixel-to-Gaussian mapping derived from the splatting process. By aggregating semantic information from multiple viewpoints, each Gaussian element is assigned corrosion-related attributes based on its visibility and semantic label consistency across different views.

The resulting semantically enriched GS models of individual components are then assembled to form an integrated bridge-level representation shown in Figure 23 (b), yielding a unified 3D damage model where spatial distribution of deterioration can be explicitly analyzed. This bottom-up evaluation approach enables structural damage assessment directly within the 3D domain, bridging the gap between image-based detection and geometry-based structural analysis.

3.3.3. Graph-based structural condition assessment

After constructing the component-wise semantic GS model, an important challenge is how to organize defect-related information in a structured manner that supports damage assessment, semantic querying, and targeted rendering. To address this, we propose a graph-based information management framework tailored to infrastructure inspection scenarios.

At the first level, following the strategy in our previous work [3], each segmented structural component is represented as a **Component** node in the graph. Component-level attributes, including component type, damage state, and representative camera viewpoints computed in Sec. 3.3.1, are attached to the corresponding nodes. Spatial adjacency relationships between components are encoded as edges, forming a component-level semantic graph that reflects the structural topology of the bridge.

Building upon this representation, we further introduce a second-level, damage-oriented subgraph to explicitly model corrosion regions. Specifically, Gaussian primitives labeled as corrosion in Sec. 3.3.2 are extracted and clustered to identify spatially coherent corrosion regions. DBSCAN is employed on the 3D positions of Gaussian centers, enabling grouping based on spatial proximity while remaining robust to irregular region shapes. Each resulting cluster is instantiated as a **CorrosionRegion** node within the corresponding component subgraph. This clustering step aims to provide an instance-level organization of corrosion for information management, rather than precise physical boundary delineation.

For each CorrosionRegion node, representative viewing angles are computed using the camera viewpoint selection strategy described in Sec. 3.3.1, allowing direct integration with the GS rendering pipeline for targeted visualization. This enables inspectors to efficiently query and render specific corrosion regions from informative viewpoints, supporting off-site inspection and condition assessment.

In addition, spatial attributes associated with each **CorrosionRegion** instance, including the voxelized spatial extent of the corresponding GS primitives, the OBB center coordinates, and the axis endpoint coordinates of the host component, are stored as properties of the corresponding nodes. These geometric attributes enable spatial reasoning at query time, such as computing the relative distance of a corrosion region to component endpoints normalized by component length, supporting fine-grained damage localization and proximity-based retrieval.

The structured semantic representation established in this framework organizes defect-related information at both the component and corrosion-region levels, enabling semantic querying, targeted rendering, and fine-grained damage localization through graph-based retrieval.

4. Case study

This section validates the proposed framework through a real-world case study on a pedestrian truss bridge located in Cardiff, Wales. Section 4.1 describes the experimental configuration, including the test bridge, the panoramic data acquisition setup, the training and test split, and the computing platform. Section 4.2 evaluates the block-wise and texture-oriented densification strategy through qualitative and quantitative comparisons, followed by a parameter sensitivity analysis. Section 4.3 then examines component-level damage characterization within the graph-based framework, covering component segmentation, corrosion segmentation, and the management of the resulting damage information.

4.1. Case study configuration

To demonstrate the practical applicability of the proposed method, we conducted experiments on a pedestrian truss bridge located in Cardiff, Wales. The bridge consists of 89 structural members in total, comprising 20 diagonal members and 13 upper chords on each lateral face, together with 12 top horizontal struts and 11 top lateral bracings. The data acquisition was performed using a DJI Phantom 4 UAV equipped with an Insta360 X4 panoramic camera, which features built-in stabilization to mitigate motion blur during flight. Each collected panoramic image has a resolution of 7680×3840 pixels. A total of 712 panoramic images were captured. Figure 5 (a) and (c) present two representative panoramic images, exemplifying the exterior and interior views of the bridge, respectively. These were converted into perspective images with a resolution of 1000×1000 pixels, among which 2,377 images contain the bridge structure. Figure 5 (b) and (d) present visualizations of the sampled perspective views. A subset of 2,204 images was successfully matched and registered through SfM, with their spatial distribution shown in Figure 6 (a). Regarding the train/test split, we selected every 8th image as a test view, with the remaining images used for training. This strategy ensures adequate spatial coverage of the bridge while maintaining a sufficient number of training samples. Due to the typical overcast weather in the UK, severe glare or harsh illumination variations were not observed, and occlusions were minimal, as reflected in the camera viewpoint distribution. All data processing was carried out on a workstation equipped with an NVIDIA RTX A6000 GPU (47.5 GB dedicated GPU memory, 63.9 GB shared GPU memory) and a 12th Gen Intel® Core™ i9-12900 CPU running at 2.40 GHz.

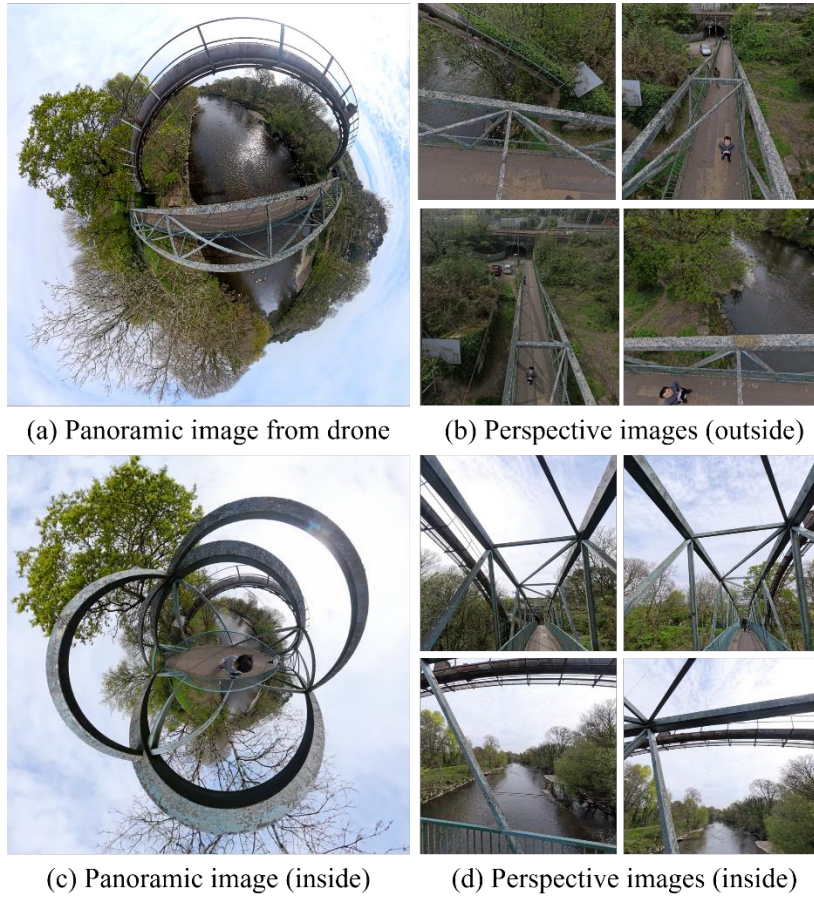


Figure 5 Collected panoramic image, generated perspective images and semantic masks.

Building upon the SfM outputs (shown in Figure 6 (a)), GS is initialized as shown in Figure 6 (b), (c), and (d). While the baseline GS model successfully captures the overall geometry and texture of both the bridge and its surrounding environment, closer inspection reveals a lack of detail in the reconstructed surface damage. Specifically, many damage areas appear as coarse color patches, indicating that the model primarily preserves low-frequency color features while failing to reconstruct high-frequency details effectively. Furthermore, the generated GS model consists of 407,157 Gaussians, among which only a small fraction correspond to the target bridge, while the majority represent background regions. This indicates that substantial optimization effort is expended on irrelevant areas, introducing distractions that hinder effective refinement of the target structure. Such inefficiency substantially limits the applicability of the baseline GS model for fine-grained surface damage characterization in structural assessment.

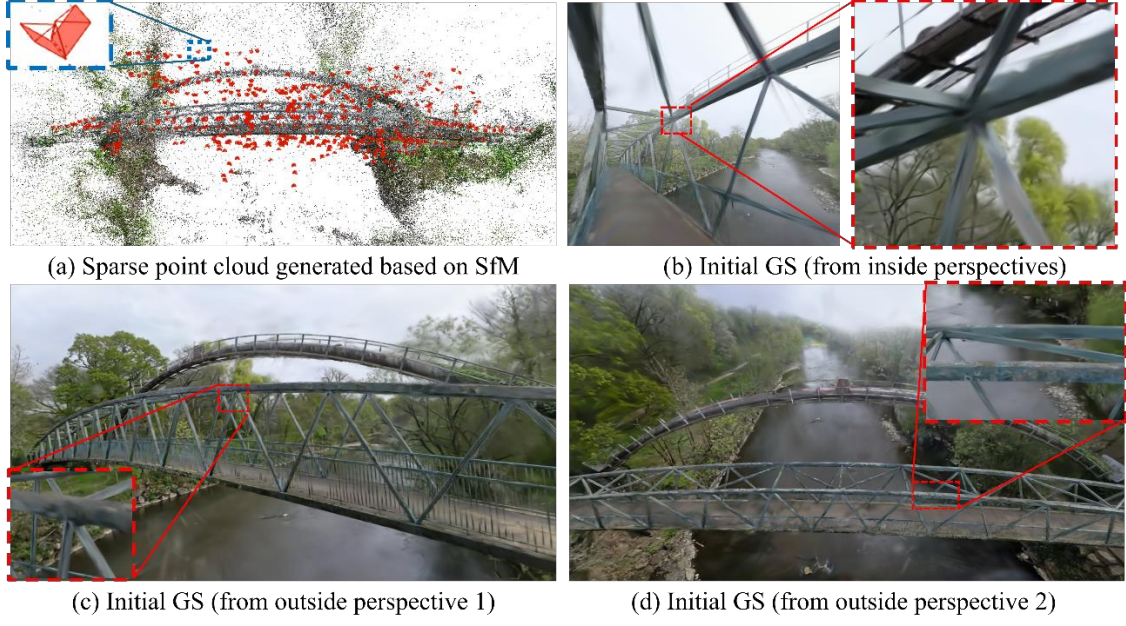


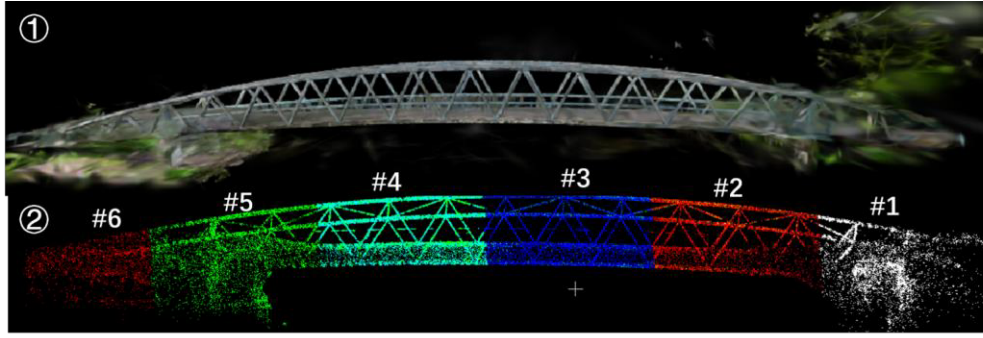
Figure 6 Sparse point cloud and initialized GS model.

4.2. Block-wise and texture-oriented GS densification

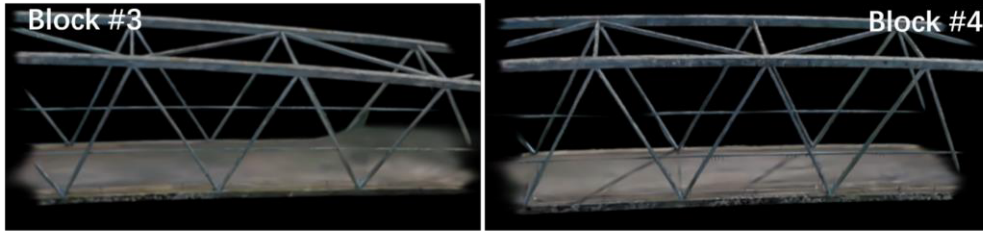
To address this limitation, we follow the methodology described in Section 3.2 to enhance the local texture expressiveness of the GS model. We first manually extract the Gaussians associated with the bridge structure. These splats are then divided into multiple blocks along the longitudinal direction of the bridge. As illustrated in Figure 7, (a) shows the extracted bridge-related blocks, (b) presents examples of rendering from two different blocks, and (c)–(d) visualize the result of viewpoint alignment. Table 1 shows the initial quantity of GS and the number of allocated viewpoints for each GS-block with $\tau_0 = 0.70$.

Table 1 Detail of GS blocks and assigned viewpoints.

