
FHGS: Feature-Homogenized Gaussian Splatting

Q. G. Duan Benyun Zhao Mingqiao Han Yijun Huang Ben M. Chen[†]

Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong

Abstract

Scene understanding based on 3D Gaussian Splatting (3DGS) has recently achieved notable advances. Although 3DGS related methods have efficient rendering capabilities, they fail to address the inherent contradiction between the anisotropic color representation of gaussian primitives and the isotropic requirements of semantic features, leading to insufficient cross-view feature consistency. To overcome the limitation, we propose *FHGS* (Feature-Homogenized Gaussian Splatting), a novel 3D feature distillation framework inspired by physical models, which freezes and distills 2D pre-trained features into 3D representations while preserving the real-time rendering efficiency of 3DGS. Specifically, our *FHGS* introduces the following innovations: Firstly, a universal feature fusion architecture is proposed, enabling robust embedding of large-scale pre-trained models' semantic features (e.g., SAM, CLIP) into sparse 3D structures. Secondly, a non-differentiable feature fusion mechanism is introduced, which enables semantic features to exhibit viewpoint independent isotropic distributions. This fundamentally balances the anisotropic rendering of gaussian primitives and the isotropic expression of features; Thirdly, a dual-driven optimization strategy inspired by electric potential fields is proposed, which combines external supervision from semantic feature fields with internal primitive clustering guidance. This mechanism enables synergistic optimization of global semantic alignment and local structural consistency. Extensive comparison experiments with other state-of-the-art methods on benchmark datasets demonstrate that our *FHGS* exhibits superior reconstruction performance in feature fusion, noise suppression, and geometric precision, while maintaining a significantly lower training time. This work establishes a novel Gaussian Splatting data structure, offering practical advancements for real-time semantic mapping, 3D stylization, and Vision-Language Navigation (VLN). Our code and additional results are available on our project page: <https://fhgs.cuastro.org/>.

1 Introduction

In recent years, scene representation particularly understanding has emerged as a prominent research focus, as it enables unmanned systems to better perceive and interpret their surrounding environments. Traditional scene representation frameworks such as Multi-View Stereo (MVS) [1] and Simultaneous Localization and Mapping (SLAM) [2] can achieve geometric reconstruction. However, these methods rely on non-differentiable pipelines and remain limited in high-level semantic perception and nonlinear feature fusion. As a result, the differentiable approaches have gradually come into focus. Among them, Neural Radiance Fields (NeRF) [3] and 3D Gaussian Splatting (3DGS) [4] have revolutionized the scene representation framework. NeRF models implicit radiance fields and learns continuous 3D spatial representations under 2D image supervision via differentiable volume rendering

[†]Corresponding author.

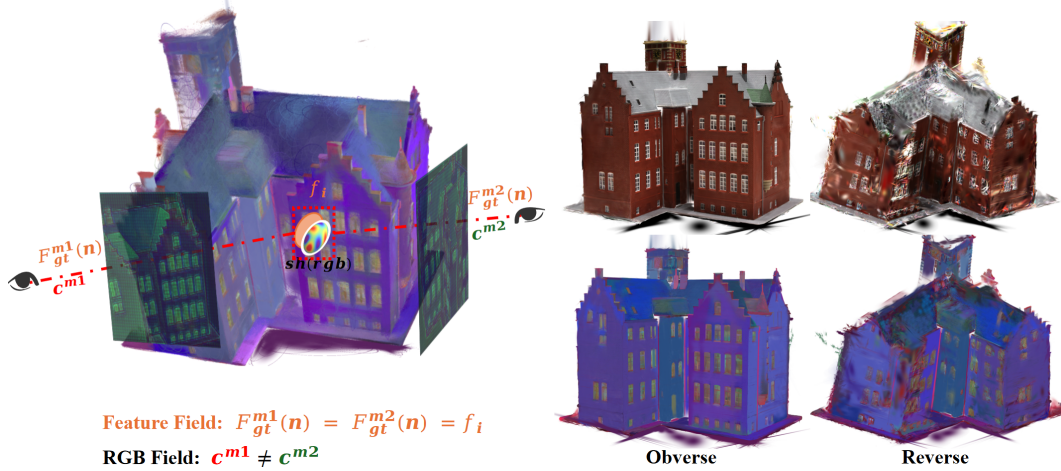


Figure 1: The left part demonstrates the inherent contradiction between the anisotropic color $\mathbf{c}$ of gaussian primitives in RGB field and the isotropic requirement of semantic features $\mathbf{f}$. The right part shows the results in obverse and reverse, indicating that *FHGS* shows superior reconstruction performance in terms of feature fusion, noise suppression, and geometric accuracy.

equations, whereas 3DGS adopts explicit anisotropic Gaussian primitives to enable high-quality reconstruction through efficient rasterization. However, these traditional neural field representations primarily focus on the fusion of RGB geometric fields, with limited exploitation of semantic features. In contrast, feature fields require maintaining semantic consistency across multiple viewpoints to prevent contradictory predictions during viewpoint transitions [5], as shown in Fig. 1.

Against this backdrop, the integration of semantic feature from transformer-based models [6, 7] fusing with NeRF and 3DGS framework has begun to emerge. NeRF-based frameworks extend radiance fields by incorporating learnable semantic feature fields, implicitly enforcing multi-view semantic consistency through continuous neural representations [8, 9]. However, their inference speed remains limited due to the dense sampling required by volumetric rendering. In contrast, 3DGS-based frameworks [10] construct explicit feature fields by directly associating semantic features with their corresponding explicit primitives. However, as shown in the Fig. 1, an inherent conflict arises between the anisotropic nature of RGB fields in their rasterization pipeline and the isotropic representation required for robust semantic features. Existing methods, whether based on implicit or explicit representations, typically treat features as fully differentiable and optimize them jointly with appearance. However, this continuous optimization may introduce inconsistencies that interfere with the self-attention mechanism in transformer, leading to feature noise and degraded rendering quality.

To address the limitations of aforementioned phenomena, we propose *FHGS* (Feature-Homogenized Gaussian Splatting), a novel feature fusion framework built upon the GS paradigm which establishes bidirectional associations between 2D semantic features and 3D feature fields, enabling end-to-end optimization for multi-view consistent feature fusion. *FHGS* preserves the efficiency and explicitness of the gaussian splatting while overcoming the limitations of rasterization-based methods designed primarily for RGB reconstruction. Specifically, each gaussian primitive is augmented with non-differentiable semantic features, which represent the frozen pre-trained features within the 3D space and directly supervised by ground-truth feature maps to enforce semantic consistency across views. To better achieve multi-view feature consistency under efficient optimization while preserving isotropic representations within the feature field, we propose a dual-driven mechanism inspired by physics-inspired principles from electric field modeling. This mechanism, composed of *External Potential Field Driving* and *Internal Feature Clustering Driving*, constrains anisotropy to photometric properties merely, while enforcing isotropy in the feature field to support consistent semantic representation.

