Spectral-GS: Taming 3D Gaussian Splatting with Spectral Entropy The Supplemental Document

LETIAN HUANG, State Key Lab for Novel Software Technology, Nanjing University, China
JIE GUO*, State Key Lab for Novel Software Technology, Nanjing University, China
JIALIN DAN, State Key Lab for Novel Software Technology, Nanjing University, China
RUOYU FU, State Key Lab for Novel Software Technology, Nanjing University, China
YUANQI LI, State Key Lab for Novel Software Technology, Nanjing University, China
YANWEN GUO, State Key Lab for Novel Software Technology, Nanjing University, China

ACM Reference Format:
Letian Huang, Jie Guo, Jialin Dan, Ruoyu Fu, Yuanqi Li, and Yanwen Guo. 2025. Spectral-GS: Taming 3D Gaussian Splatting with Spectral Entropy The Supplemental Document. *ACM Trans. Graph.* 44, 6, Article 1 (December 2025), 7 pages. https://doi.org/10.1145/3757377.3763907

A Proofs in Spectral Analysis of Gaussians

A.1 Spectral Analysis of Matrices

Mathematically, the spectrum of a matrix refers to the set of its eigenvalues [Eisenbud 2013; Golub and Van Loan 2013; Zill 2020].

Eigenvalue (Spectrum). A matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ can be eigendecomposed as follows:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{1}$$

where $\mathbf{\Lambda} = \operatorname{diag}\left(\lambda_1, \lambda_2, \ldots, \lambda_N\right)$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\mathbf{\Lambda}_{ii} = \lambda_i$. The trace of $\mathbf{A}$, denoted $\operatorname{tr}(\mathbf{A})$, is the sum of all eigenvalues, *i.e.*, $\operatorname{tr}(\mathbf{A}) = \sum_{i=1}^{N} \lambda_i$. And the determinant of $\mathbf{A}$, denoted $\det(\mathbf{A})$ or $|\mathbf{A}|$, is the product of all eigenvalues, *i.e.*, $\det(\mathbf{A}) = |\mathbf{A}| = \prod_{i=1}^{N} \lambda_i$.

Spectral radius. In mathematics, the spectral radius $\rho(\cdot)$ of a square matrix $\mathbf{A}$ is the maximum of the absolute values of its eigenvalues:

$$\rho(\mathbf{A}) = \max\left(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_N|\right). \tag{2}$$

The eigenvector corresponding to the spectral radius of $\mathbf{A}$ is commonly referred to as the principal eigenvector.

Condition number. The condition number $\kappa(\cdot)$ of a function quantifies the sensitivity of the function's output to small perturbations in its input. When selecting the spectral radius as the matrix norm (spectral norm), the condition number of a normal matrix $\mathbf{A}$ is:

$$\kappa(\mathbf{A}) = \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}\| = \frac{\rho(\mathbf{A})}{\rho_{\min}(\mathbf{A})} \tag{3}$$

where $\rho_{\min}(\mathbf{A}) = \min\left(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_N|\right)$ is the minimum of the absolute values of the eigenvalues.

Spectral entropy [Roy and Vetterli 2007; Shannon 1948; Von Neumann 2018; Wei et al. 2024]. Let $\mathbf{A}$ be a positive semi-definite matrix ($\forall\ 0 \le i \le N,\ \lambda_i \ge 0$) and the trace of $\mathbf{A}$ be positive ($\operatorname{tr}(\mathbf{A}) > 0$). Then the matrix $\mathbf{K} = \frac{\mathbf{A}}{\operatorname{tr}(\mathbf{A})}$ satisfies $\operatorname{tr}(\mathbf{K}) = 1$. The spectral entropy $\mathrm{H}(\cdot)$ is:

$$\mathrm{H}(\mathbf{A}) = \operatorname{tr}(-\mathbf{K}\ln\mathbf{K}) = -\sum_{i=1}^{N} \frac{\lambda_i}{\operatorname{tr}(\mathbf{A})} \ln \frac{\lambda_i}{\operatorname{tr}(\mathbf{A})}. \tag{4}$$

A.2 Maxima and Minima in Spectral Analysis

This section will provide a detailed proof that the condition number is minimized and the spectral entropy is maximized when $s_1 = s_2 = s_3$.

Condition number. It is evident that the spectrum of the Gaussian satisfies the following inequality:

$$0 < \min\left(s_1^2, s_2^2, s_3^2\right) \le \max\left(s_1^2, s_2^2, s_3^2\right). \tag{5}$$

Consequently, the condition number satisfies

$$\kappa(\Sigma) = \frac{\max\left(s_1^2, s_2^2, s_3^2\right)}{\min\left(s_1^2, s_2^2, s_3^2\right)} \ge 1, \tag{6}$$

with equality iff $\max\left(s_1^2, s_2^2, s_3^2\right) = \min\left(s_1^2, s_2^2, s_3^2\right)$. Therefore, we can conclude that the condition number of Σ is minimized when $s_1 = s_2 = s_3$.