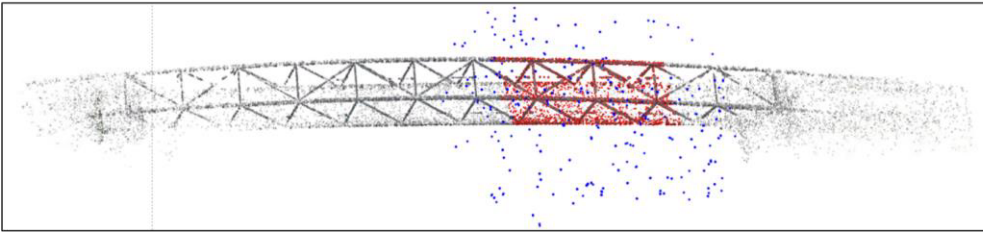
Est. (mm)	#1	#2	#3	#4	#5	#6
Num of GS	3630	26991	49860	54089	23023	8314
Num of viewpoints	385	722	471	444	229	328



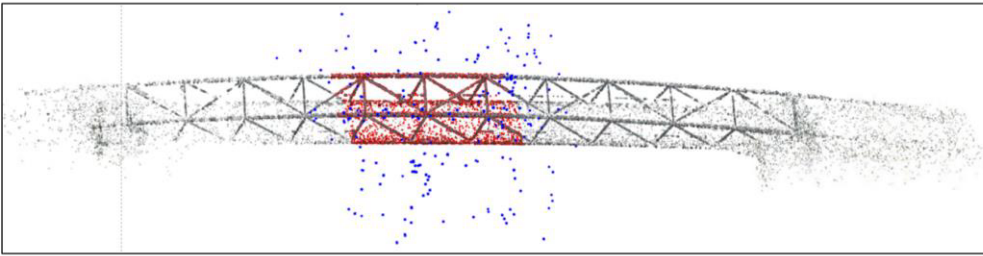
(a) Extracted bridge-related GS



(b) Rendering from GS blocks



(c) Visualization of Block #3 and corresponding viewpoints



(d) Visualization of Block #4 and corresponding viewpoints

Figure 7 Filtered GS model and detail of GS model block.

After GS-blocking and viewpoint assignment, texture-oriented GS densification is employed to enhance the texture expressiveness of each GS block. As described in Sec. 3.2.1, the ground truth for each block is extracted in two stages so as to preserve only the regions relevant to the block being optimized. Figure 9 illustrates this process. First, the semantic masks generated by SAM are used to extract the bridge-related regions from the images. Second, each block is rendered from every assigned viewpoint using the GS representation. The rendered regions are then treated as binary masks, where the visible GS regions are set as foreground (white) and the remaining parts as background (black). Figure 9(c) and (d) show the rendered masks for Block #3 and Block #4, respectively. It can be observed that these regions do not perfectly align with the structural boundaries, and are often larger due to the ellipsoidal nature of GS. To address this, we take the intersection between the semantic mask in Figure 9 (b) with the foreground regions in Figure 9(c) and (d), resulting in the final ground truth for the corresponding blocks, as illustrated in Figure 9 (e) and (f). Consequently, for each block, a dedicated

set of images is extracted, and within each image, only the regions relevant to that block are retained for the GS refinement process described in Section 3.2.3.

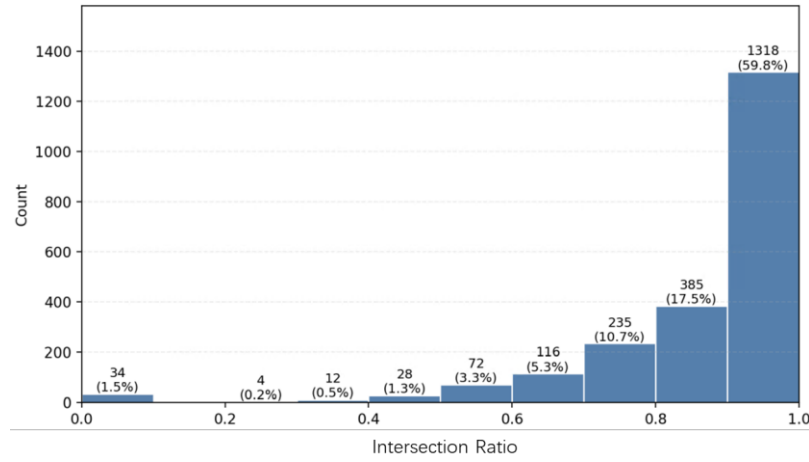


Figure 8 Distribution of ROI intersection ratios across all viewpoints.

To evaluate the robustness of the two-stage ROI extraction, we quantified the intersection ratio between the GS-rendered opacity mask (M_2) and the SAM-generated mask (M_1) across all assigned viewpoints, defined as the area of $M_1 \cap M_2$ relative to the area of M_1 . As shown in Figure 8, the distribution is heavily concentrated at high intersection ratios, indicating that the GS opacity mask and SAM mask are highly consistent across the dataset. These results confirm that the proposed two-stage masking strategy produces well-scoped and reliable ROIs in the vast majority of cases, demonstrating the robustness of the pipeline under real-world bridge inspection conditions.

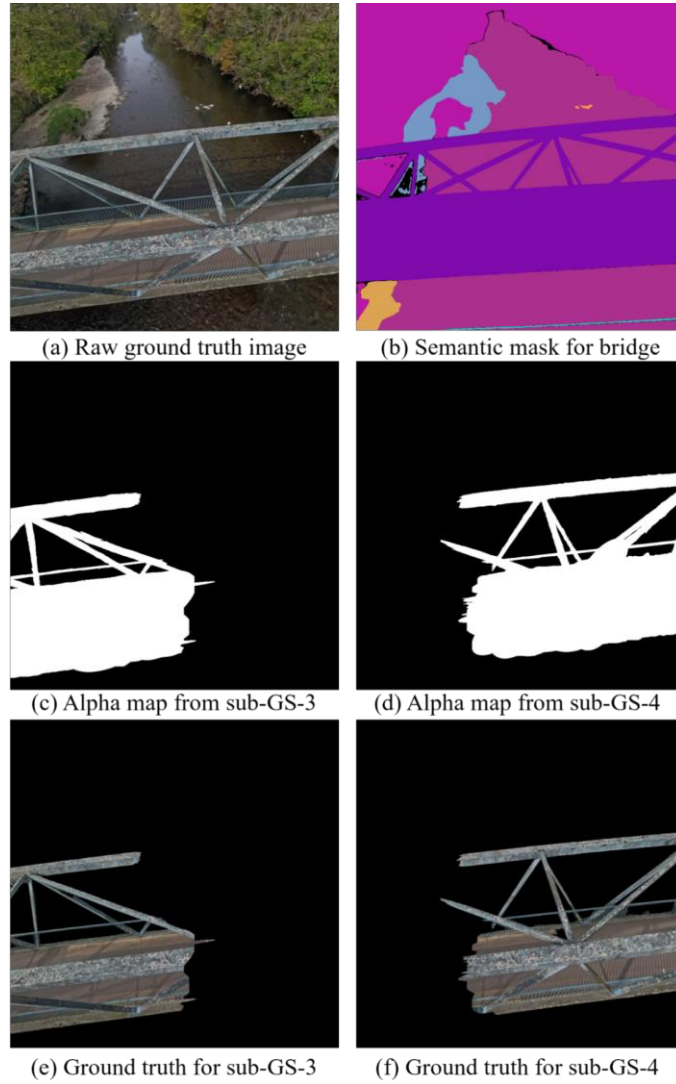


Figure 9 Illustration of masking process during block optimization

In the following session, both qualitative and quantitative evaluations are provided to demonstrate the advantages of our approach over existing methods. In particular, an ablation study is conducted to disentangle the individual contributions of block partitioning, ground-truth extraction, and local-texture-oriented Gaussian refinement. To this end, comparisons are performed against three baselines: (a) the original GS method, (b) a partitioned variant without ground-truth extraction (PGS), and (c) a partitioned variant with ground-truth extraction but with original densification strategy (PGT-GS). The subsequent sections present and discuss the results of these comparisons in detail. Unless otherwise stated, the densification-related parameter β is fixed to 0.95 for all methods to ensure a fair comparison.

4.2.1. Qualitative comparison

To begin with, a qualitative analysis is conducted. As illustrated in Figure 10, we present intermediate rendering results of local defect at different epochs during optimization. It can be observed that the proposed method effectively enhances the ability of GS to represent local damage, particularly high-frequency surface textures.

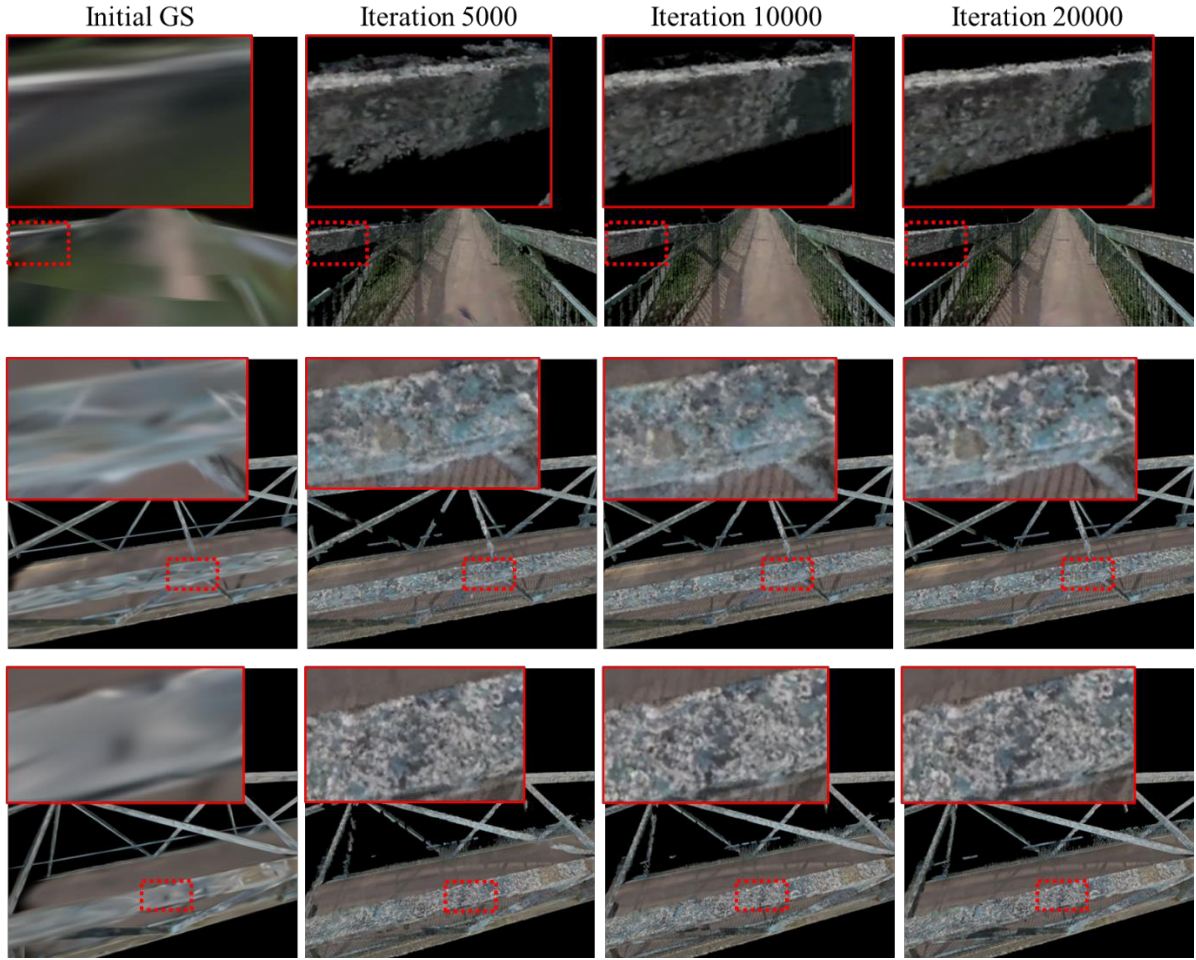


Figure 10 Visualization of the rendering results during densification

We further compare the proposed approach against three baselines, original GS, PGS, and PGT-GS, after 20,000 epochs of optimization. The results in Figure 11 reveal that the original GS model, even after extended training, fails to reconstruct fine surface textures of the structure. Combined with the results in Table 3, this suggests that the model's expressive capacity is inherently limited by computational and memory constraints, which, in the case of large-scale bridge structures, cannot be overcome simply by scaling up resources due to practical engineering restrictions.

The PGS baseline, which employs partitioned optimization without ROI extraction, achieves improved local representations of apparent damage after 20,000 epochs. However, as background regions are also optimized, the reconstruction of the target areas remains suboptimal. In contrast, the PGT-GS baseline, which incorporates partitioning and ROI extraction but following original densification strategy, produces better reconstructions of local high-frequency textures. Nevertheless, it also introduces needle-like artifacts, indicating incomplete recovery of structural details.