Extensive experiments of benchmark datasets demonstrate the proposed *FHGS* not only enhances semantic fusion quality but also improves geometric reconstruction accuracy and noise robustness through feature-driven regularization effects. The main contributions of this work are as follows:

- General feature fusion architecture: We propose a GS-based feature field fusion framework capable of integrating 2D semantic features extracted from large-scale pre-trained models (e.g., SAM [11], CLIP [12]), enabling unified optimization from low-level geometry to high-level semantics.
- Integration of non-differentiable features into GS framework: We pioneer about integrating of frozen features into the differentiable gaussian splatting methods, which fundamentally resolves the inherent contradiction between the anisotropic nature of gaussian primitives and the isotropic requirements of semantic features.
- Physics-inspired dual-drive mechanism: Inspired from electric field modeling, we design a joint optimization strategy combining external potential field driving and internal feature clustering driving, characterized by intuitive logic, computational efficiency, and strong interpretability. Additionally, the metric based on this mechanism, named FE, is proposed to evaluate the global consistency of features.
- Performance superiority: Compared with other 3DGS feature fusion frameworks on benchmark datasets, our *FHGS* achieves state-of-the-art fusion performance and enhances geometric reconstruction accuracy, while maintaining a significantly lower training time than our baseline. Therefore, our framework lays the foundation for future extensions toward real-time reconstruction and VLN applications.

2 Related Work

2.1 Novel View Synthesis

Neural Radiance Fields [3] models a continuous 3D scene representation through an implicit radiance field and a differentiable volume rendering equation supervised by 2D images. The core of NeRF is that it leverages a multilayer perceptron (MLP) to map spatial positions and viewing directions to color and density values, enabling novel view synthesis with high-quality via ray integration. Subsequent works such as Mip-NeRF [13], Instant-NGP [14], and Mip-NeRF 360 [15] further improve anti-aliasing, training speed, and scalability to large-scale unbounded scenes. However, the nature of implicit representation of NeRF-based methods [13, 14, 15] requires dense sampling and complex network inference, leading to low training and rendering efficiency that limits its applicability in real-time scenarios.

To overcome the limitations of implicit representations, 3D Gaussian Splatting (3DGS) [4] introduces an explicit scene representation by decomposing the 3D environment into a set of explicit, anisotropic gaussian primitives. Combined with the implicit pipeline, this formulation enables efficient training and rendering. Moreover, the anisotropic view-dependent appearance is further represented using spherical harmonics, conditioned on the spatial and radiometric properties of each primitive. Compared to NeRF, 3DGS eliminates the need for neural networks in the rendering pipeline, significantly improving memory efficiency and real-time performance while maintaining high-fidelity reconstruction. Extending this idea, 2D Gaussian Splatting (2DGS) [16] enhances multi-view geometric consistency by anchoring gaussian primitives to the image plane while enforcing depth consistency constraints. Despite these advances, existing GS-based frameworks remain primarily focused on geometry reconstruction, without addressing a core challenge of the utilization of semantic features. Our method *FHGS* introduces high-level semantic priors from SAM [11], enabling structure-aware guidance during reconstruction. Different from conventional GS methods that rely purely on differentiable photometric cues, our *FHGS* leverages non-differentiable, high-dimensional semantic information to guide the optimization of semantic-aware structural distributions, resulting in more precise and robust reconstructions, especially in challenging regions where appearance cues alone are insufficient.

2.2 Implicit Feature Fusion

Integrating semantic information or learned features into point-based scene representations is a well-established strategy, extensively explored in the NeRF-based works [17, 18, 19, 20, 21, 22] and now migrating to gaussian splatting. Recent attempts to incorporate features into 2D/3D gaussian splatting can be clustered into three categories. Mask fusion approaches exemplified by GaussianCut [23] and Gaussian Grouping [24] provided 2D masks onto the gaussian primitives set and employ graph-cut

or low-dimensional identity embeddings for partitioning. While this method is straightforward for interactive 2D editing, the resulting correspondence is no longer perceptually obvious in 3D space, and these method still depends on extensive manual annotation, which fails to capture any high-dimensional semantic features. External fusion schemes such as SAGA [25], Semantic Gaussians [26] and OmniSeg3D [27] distilled 2D features into 3D space through an auxiliary neural network or contrastive learning, thereby enriching semantic information at the expense of additional parameters, prolonged training, and deviation from the concise design philosophy of gaussian splatting. Feature fusion techniques including Feature 3DGS [10] and LangSplat [28] learned embeddings with individual primitive so that semantics render with color. However, these embeddings overwrite or reshape the high-dimensional tensors supplied by large segmentation models, erasing the self-attention structure and class relationships encoded therein and often introducing substantial noise that degrades segmentation quality. As a consequence, subsequent reasoning is confined to image space rather than the gaussian primitives domain.

Therefore, we model cross-view semantic coherence as a physics-inspired potential-field optimization that relocates gaussian primitives while preserving their original feature vectors. The pipeline of our *FHGS* is self-supervised and globally consistent, retains the full high-dimensional semantic tensor for downstream tasks such as segmentation, detection and multi-modal prompting, and maintains the real-time rendering performance fundamental to gaussian splatting.

3 Methodology

The proposed *FHGS* addresses the semantic distortion and efficiency bottlenecks caused by the conflict between the anisotropic rendering mechanism of gaussian splatting and the isotropic requirements of high-level semantic features. There are three core components of *FHGS*, which will be successively illustrated in this section: (1) A general-purpose feature fusion architecture that supports the integration of multi-view features. (2) A GS framework enhanced with non-differentiable features, enabling the incorporation of high-dimensional semantic priors. (3) A dual-driven feature fusion mechanism inspired by physical modeling, which guides the feature optimization process using both geometric and semantic consistency cues.

3.1 General Feature Fusion Architecture

The pipeline of the General Feature Fusion Architecture is demonstrated in Fig. 2: The Structure from Motion (SFM) process reconstructs a sparse 3D point cloud PC via Bundle Adjustment (BA) firstly. To accelerate the correspondence of 3D point cloud and 2D feature, we construct a spatial hash table $\mathcal{H}$ that indexes the projections of each 3D point pc_i across visible views M . Subsequently, pre-trained model of segmentation is used to generate 2D ground-truth feature maps $\mathbf{F}_{gt}^m$. Given a 3D point pc_i , its corresponding pixel n in a randomly selected view $m \in M$ is retrieved via the spatial hash table, and the semantic feature $\mathbf{f}_i = \mathbf{F}_{gt}^m(n)$ is sampled accordingly. Each gaussian primitive is