Spectral entropy. Using the 3D Gaussian as an example, we aim to find the maximum value of the spectral entropy in Eqn. (8) of the paper. Let t_i be $\frac{s_i^2}{\sum_{j=1}^{3} s_j^2}$. It is evident that this is a constrained optimization problem

$$\arg\max_{t_1,t_2,t_3} \mathrm{H}(\Sigma) = -\sum_{i=1}^{3} t_i \ln t_i, \quad \text{s.t.} \sum_{i=1}^{3} t_i = 1 \tag{7}$$

*Corresponding authors.

Authors' Contact Information: Letian Huang, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, lthuang@smail.nju.edu.cn; Jie Guo, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, guojie@nju.edu.cn; Jialin Dan, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, danjialin@smail.nju.edu.cn; Ruoyu Fu, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, fry@nju.edu.cn; Yuanqi Li, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, yuanqili@nju.edu.cn; Yanwen Guo, State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China, ywguo@nju.edu.cn.

ACM 1557-7368/2025/12-ART1
https://doi.org/10.1145/3757377.3763907

which can be solved using the method of Lagrange multipliers as below:

$$\arg\max_{t_1,t_2,t_3} \mathcal{F}(t_1,t_2,t_3,\lambda) = -\sum_{i=1}^{3} t_i \ln t_i + \lambda\left(\sum_{i=1}^{3} t_i - 1\right) \tag{8}$$

where λ is the Lagrange multiplier. Now we can calculate the gradient:

$$\nabla_{t_1,t_2,t_3,\lambda}\mathcal{F}(t_1,t_2,t_3,\lambda) = \left(\frac{\partial\mathcal{F}}{t_1}, \frac{\partial\mathcal{F}}{t_2}, \frac{\partial\mathcal{F}}{t_3}, \frac{\partial\mathcal{F}}{\lambda}\right) \tag{9}$$

and therefore:

$$\nabla_{t_1,t_2,t_3,\lambda}\mathcal{F}(t_1,t_2,t_3,\lambda) = \mathbf{0} \iff \begin{cases} \lambda = \ln t_1 + 1 \\ \lambda = \ln t_2 + 1 \\ \lambda = \ln t_3 + 1 \\ \sum_{i=1}^{3} t_i = 1 \end{cases}. \tag{10}$$

In summary, we can conclude that the spectral entropy of Σ is maximized when $t_1 = t_2 = t_3$ ($s_1 = s_2 = s_3$).

A.3 Relationship Between $\kappa(\cdot)$ and $\mathrm{H}(\cdot)$

We can express the spectral entropy $\mathrm{H}(\Sigma)$ as a function of the condition number $\kappa(\Sigma)$ for the 2D Gaussian:

$$\begin{aligned} \mathrm{H}(\Sigma) &= -\frac{s_1^2}{s_1^2+s_2^2}\ln\frac{s_1^2}{s_1^2+s_2^2} - \frac{s_2^2}{s_1^2+s_2^2}\ln\frac{s_2^2}{s_1^2+s_2^2} \\ &= \ln(\kappa(\Sigma)+1) - \frac{\kappa(\Sigma)\ln\kappa(\Sigma)}{\kappa(\Sigma)+1}. \end{aligned} \tag{11}$$

Then we calculate the derivative:

$$\frac{\mathrm{dH}(\Sigma)}{\mathrm{d}\kappa(\Sigma)} = -\frac{\ln\kappa(\Sigma)}{(\kappa(\Sigma)+1)^2} \le 0, \text{ where } \kappa(\Sigma) \ge 1, \tag{12}$$

which indicates that the spectral entropy of a 2D Gaussian decreases as the condition number increases. We visualize this function in Fig. 1a.

For the 3D Gaussian, assuming without loss of generality that $s_1 \le s_2 \le s_3$, we have $\kappa(\Sigma) = \frac{s_3^2}{s_1^2}$ and let $\eta = \frac{s_2^2}{s_1^2}$. We can similarly derive the function for the 3D Gaussian:

$$\mathrm{H}(\Sigma) = \ln(\kappa(\Sigma)+\eta+1) - \frac{\kappa(\Sigma)\ln\kappa(\Sigma)+\eta\ln\eta}{\kappa(\Sigma)+\eta+1} \tag{13}$$

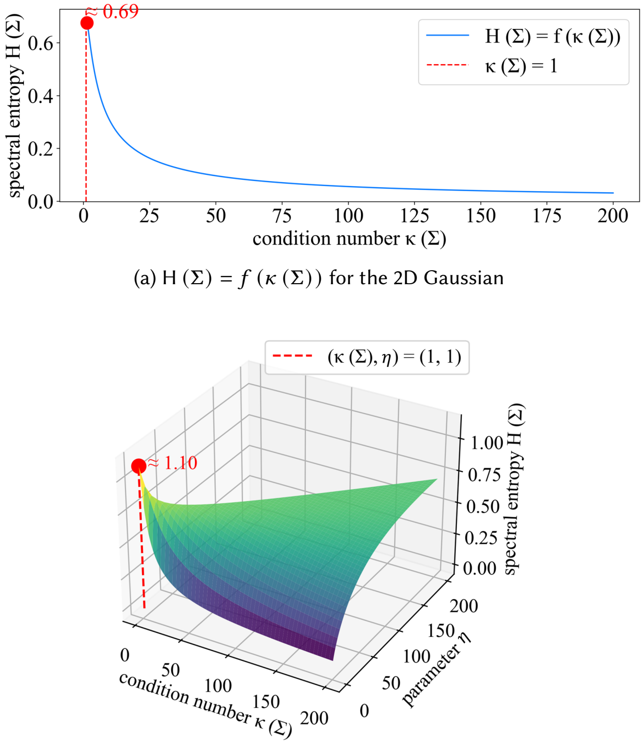
and visualize the function in Fig. 1b.

