Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

UGOD: Uncertainty-Guided Opacity and Dropout for Sparse-View 3D Gaussian Splatting

Zhihao Guo
Manchester Metropolitan University
Peng Wang
pw0038@surrey.ac.uk
University of Surrey
Zidong Chen
Imperial College London
Xiangyu Kong
University of Exeter
Yan Lyu
Southeast University
Guanyu Gao
Nanjing University of Science and Technology
Chenghao Qian
University of Leeds
Ziyang Wang
Aston University
Xinqi Fan
Manchester Metropolitan University
Liangxiu Han
Manchester Metropolitan University

Research Article

Keywords:

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11204608/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

UGOD: Uncertainty-Guided Opacity and Dropout for Sparse-View 3D Gaussian Splatting

Zhihao Guo[1], Peng Wang[2*], Zidong Chen[3], Xiangyu Kong[4], Yan Lyu[5], Guanyu Gao[6], Chenghao Qian[7], Ziyang Wang[8], Xinqi Fan[1], Liangxiu Han[1]

[1]Manchester Metropolitan University, Manchester, United Kingdom.
[2*]University of Surrey, Guildford, United Kingdom.
[3]Imperial College London, London, United Kingdom.
[4]University of Exeter, Exeter, United Kingdom.
[5]Southeast University, Nanjing, China.
[6]Nanjing University of Science and Technology, Nanjing, China.
[7]University of Leeds, Leeds, United Kingdom.
[8]Aston University, Birmingham, United Kingdom.

*Corresponding author(s). E-mail(s): pw0038@surrey.ac.uk;

Abstract

Sparse-view 3D Gaussian Splatting is prone to overfitting because limited observations leave many Gaussian primitives weakly constrained, yet their contributions are still accumulated through alpha blending. Without uncertainty estimation, the renderer cannot distinguish unreliable primitives from well-constrained ones, allowing their erroneous contributions to corrupt novel-view synthesis. We introduce **UGOD**, an uncertainty-guided framework that estimates a view-dependent uncertainty score for each Gaussian and uses it to regulate its rendering contribution. A lightweight uncertainty head conditioned on Gaussian attributes and viewing direction predicts this score, which then drives a differentiable opacity-modulation mechanism that attenuates high-uncertainty primitives before compositing. During training, a detached soft-dropout branch applies an uncertainty-controlled continuous keep mask to discourage the model from relying on poorly constrained Gaussians and thereby reduce overfitting. Crucially, detaching the uncertainty score prevents gradients from this stochastic regulariser from biasing or collapsing the uncertainty prediction. Experiments on Mip-NeRF 360 and LLFF show that UGOD improves sparse-view novel-view

synthesis while producing more compact Gaussian representations than the compared methods. These results demonstrate that Gaussian uncertainty provides an effective rendering-time control for sparse-view reconstruction.

Keywords: 3D Gaussian Splatting, sparse-view reconstruction, uncertainty estimation, novel-view synthesis, opacity modulation

1 Introduction

Novel-view synthesis (NVS) from a limited collection of images underpins applications such as digital twinning, augmented and virtual reality, and embodied robotics. 3D Gaussian Splatting (3DGS) [7] has become an attractive representation for this task because it models a scene as anisotropic Gaussian primitives that are projected, depth sorted, and alpha blended by a rasteriser. This design avoids the dense ray sampling required by many neural volumetric methods and enables high-quality real-time rendering. Its effectiveness nevertheless depends on the coverage, geometric accuracy, and appearance diversity provided by the input views.

These requirements are difficult to satisfy in sparse-view reconstruction. Structure-from-Motion (SfM) initialisation may contain only a limited set of points, and those points can be incomplete or inaccurate in weakly textured, occluded, or poorly observed regions. The resulting Gaussians receive insufficient multi-view supervision while their geometry and appearance are optimised to explain the available images. Critically, the failure is not only that geometry is incomplete: once a primitive is instantiated, its learned opacity still determines how strongly it participates in alpha blending. An unreliable Gaussian can therefore retain a large compositing weight, inject incorrect colour into the accumulated ray, and reinforce a training-view explanation that does not generalise. In other words, sparse-view artefacts arise when weakly supported contributions are repeatedly accumulated by the renderer.

Existing sparse-view 3DGS methods address this setting from complementary but incomplete angles. Depth or other priors strengthen geometric guidance when image evidence is scarce [20, 28]. Densification increases point coverage when SfM initialisation is too sparse [28]. Dropout- and pruning-based regularisers reduce the influence of selected primitives during optimisation [13, 24]. These strategies improve reconstruction by adding constraints, growing the representation, or filtering primitives in training. They do not, however, explicitly ask whether a Gaussian is uncertain for the *current* viewing direction at the moment of alpha compositing, nor do they provide a differentiable mechanism that attenuates its opacity before blending.

This gap motivates a rendering-centric view of sparse-view overfitting. Opacity in conventional 3DGS is a view-agnostic scalar jointly optimised with geometry and appearance; it does not separate a reliable contribution from one that is poorly supported under the present camera. Because alpha blending is order-dependent and cumulative, even a modest number of over-confident primitives can dominate transmittance and colour along a ray. Regulating opacity with a view-dependent uncertainty signal therefore targets the failure mode directly: unreliable contributions should be

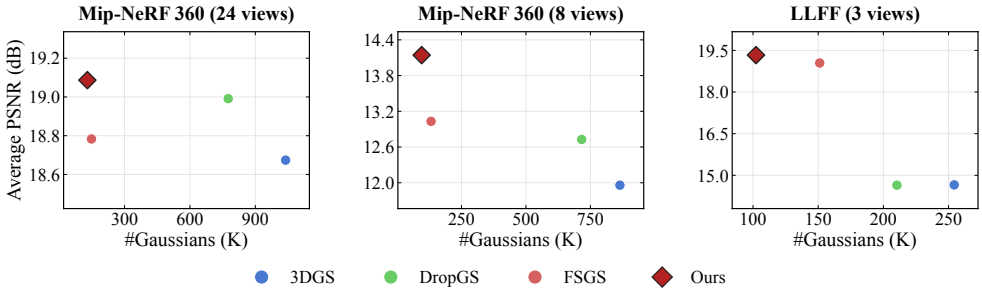

Fig. 1: Average PSNR and final-Gaussian reduction versus 3DGS* across MiP-NeRF 360 (24/8 views) and LLFF (3 views). Higher is better on every axis; UGOD leads all six.

down-weighted before they enter compositing, rather than corrected only indirectly through denser geometry or training-time dropping.

We introduce **UGOD**, an uncertainty-guided framework for sparse-view 3DGS. UGOD employs a lightweight per-Gaussian uncertainty head conditioned on Gaussian attributes, appearance features, and viewing direction to estimate view-dependent uncertainty. The normalised uncertainty drives a differentiable opacity-modulation function that attenuates the learned opacity of high-uncertainty primitives before compositing. During training, a detached soft-dropout branch converts the same uncertainty signal into a continuous keep mask, further regularising ambiguous Gaussians without allowing this stochastic branch to distort the uncertainty prediction.