By comparison, the proposed method, which integrates block partitioning, ROI extraction, and progressive densification, achieves superior representation of local surface damage. This not only mitigates the limitations observed in the baselines but also provides a robust foundation for the quantitative assessment of structural appearance deterioration.

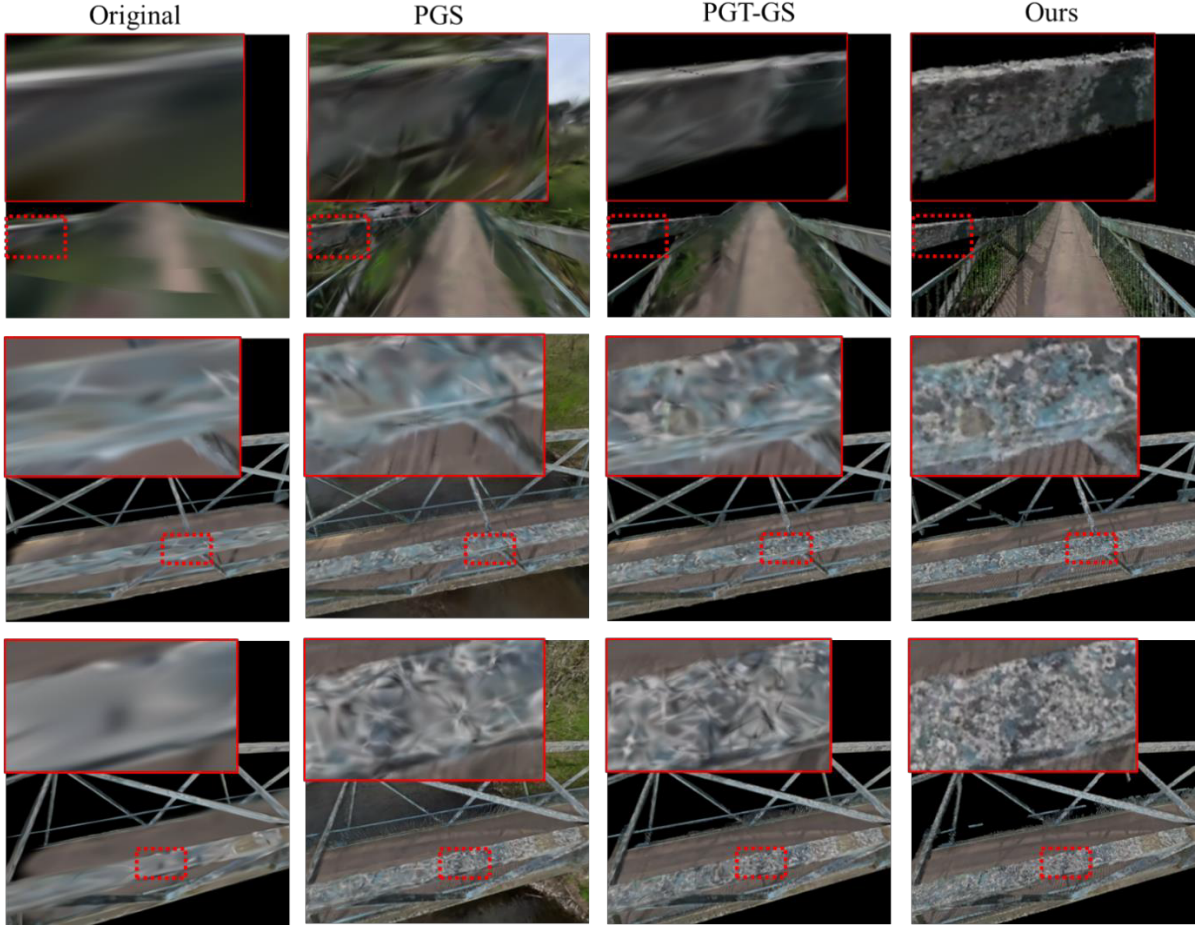


Figure 11 Comparison among different methods

Since each sub-section is optimized independently prior to global assembly, we explicitly evaluate the visual quality at block interfaces to assess inter-block consistency. As shown in Figure 12 (a), the rendering results in stitching regions remain visually coherent, with no noticeable seams or discontinuities. Compared to the corresponding ground-truth images captured near these boundaries, only minor appearance differences are observed, indicating that independent block optimization does not introduce perceptible artifacts at interfaces. Figure 12 (b) provides a detailed analysis of the spatial density distribution of Gaussian primitives after merging all sub-sections into a shared global coordinate system. The first row visualizes the overall Gaussian density across the entire reconstructed bridge, with block interfaces explicitly highlighted. Among these, interface #3 and interface #4 are selected for closer inspection and enlarged for detailed analysis. The second row illustrates the spatial overlap of Gaussian primitives contributed by the two adjacent sub-sections associated with each selected interface, demonstrating that Gaussians near block boundaries are allowed to spatially coexist rather than being truncated at block limits. The third row further visualizes the Gaussian density within these overlapping regions, where relatively higher densities can be observed. This increased density facilitates smoother blending between neighboring sub-sections and supports high-quality rendering at stitching regions, consistent with the visual results shown in Figure 12 (a).

Importantly, as evidenced by the global distribution in the first row, regions of higher Gaussian density are not confined to block interfaces. Instead, such density variations are broadly distributed across the entire bridge structure, reflecting the texture-aware and geometry-driven nature of the proposed densification strategy rather than an interface-specific bias.

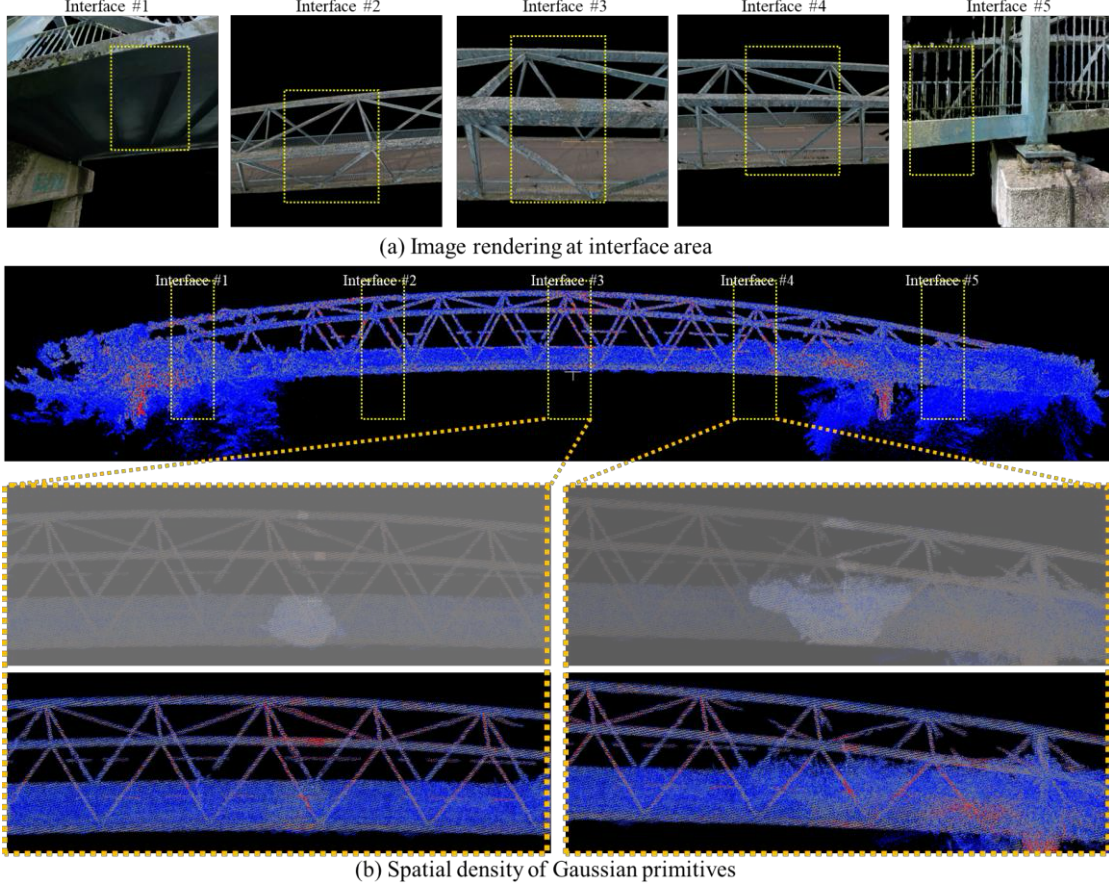


Figure 12 Detailed illustration at intersection area.

4.2.2. Quantitative comparison

Apart from qualitative comparison, quantitative comparison is also conducted from two perspectives: rendering quality and training efficiency.

From the perspective of rendering quality, Table 2 presents the quantitative comparisons between the proposed method and the three baseline approaches. The evaluation is performed across all captured viewpoints. To quantitatively assess image reconstruction quality, three standard metrics are employed: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS).

The SSIM metric, introduced earlier in Equation (12), evaluates perceptual similarity in terms of luminance, contrast, and structural consistency, while MSE and PSNR quantify pixel-level reconstruction accuracy. Their formulations are given as follows:

$$MSE(\hat{I}_j, I_j) = \frac{1}{HW} \sum_{x=1}^H \sum_{y=1}^W \|\hat{I}_j(x, y) - I_j(x, y)\|^2 \quad (24)$$

$$PNSR(\hat{I}_j, I_j) = 10 \log_{10} \left(\frac{L^2}{MSE(\hat{I}_j, I_j)} \right) \quad (25)$$

where $\hat{I}_j, I_j$ denote the predicted and ground-truth images, H and W are the image height and width, and L is the maximum possible pixel value (e.g., 255 for 8-bit images).

LPIPS measures perceptual similarity in a deep feature space extracted from a pretrained convolutional neural network and has been shown to correlate well with human visual perception, particularly for structural distortions and view-dependent artifacts commonly observed in novel view synthesis. Following common practice in neural rendering literature, LPIPS is computed using the AlexNet backbone with pretrained weights.

The results in Table 2 demonstrate that our method achieves the highest rendering accuracy among all compared approaches. This quantitative superiority is consistent with the qualitative comparisons illustrated in Figure 11, where the proposed method produces finer texture details, particularly in damaged regions

Table 2 Comparison of image rendering quality

	original GS	PGS	PGT-GS	Ours
MSE	93.64	89.65	70.61	<u>62.28</u>
PNSR	28.42 dB	28.60 dB	29.64 dB	<u>30.18 dB</u>
LPIPS	0.244	0.552	0.192	<u>0.133</u>
SSIM	0.75	0.76	0.82	<u>0.89</u>

In addition, the methods are compared in terms of training efficiency, focusing on training time, peak GPU memory consumption, and the final number of Gaussians. As summarized in Table 3, the proposed method achieves a comparable training speed to other baselines while delivering a substantial improvement in image reconstruction quality. Moreover, when considering both the number of Gaussians and reconstruction quality, the proposed method demonstrates superior efficiency, after the densification stage, it produces fewer Gaussians than the baselines while achieving higher-quality reconstructions. This further highlights the effectiveness of the proposed approach. It is important to note that the PSNR values reported in Table 3 correspond to the rendered images of individual subsections, rather than the unified GS model presented in Table 2, which reflects the performance of the merged global reconstruction.

Table 3 Comparison of training consumption

Sub Section	Method	Training Time (min)	Number of GSs	MSE	SSIM	LPIPS
sub-GS-1	PGS	23	184,881	71.28	0.84	0.225
	PGT-GS	36	161,245	82.23	0.87	0.129
	Ours	33	77,580	<u>52.91</u>	<u>0.87</u>	0.102
sub-GS-2	PGS	20	333,443	69.82	0.84	0.197
	PGT-GS	32	298,414	63.97	0.84	0.151
	Ours	39	156,037	<u>63.78</u>	<u>0.87</u>	<u>0.091</u>

sub-GS-3	PGS	22	889,107	64.96	0.88	0.106
	PGT-GS	47	304,073	53.58	0.93	0.053
	Ours	35	197,419	<u>46.65</u>	<u>0.93</u>	<u>0.044</u>
sub-GS-4	PGS	26	863,977	53.57	0.89	0.098
	PGT-GS	49	297,472	49.09	<u>0.94</u>	<u>0.053</u>
	Ours	32	201,701	<u>43.12</u>	<u>0.94</u>	<u>0.041</u>
sub-GS-5	PGS	24	991,276	63.49	0.89	0.106
	PGT-GS	31	466,437	62.22	0.91	0.044
	Ours	34	202,947	<u>52.20</u>	<u>0.94</u>	<u>0.041</u>
sub-GS-6	PGS	17	418,510	51.15	0.84	0.152
	PGT-GS	28	661,070	51.42	0.86	0.092
	Ours	32	190,824	<u>47.17</u>	<u>0.86</u>	<u>0.073</u>

In addition to evaluating overall image reconstruction quality, we further investigate the advantages of our method in reconstructing high-frequency texture details. As noted in prior studies [54], fine-scale surface textures and structural irregularities are predominantly manifested as high-frequency components in the image domain. Accordingly, we conduct a frequency-domain analysis to explicitly compare texture fidelity across methods.