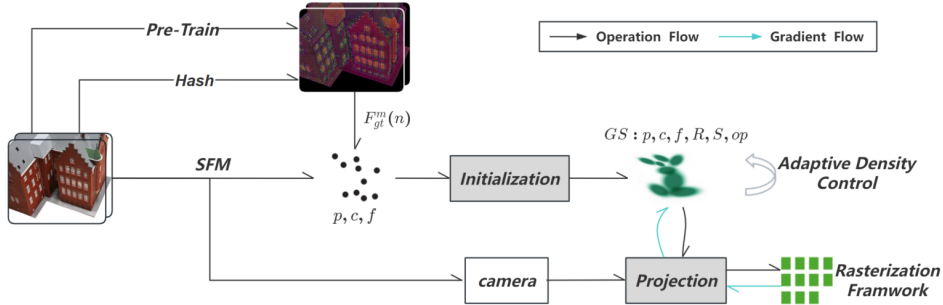


Figure 2: Pipeline of the General Feature Fusion Architecture

initialized from a point pc_i and inherits its geometric parameters, appearance attributes, and a frozen semantic feature. Specifically, a primitive is anchored at a center position $\mathbf{p}_i$ and oriented by two orthogonal tangent directions $\mathbf{t}_u$ and $\mathbf{t}_v$, with the normal vector defined as $\mathbf{t}_w = \mathbf{t}_u \times \mathbf{t}_v$. These directions form the rotation matrix $\mathbf{R}_i = [\mathbf{t}_u, \mathbf{t}_v, \mathbf{t}_w] \in \mathbb{R}^{3 \times 3}$. The spatial extent on the tangent plane is described by a planar scale vector $\mathbf{S}_i = (s_u, s_v)$. For appearance modeling, each primitive carries

an RGB color $\mathbf{c}_i \in \mathbb{R}^3$ and an opacity scalar $op_i \in [0, 1]$. Additionally, a frozen semantic feature vector $\mathbf{f}_i$ is assigned to each primitive, extracted via a non-differentiable image embedding. We write the complete representation of each gaussian primitive as:

$$\theta_i = \{\mathbf{p}_i, \mathbf{R}_i, \mathbf{S}_i, op_i, \mathbf{c}_i, \mathbf{f}_i\}. \quad (1)$$

The other symbols follow the notation of conventional 3DGS. Any point (u, v) in the tangent plane is mapped to world space by:

$$P(u, v) = \mathbf{p}_i + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v = \mathbf{H}(u, v, 1, 1)^\top \quad (2)$$

with the homogeneous matrix $\mathbf{H} \in \mathbb{R}^{4 \times 4}$ factories translation, rotation and scale. Given local coordinates $\mathbf{u} = (u, v)$, the unnormalized density is $\mathcal{G}(\mathbf{u}) = \exp(-\frac{u^2+v^2}{2})$. Then, let $\mathbf{x} = (x, y)$ be a pixel and define $\mathbf{u}(\mathbf{x})$ as the unique point in the splat's tangent plane whose homogeneous coordinates satisfy:

$$\mathbf{x} = (xz, yz, z, 1)^\top = \mathbf{W}P(u, v) = \mathbf{W}\mathbf{H}(u, v, 1, 1)^\top \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{4 \times 4}$ is the world-to-camera transformation matrix, and z denotes the depth. During the rasterization process in the GS framework, primitives that intersect with the ray l emitted from pixel n are identified. Specifically, the N primitives covered by ray l are sorted by their rendering depth, with index $i = 1$ and $i = N$ assigned to the farthest and the nearest, respectively. The final color can be computed as:

$$\mathbf{c}(\mathbf{x}) = \sum_{i=1}^N \mathbf{c}_i w_i \quad (4)$$

The weight $w_i = \alpha_i T_i$ is the dynamic differentiable parameter, while $\alpha_i = op_i \mathcal{G}_i(\mathbf{u}(\mathbf{x}))$ characterizes the intrinsic properties of gaussian primitives, and $T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$ encodes their transmittance. During the backward propagation, gradients of w_i propagate through the chain rule to drive the optimization of the geometric parameters of gaussian primitives, thereby enhancing scene reconstruction quality. As the pivotal variable linking geometry and the differentiable rasterization, w_i directly drives both reconstruction accuracy and rendering efficiency.

3.2 Non-Differentiable Features Fusion Mechanism

FHGS integrates a non-differentiable feature driving (NDFD) (orange arc pathway in Fig. 3) with the original GS framework. During the forward process, *FHGS* directly utilizes $\mathbf{F}_{gt}^m$ compute the feature loss L_{feat} based on the feature $\mathbf{f}_i$ and contribution weights w_i . It is worth noting that the forward process does not require prior feature rendering, which can further reduce the computational costs. In the backward process, although the feature $\mathbf{f}_i$ of each gaussian primitive is non-differentiable, the gradient of L_{feat} can still propagate through w_i to optimize $\{\mathbf{p}_i, \mathbf{R}_i, \mathbf{S}_i, op_i\}$, implicitly guiding gaussian primitives toward feature-consistent regions.

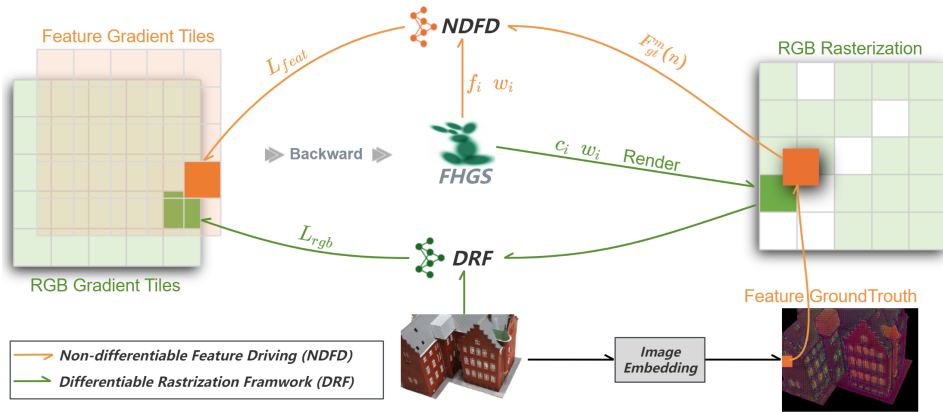


Figure 3: Schematic representation of the two mechanisms of *FHGS*: NDFD and DRF

Compared to the differentiable rasterization framework (DRF) in the conventional GS methods (green arc pathway in Fig. 3), the non-differentiable branch eliminates the need for feature rendering

during the forward process, enabling direct loss computation while preserving the efficiency of GS framework. This design brings the following characteristics: the anisotropic color rendering remains dedicated to illumination and shadow modeling, while the multi-view consistency of non-differentiable features is achieved through w -driven distribution optimization, thereby avoiding direct conflicts between the rasterization anisotropic mechanism and the isotropic requirements of semantic features. The detailed pseudo code of NDFD is given in the Appendix.

3.3 Physics-Inspired Dual-Drive Mechanism

Inspired by principles from an intuitive analogy in physical field theory, we model the $\mathbf{F}_{gt}^m$ within the rasterization as a feature field in homogeneous space $\mathbf{x}$, as defined in the Eq. 3, we consider it as an "electric field". More concretely, as illustrated in the Fig. 4, we treat the ray l emitted from pixel n as an electric field line, and define its semantic property as the ground-truth feature $\mathbf{f}_{gt} = \mathbf{F}_{gt}^m(n)$ sampled from the 2D feature map. The gaussian primitives are conceptualized as discrete "charges" carrying intrinsic features $\mathbf{f}_i$. The feature loss L_{feat} is then formulated as the potential energy loss in this electric field analogy. During the backward process, gradients drive the spatial optimization of gaussian primitives, analogous to the motion of charges under electric field forces toward regions of lower potential energy.