Across Mip-NeRF 360 and LLFF sparse-view protocols, UGOD improves novel-view synthesis quality while producing compact Gaussian representations. These results indicate that regulating compositing with view-dependent uncertainty is an effective strategy for sparse-view 3DGS. The contributions of this work are:

- A view-dependent, per-Gaussian uncertainty prediction that combines Gaussian attributes, appearance features, and viewing direction.
- A differentiable opacity gate that attenuates high-uncertainty primitives before alpha compositing.
- A gradient-detached soft-dropout regulariser that uses the learned uncertainty to stochastically regularise high-uncertainty primitives during training without directly updating the uncertainty estimator.
- Extensive sparse-view evaluation demonstrating improved novel-view synthesis quality and Gaussian compactness relative to the compared methods.

2 Related Work

2.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents a scene using anisotropic Gaussian primitives that are projected, depth sorted, and alpha blended in image space. Its

rasterisation-based renderer provides an attractive balance between rendering quality and speed [7]. Subsequent studies have improved camera handling [23], point management, and rendering quality [2, 21, 26]. These advances make 3DGS effective for dense-view reconstruction, but the quality of the learned representation remains sensitive to the coverage and accuracy of its initialisation and observations.

Opacity is optimised jointly with geometry and appearance, and determines each primitive's contribution to alpha compositing. Existing studies have examined the distinction between opacity and extinction formulations [3], volumetrically consistent Gaussian rasterisation [16], and material-aware opacity modelling [22]. These directions improve the physical interpretation or material dependence of the rendering weights. They do not, however, explicitly ask whether a Gaussian is reliable for a particular viewing direction when the scene is weakly constrained. Our work instead models this view-dependent uncertainty and uses it to regulate opacity before alpha compositing.

2.2 Sparse 3DGS Reconstruction

Sparse-view reconstruction has been addressed through complementary forms of geometric guidance and regularisation. Prior-based methods use additional signals to compensate for incomplete image observations: SparseGS [20], for example, combines depth and diffusion priors. A second line of work seeks to improve the representation before or during optimisation. FSGS [28] densifies the initial point cloud to increase geometric coverage, which is particularly helpful when SfM produces only a small or unevenly distributed set of initial points. PcdGS [27] similarly densifies sparse SfM initialisation using mask and monocular-depth cues, whereas LiDAR-3DGS [10] supplements it with an aligned LiDAR point cloud when an additional sensor is available. These approaches improve the geometric support available at initialisation.

Other approaches act directly on the Gaussian primitives. DropGaussian [13] regularises optimisation by selectively dropping Gaussians, allowing the remaining primitives to receive more informative gradients. CoR-GS [24] uses geometric and rendering disagreement to guide co-pruning and pseudo-view co-regularisation. SuraGS [15] uses surface-aware primitives for few-shot synthesis, while HR-2DGS [17] combines depth and normal regularisation in a 2DGS representation. StruGS [12] instead targets structural consistency through multi-view guidance and depth-balanced optimisation. Together, these methods show that sparse-view performance depends not only on improving initial geometry but also on controlling which primitives are retained and how they contribute during learning. Their regularisation criteria, however, do not provide a view-dependent estimate of each primitive's uncertainty at the point of rendering. Consequently, the renderer cannot distinguish well-supported primitives from those unreliable in the current view: both are assigned comparable importance during compositing. Under cumulative alpha blending, an unreliable primitive can then incorrectly consume transmittance or occlude reliable evidence, producing view-dependent artefacts in the rendered image.

2.3 Uncertainty Estimation in Gaussian Splatting

Recent uncertainty-estimation methods for Gaussian splatting pursue objectives that differ from sparse-view reconstruction control. Stochastic formulations, such as variational multi-scale 3DGS [9], model distributions over Gaussian parameters and estimate uncertainty from the resulting samples. Parameter-space methods instead use information measures: FisherRF [6] derives uncertainty from Fisher information, while POp-GS [19] uses P-optimality for next-best-view selection. PRIMU [5] predicts novel-view uncertainty from primitive-based error and coverage features. These methods principally estimate uncertainty for view selection or uncertainty-map prediction, rather than using it to control the contribution of primitives during reconstruction.

Most closely related, Galappaththige *et al.* [4] propose a post-hoc predictive-photometric uncertainty method. They freeze a trained 3DGS representation and fit view-dependent per-primitive uncertainty channels to training-view reconstruction residuals using a regularised least-squares objective; the learned channels are rasterised into pixel-wise uncertainty maps. UGOD has a different purpose and learning mechanism. We learn the uncertainty head jointly with sparse-view reconstruction from Gaussian attributes and the viewing direction, rather than fitting residuals after training. Its uncertainty is then used online to modulate opacity before alpha compositing and to drive detached soft dropout during training. Consequently, UGOD targets the quality–compactness trade-off of sparse-view reconstruction, whereas post-hoc predictive uncertainty methods preserve a fixed renderer and target uncertainty estimation for downstream decisions.

3 Method

3.1 Preliminaries

3D Gaussian Splatting. 3DGS initialises a set of anisotropic Gaussians from an SfM point cloud. Each Gaussian is parameterised by its position, rotation, scale, opacity, and appearance. For a spatial location $\mathbf{x} \in \mathbb{R}^3$, a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is

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

Here $\boldsymbol{\mu} \in \mathbb{R}^3$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is a positive semi-definite covariance matrix parameterised as $\boldsymbol{\Sigma} = \mathbf{RSS}^\top\mathbf{R}^\top$, where $\mathbf{R}$ is an orthogonal rotation matrix and $\mathbf{S}$ is a diagonal scale matrix. In the implementation, $\mathbf{R}_i$ is parameterised by a unit quaternion $\mathbf{q}_i$, and $\mathbf{S}_i$ by a scale vector $\mathbf{s}_i$. We use these compact per-Gaussian parameterisations below.

The Gaussians are projected to the image plane by splatting-based rasterisation [29]. A first-order approximation of the projection at the Gaussian mean gives the projected covariance

$$\boldsymbol{\Sigma}' = \mathbf{JW\Sigma W}^\top\mathbf{J}^\top, \tag{2}$$

where $\mathbf{J}$ is the Jacobian of the local projective transformation and $\mathbf{W}$ is the view transformation matrix. For pixel coordinate $\mathbf{x}$, let $G_i^{\mathrm{2D}}(\mathbf{x})$ denote the projected footprint

of the i-th Gaussian and $o_i \in (0,1)$ its learned opacity. Its screen-space compositing weight and transmittance are

$$\alpha_i(\mathbf{x}) = o_i G_i^{\mathrm{2D}}(\mathbf{x}), \qquad T_i(\mathbf{x}) = \prod_{j<i} \big(1 - \alpha_j(\mathbf{x})\big), \tag{3}$$

where Gaussians are ordered from front to back. The rendered RGB colour is

$$\mathbf{C}(\mathbf{x}) = \sum_{i=1}^{n} T_i(\mathbf{x})\, \alpha_i(\mathbf{x})\, \mathbf{c}_i. \tag{4}$$

Opacity in 3DGS. The learned opacity o_i is optimised jointly with the other Gaussian attributes. It determines the screen-space weight α_i in Eq. 3 and influences density control during training: primitives with very low opacity are pruned, while the remaining primitives continue to refine the representation. Under sparse or noisy initialisation, however, this view-independent scalar cannot distinguish a reliably supported contribution from one that is ambiguous for the current camera. The following section introduces view-dependent uncertainty to address this limitation.