Figure 13 presents both spatial-domain and frequency-domain visualizations for our method and representative benchmark approaches. In the frequency-domain representations, components near the center correspond to low-frequency information, while those closer to the image boundaries capture high-frequency details. As illustrated, our method preserves noticeably stronger high-frequency responses, indicating superior reconstruction of fine-grained textures and local surface details.

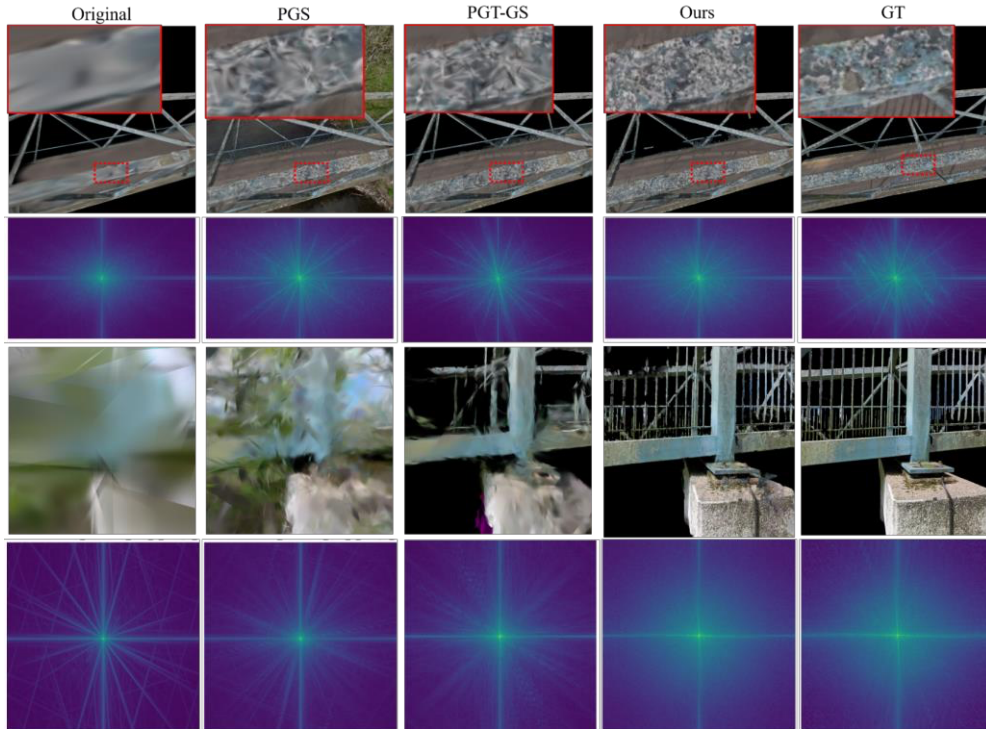


Figure 13 Comparison of rendered image and magnitude spectra.

In addition to evaluating overall image reconstruction quality using conventional perceptual metrics, we further conduct a frequency-domain analysis to specifically assess the fidelity of high-frequency texture reconstruction. This analysis is motivated by the observation that fine-scale surface details and material textures, particularly those associated with local defects, are predominantly encoded in the high-frequency components of images.

To quantitatively support this observation, we introduce two complementary frequency-domain metrics: high-frequency energy ratio (HFER) and relative high-frequency preservation (RHFP). Let $F(\cdot)$ denote the FFT operator, and let Ω_{HF} represent a predefined high-frequency band in the normalized frequency domain. This band is defined radially by a lower and upper cutoff (r_{low}, r_{high}) relative to the Nyquist frequency, ensuring consistent frequency selection across different image resolutions. To jointly capture fine-grained textures and damage-sensitive structural transitions (e.g., corroded boundaries and material discontinuities), we set $r_{low} = 0.15$ and $r_{high} = 0.5$. This range balances sensitivity to local surface irregularities while avoiding domination by low-frequency geometric structures.

We first measure the proportion of spectral energy concentrated in the high-frequency band using the HFER. For an image I , HFER is defined as:

$$HFER(I) = \frac{\sum_{(u,v) \in \Omega_{HF}} |F(I)_{u,v}|^2}{\sum_{(u,v)} |F(I)_{u,v}|^2} \quad (26)$$

We compute $HFER_{gt}$ and $HFER_{pred}$ for the ground-truth and reconstructed images, respectively. To further reduce scene-dependent spectral bias, we additionally compute a relative ratio:

$$HFER_{rel} = \frac{HFER_{pred}}{HFER_{gt}} \quad (27)$$

which directly reflects how well the reconstructed image preserves the proportion of high-frequency energy present in the ground truth. Values closer to 1 indicate better preservation of high-frequency content.

On the other hand, while HFER captures the amount of high-frequency energy, it does not measure spectral alignment. We therefore introduce the HFSE to quantify the discrepancy between reconstructed and ground-truth spectra within the high-frequency band:

$$HFSE(I) = \frac{\sum_{(u,v) \in \Omega_{HF}} (|F(\hat{I})_{u,v}| - |F(I)_{u,v}|)}{\sum_{(u,v) \in \Omega_{HF}} |F(I)_{u,v}|} \quad (26)$$

This metric computes a normalized $L1$ distance between magnitude spectra in the high-frequency band, making it particularly sensitive to local texture discrepancies. Lower HFSE values indicate closer spectral agreement and more faithful reconstruction of fine-grained textures.

Table 4 summarizes the quantitative results over all test images. Our method outperforms the compared baselines on both metrics, suggesting a clear advantage in high-frequency texture reconstruction and fine-detail preservation.

Table 4 Quantitative comparison of high-frequency texture reconstruction.

	original GS	PGS	PGT-GS	Ours
$HFER_{rel}$	0.098 ± 0.082	0.259 ± 0.060	0.546 ± 0.187	<u>0.819 ± 0.132</u>
$HFSE$	0.690 ± 0.084	0.741 ± 0.071	0.566 ± 0.065	<u>0.468 ± 0.066</u>

To further validate the effectiveness of the proposed densification strategy compared to conventional methods, we also analyze its influence on the geometric distribution of Gaussians, as the shape of each Gaussian plays a crucial role in determining the final rendering fidelity. As illustrated in Figure 14, the first and second subfigures show the statistical distributions of the Scaling Value (length of the longest-axis) and the aspect ratio (the ratio between the longest and shortest axes) of all GSs during densification. It can be observed that although both the proposed and conventional densification methods effectively reduce the length of the Gaussians’ major axes, thereby controlling their overall size, the proposed method additionally improves the aspect ratio of Gaussians, which effectively mitigates needle-like artifacts in the rendered results.

The third subfigure presents the temporal evolution of the total number of GSs throughout the densification process, along with the proportion of GSs pruned at each iteration. As shown, the proposed method maintains an advantage in terms of Gaussian quantity compared with the conventional approach, which may result from the higher selection criteria adopted during densification, leading to a correspondingly higher pruning ratio.

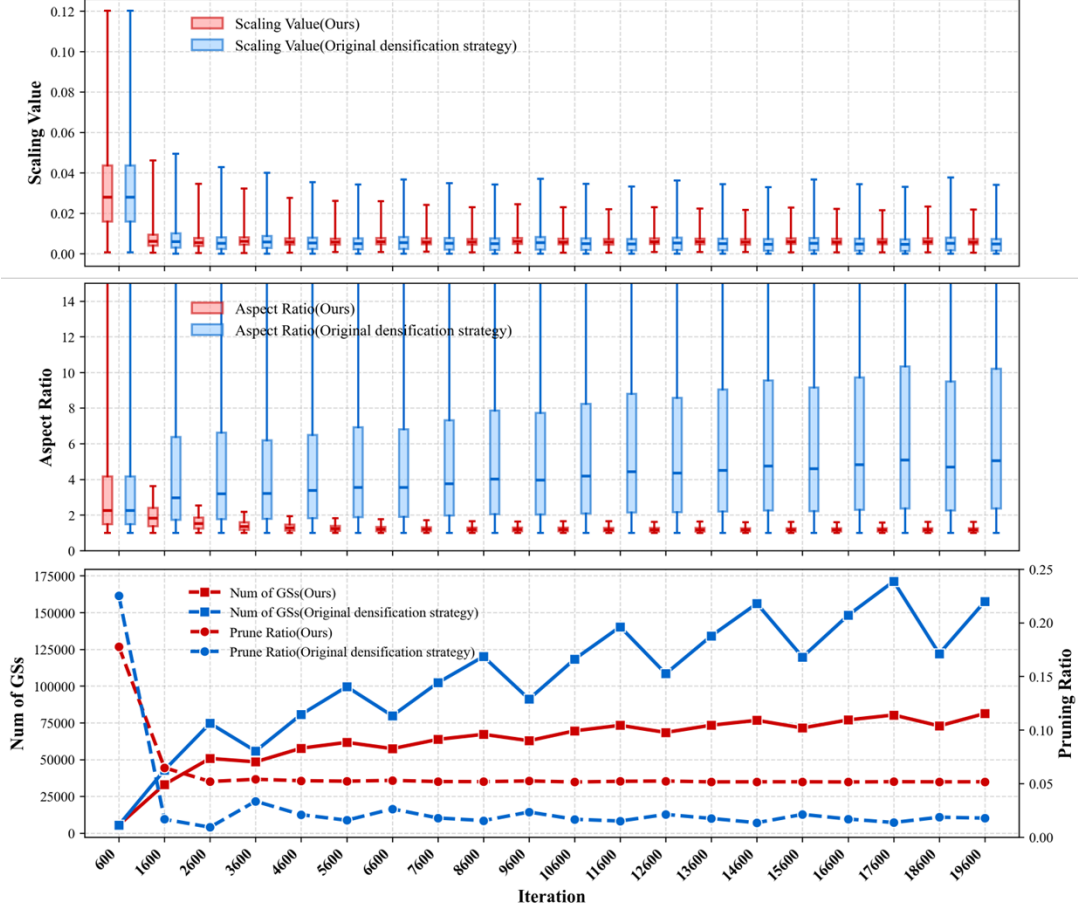


Figure 14 Comparison of GS densification process.

4.2.3. Parameter sensitivity analysis

This section investigates the sensitivity of our reconstruction pipeline to several key parameters, including the viewpoint assignment threshold τ_0 , the densification threshold β controlling the subdivision process, and the weighting λ_{grad} of the texture-aware term in the loss function. The primary goal is to quantify how variations in these parameters affect reconstruction quality.

● Sensitivity to the Viewpoint Assignment Threshold τ_0

To analyze the influence of the SSIM-based viewpoint assignment threshold τ_0 , we evaluate four representative values: $\tau_0 \in \{0.6, 0.7, 0.8, 0.9\}$. For each setting, viewpoints are reassigned to Block#3 and Block#4, and the number of images allocated under different thresholds is reported in Table X.

Based on these assignments, Gaussian Splatting densification is performed using identical default parameters, and reconstruction quality is quantitatively evaluated using LPIPS (perceptual similarity) and HFSE (high-frequency spectral energy).

As shown in Table 5, increasing τ_0 results in a larger number of assigned images for each section. However, this increase in viewpoint count does not lead to a commensurate improvement in reconstruction quality across either LPIPS or HFSE. This observation indicates that many of the additionally included images capture only marginal portions of the target region, and therefore

contribute limited structural or textural information to the reconstruction. These results support the use of a moderate threshold, which balances computational efficiency and reconstruction fidelity.

Table 5 Impact of τ_0 on reconstruction quality.

Sub Section	τ_0	Image Num	LPIPS	HFSR
sub-GS-3	0.90	641	0.0456	0.571
	0.80	535	0.0546	0.568
	0.70	471	0.0448	0.524
	0.60	295	0.1004	0.845
sub-GS-4	0.90	703	0.0498	0.536
	0.80	549	0.0451	0.526
	0.70	444	0.0412	0.508
	0.60	356	0.0795	0.614

● Sensitivity to the Densification Control Parameter β

As described in Section 3.2.3, the three densification-related thresholds in our pipeline are jointly controlled by a single parameter β . Consequently, we focus on varying β to examine its effect on the training process and geometric evolution of Gaussian primitives.

Figure 15 illustrates the reconstruction results obtained under different values of β . A larger β generally yields improved rendering quality. Mechanistically, β determines the aggressiveness of Gaussian removal during densification: smaller values correspond to a higher proportion of Gaussian primitives being deleted.

Figure 16 further visualizes the geometric evolution of Block#6 during densification. As shown in the first two subfigures, smaller β values lead to faster correction of anomalous Gaussians, including excessively large primitives and extreme aspect ratios. However, the third subfigure reveals that when β is set to 65% or 75%, the total number of Gaussians decreases over training, indicating overly aggressive pruning. This behavior is consistent with the degradation in reconstruction quality observed in Figure 15, where excessive Gaussian removal leads to loss of rendering performance. Therefore, the adopted value $\beta = 95\%$ represents a reasonable compromise that effectively suppresses outliers while preserving sufficient geometric capacity for high-fidelity reconstruction.