A.4 View-Inconsistency in Filtering

The Jacobian matrix of the local affine approximation is as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{f_x}{\mu_z} & 0 & -\frac{f_x\mu_x}{\mu_z^2} \\ 0 & \frac{f_y}{\mu_z} & -\frac{f_y\mu_y}{\mu_z^2} \end{bmatrix} = \begin{bmatrix} \frac{f_x}{\mu_z} & 0 & 0 \\ 0 & \frac{f_y}{\mu_z} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{\mu_x}{\mu_z} \\ 0 & 1 & -\frac{\mu_y}{\mu_z} \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

where f_x, f_y denote the intrinsic parameters of the camera model and $\boldsymbol{\mu}' = \begin{bmatrix}\mu_x & \mu_y & \mu_z\end{bmatrix}^\top$ is the position of the 3D Gaussian in the camera space. We assume that the position of the Gaussian projected onto the $z = 1$ plane, *i.e.*, $\left[\frac{\mu_x}{\mu_z}, \frac{\mu_y}{\mu_z}\right]^\top$, remains unchanged during the camera zoom-in. Since the covariance matrix in the camera space Σ' also remains unchanged when zooming in, the following matrix is a constant matrix



(a) $\mathrm{H}(\Sigma) = f(\kappa(\Sigma))$ for the 2D Gaussian

(b) $\mathrm{H}(\Sigma) = f(\kappa(\Sigma, \eta))$ for the 3D Gaussian

Fig. 1. Visualization of the function $\mathrm{H}(\Sigma) = f(\kappa(\Sigma))$ for the 2D Gaussian and $\mathrm{H}(\Sigma) = f(\kappa(\Sigma, \eta))$ for the 3D Gaussian.

$$\Sigma'' = \begin{bmatrix} 1 & 0 & -\frac{\mu_x}{\mu_z} \\ 0 & 1 & -\frac{\mu_y}{\mu_z} \\ 0 & 0 & 1 \end{bmatrix} \Sigma' \begin{bmatrix} 1 & 0 & -\frac{\mu_x}{\mu_z} \\ 0 & 1 & -\frac{\mu_y}{\mu_z} \\ 0 & 0 & 1 \end{bmatrix}^\top = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \tag{15}$$

where $a, b, c, d, e, f \in [0, +\infty)$. Then we obtain:

$$\Sigma_{\text{filter}} = \mathbf{J}\Sigma'\mathbf{J}^\top + \sigma\mathbf{I} \approx \begin{bmatrix} \sigma + \frac{af_x^2}{\mu_z^2} & \frac{bf_xf_y}{\mu_z^2} \\ \frac{bf_xf_y}{\mu_z^2} & \sigma + \frac{df_y^2}{\mu_z^2} \end{bmatrix}. \tag{16}$$

Then we can compute the condition number of the matrix:

$$\begin{aligned} \kappa(\Sigma_{\text{filter}}) &= \frac{\frac{1}{2}\mathrm{tr}(\Sigma_{\text{filter}}) + \sqrt{\frac{1}{4}\mathrm{tr}^2(\Sigma_{\text{filter}}) - |\Sigma_{\text{filter}}|}}{\frac{1}{2}\mathrm{tr}(\Sigma_{\text{filter}}) - \sqrt{\frac{1}{4}\mathrm{tr}^2(\Sigma_{\text{filter}}) - |\Sigma_{\text{filter}}|}} \\ &= \frac{2\sigma + \left(a + d\frac{f_y^2}{f_x^2} + \sqrt{\left(a - d\frac{f_y^2}{f_x^2}\right)^2 + 4b^2\frac{f_y^2}{f_x^2}}\right)\frac{f_x^2}{\mu_z^2}}{2\sigma + \left(a + d\frac{f_y^2}{f_x^2} - \sqrt{\left(a - d\frac{f_y^2}{f_x^2}\right)^2 + 4b^2\frac{f_y^2}{f_x^2}}\right)\frac{f_x^2}{\mu_z^2}} \end{aligned} \tag{17}$$

which is a function of $\frac{f_x^2}{\mu_z^2}$ due to the other values remaining constant during the camera zoom-in. And the derivative of the function is:

$$\frac{\mathrm{d}\kappa\left(\Sigma_{\text{filter}}\right)}{\mathrm{d}\frac{f_x^2}{\mu_z^2}} = \frac{4p\sigma}{\left(2\sigma + \left(a + d\frac{f_y^2}{f_x^2} - p\right)\frac{f_x^2}{\mu_z^2}\right)^2} > 0 \tag{18}$$

where $p = \sqrt{\left(a - d\frac{f_y^2}{f_x^2}\right)^2 + 4b^2\frac{f_y^2}{f_x^2}}$. This indicates that the condition number of a 2D Gaussian increases when zooming in the camera (increasing the $\frac{f_x}{\mu_z}$). And according to the relationship between spectral entropy and the condition number derived in Sec. A.3, the spectral entropy $\mathrm{H}\left(\Sigma_{\text{filter}}\right)$ also decreases when zooming in the camera. The curve of $\kappa\left(\Sigma_{\text{filter}}\right)$ as a function of $\frac{f_x^2}{\mu_z^2}$ is visualized in Fig. 4 of the paper.

A.5 k in 3D Shape-Aware Splitting

For the 3D Gaussian, assuming without loss of generality that $s_1 \leq s_2 \leq s_3$, we can derive the following:

$$\kappa\left(\Sigma_{\text{split}}\right) = \frac{\max\left(\frac{s_3^2}{(k+k_0)^2}, \frac{s_2^2}{k_0^2}\right)}{\min\left(\frac{s_3^2}{(k+k_0)^2}, \frac{s_1^2}{k_0^2}\right)} = \begin{cases} \frac{k_0^2 s_3^2}{(k+k_0)^2 s_1^2}, & \text{if } \frac{k+k_0}{k_0} \leq \frac{s_3}{s_2} \\ \frac{s_2^2}{s_1^2}, & \text{if } \frac{s_3}{s_2} < \frac{k+k_0}{k_0} \leq \frac{s_3}{s_1} \\ \frac{(k+k_0)^2 s_2^2}{k_0^2 s_3^2}, & \text{if } \frac{k+k_0}{k_0} > \frac{s_3}{s_1} \end{cases}, \tag{19}$$

$$\kappa\left(\Sigma\right) = \frac{s_3^2}{s_1^2}. \tag{20}$$

Clearly, we have:

$$\frac{k_0^2 s_3^2}{\left(k+k_0\right)^2 s_1^2} < \frac{s_3^2}{s_1^2} = \kappa\left(\Sigma\right), \quad \frac{s_2^2}{s_1^2} \leq \frac{s_3^2}{s_1^2} = \kappa\left(\Sigma\right). \tag{21}$$

And when Eqn. (12) in the paper is satisfied, we can derive the following:

$$\frac{k+k_0}{k_0} < \frac{\rho^{\frac{3}{2}}\left(\Sigma\right)}{\sqrt{|\Sigma|}} = \frac{s_3^2}{s_2 s_1}, \tag{22}$$

$$\frac{\left(k+k_0\right)^2 s_2^2}{k_0^2 s_3^2} < \frac{\left(\frac{s_3^2}{s_2 s_1}\right)^2 s_2^2}{s_3^2} = \frac{s_3^2}{s_1^2} = \kappa\left(\Sigma\right). \tag{23}$$

In summary, we can conclude that the condition number after splitting does not exceed that before splitting, *i.e.*, $\kappa\left(\Sigma_{\text{split}}\right) \leq \kappa\left(\Sigma\right)$, when Eqn. (12) in the paper is satisfied.

A.6 2D View-Consistent Filtering Kernel

We prove the view-consistency of our filter:

$$\begin{aligned} \kappa\left(\Sigma_{\text{train}}\right) &= \frac{\rho\left(\mathbf{J}_{\text{train}}\Sigma'\mathbf{J}_{\text{train}}^{\top}\right) + \rho\left(\sigma\mathbf{I}\right)}{\rho_{\min}\left(\mathbf{J}_{\text{train}}\Sigma'\mathbf{J}_{\text{train}}^{\top}\right) + \rho_{\min}\left(\sigma\mathbf{I}\right)} \\ &= \frac{\rho\left(\mathbf{J}_{\text{test}}\Sigma'\mathbf{J}_{\text{test}}^{\top}\right) + \rho\left(\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)\sigma\mathbf{I}\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}\right)}{\rho_{\min}\left(\mathbf{J}_{\text{test}}\Sigma'\mathbf{J}_{\text{test}}^{\top}\right) + \rho_{\min}\left(\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)\sigma\mathbf{I}\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}\right)} \\ &= \frac{\rho\left(\Sigma_{\text{test}}\right) + \rho\left(\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)\sigma\mathbf{I}\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}\right)}{\rho_{\min}\left(\Sigma_{\text{test}}\right) + \rho_{\min}\left(\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)\sigma\mathbf{I}\left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}\right)} \\ &= \kappa\left(\Sigma_{\text{test}}\right). \end{aligned} \tag{24}$$

In Sec. A.4, when $\sigma \propto \frac{f_x^2}{\mu_z^2}$, the function $\kappa\left(\Sigma_{\text{filter}}\right)$ and $\mathrm{H}\left(\Sigma_{\text{filter}}\right)$ are constant. Therefore, we can approximate this operation using the filter kernel function $\sigma\left(f_x, \mu_z\right) = \sigma_0\frac{f_x^2}{\mu_z^2}$, where σ_0 is a constant.