3.2 Per-Gaussian Uncertainty Prediction

We augment 3D Gaussian Splatting (3DGS) [8] with a lightweight per-Gaussian uncertainty head $\mathcal{M}_\theta$. The predicted uncertainty serves two coupled but *gradient-decoupled* roles: (i) a differentiable *opacity-modulation function* that attenuates the opacity of high-uncertainty Gaussians and trains the head through the photometric loss, and (ii) a detached, training-only *soft dropout* that acts as a tail regulariser on the least certain Gaussians. A PSNR-based early-stopping rule freezes the head once it stops helping, so the learned uncertainty stays meaningful and never destabilises the underlying geometry.

Notation and overview.

For each visible Gaussian $i \in \mathcal{V}$, the uncertainty head predicts a raw uncertainty u_i, which is robustly normalised to $\tilde{u}_i$. We denote by $\breve{u}_i$ the gradient-detached copy of $\tilde{u}_i$: it equals $\tilde{u}_i$ in the forward pass but satisfies $\partial \breve{u}_i / \partial \tilde{u}_i = 0$ during back-propagation. The normalised value drives a differentiable opacity-modulation factor g_i, while $\breve{u}_i$ drives a training-only dropout probability p_i and soft keep mask m_i. Thus, the uncertainty pathway is $u_i \to \tilde{u}_i \to g_i$ and $\breve{u}_i \to p_i \to m_i$.

3.2.1 Uncertainty Head: Inputs and Architecture

Each Gaussian G_i carries a mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$, an (activated) scale $\mathbf{s}_i \in \mathbb{R}^3$, a DC colour $\mathbf{c}_i^{\mathrm{dc}} \in \mathbb{R}^3$, and higher-order spherical harmonic (SH) coefficients. Concretely, for SH degree D we denote the (non-DC) coefficients as $\{\mathbf{c}_i^{(\ell)}\}_{\ell=1}^{L}$ with $\mathbf{c}_i^{(\ell)} \in \mathbb{R}^3$ and $L = D(D+2)$.

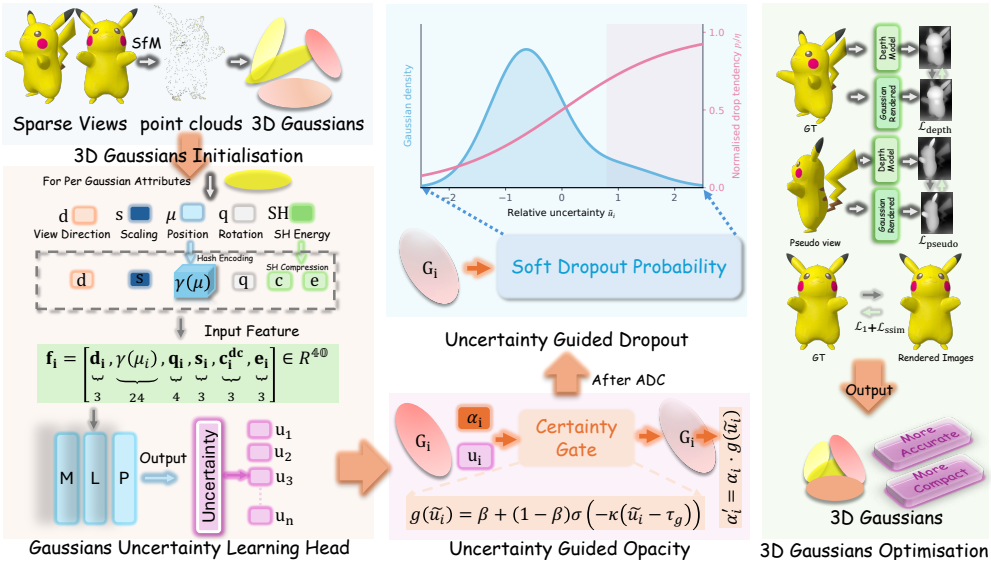

Fig. 2: **Overview of UGOD.** Sparse input views are reconstructed by SfM and initialise 3D Gaussian primitives (left). For each visible Gaussian, the uncertainty head combines the viewing direction, HashGrid position encoding, geometric attributes, and compressed SH appearance to predict a view-dependent uncertainty u_i (lower left). This score drives two gradient-decoupled mechanisms (centre): normalised uncertainty differentiably gates opacity before compositing, whereas its detached copy determines the training-only Concrete soft-dropout probability. The resulting Gaussians are jointly optimised using supervised depth, pseudo-view depth, and photometric losses (right). At inference, soft dropout and relative-uncertainty gating are disabled, and deterministic raw-uncertainty opacity modulation is used.

SH compression.

Directly feeding all SH coefficients to the uncertainty head would introduce a high-dimensional, highly correlated appearance input: the uncertainty prediction should depend on how "view-dependent" a Gaussian is, not on the specific orientation of its SH lobes. We therefore keep the DC colour $\mathbf{c}_i^{\mathrm{dc}}$ unchanged and summarise the remaining SH bands by a per-channel magnitude statistic.

Specifically, we compute a *rest-energy* vector $\mathbf{e}_i \in \mathbb{R}^3$ by averaging the element-wise absolute values of the non-DC coefficients across all bands:

$$\mathbf{e}_i \;=\; \frac{1}{L}\sum_{\ell=1}^{L}\left|\mathbf{c}_i^{(\ell)}\right|, \tag{5}$$

where $|\cdot|$ is applied element-wise to the RGB channels. Intuitively, $\mathbf{e}_i$ measures the overall strength of view-dependent effects per colour channel while discarding directional details. This reduces the SH block from $3L$ scalars (e.g., $3 \times 15 = 45$ values for

degree $D = 3$) to 3 scalars. Together with the DC colour, we obtain a compact 6-D appearance descriptor $[\mathbf{c}_i^{\mathrm{dc}}, \mathbf{e}_i]$ used as part of the head input.

Input feature.

Given the camera centre $\mathbf{o}$, we form the unit view direction $\mathbf{d}_i = (\boldsymbol{\mu}_i - \mathbf{o})/\|\boldsymbol{\mu}_i - \mathbf{o}\|$ and a multi-resolution hash encoding $\gamma(\boldsymbol{\mu}_i) \in \mathbb{R}^{24}$ of the position. The head input is the 40-dimensional concatenation

$$\mathbf{f}_i = \left[\mathbf{d}_i,\, \gamma(\boldsymbol{\mu}_i),\, \mathbf{q}_i,\, \mathbf{s}_i,\, \mathbf{c}_i^{\mathrm{dc}},\, \mathbf{e}_i\right] \in \mathbb{R}^{40}. \tag{6}$$

The respective dimensions of these components are 3, 24, 4, 3, 3, and 3.