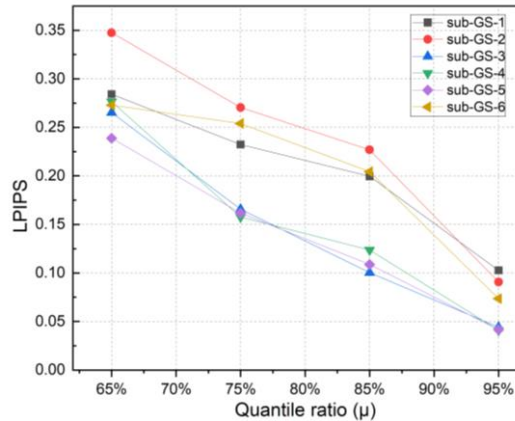


Figure 15 Comparison of LPIPS under different β value

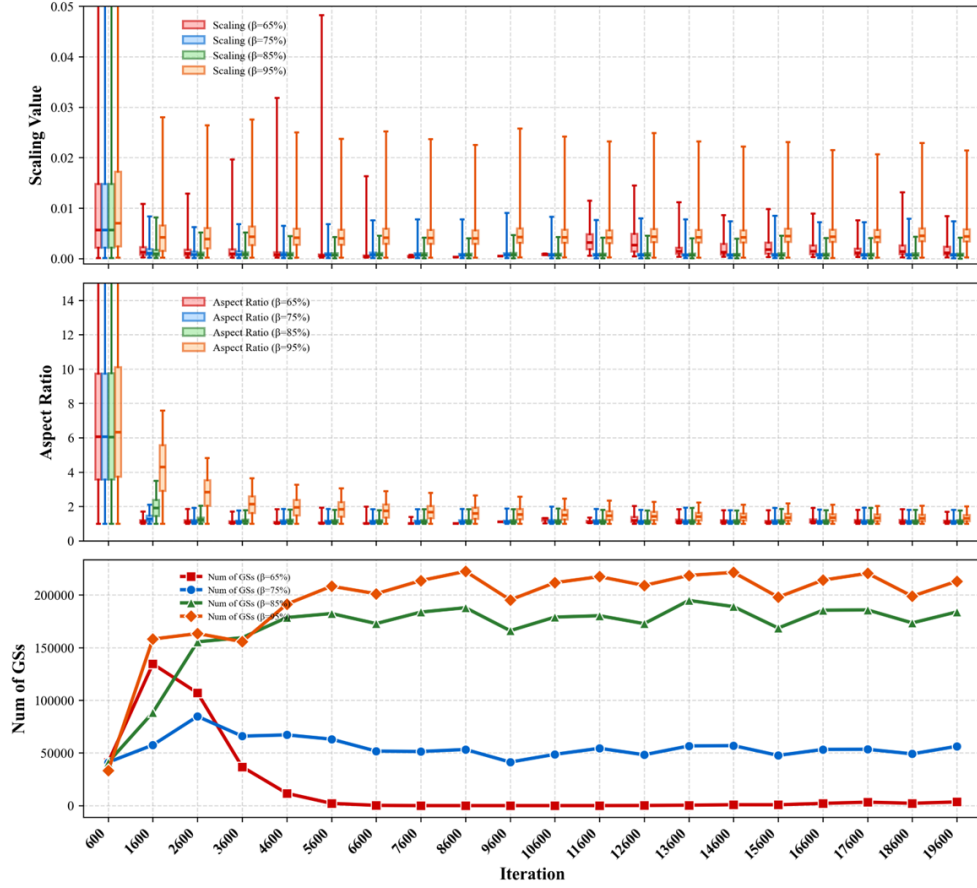


Figure 16 Comparison of GS densification process under different β value

● Sensitivity to the Weight of the Texture-Aware Loss Term λ_{grad}

To evaluate the effect of the texture-aware loss weighting, we train the model with four different values: $\lambda_{grad} \in \{0, 0.25, 0.5, 0.75\}$. Figure 17 reports quantitative comparisons in terms of overall perceptual reconstruction quality (LPIPS) and high-frequency texture fidelity (HFSE).

Compared to the baseline configuration without texture-aware loss ($\lambda_{grad} = 0$), introducing the texture-aware term consistently improves performance on both metrics. However, excessively large weights lead to degradation in both overall reconstruction quality and high-frequency detail preservation, indicating overemphasis on local gradients at the expense of global structural consistency. Based on these observations, the selected value ($\lambda_{grad} = 0.25$) achieves the best trade-off between global fidelity and local texture enhancement.

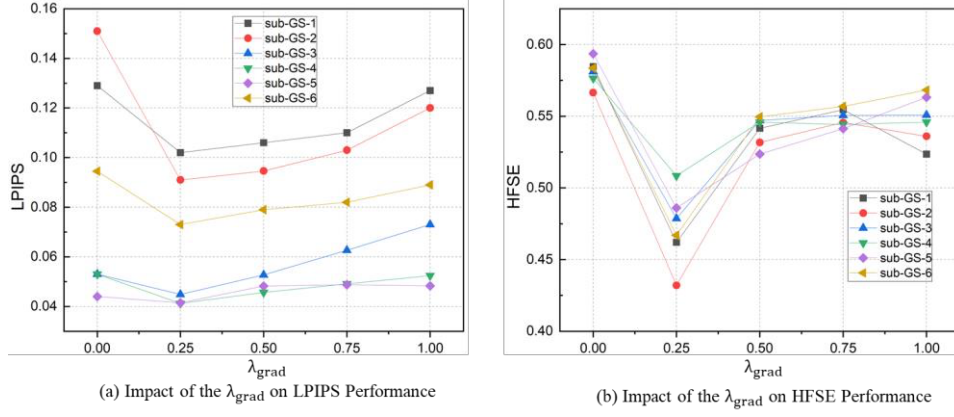


Figure 17 Impact of λ_{grad} on image rendering.

To further assess the generality of the proposed texture-aware strategy, we replace the Sobel operator with an alternative high-frequency texture extraction method, namely the Laplacian operator [55]. The weighting parameter is kept identical to the main experiments ($\lambda_{grad} = 0.25$) to ensure a fair comparison.

As reported in Table 6, the Laplacian-based variant consistently outperforms the baseline without any texture-aware loss in both LPIPS and HFSE. This result demonstrates that the observed performance gains are not tied to a specific gradient operator but rather stem from the general principle of incorporating high-frequency texture supervision into the optimization process.

Table 6 Comparison of LPIPS and HFSE of different texture representations within loss function

Sub Section	Method	LPIPS	HFSE
sub-GS-1	original GS	0.129	0.584
	Ours	0.102	0.462
	Laplace-based	<u>0.116</u>	<u>0.567</u>
sub-GS-2	original GS	0.151	0.566
	Ours	0.091	0.432
	Laplace-based	<u>0.128</u>	<u>0.535</u>
sub-GS-3	original GS	0.053	0.581
	Ours	0.045	0.478
	Laplace-based	<u>0.051</u>	<u>0.558</u>
sub-GS-4	original GS	0.053	0.576
	Ours	0.041	<u>0.508</u>
	Laplace-based	<u>0.049</u>	<u>0.548</u>
sub-GS-5	original GS	0.044	0.593
	Ours	0.041	0.486
	Laplace-based	<u>0.043</u>	<u>0.537</u>
sub-GS-6	original GS	0.092	0.583
	Ours	0.073	0.467
	Laplace-based	<u>0.089</u>	<u>0.533</u>

4.3. Component-level damage characterization within the graph-based framework

After constructing the high-fidelity GS model, multi-granularity semantic information is embedded to enable detailed damage assessment. As outlined in Sec. 3.3, two types of semantic information are considered: component-level semantics and damage-related semantics. This section presents the segmentation results for these two categories.

4.3.1. Component segmentation

As illustrated in Figure 18 and following the methodology presented in Section 3.3.1, the point cloud processing procedure was implemented through a multi-stage approach. First, as shown in Figure 18 (a), the original point cloud was segmented into distinct clusters using DBSCAN. Subsequently, Principal Component Analysis (PCA) was applied to each cluster to determine its principal direction, with further grouping conducted based on these principal directions using the direction-based clustering algorithm detailed in Algorithm 1. This process resulted in five primary directional orientations, as illustrated as arrows in Figure 18 (b), representing the two main extension directions of the lateral chord members, one extension direction of the top horizontal braces, and two extension directions of the top diagonal braces.

Based on these extension directions, spatial line segments formed by the clusters along each directional orientation were computed using Algorithm 2. Point clouds distributed in the vicinity of these spatial line segments were consolidated into 89 groups through spatial refinement procedures, thereby generating the grouping results corresponding to the web members, as shown in Figure 18 (b). For the remaining point clouds, DBSCAN was applied once again for main arch segmentation, with results presented in Figure 18 (c). The final comprehensive grouping results are displayed in Figure 18 (d). Subsequently, each identified component can be rendered for detailed off-site inspection, and Figure 19 illustrates the GS rendering results for components AL-4 to AL-6 and AR-4 to AR-6. This component-level representation provides a solid foundation for the subsequent construction of the semantic graph and the retrieval of associated damage information.

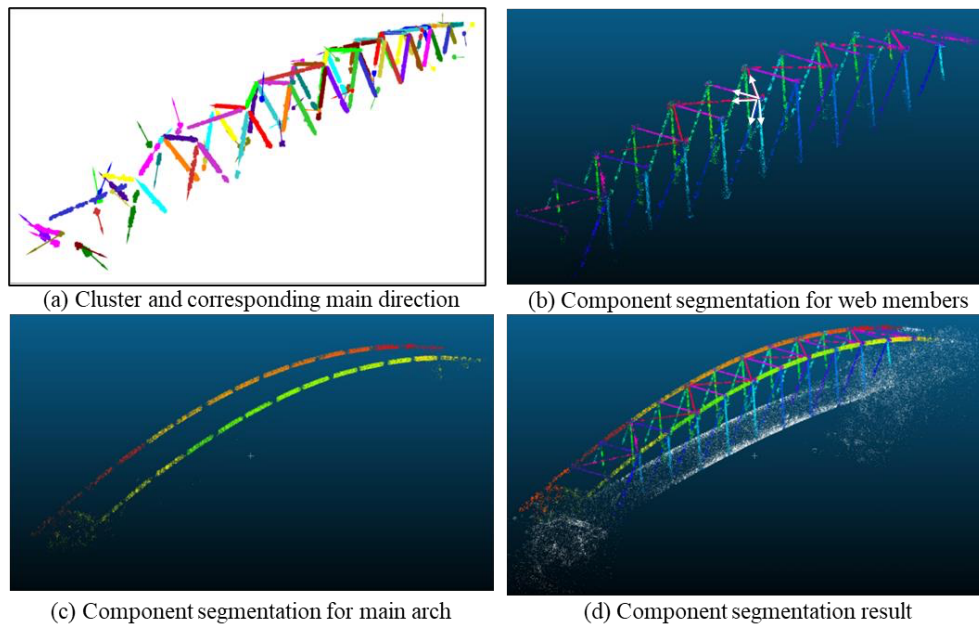


Figure 18 Process of component segmentation.

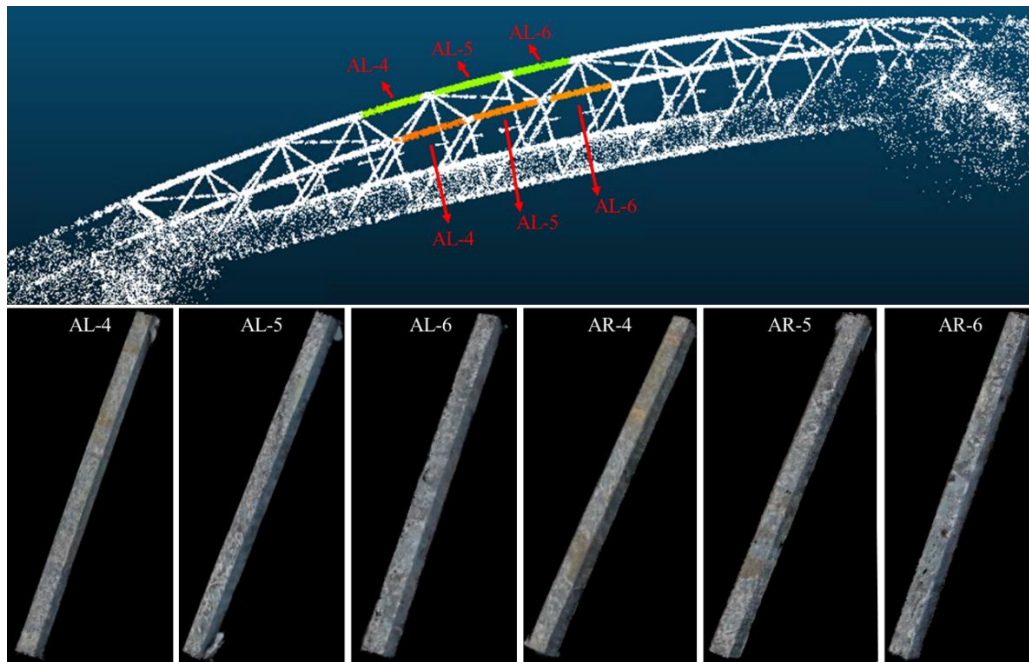


Figure 19 Example of component rendering and corrosion-related semantic mask.