External Potential Field Constraint: Following the logic of NDFD, we compute the cumulative similarity between the features $\mathbf{f}_i$ intersecting with ray l and the ground-truth feature $\mathbf{f}_{gt}$, constructing a similarity loss during the forward process:

$$L_{gt} = \sum_{i=1}^N w_i \sigma_i \quad (5)$$

To eliminate the inherent contradiction between gaussian primitives and ground-truth semantics in the feature space, *FHGS* introduces a similarity-based activation function $\sigma_i = \frac{1}{1+e^{k(\varphi-\lambda)}}$, where $\varphi = \cos \langle \mathbf{f}_i, \mathbf{f}_{gt} \rangle$. This sigmoid function maps feature similarity into a polarity-like response, analogous to the binary behavior of electric charges. More detailed explanation of sigmoid function are given in the Appendix.

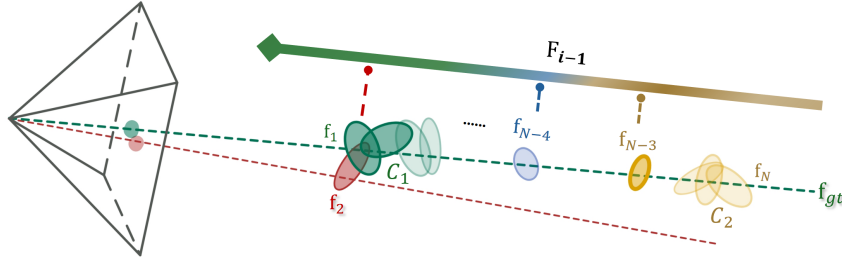


Figure 4: The illustration of proposed Dual-Drive Mechanism: The color of each gaussian primitive and ray represents their feature properties, and the transparency represents the magnitude of the weight w_i of the primitive on the ray. $\mathbf{f}_{N-3}$ exhibits similarity to posterior accumulated values $\mathbf{F}_{i-1}$, which is the value of the cluster C_2 only constrained by L_{gt} ; $\mathbf{f}_{N-4}$ represents the inter-cluster noise points of C_1 and C_2 suppressed by L_{gt} and L_{cf} ; $\mathbf{f}_2$ corresponds to the internal noise points from cluster C_1 , where both L_{gt} and L_{cf} effectively optimize the distribution of C_1 .

Internal Clustering Driving: In order to suppress noise, enhance semantic coherence, and quantify the semantic feature entropy at pixel n , we simplify the bidirectional traversal of feature similarity between gaussian primitives during the rasterization process as:

$$\begin{aligned} L_{cf} &= \sum_{i=1}^N \sum_{j=1}^{i-1} \sigma_i w_i w_j (1 - \mathbf{f}_i \cdot \mathbf{f}_j) \\ &= \sum_{i=1}^N \sigma_i w_i (W_{i-1} - \mathbf{F}_{i-1} \cdot \mathbf{f}_i) \end{aligned} \quad (6)$$

The detailed derivation process of L_{cf} can be found in the Appendix. Since both $\mathbf{f}_i$ and $\mathbf{f}_j$ are normalized, $\cos \langle \mathbf{f}_i, \mathbf{f}_j \rangle$ simplifies to $\mathbf{f}_i \cdot \mathbf{f}_j$. We can obtain the cumulative weight $W_n = \sum_{i=1}^n w_i$ and cumulative feature $\mathbf{F}_n = \sum_{i=1}^n w_i \mathbf{f}_i$ along the ray from far to near. In addition, each current feature $\mathbf{f}_i$ is compared only with the cumulative feature $\mathbf{F}_{i-1}$. This avoids spurious repulsion across unrelated objects and reduces complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. Furthermore, the cumulative weight W_{i-1} encodes the rendering contributions of farther gaussian primitives, implicitly modeling depth hierarchy. More specifically, the similarity activation function σ_i suppresses interference from background clutter, preventing incorrect contributions to foreground semantic clusters (e.g., C_1 and $\mathbf{f}_1$ in the Fig. 4). This mechanism achieves local semantic coherence by anchoring primitives that are semantically consistent with $\mathbf{f}_{gt}$ (e.g., $\mathbf{f}_1$), while repelling dissimilar ones, thereby reducing feature conflicts and reinforcing cluster purity. It effectively suppresses internal noise (e.g., $\mathbf{f}_2$ in the Fig. 4) and eliminates irrelevant outliers in space (e.g., C_2 and $\mathbf{f}_{N-3}$ in the Fig. 4), resulting in cleaner and more compact semantic regions.

The aforementioned two driving methods, together with L_{rgb} , jointly constrain the semantic fusion process of 3D scenes. The external potential field driving ensures semantic consistency across views, while the internal clustering suppresses outlier noise and enhances intra-cluster coherence. Moreover, the internal-clustering term L_{cf} refines the fine-grained details captured by L_{gt} and accelerates its convergence. Two hyper-parameters λ_1 and λ_2 are manually selected to balance the contribution of external semantic guidance and internal clustering regularization, respectively. Finally, we define the overall loss L as:

$$L = L_{rgb} + \lambda_1 L_{gt} + \lambda_2 L_{cf}$$

Under the NDFD mechanism, gradient with respect to w_k not only influence the geometry and appearance of local primitive but also affect the spatial distribution of subsequent gaussian primitives in the backward traversal. The gradient can be obtained by:

$$\frac{\partial L_{cf}}{\partial w_k} = \sigma_k(W_{k-1} - \mathbf{F}_{k-1} \cdot \mathbf{f}_k) + \sum_{i=k+1}^N \sigma_i w_i (1 - \mathbf{f}_i \cdot \mathbf{f}_k)$$

The symmetry between the forward and backward passes allows cumulative terms computed during the forward traversal to be directly reused in gradient calculations (see Appendix), eliminating redundant passes and preserving the $\mathcal{O}(N)$ complexity of both processes.

SS

4 Experiments

We implement *FHGS* within a 2DGS-based framework, deploying tailor-made CUDA kernels to accelerate the proposed feature-fusion operations. We use the image embedding of SAM [11] as input to the feature. The original 2DGS renderer is retained to export depth-distortion maps, depth maps, normal maps, and mesh reconstructions, which serve as the inputs to our quantitative and qualitative evaluations. In the sigmoid activation function, the similarity threshold and slope are empirically fixed to $\lambda = 0.5$ and $k = 20$, respectively, ensuring stable binarization of the feature-matching score σ that governs the polarity of gaussian primitive. For benchmarking, we adopt Feature3DGS [10] as the baseline. Following its protocol, we report the L_1 feature loss FL1 (lower values indicating higher feature similarity) under the same rendering pipeline, where smaller FL1 values signify better feature fusion. Cross-view consistency is further assessed with the ground-truth entropy metric L_{gt} (Eq. 5); lower L_{gt} scores indicate tighter multi-view alignment. To ensure fair comparisons, all experiments are executed on a workstation equipped with a single NVIDIA GeForce RTX 4090 (24 GB) and an AMD Ryzen 9 9950X (16 cores). In addition, we use identical feature-extraction pipelines together with the default 2DGS optimizer settings (learning rate, iteration count, batch size) for both the baseline and our method, thereby eliminating performance biases due to hyperparameter tuning or feature-generation differences.