Network.

The position is encoded with a multi-resolution hash grid (6 levels, 4 features/level, base resolution 16, per-level scale 1.5, $\log_2$ hashmap size 15), yielding the 24-D code $\gamma(\cdot)$. The 40-D feature is mapped to a scalar logit by a fully fused MLP $\mathcal{M}_\theta : \mathbb{R}^{40} \to \mathbb{R}$, where θ denotes its learnable parameters. The MLP has two hidden layers of 32 neurons, each followed by a LeakyReLU activation. Let $\sigma(x) = (1 + \exp(-x))^{-1}$ denote the logistic sigmoid. We use a single-head configuration that emits one view-conditioned uncertainty score per visible Gaussian,

$$\begin{aligned} u_i &= \sigma\big(\mathcal{M}_\theta(\mathbf{f}_i)\big) \in (0, 1), \\ u_i &\leftarrow \mathrm{clip}\big(u_i,\, 10^{-3},\, 1 - 10^{-3}\big)\,. \end{aligned} \tag{7}$$

This score is not a calibrated epistemic-uncertainty estimate: it is learned only through the photometric objective via differentiable opacity modulation, without an uncertainty target or calibration loss.

We robustly normalise the raw uncertainties over the visible Gaussian set $\mathcal{V}$:

$$\tilde{u}_i = \mathrm{clip}\left(\frac{u_i - \mathrm{med}_{j\in\mathcal{V}}(u_j)}{\mathrm{MAD}_{j\in\mathcal{V}}(u_j) + \delta}, [-c, c]\right), \tag{8}$$

where med denotes the median, and the median absolute deviation is

$$\mathrm{MAD}_{j\in\mathcal{V}}(u_j) = \mathrm{med}_{j\in\mathcal{V}} |u_j - \mathrm{med}_{k\in\mathcal{V}}(u_k)|\,. \tag{9}$$

This robust scale estimate limits the influence of a small number of high-uncertainty Gaussians; $\delta > 0$ prevents division by zero and $c = 2$ bounds the relative uncertainty. Both the opacity-modulation function and the dropout module use $\tilde{u}_i$.

3.2.2 Uncertainty-Guided Opacity Modulation

We attenuate opacity rather than directly removing Gaussians, so that unreliable primitives contribute less to alpha compositing while still receiving photometric supervision. We map relative uncertainty to a monotonically decreasing, differentiable

gate:

$$g_i = \beta + (1 - \beta)\,\sigma\big(-\kappa(\tilde{u}_i - \tau_g)\big), \tag{10}$$

where κ controls the transition sharpness, τ_g specifies the relative-uncertainty threshold, and β is the minimum gate value. Thus, low-uncertainty Gaussians have g_i close to 1 and retain their learned opacity, whereas high-uncertainty Gaussians approach the floor β. The floor prevents modulation alone from fully removing a Gaussian. The gate modulates the learned opacity explicitly as

$$o_i^{\text{gate}} = o_i\, g_i, \tag{11}$$

so high uncertainty reduces the primitive's screen-space alpha and hence its colour and transmittance contribution before compositing. The relative-uncertainty modulation is enabled only after a warm-up of T_g iterations. Before warm-up, and during evaluation, the implementation uses the base raw-uncertainty opacity operation

$$o_i^{\text{raw}} = (1 - u_i)o_i, \qquad \alpha_i^{\text{raw}}(\mathbf{x}) = o_i^{\text{raw}} G_i^{\text{2D}}(\mathbf{x}). \tag{12}$$

The two formulations are intentionally assigned different roles. During optimisation, the relative gate is a distribution-adaptive regulariser: its median–MAD normalisation makes the threshold responsive to the *relative* reliability of currently visible Gaussians, even as densification and pruning change the scale of predicted uncertainties. The floor β additionally prevents premature removal of primitives and preserves useful gradients. At inference, however, this normalisation would make a Gaussian's opacity depend on which other primitives happen to be visible in the same view. We therefore replace the training-time relative gate with the raw-uncertainty rule in Eq. 12, which gives each Gaussian a deterministic, per-primitive attenuation independent of population statistics. Thus, relative gating stabilises optimisation, whereas the raw rule provides a well-defined deployed renderer for all reported evaluations. The modulated screen-space weight is

$$\alpha_i^{\text{gate}}(\mathbf{x}) = o_i^{\text{gate}} G_i^{\text{2D}}(\mathbf{x}). \tag{13}$$

Crucially, $\tilde{u}_i$ is *not* detached on this branch, so the photometric gradient flows through g_i back into $\mathcal{M}_\theta$: this is the sole supervisory signal that teaches the head what "uncertain" means. The relative-uncertainty modulation is used during training after warm-up, evaluation uses Eq. 12.

Rasterised uncertainty map.

To visualise the learned per-Gaussian uncertainty in image space, we rasterise an uncertainty map U by projecting the predicted values $\{u_i\}$ with the same α-blending pipeline used for RGB, while keeping the Gaussian geometry fixed. Formally, for a pixel $\mathbf{x}$,

$$U(\mathbf{x}) = \sum_i T_i(\mathbf{x})\,\alpha_i(\mathbf{x})\,u_i, \tag{14}$$

where T_i and α_i follow the standard 3DGS compositing (Eq. 3); thus, U uses unmodulated compositing weights and is distinct from the training and inference weights

defined in Supplementary Eq. (11). This map is used only for analysis. During training, opacity is modulated by the relative-uncertainty function after warm-up, whereas soft dropout remains training-only.

3.2.3 Detached Soft Dropout (Tail Regulariser)

Opacity modulation and soft dropout have complementary roles. The opacity gate is a deterministic rendering mechanism: on every training view, it continuously down-weights an uncertain Gaussian while preserving its contribution and the gradient that learns its uncertainty. This alone does not prevent the representation from repeatedly relying on the same weakly constrained Gaussians. Soft dropout instead acts as a stochastic training-only regulariser: occasionally suppressing these Gaussians forces the remaining, better-supported primitives to explain the observations and reduces co-adaptation to unreliable evidence. Accordingly, it assigns greater regularisation to high-uncertainty Gaussians. The per-Gaussian drop probability uses the gradient-detached uncertainty $\breve{u}_i$ defined above,

$$p_i = r(t)\,\eta\,\sigma(\breve{u}_i - \tau_d), \tag{15}$$

where η scales the uncertainty influence, τ_d is a drop threshold, and $r(t) = \mathrm{clip}\big((t - T_d)/R,\, 0,\, 1\big)$ is a linear ramp that starts dropout after a fixed warm-up at $T_d = 1200$ and reaches its full value over $R = 500$ iterations. Before applying the logit, we clip $p_i \leftarrow \mathrm{clip}(p_i, \varepsilon_p, 1 - \varepsilon_p)$ with $\varepsilon_p > 0$. We then convert p_i into a differentiable *concrete* keep mask. We first form the perturbed logit