4.3.2. Corrosion segmentation

This section describes the detailed process of image-based corrosion segmentation, including dataset preparation, model finetune, and semantic embedding.

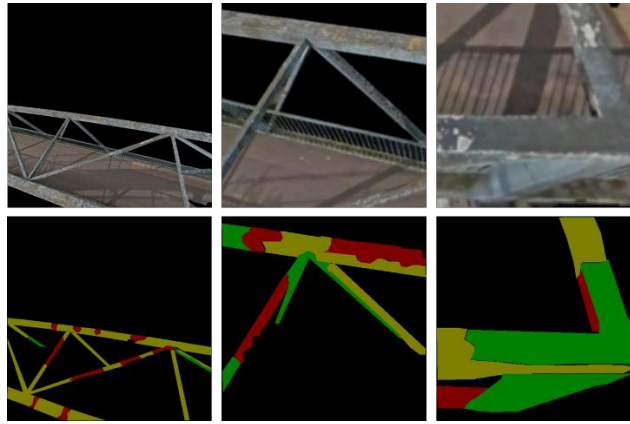


Figure 20 Examples of generated dataset

Dataset preparation: As shown in Figure 20, unlike generic corrosion datasets, the studied bridge surfaces not only exhibit corrosion but are also extensive biological colonization, which further accelerate material degradation [56]. Accordingly, the segmentation task was defined as a three-class problem, consisting of *corroded regions*, *moss-covered regions*, and *healthy surface regions*. Among the 2,377 perspective images described in Section 4.1, 83 representative images were manually annotated for damage labeling.

However, as illustrated in Figure 20, large portions of the bridge are dominated by moss-covered areas, leading to a class imbalance issue where both corrosion and healthy regions occupy a relatively small proportion of the total surface area. To mitigate this imbalance and improve segmentation efficiency, we applied data augmentation and extracted additional cropped patches, resulting in a total of 117 images for training and validation. Representative samples and corresponding annotations are shown in Figure 18, where red denotes corroded regions, yellow indicates vegetation-covered areas, and green represents healthy regions.

Importantly, the reported mIoU metrics for corrosion and moss are computed on a held-out test set with 50 images, distinct from both the training and validation sets. This ensures that the reported performance reflects generalization rather than overfitting, even with the relatively small original dataset. Confusion matrices and representative error cases are additionally provided in the supplementary material to illustrate the method's limitations and typical misclassification patterns.

Model finetune: The DeeplabV3+ model was initialized with pretrained weights from a large-scale image dataset [15] to leverage its strong feature extraction capability. Building upon our prior work [15], in which the model had already been trained on corrosion segmentation tasks, we further fine-tuned this initialized model to adapt it to the specific bridge scenarios investigated in this study. A two-stage fine-tuning strategy was adopted, consisting of a frozen phase (epochs 0–50, batch size = 8) and an unfrozen phase (epochs 50–200, batch size=2). During the frozen phase, the backbone weights were kept fixed to stabilize feature extraction and prevent premature overfitting of the pretrained encoder to the limited domain-specific dataset. After convergence of the decoder layers, all parameters were

1010 unfrozen to enable end-to-end optimization, allowing the network to adapt its feature representations to
1011 corrosion-specific visual patterns. The detailed hyperparameter settings are summarized in Table 7.

1012 Table 7 Detail of finetune strategy

Parameter	Value / Setting
Backbone	Xception
Optimizer	Stochastic Gradient Descent (SGD)
Initial Learning Rate	5×10^{-4}
Minimum Learning Rate	5×10^{-6}
Training Strategy	Two-stage (Freeze $\rightarrow$ Unfreeze) Frozen Phase: Epochs 0–50, Batch size 8 Unfrozen Phase: Epochs 50–200, Batch size 2

1013 The training process is illustrated in Figure 21, which shows the convergence of the loss function,
1014 demonstrating stable optimization behavior throughout training. The quantitative segmentation results
1015 are summarized in Table 8, presenting performance across the three damage categories, *healthy*, *moss-*
1016 *covered*, and *corroded regions*, in terms of mean Intersection over Union (mIoU), mean Pixel Accuracy
1017 (mPA), Precision, and Recall. Furthermore, Figure 22 presents the confusion matrices of the validation
1018 results, which provide a detailed view of the inter-class prediction behavior. The strong diagonal
1019 dominance and reduced inter-class confusion indicate that the proposed method achieves reliable
1020 discrimination among different surface conditions. These results further confirm the effectiveness of the
1021 proposed approach for corrosion segmentation in bridge inspection scenarios.

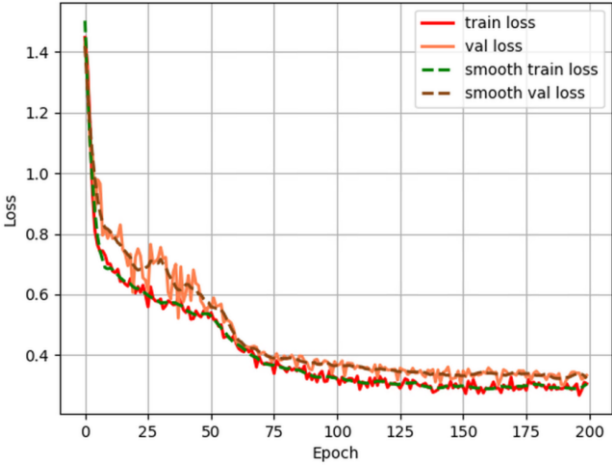


Figure 21 Process of model finetuning

1022 Table 8 Quantitative Evaluation of Segmentation Performance for Each Damage Category

Metric	Health	Moss-covered (Plant)	Corrosion	Background
mIoU	0.55	0.75	0.63	0.94
mPA	0.67	0.90	0.68	0.96
mPresision	0.75	0.82	0.90	0.97
Recall	0.67	0.90	0.68	0.96

1023

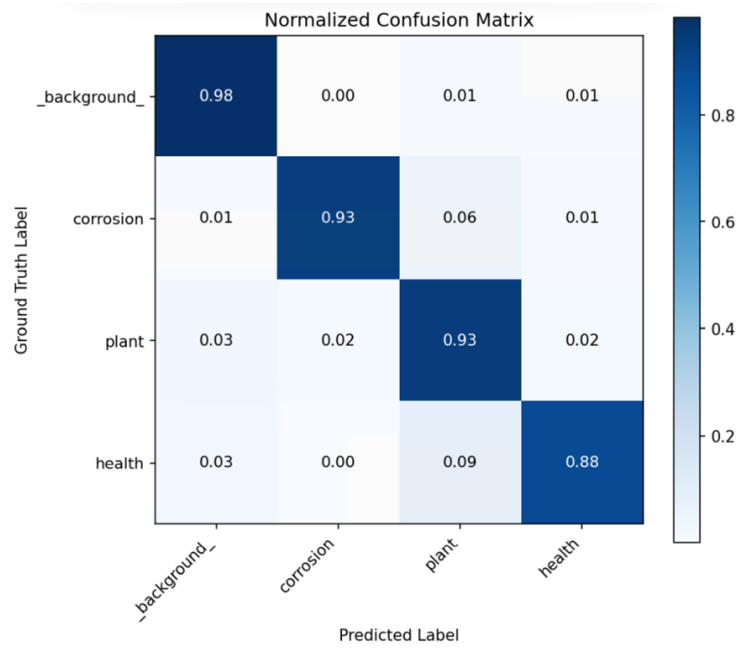


Figure 22 Confusion matrices of validation results.

Semantic embedding: Following the completion of image-based damage segmentation, the 2D semantic masks generated by the deep learning model are embedded into the GS representation using the semantic embedding procedure described in Gaga [48]. This integration enables a bottom-up evaluation of structural damage within the 3D domain.

As illustrated in Figure 23(a), for each component, multiple viewpoint renderings are first generated from the corresponding GS model to capture comprehensive surface coverage. The trained segmentation network is then applied to these rendered images to produce pixel-level semantic masks that delineate corrosion- and vegetation-affected regions. Subsequently, the semantic embedding method is employed to map the 2D semantic information back onto the GS model, establishing a one-to-one correspondence between image pixels and 3D Gaussian elements. The resulting semantic GS models of individual components are then assembled to form the integrated bridge-level representation, as shown in Figure 23 (b), which provides a unified 3D damage representation.

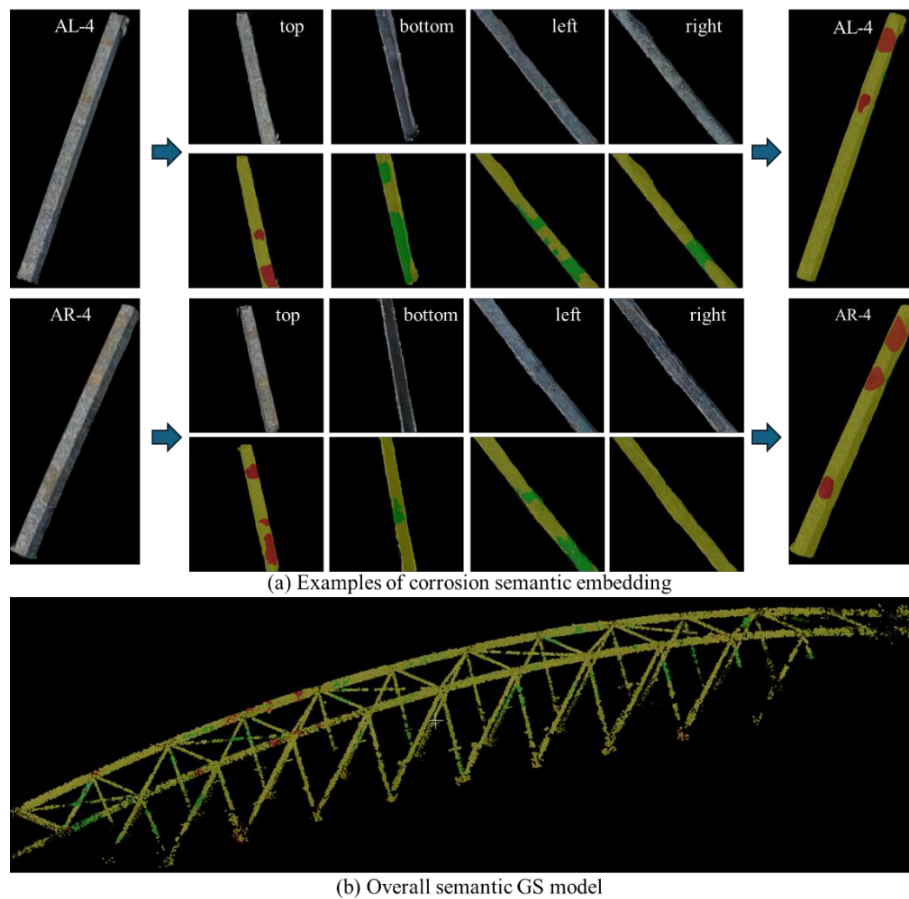


Figure 23 Process of Semantic GS generation: (a) corrosion semantic embedding process (take AR-4 and AL-4 as example); (b) generated semantic GS model.

4.3.3. Damage information management based on GS

Based on the semantic GS representation shown in Figure 23, and following the methodology described in Section 3.3.3, we construct the semantic graph illustrated in Figure 24 to systematically organize structural damage information. Figure 24(a)–(d) present the semantic attributes of the GS model from multiple perspectives, including component type, corrosion ratio, health ratio, and biological colonization ratio. These attribute layers enable stratified analysis of structural conditions and support prioritized inspection planning based on degradation severity.

Since individual components often contain multiple spatially distinct corrosion regions, each localized defect is stored as a dedicated node within the subgraph associated with its parent component node. These corrosion-specific nodes, visualized in black in the figure, preserve fine-grained spatial and textural information essential for defect-level analysis. This hierarchical organization, spanning bridge-level, component-level, and defect-level nodes, facilitates both coarse-grained queries (e.g., "all severely corroded girders") and fine-grained damage inspection (e.g., "corrosion patches exceeding 15% area on a specific girder").