4.1 Comparative experiment

To verify the generalization and robustness of our method, we conduct systematic experiments on a range of public datasets covering both indoor and outdoor environments. For indoor evaluations, we

Table 1: Quantitative results comparison on indoor scenes

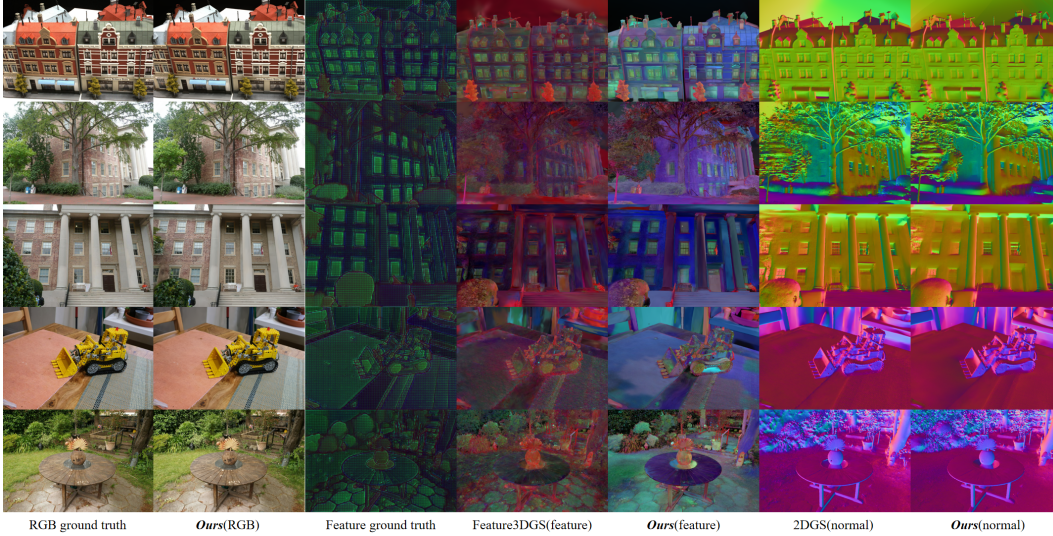
Method	DTU-24 [29]				DTU-37 [29]				MN360-kitchen [15]			
	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$
2DGS	30.1	1.35	0.61	6.1m	30.5	1.31	0.52	6.3m	30.2	1.32	0.79	6.5m
Feature3DGS	31.5	0.52	0.24	82.2m	31.1	0.88	0.31	73.2m	31.7	0.63	0.31	113.2m
FHGS (ours)	30.9	0.15	0.22	5.2m	30.8	0.21	0.18	5.7m	30.8	0.23	0.21	5.1m

evaluate our method on DTU (scans 24, 37) [29] and Mip-NeRF 360 (Kitchen) [15], as the results shown in Table 1, while outdoor evaluations are performed on Mip-NeRF 360 (Garden, Stump) and Tanks and Temples (TnT Caterpillar) [30], the results are shown in the Table 2. All input images are uniformly downsampled to a maximum side length of 1,000 pixels to balance computational efficiency and reconstruction accuracy. Sparse point clouds are initialized with COLMAP [31] are used for SfM, with a fixed iteration count of 10,000 to ensure optimization consistency. During testing, Feature3DGS [10] failed in TnT [30] due to its huge utilization of GPU memory.

The experimental results demonstrate that *FHGS* reduces training time by $15\times$ relative to Feature3DGS, improves performance by 8–10% over standard 2DGS, and maintains real-time rendering at ≥ 60 FPS. In terms of feature fusion quality, *FHGS* achieves the same performance comparable to Feature3DGS in the FL1 metrics, validating its effectiveness in feature similarity measurement. *FHGS* also exhibits superior FE metrics (lower values denote stronger cross-view consistency), highlighting its advantage in semantic coherence across viewpoints.

Table 2: Quantitative results comparison on outdoor scenes

Method	COLMAP [31]				MN360-Garden [15]				TnT-Caterpillar [30]			
	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$
2DGS	27.4	1.73	0.83	10.16m	31.3	1.67	0.75	6.3m	26.8	1.72	0.76	5.2m
Feature3DGS	28.2	0.55	0.42	181m	31.6	0.65	0.33	155.4m	-	-	-	-
FHGS (ours)	26.5	0.25	0.24	7.8m	30.6	0.25	0.18	6.1m	26.6	0.21	0.41	5.2m

Figure 5: Qualitative results to compare our *FHGS* with Feature3DGS [10] in feature map and 2DGS [16] in normal map.

To visually assess fusion quality, we map channels 15, 28 and 31 of the image embedding features to RGB for rendering. The visualization comparison results in Fig. 5 further demonstrates the superiority of our *FHGS* which produces uniform feature distributions with minimal noise, smooth semantic transitions, and clear boundaries. For geometric reconstruction, our method effectively suppresses noise and drives geometric structures to converge toward thin planar surfaces, ultimately achieving high-precision surface reconstruction comparable to MVS. Furthermore, the quantitative results show

that our method achieves higher PSNR values than the baseline in indoor scenes, but lower PSNR in outdoor scenarios. This improvement arises because low-semantic features such as soil and shadow regions are effectively filtered out during reconstruction (See Appendix).

Our training is faster and uses less GPU memory in Table 3. These results conclusively demonstrate that *FHGS* significantly enhances semantic-geometric consistency in 3D scene representations while preserving real-time rendering efficiency through the proposed novel fusion mechanism and optimization strategy. More results of experiments can be found in the Appendix.

Table 3: Quantitative results between *FHGS*, 3DGS, 2DGS and Feature3DGS on the DTU [29], we report chamfer distance, PSNR (training-set view), reconstruction time, model size and point number.

Methods	CD $\uparrow$	PSNR $\uparrow$	Time $\downarrow$	PN $\downarrow$	MB (Storage)
3DGS	1.96	35.76	11.2m	532k	113
2DGS	0.83	33.42	5.5m	342k	52
Feature3DGS	1.85	35.25	>24h	642k	745
<i>FHGS (ours)</i>	0.75	34.21	4.8m	196k	183

4.2 Ablation Study

The ablation study is conducted on the scan24 of DTU dataset [29] with 10,000 training iterations to investigate the effects of the loss functions L_{gt} and L_{cf} on feature fusion, geometric reconstruction, and optimization efficiency (illustrated in Table 4). Fig. 6 (a) illustrates the result of image embedding from SAM [11]. The experimental results indicate that these two loss terms serve complementary roles: As illustrated in the Fig. 6 (d), when L_{gt} and L_{cf} are both disabled, and visualizing the feature through the default rendering logic, the resulting feature map diverges markedly from the ground truth and appears cluttered. When only L_{cf} is removed, although the optimization proceeds faster, the model suffers from semantic contamination and overfitting at the surface level. As shown in the Fig. 6 (c), numerous valid gaussian primitives are incorrectly discarded, leading to excessive transparency in the reconstructed geometry and severe degradation in reconstruction quality. When both L_{gt} and L_{cf} are jointly applied, the framework achieves an optimal balance: the feature consistency metric FE improves, geometric structures converge toward thin and planar forms, and convergence speed increases. Fig. 6 (b) has shown that under this configuration, the distribution of the feature field is uniform and dense, semantic boundaries are sharp and well-defined, and the reconstructed surfaces retain detailed geometric information. These findings validate the effectiveness of the dual-loss collaborative optimization strategy.