B Spectral-GS Algorithm

Our method is summarized in Algorithm 1 and Algorithm 2. We mark the enhancements in related methods [Bulò et al. 2024; Huang et al. 2024b; Hyung et al. 2024; Kheradmand et al. 2024; Radl et al. 2024; Tu et al. 2025] to show their orthogonality to our approach.

C Details of Experiments

C.1 Additional Comparisons

Figures 4-7 illustrate additional qualitative comparisons over the Blender Dataset [Mildenhall et al. 2020], Deep Blending [Hedman et al. 2018], scenes captured by ourselves and the Mip-NeRF 360 Dataset [Barron et al. 2022]. Differences in quality are highlighted by insets. For better comparison, the metrics are annotated below some images. Tables 1-2 list LPIPS and spectral entropy for our evaluation over all considered approaches and scenes [Hedman et al. 2018; Knapitsch et al. 2017; Mildenhall et al. 2020].

Table 1. The LPIPS↓ and spectral entropy↑ scores for each scene of the Blender Dataset [Mildenhall et al. 2020].

LPIPS↓	Hotdog	Chair	Ship	Lego	Materials	Drums	Ficus	Mic	Avg.
3DGS [Kerbl et al. 2023]	0.204	0.213	0.310	0.189	0.141	0.190	0.111	0.063	0.178
Mip-Splatting [Yu et al. 2024]	0.126	0.114	0.216	0.111	0.113	0.139	0.065	0.031	0.114
Pixel-GS [Zhang et al. 2024]	0.210	0.215	0.312	0.192	0.146	0.196	0.100	0.063	0.179
Analytic-Splatting [Liang et al. 2024]	0.165	0.174	0.282	0.150	0.141	0.177	0.120	0.053	0.158
Analytic.+3D Filter [Liang et al. 2024]	0.133	0.112	0.219	0.119	0.115	0.139	0.071	0.031	0.117
Ours	0.099	0.098	0.196	0.098	0.100	0.134	0.063	0.028	0.102

Spectral Entropy↑	Hotdog	Chair	Ship	Lego	Materials	Drums	Ficus	Mic	Avg.
3DGS [Kerbl et al. 2023]	0.152	0.083	0.141	0.189	0.242	0.138	0.281	0.088	0.164
Mip-Splatting [Yu et al. 2024]	0.339	0.397	0.338	0.361	0.397	0.378	0.459	0.238	0.363
Pixel-GS [Zhang et al. 2024]	0.141	0.086	0.131	0.170	0.193	0.141	0.271	0.078	0.151
Analytic-Splatting [Liang et al. 2024]	0.234	0.222	0.210	0.241	0.285	0.212	0.260	0.170	0.227
Analytic.+3D Filter [Liang et al. 2024]	0.332	0.390	0.334	0.354	0.397	0.252	0.332	0.170	0.320
Ours	0.697	0.992	0.996	0.970	0.967	0.969	0.986	0.991	0.946

Algorithm 1 Spectral-GS Algorithm
W, H: width and height of the training or testing images

```
$M, S, C, O \leftarrow$ Gaussians()    ▷ Pos, Covs, Colors, Opacs
if is testing then
    $V \leftarrow$ TestingView()
    $M_p, S_p \leftarrow$ SplatGaussian($W, H, M, S, V$)    ▷ Can be combined with other methods [Huang et al. 2024a; Tu et al. 2025]
    $S_f, O_f \leftarrow$ SpectralBasedFilter($S_p, O, V$)    ▷ Filter, Ours
    DepthBasedRadixSort($M_p, S_f, O_f, C, M$)    ▷ Can be combined with Radl et al. [2024]
    $I \leftarrow$ Rasterize($W, H, M_p, S_f, C, O_f, V$)
else
    $i \leftarrow 0$    ▷ Iteration Count
    while not converged do
        $V, \hat{I} \leftarrow$ TrainingView()    ▷ Camera and Image
        $M_p, S_p \leftarrow$ SplatGaussian($W, H, M, S, V$)    ▷ Can be combined with other methods [Huang et al. 2024a; Tu et al. 2025]
        $S_f, O_f \leftarrow$ SpectralBasedFilter($S_p, O, V$)    ▷ Filter, Ours
        DepthBasedRadixSort($M_p, S_f, O_f, C, M$)    ▷ Can be combined with Radl et al. [2024]
        $I \leftarrow$ Rasterize($W, H, M_p, S_f, C, O_f, V$)
        $\mathcal{L} \leftarrow$ Loss($I, \hat{I}$)    ▷ Compute Loss, can be combined with Hyung et al. [2024]; Xie et al. [2024]
        $M, S, C, O \leftarrow$ Adam($\nabla\mathcal{L}$)    ▷ Backprop & Step, can be combined with Kheradmand et al. [2024]
        if IsRefinementIteration($i$) then
            for all $\mathcal{G}^{3D}(\boldsymbol{\mu}, \Sigma, c, o)$ in $(M, S, C, O)$ do
                if $o < \epsilon_o$ or IsInvalidSpectrum($o, \Sigma$) then
                    SpectralBasedPruneGaussian()    ▷ Ours
                end if
                if $\|\nabla_{\mu_{\text{proj}}}\mathcal{L}\|_F > \tau_{\text{loss}}$ then    ▷ Densify, can be combined with Zhang et al. [2024]
                    if $\rho(\Sigma) > \tau_{\text{radius}}$ then    ▷ Split
                        LossBasedSplitGaussian($\boldsymbol{\mu}, \Sigma, c, o$)
                    else    ▷ Clone
                        LossBasedCloneGaussian($\boldsymbol{\mu}, \Sigma, c, o$)
                    end if
                end if
                if $\mathrm{H}(\Sigma) < \tau_{\text{spectral}}$ then    ▷ Densify, Ours
                    SpectralBasedSplitGaussian($\boldsymbol{\mu}, \Sigma, c, o$)    ▷ Ours
                end if
            end for
        end if
        $i \leftarrow i + 1$
    end while
end if
```