$$z_i = \mathrm{logit}(p_i) + \mathrm{logit}(\varepsilon_i),$$

where $\varepsilon_i \sim \mathcal{U}(\varepsilon_p, 1 - \varepsilon_p)$ is independent uniform noise. The keep mask is then

$$m_i = \mathrm{clip}_{[m_{\min},\, m_{\max}]}\left[1 - \sigma\left(\frac{z_i}{T_{\mathrm{conc}}}\right)\right], \tag{16}$$

where T_{conc} is the temperature. We apply the mask as $\alpha_i^{\mathrm{train}}(\mathbf{x}) = \alpha_i^{\mathrm{gate}}(\mathbf{x}) m_i$. This stochastic masking is useful only during optimisation: by requiring the scene to remain explainable under occasional removal of uncertain primitives, it discourages co-adaptation and over-reliance on weakly constrained evidence. At test time, the objective is instead to render the best deterministic estimate of the complete scene. Consequently, we disable dropout and retain every primitive. This does not discard uncertainty: a high score indicates comparatively unreliable, rather than certainly invalid, evidence, so hard removal could eliminate useful detail. Rendering therefore uses $\alpha_i^{\mathrm{raw}}(\mathbf{x})$ from Eq. 12 to deterministically attenuate each uncertain primitive in proportion to its score, rather than randomly dropping it.

Gradient flow.

The two mechanisms share one scalar u_i but are deliberately decoupled: the opacity-modulation branch is differentiable and back-propagates the rendering loss into $\mathcal{M}_\theta$,

whereas the dropout branch consumes $\breve{u}_i$ and therefore *never* reshapes the uncertainty. This prevents the degenerate solution in which the head inflates u_i merely to be dropped, and lets the head learn a photometrically useful uncertainty score while dropout remains a pure regulariser. Because $\breve{u}_i$ is gradient-detached, the dropout-probability path does not update the uncertainty head; it does not detach the rendered mask from the Gaussian parameters. Consequently, the Concrete mask regularises Gaussian opacity, geometry, and appearance through the rendered photometric loss.

3.2.4 Training Objective

Following FSGS [28], Full UGOD adopts DPT-based monocular depth regularisation [14] to stabilise geometry under sparse observations; it is an adopted training component rather than a contribution of UGOD. A frozen DPT-Hybrid estimator predicts a depth prior D once for each training image during dataset loading. Let $\hat{D}$ be the rasterised depth and $\rho(\cdot,\cdot)$ denote Pearson correlation. To accommodate the scale and sign ambiguity of monocular depth, we use

$$\mathcal{L}_{\text{depth}} = \min\Big\{1-\rho(-D,\hat{D}),\; 1-\rho\Big((D+200)^{-1},\hat{D}\Big)\Big\}. \tag{17}$$

Between iterations 500 and 5500, we additionally sample pseudo cameras, render RGB and depth, and apply the frozen DPT-Hybrid estimator to the rendered RGB. This provides a pseudo-view depth prior for regularising geometry beyond the observed training views. The pseudo-view depth loss is

$$\mathcal{L}_{\text{pseudo}} = 1-\rho\Big(\hat{D}_{\text{p}},-D_{\text{p}}\Big), \tag{18}$$

where D_{p} is the DPT prediction for the rendered pseudo view and $\hat{D}_{\text{p}}$ is its rasterised depth. This term is linearly ramped during its first 500 active iterations. DPT is a fixed target: gradients from $\mathcal{L}_{\text{pseudo}}$ flow through the rendered depth, not through the DPT preprocessing path.

The Gaussians and uncertainty head are jointly optimised with

$$\begin{aligned}\mathcal{L} = {} & (1-\lambda)\,\mathcal{L}_1 + \lambda\,\big(1-\text{SSIM}\big) \\ & + \beta_d(t)\mathcal{L}_{\text{depth}} + \alpha(t)\lambda_p\mathcal{L}_{\text{pseudo}},\end{aligned} \tag{19}$$

where $\lambda = 0.2$, $\beta_d(t) = 0.05$ through iteration 5500 and 0.001 thereafter, and $\alpha(t)$ is the pseudo-view ramp. We use $\lambda_p = 0.5$ by default and $\lambda_p = 0.03$ for outdoor MiP-NeRF 360 scenes. The uncertainty head is supervised implicitly through the differentiable opacity-modulation function in Eq. 10; we use no auxiliary uncertainty-calibration loss in the reported experiments. The head uses its own Adam optimiser (cosine-annealed learning rate), independent of the 3DGS optimiser. We apply validation-based early stopping to the uncertainty head; its criterion and hyper-parameters are provided in the supplementary material.

The complete training and inference procedure is provided in Supplementary Algorithm 1.

Default hyper-parameters.

The complete uncertainty-head, gating, dropout, and early-stopping settings are reported in Supplementary Table 1.

4 Experiments

4.1 Experimental Setup

Datasets and protocols. We evaluate sparse-view NVS on MiP-NeRF 360 [1] and LLFF [11]. We follow FSGS [28] for its 24-view MiP-NeRF 360 and 3-view LLFF protocols, and additionally evaluate the more challenging 8-view MiP-NeRF 360 setting. For MiP-NeRF 360, we use the six scenes reported in Supplementary Tables 2 and 3, using predefined disjoint training and evaluation cameras.

For LLFF, we use three COLMAP-reconstructed training views per scene and evaluate on the corresponding held-out views, as detailed in Supplementary Table 4. LLFF images are downsampled by a factor of eight along both spatial dimensions. For MiP-NeRF 360, images wider than 1600 pixels are resized to width 1600 while preserving aspect ratio, whereas smaller images retain their original resolution. These camera splits and image resolutions define our evaluation protocol; consequently, absolute metrics need not match prior reports that use different splits or input resolutions.

Implementation. We implement UGOD in PyTorch on an NVIDIA RTX A100 and train all benchmarked methods for 6,000 iterations. The uncertainty head uses the $(6, 0, 0, 0)$ HashGrid configuration selected by the ablation in Table 3; its remaining settings follow Supplementary Table 1. We reset opacity at iterations 2001 and 5001 for 6000-iteration training. The uncertainty head is frozen after two consecutive validation evaluations for which PSNR decreases relative to the preceding evaluation ($\Delta\text{PSNR} < \epsilon = 0$); Gaussian parameters continue to be optimised. In tables and comparison figures only, 3DGS* denotes the official 3DGS baseline evaluated under the same sparse-view protocol. Each table specifies the compared variants: *Ours (full)* denotes the depth-assisted model with uncertainty-guided soft dropout.

Metrics. We report PSNR, SSIM [18], and LPIPS [25] on held-out views. We additionally report the final Gaussian count and its reduction relative to 3DGS* and DropGS where available, since sparse-view reconstruction requires a trade-off between rendering quality and representation compactness.

4.2 Quantitative Comparison

We quantitatively compare Full UGOD with 3DGS* [7], DropGS [13], and FSGS [28]. Table 1 first reports scene-averaged results, and Supplementary Tables 2, 3, and 4 provide the corresponding per-scene comparisons.