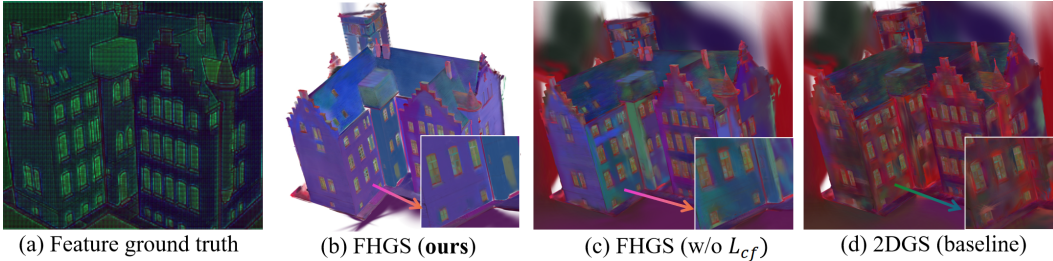


Figure 6: Ablation study on the scan24 of DTU dataset [29].

Table 4: Quantitative analysis of ablation experiments on DTU scan24

Methods	PSNR $\uparrow$	FE $\downarrow$	FL1 $\downarrow$	Time $\downarrow$	PN $\downarrow$
<i>FHGS (ours)</i>	30.9	0.15	0.16	5.2m	214k
FHGS (w/o L_{cf})	27.2	0.10	0.21	4.3m	217k
2DGS (baseline)	30.1	1.35	0.46	6.1m	329k

5 Conclusion and Discussion

We introduce a Gaussian splatting based framework named *FHGS*, which incorporates a non-differentiable feature-driven regularization term to enforce multi-view semantic consistency. *FHGS* markedly boosts multi-view feature alignment and geometric reconstruction quality while maintaining real-time performance, as demonstrated by extensive experiments on diverse indoor and outdoor datasets. While our *FHGS* successfully achieves multi-view consistent and accurate geometric reconstruction, it still has some limitations: our methods remains sensitive to the manually tuned similarity-activation parameters λ and k ; its hash-table and cumulative-weight buffers incur considerable GPU memory in large-scale scenes. In future work, we plan to explore adaptive parameter learning strategies to reduce dependence on manual tuning, and to develop memory-efficient and compact representations to enhance scalability in large-scale environments.

Acknowledgments and Disclosure of Funding

This study was supported by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics, and in part by the Research Grants Council of Hong Kong SAR under Grants 14206821, 14217922 and 14209623.

References

- [1] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.
- [2] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, Yunfan Ren, Rong Wang, Fanle Meng, and Fu Zhang. Fast-livo2: Fast, direct lidar-inertial-visual odometry. *IEEE Transactions on Robotics*, 41: 326–346, 2025. doi: 10.1109/TRO.2024.3502198.
- [3] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- [5] Shi-Sheng Huang, Haoxiang Chen, Jiahui Huang, Hongbo Fu, and Shi-Min Hu. Real-time globally consistent 3d reconstruction with semantic priors. *IEEE Transactions on Visualization and Computer Graphics*, 29(4):1977–1991, 2023. doi: 10.1109/TVCG.2021.3137912.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553.
- [8] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, June 2023.
- [9] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.

- [10] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suyu You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.
- [11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
- [13] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.
- [14] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127.
- [15] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2022. doi: 10.1109/CVPR52688.2022.00539.
- [16] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428.
- [17] Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. NeRF-SOS: Any-view self-supervised object segmentation on complex scenes. In *The Eleventh International Conference on Learning Representations*, 2023.
- [18] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lurf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023.
- [19] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- [20] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022.
- [21] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.
- [22] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, June 2023.
- [23] Umangi Jain, Ashkan Mirzaei, and Igor Gilitschenski. Gaussiancut: Interactive segmentation via graph cut for 3d gaussian splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [24] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024.

- [25] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. SAGA: Segment any 3d gaussians. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2):1971–1979, 2025. doi: 10.1609/aaai.v39i2.32193.
- [26] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting, 2024.
- [27] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20612–20622, June 2024.
- [28] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20051–20060, June 2024.
- [29] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2014. doi: 10.1109/CVPR.2014.59.
- [30] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. 36(4), 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073599.
- [31] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The main claims made in our abstract and introduction correctly reflect the contribution and scope of our paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper comprehensively discuss the limitation in the section of Conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: All the formulas in the paper are numbered and cross-referenced, the detailed proofs of core theorems are given in the section of Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We describe the two architectures in combination with the schematic diagrams in the paper. We discuss and explain our proposed algorithm through experiments on a variety of datasets.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We have prepared all the relevant code and ensured that they can be easily reproduced. If our paper is accepted, we will immediately open source all the data and code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have specified all the training and test details necessary to understand the results in the beginning of EXPERIMENTS.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided the sufficient information about the computer sources used in our experiments in the section of EXPERIMENTS.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: All the research conducted in our paper conform the NeurIPS Code of Ethics in every respect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All assets mentioned in the article have been correctly cited, explicitly mentioned and respected, and comply with all licensing and usage terms.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have provided detailed documentation for all proposed assets, and once the article is accepted, we will open source it together with all the assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [No]

Justification: [NA]

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Detailed Explanation of Non-Differentiable Feature Driving (NDFD)

Algorithm 1 General Feature Fusion and Densification Framework

```

Fgt ← GETFEATUREFROMSAM(Igt)                                ▷ Features
(p, H) ← GETPOINTFROMSFM(Igt)                                ▷ Positions & Hash
f ← POINTSFEATUREFUSION(p, F, H)                            ▷ Point Features
(R, S, c, op) ← INITATTRIBUTES()                            ▷ Rotations, Scales, Color, Opacity
i ← 0                                                         ▷ Iteration Counter
while not converged do
  (m, Igtm, Fgtm) ← SAMPLETRAININGVIEW()                ▷ Camera, Image, Feature
  Irem ← DRF(p, R, S, op, c, m)                        ▷ Rasterization
  Lrgb ← LOSS(Igtm, Irem)                                ▷ Photometric Loss
  (Lgt, Lcf) ← NDFD(p, R, S, op, f, Fgtm, m)        ▷ Semantic Loss
  L ← Lrgb + Lgt + Lcf                                ▷ Total Loss
  (p, R, S, op, c) ← ADAM(∇L)                            ▷ Update
  if ISREFINEMENTITERATION(i) then
    DENSIFICATION(p, R, S, op, c, f)                ▷ Adaptive Density
  end if
  i ← i + 1
end while

```

Details of the Rasterizer: Our implementation builds directly on the GPU rasterizer proposed in 3D Gaussian Splatting. Following that design, the image plane of size $w \times h$ is partitioned into 16×16 px tiles. Each gaussian primitive that overlaps a tile is *duplicated* for that tile and assigned a 64-bit key whose lower 32 bits encode depth and upper bits encode the tile index. A single parallel radix sort on these keys resolves global depth order and produces a compact, per-tile, depth-sorted list of instances; a second pass identifies the start–end range for each tile (see CULLGAUSSIAN, DUPLICATEWITHKEYS, and SORTBYKEYS in 3DGS). This eliminates sequential primitive traversal and maximizes GPU utilization.