Algorithm 2 SpectralBasedSplitGaussian
$\boldsymbol{\mu}, \Sigma, c, o$: position, covariance, color and opacity
k, k_0, K: splitting hyperparameters

```
$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$    ▷ Sampling based on the PDF
for $i \in [1, 3]$ do
    $k_i \leftarrow k \cdot \mathbb{1}\left\{s_i^2 = \rho(\Sigma)\right\} + k_0$    ▷ Anisotropic coefficient
end for
$o_1, o_2, \cdots, o_K \leftarrow o$    ▷ Can be combined with other methods [Bulò et al. 2024; Kheradmand et al. 2024]
$c_1, c_2, \cdots, c_K \leftarrow c$
$\mathbf{R}, s_1, s_2, s_3 \leftarrow \Sigma$    ▷ Eigendecompose
$\Sigma_1, \Sigma_2, \ldots, \Sigma_K \leftarrow \mathbf{R}\mathrm{diag}\left(\frac{1}{k_1^2}s_1^2, \frac{1}{k_2^2}s_2^2, \frac{1}{k_3^2}s_3^2\right)\mathbf{R}^\top$
```

Table 2. The LPIPS↓ and spectral entropy↑ for each scene of Tanks & Templates [Knapitsch et al. 2017] and Deep Blending [Hedman et al. 2018].

	LPIPS↓				Spectral Entropy↑			
	Train	Truck	Drjohnson	Playroom	Train	Truck	Drjohnson	Playroom
3DGS [Kerbl et al. 2023]	0.232	0.221	0.215	0.215	0.316	0.254	0.311	0.276
Mip-Splatting [Yu et al. 2024]	0.212	0.166	0.213	0.201	0.416	0.544	0.377	0.349
Pixel-GS [Zhang et al. 2024]	0.228	0.193	0.234	0.215	0.334	0.286	0.305	0.280
Analytic-Splatting [Liang et al. 2024]	0.231	0.176	0.217	0.205	0.336	0.377	0.334	0.292
Analytic.+3D Filter [Liang et al. 2024]	0.218	0.176	0.211	0.205	0.346	0.518	0.336	0.321
Ours	0.190	0.129	0.203	0.186	0.808	0.881	0.776	0.794

C.2 Additional Discussions

Number of Gaussians. Since our 3D shape-aware splitting increases the number of Gaussians to capture high-frequency details and prunes degenerated Gaussians with excessively low spectral entropy directly, we conduct additional quantitative experiments regarding the number of Gaussians and image quality. As shown in Fig. 2, our method outperforms others [Kerbl et al. 2023; Yu et al. 2024] when the number of Gaussians is the same. When the number of Gaussians increases, our splitting algorithm reduces needle-like artifacts, producing renderings with more details. In contrast, although other methods utilize a larger number of Gaussians, they still lack shape awareness, resulting in needle-like artifacts.

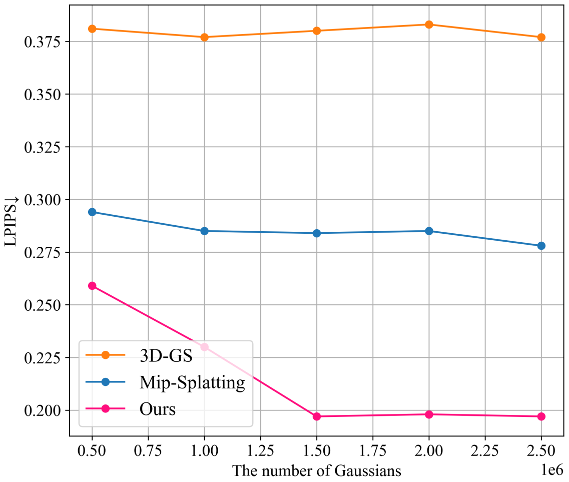


Fig. 2. Relationship between the number of Gaussians and image quality (LPIPS↓ [Zhang et al. 2018]) across different methods on Ball.