LLFF. On the more challenging 3-view protocol, Full UGOD achieves the best scene-average PSNR (19.33), SSIM (0.637), and LPIPS (0.245), while using 102,280 Gaussians on average 2.5× fewer than 3DGS*. It obtains the highest PSNR on six of

Table 1: Scene-average results on LLFF (3 views) and MiP-NeRF 360 (8/24 views). Reduction is relative to 3DGS*; best and second-best values are bold and underlined.

Setting	Method	PSNR↑	SSIM↑	LPIPS↓	#Gaussians	vs 3DGS*
LLFF (3-view)	3DGS*	14.6543	0.4379	0.3990	254,350	–
	DropGS	14.6443	0.4557	0.3894	210,414	1.2×
	FSGS (w/ depth)	<u>19.0457</u>	<u>0.6230</u>	<u>0.2514</u>	<u>151,173</u>	<u>1.7×</u>
	Ours (full)	**19.3300**	**0.6370**	**0.2449**	**102,280**	**2.5×**
Mip360 (8-view)	3DGS*	11.9591	0.2747	<u>0.6095</u>	864,393	–
	DropGS	12.7267	0.3193	**0.6026**	716,133	1.2×
	FSGS (w/ depth)	<u>13.0309</u>	<u>0.3588</u>	0.6197	<u>132,090</u>	<u>6.5×</u>
	Ours (full)	**14.1435**	**0.3682**	0.6179	**95,662**	**9.0×**
Mip360 (24-view)	3DGS*	18.6739	<u>0.5614</u>	**0.4289**	1,038,952	–
	DropGS	<u>18.9911</u>	**0.5629**	<u>0.4520</u>	775,653	1.3×
	FSGS (w/ depth)	18.7833	0.5315	0.5245	<u>148,296</u>	<u>7.0×</u>
	Ours (full)	**19.0867**	0.5365	0.5203	**129,498**	**8.0×**

seven scenes and the smallest representation on six scenes. *Orchids* is the exception for rendering quality, where FSGS achieves the best PSNR and LPIPS (with tied best SSIM), while on *Room* FSGS uses fewer Gaussians and achieves the best SSIM and LPIPS. These per-scene results show that the gain remains scene dependent, but is most consistent under severe view sparsity.

MiP-NeRF 360. Under the 8-view protocol, Full UGOD obtains the highest average PSNR (14.14) and SSIM (0.368), with 95,662 final Gaussians on average. This is 9.0× fewer Gaussians than 3DGS* and 7.5× fewer than DropGS. The per-scene results show the highest PSNR on five of the six scenes and the smallest representation on three; on *bicycle*, for example, Full UGOD improves PSNR from 11.01 to 13.11 while reducing the Gaussian count from 1,460,812 to 59,483. DropGS achieves the best average LPIPS, and FSGS attains the best PSNR, SSIM, and LPIPS on *counter*; therefore, the 8-view result is not a uniform improvement for every metric.

With 24 views, Full UGOD still gives the highest average PSNR (19.09) and the smallest average representation (129,498 Gaussians, 8.0× fewer than 3DGS*). It has the lowest Gaussian count on five of the six scenes and the highest PSNR on *bicycle*, *counter*, and *kitchen*. The detailed per-scene analysis in Supplementary Table 3 indicates that, with more observations, uncertainty attenuation can trade fine detail for representation compactness. Thus, in this less-sparse setting, the results demonstrate a quality–compactness trade-off rather than across-the-board metric superiority.

4.3 Ablation Study

Component ablation. Table 2 summarises the component rows in Supplementary Tables 5, 6, and 7: for each sparse-view protocol, it compares uncertainty-guided opacity modulation with and without soft dropout, and the full configuration, in terms of scene-average PSNR, SSIM, LPIPS, and final Gaussian count. In the 8-view setting, the soft-dropout comparison is averaged over the five scenes where both variants are

Table 2: Scene-average component ablation. N is the number of scenes averaged; the 8-view soft-dropout variants use five scenes, while the full model uses all six. Best and second-best values are bold and underlined.

Setting	Variant	N	PSNR↑	SSIM↑	LPIPS↓	#Gaussians↓
LLFF (3-view)	Ours	7	<u>18.2386</u>	<u>0.5989</u>	<u>0.2701</u>	<u>180,905</u>
	w/o soft dropout	7	17.9529	0.5700	0.2877	183,460
	Ours (full)	7	**19.3300**	**0.6370**	**0.2449**	**102,280**
Mip360 (8-view)	w/o soft dropout	5	<u>13.9438</u>	<u>0.3672</u>	**0.6074**	108,008
	w/ soft dropout	5	12.9741	0.3475	0.6267	<u>102,644</u>
	Ours (full)	6	**14.1435**	**0.3682**	<u>0.6179</u>	**95,662**
Mip360 (24-view)	w/o soft dropout	6	<u>19.0752</u>	**0.5521**	**0.4744**	243,836
	w/ soft dropout	6	18.7457	<u>0.5429</u>	<u>0.4760</u>	<u>244,322</u>
	Ours (full)	6	**19.0867**	0.5365	0.5203	**129,498**

available; the bicycle scene instead reports modulation-only and full results. Adding soft dropout alone reduces the average Gaussian count from 108,008 to 102,644 but also lowers PSNR from 13.94 to 12.97. The full model obtains the best 8-view average PSNR and SSIM while using 95,662 Gaussians; because it also includes depth supervision, this comparison measures the complete system rather than an isolated dropout effect. At 24 views, the full model improves PSNR and compactness but not average SSIM or LPIPS. In contrast, under the LLFF 3-view protocol, the full configuration improves all three image metrics and reduces the average Gaussian count to 102,280. These results support using the complete configuration in the most severely sparse setting while making the quality–compactness trade-off explicit elsewhere.

HashGrid input configuration. We conduct a controlled ablation on the MiP-NeRF 360 *kitchen* scene with 24 training views, holding all non-encoding settings fixed (Table 3). P, S, R, and V denote the number of HashGrid levels assigned to position, scale, rotation, and view direction, respectively. Position-only encoding with six levels, namely $(P, S, R, V) = (6, 0, 0, 0)$, achieves the best PSNR (15.0757), SSIM (0.4628), and LPIPS (0.6396). Adding one view-direction level to the five-level position encoding improves the result over $(5, 0, 0, 0)$, but remains inferior to $(6, 0, 0, 0)$. Allocating levels to scale and rotation, or increasing the position encoding to seven levels, degrades all three metrics in this controlled setting. We therefore use $(P, S, R, V) = (6, 0, 0, 0)$: only position is HashGrid encoded, while the remaining attributes are concatenated in their raw form.

4.4 Qualitative Comparison

Figure 3 presents qualitative comparisons on Mip-NeRF 360 (24-view *stump* and 8-view *bicycle*) and LLFF (3-view *horns*, *flower*, and *fern*). Each rendered view is annotated with the displayed-view PSNR and the final Gaussian count of the corresponding model.