Algorithm 2 Non-Differentiable Feature Driving Mechanism

```

function NDFD(p, R, S, op, f, Fgtm, m)
  x ← HOMOGENIZATION(m)                                ▷ Camera Homogenization
  g ←  $\mathcal{G}_i(\mathbf{u}(\mathbf{x}))$  ← 2DSCREENGAUSSIANS(p, R, S, x)    ▷ Screen-space Gaussians
  T ← CREATETILES(m)                                    ▷ Tile Grid
  (I, K) ← DUPLICATEWITHKEYS(g, T)                    ▷ Indices & Keys
  SORTBYKEYS(K, I)                                      ▷ Global Sort
  Tr ← IDENTIFYTILERANGES(T, K)                      ▷ Tile Ranges
  Lgt ← 0, Lcf ← 0                                    ▷ Initilize Loss Buffers
  for all t ∈ T do
    for all i ∈ t do
      r ← GETTILERANGE(Tr, t)                        ▷ Index Range in K
      for all j ∈ r do
         $\sigma_j$  ← SIGMOID(Fgtm(i), fj)                ▷ Polarity Response
        wj ← WEIGHTCALC(gj, opj)                ▷ Opacity-weighted Area
        Lgt[i] += EPFC(wj, fj,  $\sigma_j$ )            ▷ External Potential
        Lcf[i] += ICD(wj, fj,  $\sigma_j$ )            ▷ Internal Clustering
      end for
    end for
  end for
  return (Lgt, Lcf)
end function

```

Non-Differentiable Feature Driving (NDNF): Alg. 2 augments the aforementioned rasterizer with a feature-centric branch that runs entirely on the sorted gaussian primitives lists and never invokes

α blending. Given the current view m , camera homogenization first projects gaussian primitives means into screen space, after which key generation and radix sorting produce per-tile ranges. For every pixel i in a tile t , we then traverse the corresponding range r in front-to-back order. A sigmoid activation $\sigma_j = \text{SIGMOID}(\mathbf{F}_{gt}^m(i), f_j)$ converts the cosine similarity between the frozen feature f_j of the j -th gaussian primitives and the ground-truth embedding $\mathbf{F}_{gt}^m(i)$ into a charge-like polarity. The raster weight w_j , which combines projected area and opacity exactly as in α blending, is accumulated only by this feature branch. Then, two loss terms are computed: the external-potential loss L_{gt} attracts σ_j -weighted features toward $\mathbf{F}_{gt}^m(i)$, whereas the internal-clustering loss L_{cf} applies the cumulative-feature rule to penalize incoherent neighbors. These loss buffers are initialized once per frame and updated atomically in the innermost loop, so no intermediate feature image is rendered. During back-propagation, gradients propagate solely through the weights w_j , which reuse the same cumulative prefix employed for α -blending in the forward traversal, thereby retaining the $\mathcal{O}(N)$ complexity of the original rasterizer.

A.1 Derivation of the feature similarity

Internal clustering loss L_{cf} : For a given pixel p , let $\{(w_i, f_i)\}_{i=1}^N$ be the set of gaussian primitive whose screen-space footprints cover that pixel, where w_i is the weight and $f_i \in \mathbb{R}^d$ is the frozen semantic feature of the i -th primitive. The internal-clustering loss:

$$L_{cf} = \sum_{i=1}^N \sum_{j=1}^N w_i w_j (1 - \cos\langle f_i \cdot f_j \rangle) \quad (7)$$

computes the entropy of the local feature distribution by accumulating the weighted cosine dissimilarity between every pair of primitives. The process of minimizing L_{cf} pushes feature vectors of neighboring primitive to align, suppresses noisy outliers, and tightens semantic coherence within the pixel neighborhood. And simultaneously, this process allows primitives belonging to different objects to repel each other through their low cosine similarity.

We further convert it to an $\mathcal{O}(N)$ backward traversal by noting that feature vectors are normalized, therefore $\cos\langle f_i, f_j \rangle = f_i \cdot f_j$. We can rearranged the representation of L_{cf} as:

$$\begin{aligned} L_{cf} &= \sum_{i=1}^N \sum_{j=1}^{i-1} \sigma_i w_i w_j (1 - \cos\langle f_i \cdot f_j \rangle) \\ &= \sum_{i=1}^N \sigma_i w_i \left(\sum_{j=1}^{i-1} w_j - \sum_{j=1}^{i-1} w_j f_j \cdot f_i \right) \\ &= \sum_{i=1}^N \sigma_i w_i \left(\sum_{j=1}^{i-1} w_j - \sum_{j=1}^{i-1} w_j f_j \cdot f_i \right) \\ &= \sum_{i=1}^N \sigma_i w_i (W_{i-1} - F_{i-1} \cdot f_i) \end{aligned} \quad (8)$$

where the cumulative weight W_{i-1} and cumulative feature F_{i-1} are updated one time in each step during the front-to-back blend. The final form of L_{cf} retains the physical meaning of pairwise semantic attraction–repulsion so evaluates in $\mathcal{O}(N)$ time. The cumulative values of W_{N-1} and F_{N-1} are recorded.

A.2 Calculation process of the gradients

As derived in the main text, we obtain the partial derivatives:

$$\frac{\partial L_{cf}}{\partial w_k} = \sigma_k (W_{k-1} - \mathbf{F}_{k-1} \cdot \mathbf{f}_k) + \sum_{i=k+1}^N \sigma_i w_i (1 - \mathbf{f}_i \cdot \mathbf{f}_k) \quad (9)$$

In the forward pass the gaussian primitive is processed in descending depth order from the farthest to the nearest with respect to the camera. The backward pass visits the same primitive in the reverse

order. Because the index k is defined with respect to the forward ordering, we re-index the backward traversal by a new counter $q = 1, \dots, N$. Exploiting this forward-backward symmetry, the gradient of the internal-clustering loss with respect to the weight of the current primitive can be rewritten as:

$$\frac{\partial L_{cf}}{\partial w_q} = \sigma_q((W_N - W_q) - (\mathbf{F}_N - \mathbf{F}_q) \cdot \mathbf{f}_q) + \sum_{i=1}^{q-1} \sigma_i w_i (1 - \mathbf{f}_i \cdot \mathbf{f}_q) \quad (10)$$

Based on the $w_i = \alpha_i T_i$, $\alpha_i = \text{opi}\mathcal{G}_i(\mathbf{u}(\mathbf{x}))$, $T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$, we can obtain:

$$\frac{\partial L_{gt}}{\partial \alpha_k} = \frac{\partial L_{gt}}{\partial w_k} \cdot \frac{\partial w_k}{\partial \alpha_k} = T_k \cdot \frac{\partial L_{gt}}{\partial w_k} - \frac{1}{1 - \alpha_k} \sum_{i=k+1}^N \frac{\partial L_{gt}}{\partial w_i} \cdot w_i \quad (11)$$

Analogously to the equation above, we can also obtain:

$$\frac{\partial L_{gt}}{\partial \alpha_q} = \frac{\partial L_{gt}}{\partial w_q} \cdot \frac{\partial w_q}{\partial \alpha_q} = T_q \cdot \frac{\partial L_{gt}}{\partial w_q} - \frac{1}{1 - \alpha_q} \sum_{i=0}^{q-1} \frac{\partial L_{gt}}{\partial w_i} \cdot w_i \quad (12)$$

B Comprehensive Results of Experiments

We conduct a detailed comparison between our method and Feature3DGS on the DTU indoor dataset. As shown in the Fig. 7, our method yields more uniform feature distributions and sharper boundaries. Moreover, it effectively suppresses background clutter, which remains prominent in Feature3DGS. The enhanced clarity and selectivity of our features also benefit downstream tasks such as segmentation and reconstruction. These observations highlight the strength of our feature driving mechanism in promoting structural coherence and semantic focus.