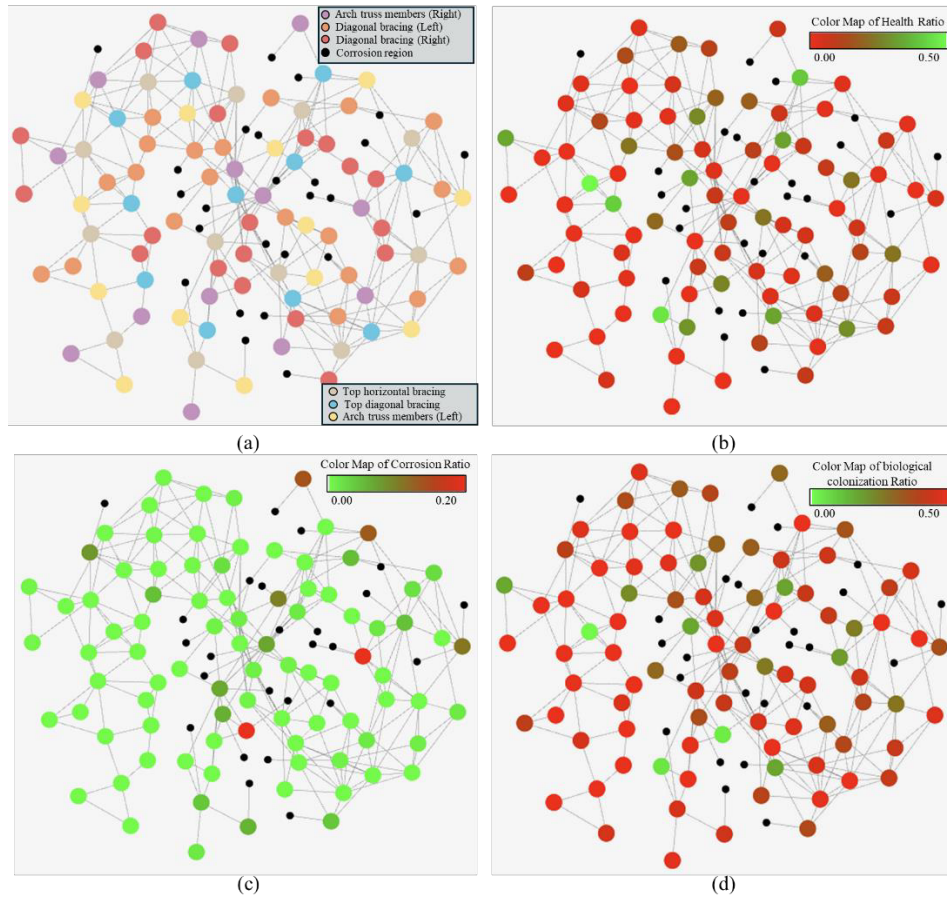


Figure 24 Ggenerated semantic graph for target bridge structure: (a) from the perspective of component type; (b) from the perspective of health ratio; (c) from the perspective of corrosion ratio; (d) from the perspective of risky ratio.

Building upon the constructed semantic graph, the framework enables semantic-driven information retrieval integrated with targeted GS-based rendering, supporting remote inspection workflows and access to historical structural condition records [6]. As illustrated in Figure 25, the pipeline links three key modules: (1) the semantic graph, which organizes components and corrosion regions into a hierarchical structure; (2) the retrieval engine, which interprets user queries and identifies relevant semantic nodes, implemented in this study using Neo4j and Cypher; and (3) the rendering module, which synthesizes component- or defect-specific views using the GS model and corresponding viewpoints. Through this coordinated workflow, semantic queries can directly trigger localized visualization of damage regions, enabling focused inspection, efficient information access, and fine-grained condition interpretation.

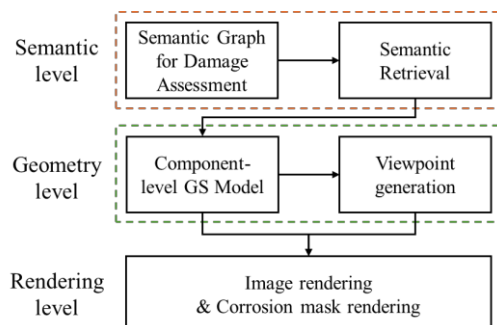


Figure 25 Pipeline of semantic-driven image rendering.

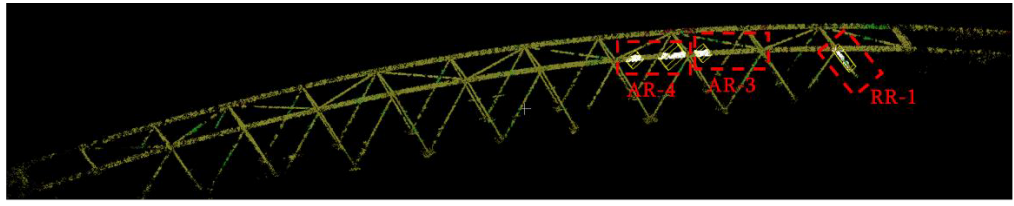
Table 9 provides representative examples of Cypher-based semantic queries, demonstrating capabilities such as filtering components by damage severity and identifying spatial relationships between defects and critical structural elements. Figure 26 further illustrates how semantic retrieval can be combined with Gaussian Splatting to achieve targeted rendering of specific corrosion regions. By querying individual defect nodes and rendering only their associated Gaussian primitives, inspectors can obtain spatially isolated, high-fidelity visualizations of damage areas within their local geometric context. This filtered visualization eliminates visual clutter from the full-scale reconstruction, allowing inspectors to focus on damage-critical regions and support more informed maintenance decision-making.

These results demonstrate the effectiveness of the proposed semantic graph in enabling fine-grained damage management and remote visual examination. Furthermore, the structured semantic representation established in this work provides a practical foundation for retrieving and rendering damage-related information, supporting engineering decision-making and future integration with more advanced inspection workflows.

1074
1075

Table 9 Examples of information retrieval.

Retrieval Task Description	Cypher code	Retrieval result
Identify the most severely corroded component.	<pre>MATCH (c: Component) WITH c ORDER BY c.corrosion_ratio DESC LIMIT 1 MATCH (c)-[:HAS_CORROSION]->(cor) RETURN c AS most_corroded, collect(cor) AS corrosion_regions;</pre>	
Retrieve the spatial distribution of corrosion regions on the three most severely corroded component.	<pre>MATCH (c: Component) WITH c ORDER BY c.corrosion_ratio DESC LIMIT 3 MATCH (c)-[:HAS_CORROSION]->(cor) RETURN c AS most_corroded, collect(cor) AS corrosion_regions;</pre>	
Retrieve corrosion regions located within 10% of the total length from their host component	<pre>MATCH (comp)-[:HAS_CORROSION]->(corr) WITH comp, corr, comp.axis_end0 AS e0, comp.axis_end1 AS e1, corr.obb_center AS c WITH comp, corr, c, e0, e1, sqrt((e1[0]-e0[0])^2 + (e1[1]-e0[1])^2 + (e1[2]-e0[2])^2) AS comp_length, sqrt((c[0]-e0[0])^2 + (c[1]-e0[1])^2 + (c[2]- e0[2])^2) AS d0, sqrt((c[0]-e1[0])^2 + (c[1]-e1[1])^2 + (c[2]- e1[2])^2) AS d1 WITH comp, corr, comp_length, CASE WHEN d0 <= d1 THEN d0 ELSE d1 END AS distance_to_nearest_end WHERE distance_to_nearest_end <= 0.1 * comp_length RETURN comp, corr</pre>	



(a) Visualization of result for retrieval #2



(b) Rendering for AR-4

(1st -most severely corroded component)



(c) Rendering for AR-3

(2nd -most severely corroded component)



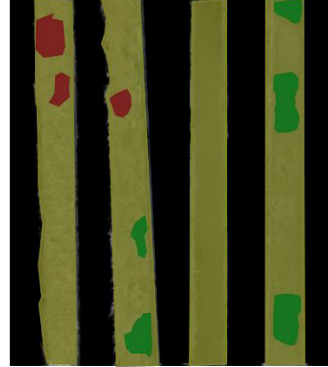
(d) Rendering for RR-1

(3rd -most severely corroded component)



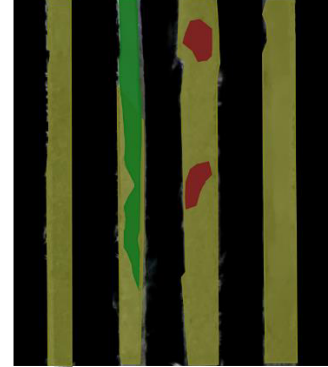
(b) Damage distribution on AR-4

(1st -most severely corroded component)



(c) Damage distribution on AR-3

(2nd -most severely corroded component)



(d) Damage distribution on RR-1

(3rd -most severely corroded component)

Figure 26 Information Retrieval based on Damage Ratio

5. Conclusion and future work

This study presented BridgeGS, a Gaussian Splatting-based framework that advances bridge surface inspection by enabling high-fidelity 3D reconstruction, defect-sensitive texture modeling, and structured semantic integration, capabilities that have remained difficult to achieve with existing image- or point-cloud-based pipelines. Through panoramic image-based GS initialization, block-wise GS refinement, and texture-oriented densification, BridgeGS reconstructs large-scale bridge structures with significantly enhanced preservation of corrosion-relevant surface details. By further integrating component segmentation, corrosion segmentation, and graph-based semantic organization, the framework supports component-aligned defect interpretation and efficient semantic-guided rendering within a unified 3D representation.

Experiments on a real-world truss bridge demonstrate that BridgeGS not only improves texture fidelity and defect visibility but also provides actionable semantic structures for targeted rendering and damage interpretation. These results highlight BridgeGS as a promising direction for next-generation

digital inspection workflows, with potential to enhance surface defect quantification, spatial localization, long-term condition tracking, and integration into asset-management systems.

Despite the promising results, several limitations suggest directions for future research:

First, the truss bridge scanning was conducted using manually piloted UAVs, resulting in uneven image distribution and heterogeneous reconstruction quality. Future work will explore automated acquisition strategies, such as autonomous UAV flight planning and fully panoramic imaging, to achieve more uniform and systematic coverage.

Second, although the proposed texture-oriented densification improves the ability of GS to capture high-frequency surface details, further research is needed to better balance global geometric consistency with localized texture fidelity and to incorporate more advanced texture priors. In addition, the current gradient-domain supervision relies on sufficient image contrast; extending the framework to handle low-illumination conditions, such as bridge undersides with limited lighting during acquisition, represents an important robustness challenge for future investigation.

Third, the semantic enrichment pipeline relies primarily on 2D segmentation and geometry-based grouping. Future work will investigate unified geometry- and learning-based semantic enrichment strategies, enabling GS models to carry richer multi-criteria semantics and evolve toward fully integrated 3D information models capable of supporting more comprehensive structural inspection and evaluation. Furthermore, while the current segmentation model achieves reliable class discrimination, incorporating uncertainty-aware semantic embedding strategies could better quantify and mitigate the propagation of segmentation errors into downstream 3D damage quantification, and is identified as a meaningful direction for future development.

Fourth, while our framework achieves high-fidelity reconstruction, the resulting model size may limit deployment in resource-constrained scenarios. Exploring defect-aware compression strategies could further enhance the practical applicability of this approach, enabling deployment on edge devices and mobile platforms while maintaining the preservation of damage-critical features.

Acknowledgements

This work was supported by the Cardiff School of Engineering and the Cardiff University–China Scholarship Council (CSC) Joint Scholarship Programme. Additional support was provided by the Knowledge Transfer Partnership (KTP) project *DIGIBRIDGE: BIM and Digital Twins in Support of Smart Bridge Structural Surveying*, and by a Scientific Research Project supported by the Jiangsu Communications Holdings Co., Ltd. (Grant No. 2025JKY10).