Robustness to needle-like stuff. As illustrated in Fig. 3, 3DGS [Kerbl et al. 2023] does not directly use a single needle-like Gaussian to fit such needle-like stuff, which may still produce artifacts in such cases. And our method does not lead to undesirable consequences due to the constraints on Gaussian spectral entropy, while still achieving high-fidelity rendering without noticeable artifacts.

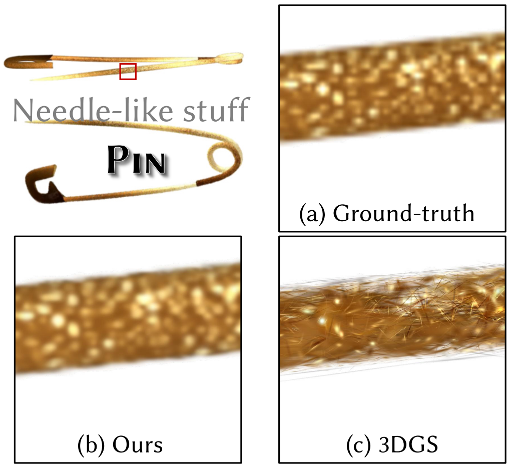


Fig. 3. Robustness of our method to needle-like stuff.

References

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

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. 2024. Revising densification in gaussian splatting. In *European Conference on Computer Vision*.

David Eisenbud. 2013. *Commutative algebra: with a view toward algebraic geometry*. Vol. 150. Springer Science & Business Media.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–15.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024b. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. 2024a. On the error analysis of 3d gaussian splatting and an optimal projection strategy. In *European conference on computer vision*. Springer, 247–263.

Junha Hyung, Susung Hong, Sungwon Hwang, Jaeseong Lee, Jaegul Choo, and Jin-Hwa Kim. 2024. Effective rank analysis and regularization for enhanced 3d gaussian splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023).

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 2024. 3D Gaussian Splatting as Markov Chain Monte Carlo. *Advances in Neural Information Processing Systems* 37 (2024), 80965–80986.

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

Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. 2024. Analytic-Splatting: Anti-Aliased 3D Gaussian Splatting via Analytic Integration. In *European conference on computer vision*.

B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*.

Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. 2024. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–17.

Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*. IEEE, 606–610.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

Xuechang Tu, Lukas Radl, Michael Steiner, Markus Steinberger, Bernhard Kerbl, and Fornando de la Torre. 2025. VRsplat: Fast and Robust Gaussian Splatting for Virtual Reality. *Proc. ACM Comput. Graph. Interact. Tech.* 8, 1, Article 1 (2025).

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. 2022. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5481–5490.

John Von Neumann. 2018. *Mathematical foundations of quantum mechanics: New edition*. Vol. 53. Princeton university press.

Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. 2024. Large language model evaluation via matrix entropy. *arXiv preprint arXiv:2401.17139* (2024).

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. 2024. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4389–4398.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-Splatting: Alias-free 3D Gaussian Splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. 2024. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*.

Dennis G Zill. 2020. *Advanced engineering mathematics*. Jones & Bartlett Learning.

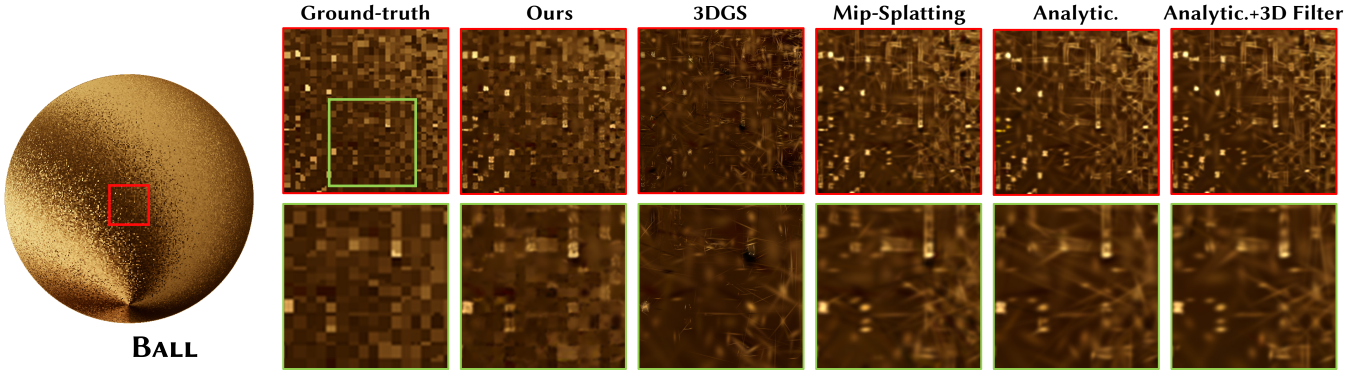


Fig. 4. **Qualitative comparisons on the synthetic scene [Verbin et al. 2022] with modified textures.** Differences in quality highlighted by insets.