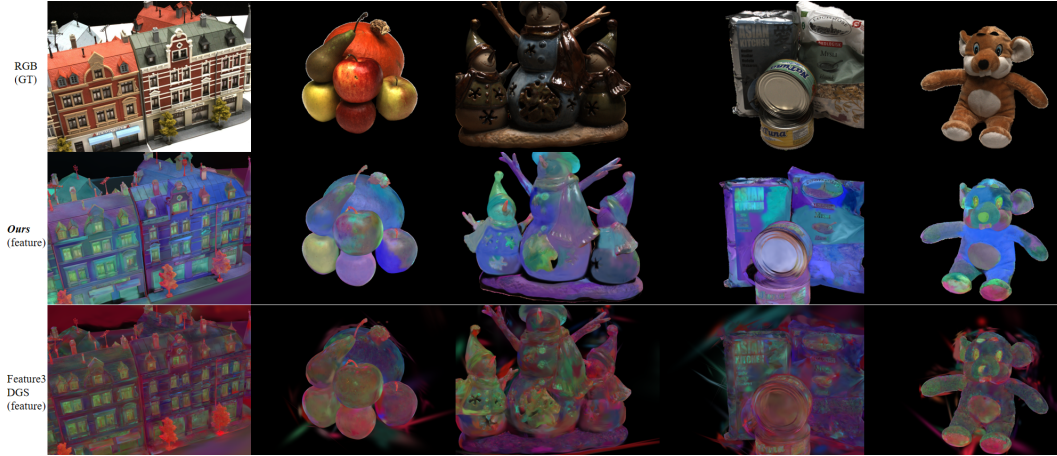


Figure 7: Qualitative results to compare our *FHGS* with Feature3DGS in feature field. The results shown that *FHGS* achieves better feature extraction with more uniform feature distributions, shaper boundaries and cleaner background.

We further evaluate our method against Feature3DGS on challenging outdoor scenes from the TnT and MipNeRF360 datasets. As shown in Fig. 8, our method consistently delivers more coherent and spatially uniform feature fields, with significantly clearer object boundaries and effective suppression of background noise. In addition to semantic features, we also visualize the surface normal maps extracted from our reconstruction, which exhibit plausible geometric structures and fine-grained surface details. These results demonstrate the robustness of our method under natural lighting, large-scale geometry, and high-frequency textures, confirming its generalization to diverse outdoor environments.

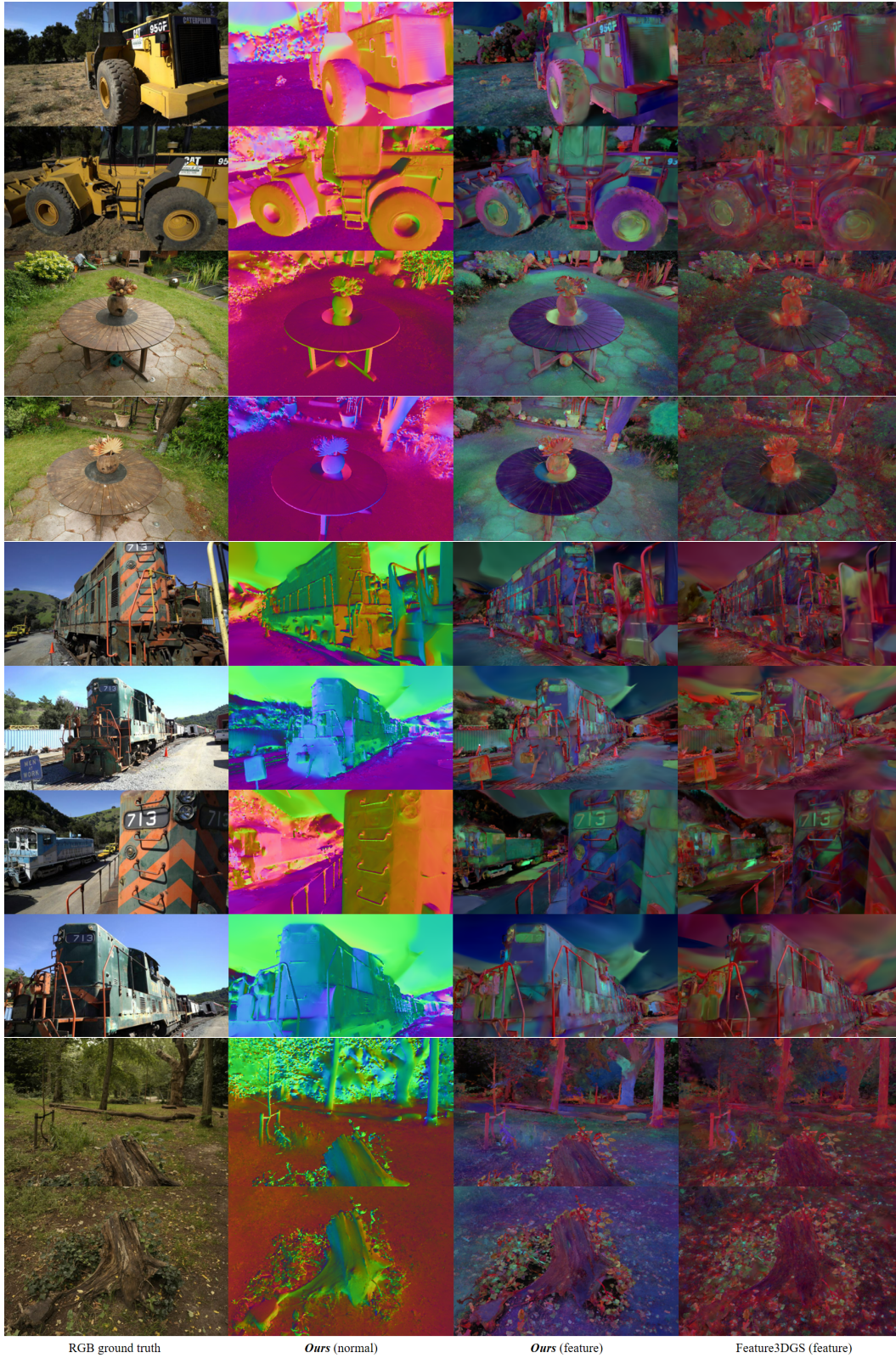


Figure 8: Qualitative comparison on outdoor scenes from TnT and MipNeRF360