Fig. 3: Qualitative comparison on MiP-NeRF 360 (24/8 views) and LLFF (3 views). Each non-GT panel reports the displayed-view PSNR and final Gaussian count.

On the 8-view *bicycle* scene, 3DGS* and DropGS exhibit substantial floaters and blurred structures, whereas FSGS recovers more coherent geometry. Full UGOD more clearly recovers bicycle spokes and the background fence in this example, with a view-wise PSNR of 13.11 dB and 59.5K Gaussians, compared with 11.01 dB and 1460.8K for 3DGS*, 12.24 dB and 1000.2K for DropGS, and 11.05 dB and 142.1K for FSGS.

The same quality–compactness pattern is visible in the LLFF examples. On *horns*, *flower*, and *fern*, Full UGOD attains higher displayed-view PSNR than FSGS (20.11 vs. 19.52 dB, 21.98 vs. 21.47 dB, and 18.31 vs. 17.85 dB, respectively) while using fewer Gaussians (71.1K vs. 99.2K, 93.4K vs. 244.6K, and 93.8K vs. 117.4K, respectively). The 24-view *stump* example shows a smaller visual and numerical margin: Full UGOD reaches 18.11 dB with 142.8K Gaussians, compared with 18.03 dB and 149.3K for FSGS. These examples are consistent with the quantitative results: the proposed model can improve the quality–compactness trade-off under sparse-view supervision, although the margin varies across scenes.

Table 3: Controlled HashGrid input-encoding ablation on MiP-NeRF 360 *kitchen* with 24 training views. P, S, R, and V are the numbers of encoding levels assigned to position, scale, rotation, and view direction. **Bold** indicates the best result.

P	S	R	V	PSNR ↑	SSIM ↑	LPIPS ↓
5	0	0	0	12.8370	0.4092	0.6619
5	0	0	1	14.7173	0.4575	0.6400
6	0	0	0	**15.0757**	**0.4628**	**0.6396**
6	1	1	0	14.6207	0.4508	0.6461
7	0	0	0	10.9079	0.4070	0.6758

4.5 Uncertainty Dynamics and Opacity Response

To characterise the signal learned through the photometric objective, we track the uncertainty head throughout a representative three-view training run on the *fern* scene from LLFF (Fig. 4). This analysis concerns the evolution and use of the predicted uncertainty, rather than calibration error. The raw uncertainty distribution initially has a broad mode near $u = 0.45$, but progressively concentrates near $u = 0.1$ while retaining a high-uncertainty tail (Fig. 4a). Thus, training does not uniformly reduce all uncertainty predictions: most Gaussians become low-uncertainty, whereas a subset remains uncertain.

The proportion of relative uncertainties satisfying $\tilde{u}_i > 0.8$ rises from approximately 27% at 1k iterations to 40% at 2k, coinciding with the densification period, and then decreases to approximately 36% at 6k (Fig. 4b). This trajectory is consistent with densification introducing initially less-constrained Gaussians, after which photometric optimisation separates the low-uncertainty majority from a persistent uncertain tail. At 6k iterations, larger predicted uncertainties are associated with smaller gate multipliers (Fig. 4c). The response approaches the configured multiplier floor $\beta = 0.70$ for high-uncertainty Gaussians, directly illustrating their attenuation through the differentiable opacity-modulation pathway in Eq. 10.

4.6 Photometric Residuals and Uncertainty Visualisation

Only UGOD contains the uncertainty head and can therefore produce the predicted uncertainty map $U(\mathbf{x})$ in Eq. 14. Its scalar u_i is a view-dependent signal predicted from Gaussian attributes and viewing direction, and is only indirectly supervised through the opacity-modulation pathway.

For 3DGS*, DropGS, and FSGS, Figure 5 instead shows the photometric residual map

$$E(\mathbf{x}) = \left\| \hat{\mathbf{C}}(\mathbf{x}) - \mathbf{C}_{\mathrm{GT}}(\mathbf{x}) \right\|_1 . \tag{20}$$

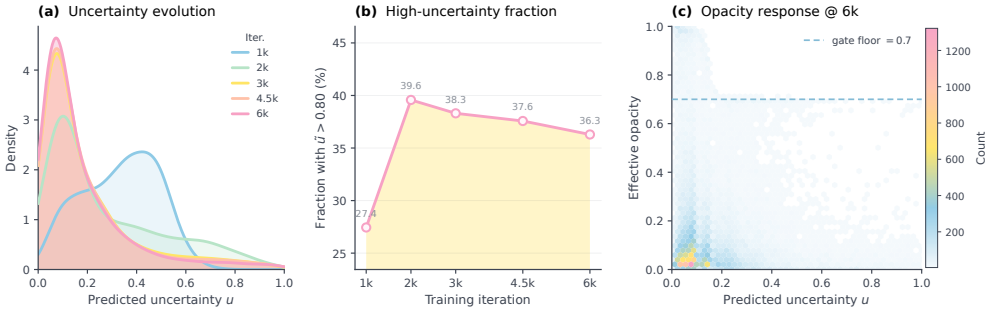

Fig. 4: Uncertainty dynamics for the LLFF *fern* scene with three training views: raw-uncertainty distributions (left), the fraction with $\tilde{u}_i > 0.8$ (centre), and the uncertainty–opacity response at 6k iterations (right).

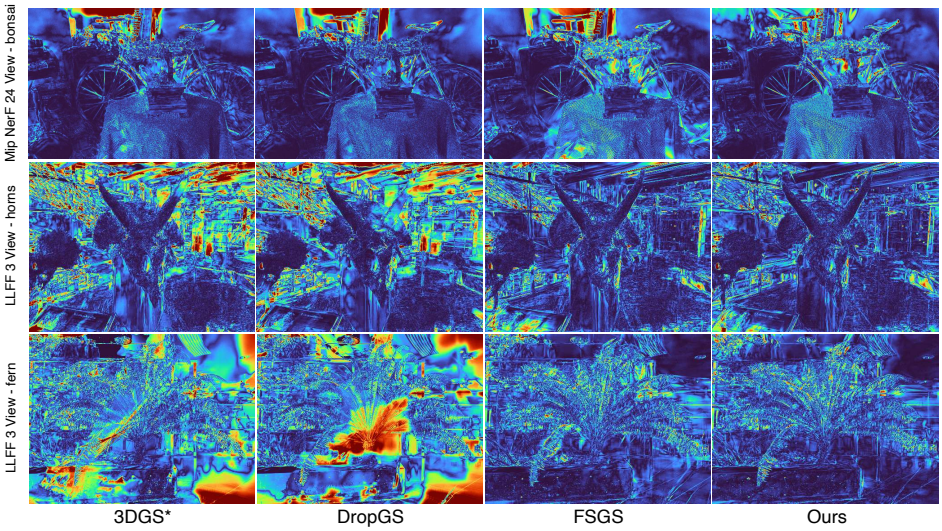

Fig. 5: Photometric residuals for 3DGS*, DropGS, and FSGS, and UGOD's predicted uncertainty. Rows show MiP-NeRF 360 *bonsai* (24 views) and LLFF *horns* and *fern* (3 views).