Reference

- [1] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Automation in Construction* 94 (2018) 112–126. <https://doi.org/10.1016/j.autcon.2018.06.006>.
- [2] F. Wang, Y. Zou, X. Chen, C. Zhang, L. Hou, E. del Rey Castillo, J.B.P. Lim, Rapid in-flight image quality check for UAV-enabled bridge inspection, *ISPRS Journal of Photogrammetry and Remote Sensing* 212 (2024) 230–250. <https://doi.org/10.1016/j.isprsjprs.2024.05.008>.
- [3] J. Liu, H. Li, C. Chai, K. Chen, D. Wang, A LLM-informed multi-agent AI system for drone-based visual inspection for infrastructure, *Advanced Engineering Informatics* 68 (2025) 103643. <https://doi.org/10.1016/j.aei.2025.103643>.
- [4] Z. Xia, J. Shu, W. Ding, Y. Gao, Y. Duan, C.J. Debono, V. Prakash, D. Seychell, R.P. Borg, Complete-coverage path planning for surface inspection of cable-stayed bridge tower based on building information models and climbing robots, *Computer Aided Civil Eng* (2025) mice.13469. <https://doi.org/10.1111/mice.13469>.
- [5] J.J. Rubio, T. Kashiwa, T. Laiteerapong, W. Deng, K. Nagai, S. Escalera, K. Nakayama, Y. Matsuo, H. Prendinger, Multi-class structural damage segmentation using fully convolutional networks, *Computers in Industry* 112 (2019) 103121. <https://doi.org/10.1016/j.compind.2019.08.002>.
- [6] J. Liu, H. Li, D. Wang, C. Chai, Y. Dong, Semantic-PolygonGraph driven context-aware coverage path planning for infrastructure visual inspection, *Advanced Engineering Informatics* 68 (2025) 103580. <https://doi.org/10.1016/j.aei.2025.103580>.
- [7] P. Stałowska, C. Suchocki, M. Rutkowska, Crack detection in building walls based on geometric and radiometric point cloud information, *Automation in Construction* 134 (2022) 104065. <https://doi.org/10.1016/j.autcon.2021.104065>.
- [8] Q. Wang, X. Wang, Q. He, J. Huang, H. Huang, P. Wang, T. Yu, M. Zhang, 3D tensor-based point cloud and image fusion for robust detection and measurement of rail surface defects, *Automation in Construction* 161 (2024) 105342. <https://doi.org/10.1016/j.autcon.2024.105342>.
- [9] C. Lin, Y. Chen, K. Itakura, S. Maharjan, P. Chun, Bridge inspection using image–point cloud fusion with image filtering, damage detection and 3D registration, *Automation in Construction* 180 (2025) 106538. <https://doi.org/10.1016/j.autcon.2025.106538>.
- [10] H. Li, Y. Chen, J. Liu, C. Che, Z. Meng, H. Zhu, High-resolution model reconstruction and bridge damage detection based on data fusion of unmanned aerial vehicles light detection and ranging data imagery, *Computer Aided Civil Eng* 39 (2024) 1197–1217. <https://doi.org/10.1111/mice.13133>.
- [11] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, (2020). <https://doi.org/10.48550/arXiv.2003.08934>.
- [12] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, 3D Gaussian Splatting for Real-Time Radiance Field Rendering, (2023). <http://arxiv.org/abs/2308.04079> (accessed October 21, 2024).
- [13] X. Pan, T.T.Y. Yang, J. Li, C. Ventura, C. Málaga-Chuquitaype, C. Li, R.K.L. Su, S. Brzev, A review of recent advances in data-driven computer vision methods for structural damage evaluation: algorithms, applications, challenges, and future opportunities, *Arch Computat Methods Eng* (2025). <https://doi.org/10.1007/s11831-025-10279-8>.
- [14] E. Ichi, S. Dorafshan, Effectiveness of infrared thermography for delamination detection in reinforced concrete bridge decks, *Automation in Construction* 142 (2022) 104523. <https://doi.org/10.1016/j.autcon.2022.104523>.
- [15] C. Chai, Y. Gao, G. Xiong, J. Liu, H. Li, Domain knowledge-driven image captioning for bridge damage description generation, *Automation in Construction* 174 (2025) 106116. <https://doi.org/10.1016/j.autcon.2025.106116>.
- [16] G. Su, Y. Qin, H. Xu, J. Liang, Automatic real-time crack detection using lightweight deep learning models, *Engineering Applications of Artificial Intelligence* 138 (2024) 109340. <https://doi.org/10.1016/j.engappai.2024.109340>.
- [17] R. Carvalho, R. Nascimento, T. D’Angelo, S. Delabrida, A.G.C. Bianchi, R.A.R. Oliveira, H. Azpúrua, L.G. Uzeda Garcia, A UAV-Based Framework for Semi-Automated Thermographic Inspection of Belt Conveyors in the Mining Industry, *Sensors* 20 (2020) 2243. <https://doi.org/10.3390/s20082243>.

- [18] W. Ding, H. Yang, K. Yu, J. Shu, Crack detection and quantification for concrete structures using UAV and transformer, *Automation in Construction* 152 (2023) 104929. <https://doi.org/10.1016/j.autcon.2023.104929>.
- [19] N. Kassotakis, V. Sarhosis, M.V. Peppas, J. Mills, Quantifying the effect of geometric uncertainty on the structural behaviour of arches developed from direct measurement and Structure-from-Motion (SfM) photogrammetry, *Engineering Structures* 230 (2021) 111710. <https://doi.org/10.1016/j.engstruct.2020.111710>.
- [20] J.L. Schönberger, J.-M. Frahm, Structure-from-motion revisited, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. <https://doi.org/10.1109/CVPR.2016.445>.
- [21] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, V. Rodehorst, Framework for automated UAS-based structural condition assessment of bridges, *Automation in Construction* 97 (2019) 77–95. <https://doi.org/10.1016/j.autcon.2018.10.006>.
- [22] M. Goesele, B. Curless, S.M. Seitz, Multi-View Stereo Revisited, in: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, IEEE, New York, NY, USA, 2006: pp. 2402–2409. <https://doi.org/10.1109/CVPR.2006.199>.
- [23] Y. Xu, J. Zhang, UAV-based bridge geometric shape measurement using automatic bridge component detection and distributed multi-view reconstruction, *Automation in Construction* 140 (2022) 104376. <https://doi.org/10.1016/j.autcon.2022.104376>.
- [24] D. Ma, H. Fang, N. Wang, G. Pang, B. Li, J. Dong, X. Jiang, A low-cost 3D reconstruction and measurement system based on structure-from-motion (SfM) and multi-view stereo (MVS) for sewer pipelines, *Tunnelling and Underground Space Technology* 141 (2023) 105345. <https://doi.org/10.1016/j.tust.2023.105345>.
- [25] J. Chen, Y.K. Cho, CrackEmbed: Point feature embedding for crack segmentation from disaster site point clouds with anomaly detection, *Advanced Engineering Informatics* 52 (2022) 101550. <https://doi.org/10.1016/j.aei.2022.101550>.
- [26] M. Makuch, P. Gawronek, 3D Point Cloud Analysis for Damage Detection on Hyperboloid Cooling Tower Shells, *Remote Sensing* 12 (2020) 1542. <https://doi.org/10.3390/rs12101542>.
- [27] C. Zhang, J. Shu, H. Zhang, Y. Ning, Y. Yu, Estimation of load-carrying capacity of cracked RC beams using 3D digital twin model integrated with point clouds and images, *Engineering Structures* 310 (2024) 118126. <https://doi.org/10.1016/j.engstruct.2024.118126>.
- [28] S. Zhao, F. Kang, J. Li, Intelligent segmentation method for blurred cracks and 3D mapping of width nephograms in concrete dams using UAV photogrammetry, *Automation in Construction* 157 (2024) 105145. <https://doi.org/10.1016/j.autcon.2023.105145>.
- [29] R. Zeibak-Shini, R. Sacks, L. Ma, S. Filin, Towards generation of as-damaged BIM models using laser-scanning and as-built BIM: First estimate of as-damaged locations of reinforced concrete frame members in masonry infill structures, *Advanced Engineering Informatics* 30 (2016) 312–326. <https://doi.org/10.1016/j.aei.2016.04.001>.
- [30] D. Isailović, V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, J. Döllner, Bridge damage: Detection, IFC-based semantic enrichment and visualization, *Automation in Construction* 112 (2020) 103088. <https://doi.org/10.1016/j.autcon.2020.103088>.
- [31] A. Vetrivel, M. Gerke, N. Kerle, G. Vosselman, Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images, *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015) 61–78. <https://doi.org/10.1016/j.isprsjprs.2015.03.016>.
- [32] J.-L. Xiao, J.-S. Fan, Y.-F. Liu, B.-L. Li, J.-G. Nie, Region of interest (ROI) extraction and crack detection for UAV-based bridge inspection using point cloud segmentation and 3D-to-2D projection, *Automation in Construction* 158 (2024) 105226. <https://doi.org/10.1016/j.autcon.2023.105226>.
- [33] H. Song, X. Zhu, H. Li, G. Yang, Multimodal deep learning-based automatic generation of repair proposals for steel bridge shallow damage, *Automation in Construction* 171 (2025) 105961. <https://doi.org/10.1016/j.autcon.2025.105961>.
- [34] Z. Liang, J.C.F. Chan, J. Zhang, Z. Liang, B. Wang, M. Wang, J.C.P. Cheng, Multimodal LLM-driven language-embedded 3D gaussian splatting for semantic and realistic digitization of historical buildings, *Automation in Construction* 181 (2026) 106628. <https://doi.org/10.1016/j.autcon.2025.106628>.

- [35] X. Luo, Y. Huang, H. Li, Y. Mei, F. Zhang, M. Ma, A lightweight neural radiance field model with automatic semantic segmentation for post-earthquake building three-dimensional reconstruction, *Computer Aided Civil Eng* (2025) mice.70031. <https://doi.org/10.1111/mice.70031>.
- [36] G. Kim, Y. Cha, 3D Pixelwise damage mapping using a deep attention based modified Nerfacto, *Automation in Construction* 168 (2024) 105878. <https://doi.org/10.1016/j.autcon.2024.105878>.
- [37] Y. Liu, H. Guan, C. Luo, L. Fan, N. Wang, J. Peng, Z. Zhang, CityGaussian: Real-time High-quality Large-Scale Scene Rendering with Gaussians, (2024). <https://doi.org/10.48550/arXiv.2404.01133>.
- [38] A. Guédon, V. Lepetit, SuGaR: Surface-Aligned Gaussian Splatting for Efficient 3D Mesh Reconstruction and High-Quality Mesh Rendering, (2023). <http://arxiv.org/abs/2311.12775> (accessed October 21, 2024).
- [39] H. Chen, C. Li, G.H. Lee, NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance, (2023). <https://doi.org/10.48550/arXiv.2312.00846>.
- [40] Z. Yang, H. Yang, Z. Pan, L. Zhang, Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting, (2024). <https://doi.org/10.48550/arXiv.2310.10642>.
- [41] M. Ye, M. Danelljan, F. Yu, L. Ke, Gaussian Grouping: Segment and Edit Anything in 3D Scenes, (2024). <https://doi.org/10.48550/arXiv.2312.00732>.
- [42] Y. Song, H. Lin, J. Lei, L. Liu, K. Daniilidis, HDGS: Textured 2D Gaussian Splatting for Enhanced Scene Rendering, (2024). <https://doi.org/10.48550/arXiv.2412.01823>.
- [43] Y. Liu, C. Luo, Z. Mao, J. Peng, Z. Zhang, CityGaussianV2: Efficient and geometrically accurate reconstruction for large-scale scenes, (2024). <https://doi.org/10.48550/arXiv.2411.00771>.
- [44] W. Liu, Y. Zhong, Y. Li, X. Chen, J. Cui, H. Zhang, L. Xu, X. Lou, Y. Shi, J. Yu, Y. Zhang, CityGo: Lightweight Urban Modeling and Rendering with Proxy Buildings and Residual Gaussians, (2025). <https://doi.org/10.48550/arXiv.2505.21041>.
- [45] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, Z. Wang, LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction and 200+ FPS, (2024). <https://doi.org/10.48550/arXiv.2311.17245>.
- [46] Y. Chen, Q. Wu, W. Lin, M. Harandi, J. Cai, HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression, (2024). <https://doi.org/10.48550/arXiv.2403.14530>.
- [47] H. Liu, Y. Wang, C. Li, R. Cai, K. Wang, W. Li, P. Molchanov, P. Wang, Z. Wang, FlexGS: Train Once, Deploy Everywhere with Many-in-One Flexible 3D Gaussian Splatting, (2025). <https://doi.org/10.48550/arXiv.2506.04174>.
- [48] W. Lyu, X. Li, A. Kundu, Y.-H. Tsai, M.-H. Yang, Gaga: Group Any Gaussians via 3D-aware Memory Bank, (2024). <https://doi.org/10.48550/arXiv.2404.07977>.
- [49] B. Chao, H.-Y. Tseng, L. Porzi, C. Gao, T. Li, Q. Li, A. Saraf, J.-B. Huang, J. Kopf, G. Wetzstein, C. Kim, Textured Gaussians for Enhanced 3D Scene Appearance Modeling, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. <https://doi.org/10.48550/arXiv.2411.18625>.
- [50] L. Qi, J. Kuen, W. Guo, T. Shen, J. Gu, J. Jia, Z. Lin, M.-H. Yang, High-Quality Entity Segmentation, (2023). <https://doi.org/10.48550/arXiv.2211.05776>.
- [51] T. Czerniawski, B. Sankaran, M. Nahangi, C. Haas, F. Leite, 6D DBSCAN-based segmentation of building point clouds for planar object classification, *Automation in Construction* 88 (2018) 44–58. <https://doi.org/10.1016/j.autcon.2017.12.029>.
- [52] H. Fu, D. Meng, W. Li, Y. Wang, Bridge Crack Semantic Segmentation Based on Improved Deeplabv3+, *JMSE* 9 (2021) 671. <https://doi.org/10.3390/jmse9060671>.
- [53] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, (2018). <https://doi.org/10.48550/arXiv.1802.02611>.
- [54] F.M. Nejad, N. Karimi, H. Zakeri, Automatic image acquisition with knowledge-based approach for multi-directional determination of skid resistance of pavements, *Automation in Construction* 71 (2016) 414–429. <https://doi.org/10.1016/j.autcon.2016.08.003>.
- [55] X. Wang, Laplacian Operator-Based Edge Detectors, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007) 886–890. <https://doi.org/10.1109/TPAMI.2007.1027>.
- [56] L.R. Irudhiyarajah, A. Amarasinghe, I.R. Upasiri, Predicting the Durability of Walling Units by Identifying Moss Growth Through Image Processing, in: R. Dissanayake, P. Mendis, S. De Silva,

1286 S. Fernando, U. Attanayake, P. Gajanayake (Eds.), Proceedings of the 15th International
1287 Conference on Sustainable Built Environment, Springer Nature Singapore, Singapore, 2025: pp.
1288 527–543.