These baseline heatmaps measure image-space reconstruction error, not predicted uncertainty. The visualisation consequently compares residual patterns for the baselines with the predicted uncertainty pattern of UGOD on MiP-NeRF 360 *bonsai* (24 views) and LLFF *horns* and *fern* (three training views). The baseline residuals are concentrated around under-constrained structures and backgrounds, whereas UGOD's predicted map assigns lower accumulated uncertainty to most retained scene content.

5 Conclusion

We introduced UGOD, an uncertainty-guided framework for sparse-view 3D Gaussian Splatting. UGOD learns view-dependent per-Gaussian uncertainty from Gaussian attributes and the viewing direction, then uses this signal in two complementary mechanisms. First, uncertainty-guided opacity modulation continuously attenuates high-uncertainty contributions before alpha compositing. Second, gradient-detached soft dropout regularises high-uncertainty primitives during training without directly altering the uncertainty estimator. Experiments on MiP-NeRF 360 and LLFF show that this design improves the quality–compactness trade-off in the evaluated sparse-view protocols, with the most consistent gains under severe view sparsity. The per-scene results further show that the benefit is metric and scene dependent, especially when additional input views reduce the reconstruction ambiguity.

Supplementary information. The online version contains supplementary material available at the journal website.

Acknowledgements. This work was supported by Manchester Metropolitan University Student Scholarship.

Statements and Declarations

Funding. This work was supported by Manchester Metropolitan University Student Scholarship.

Competing interests. The authors declare that they have no competing interests.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The source code and configuration files needed to reproduce the experiments in this paper will be released publicly upon acceptance of the manuscript. The MiP-NeRF 360 and LLFF datasets used in this study are publicly available from their original releases.

Materials availability. Not applicable.

Code availability. The source code will be released publicly upon acceptance of the manuscript.

Author contribution. Zhihao Guo: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. Peng Wang: Conceptualization, Methodology, Supervision, Writing – review & editing, Project administration. Zidong Chen: Software, Validation, Investigation, Writing – review & editing. Xiangyu Kong: Validation, Investigation, Writing – review & editing. Yan Lyu: Validation, Investigation, Writing – review & editing. Guanyu Gao: Validation, Investigation, Writing – review & editing. Chenghao Qian: Validation, Investigation, Writing – review & editing. Ziyang Wang: Validation, Investigation, Writing – review & editing. Xinqi Fan: Supervision, Writing – review &

editing, Funding acquisition. Liangxiu Han: Supervision, Writing – review & editing, Funding acquisition, Project administration. All authors read and approved the final manuscript.

References

[1] Barron JT, Mildenhall B, Verbin D, et al (2022) Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5470–5479

[2] Bulò SR, Porzi L, Kontschieder P (2024) Revising densification in gaussian splatting. arXiv preprint arXiv:240406109

[3] Celarek A, Kopanas G, Drettakis G, et al (2025) Does 3d gaussian splatting need accurate volumetric rendering? arXiv preprint arXiv:250219318

[4] Galappaththige CJ, Gottwald T, Stehr P, et al (2026) Predictive photometric uncertainty in gaussian splatting for novel view synthesis. In: European Conference on Computer Vision

[5] Gottwald T, Heinert E, Stehr P, et al (2026) Primu: Uncertainty estimation for novel views in gaussian splatting from primitive-based representations of error and coverage. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11871–11880

[6] Jiang W, Lei B, Daniilidis K (2023) Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. arXiv preprint arXiv:231117874

[7] Kerbl B, Kopanas G, Leimkühler T, et al (2023) 3d gaussian splatting for real-time radiance field rendering. ACM Trans Graph 42(4):139–1

[8] Kerbl B, Kopanas G, Leimkühler T, et al (2023) 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics 42(4). URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[9] Li R, Cheung Ym (2024) Variational multi-scale representation for estimating uncertainty in 3d gaussian splatting. Advances in Neural Information Processing Systems 37:87934–87958

[10] Lim H, Chang H, Choi JB, et al (2025) Lidar-3dgs: Lidar reinforcement for multimodal initialization of 3d gaussian splats. Computers & Graphics 132:104293. https://doi.org/10.1016/j.cag.2025.104293

[11] Mildenhall B, Srinivasan PP, Tancik M, et al (2019) Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4759–4768

[12] Pang G, Li K, Zhang G, et al (2025) Strugs: Structurally consistent 3d gaussian splatting with targeted optimization strategies. Computers & Graphics 132:104440. https://doi.org/10.1016/j.cag.2025.104440

[13] Park H, Ryu G, Kim W (2025) Dropgaussian: Structural regularization for sparse-view gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp 21600–21609

[14] Ranftl R, Bochkovskiy A, Koltun V (2021) Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 12179–12188

[15] Shen J, Feng T, Dong H, et al (2025) Surags: Toward efficient few-shot novel view synthesis via surface-aware gaussian splatting. Computers & Graphics 131:104349. https://doi.org/10.1016/j.cag.2025.104349

[16] Talegaonkar C, Belhe Y, Ramamoorthi R, et al (2024) Volumetrically consistent 3d gaussian rasterization. arXiv preprint arXiv:241203378

[17] Tang Y, Yan J, Li Y, et al (2025) Hr-2dgs: Hybrid regularization for sparse-view 3d reconstruction with 2d gaussian splatting. Computers & Graphics 133:104444. https://doi.org/10.1016/j.cag.2025.104444

[18] Wang Z, Bovik AC, Sheikh HR, et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4):600–612

[19] Wilson J, Almeida M, Mahajan S, et al (2025) Pop-gs: Next best view in 3d-gaussian splatting with p-optimality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3646–3655

[20] Xiong H, Muttukuru S, Xiao H, et al (2025) Sparsegs: Sparse view synthesis using 3d gaussian splatting. In: International Conference on 3D Vision

[21] Yang H, Zhang C, Wang W, et al (2024) Gaussian splatting with localized points management. CoRR

[22] Yong S, Manivannan VNP, Kerbl B, et al (2025) Omg: Opacity matters in material modeling with gaussian splatting. arXiv preprint arXiv:250210988

[23] Yu Z, Chen A, Huang B, et al (2024) Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 19447–19456

[24] Zhang J, Li J, Yu X, et al (2024) Cor-gs: sparse-view 3d gaussian splatting via co-regularization. In: European Conference on Computer Vision, Springer, pp 335–352

[25] Zhang R, Isola P, Efros AA, et al (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 586–595

[26] Zhang Z, Hu W, Lao Y, et al (2024) Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. arXiv preprint arXiv:240315530

[27] Zhao J, Li C, Zhang S, et al (2025) Pcdgs: A point cloud densification method for gaussian splatting. Computers & Graphics 130:104241. https://doi.org/10.1016/j.cag.2025.104241

[28] Zhu Z, Fan Z, Jiang Y, et al (2024) Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: European conference on computer vision, Springer, pp 145–163

[29] Zwicker M, Pfister H, Van Baar J, et al (2001) Surface splatting. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp 371–378

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ugodsupplementary.pdf