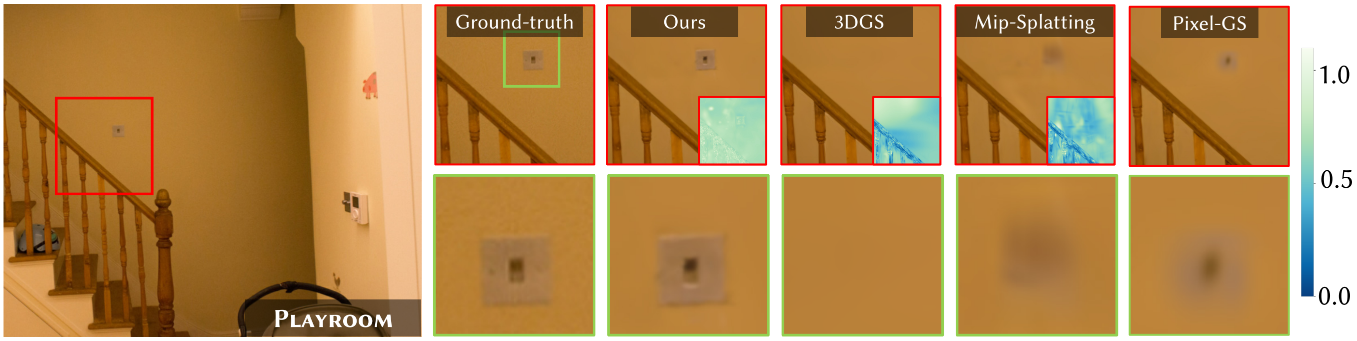


Fig. 5. **Qualitative comparisons on the Deep Blending Dataset [Hedman et al. 2018].**

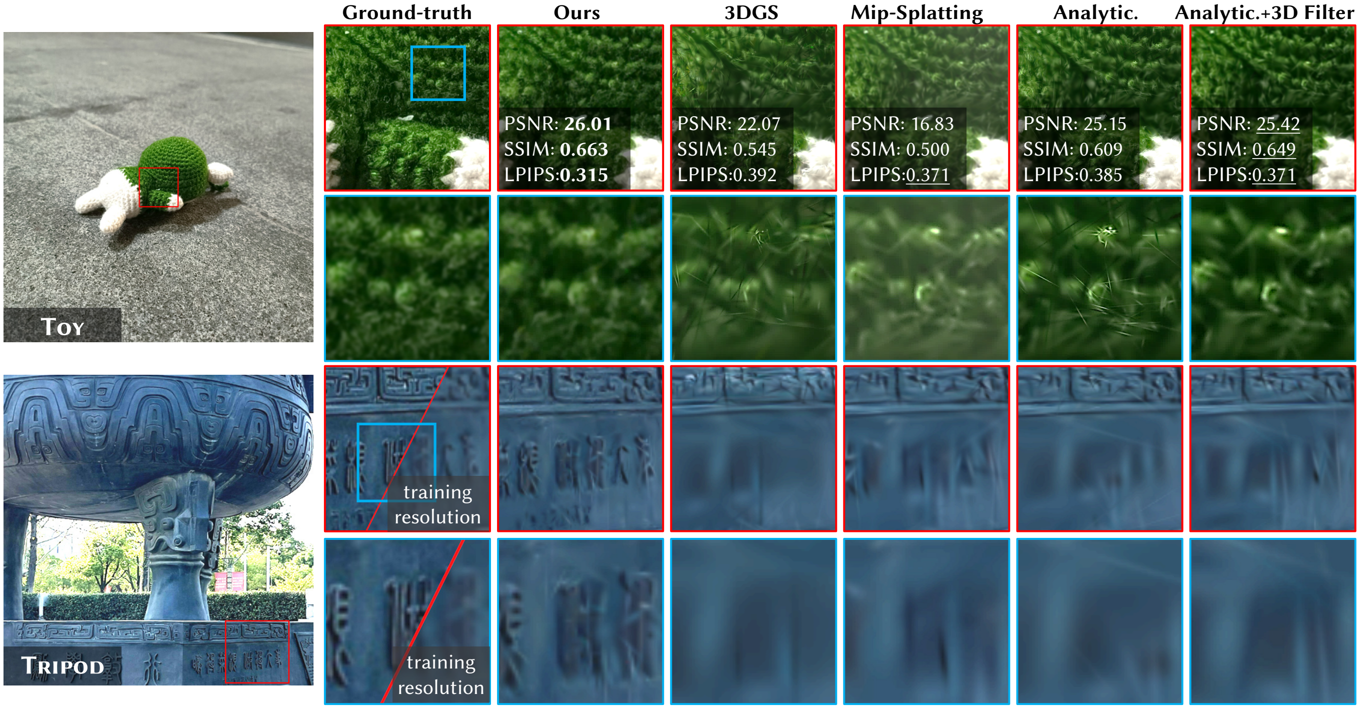


Fig. 6. **Qualitative comparisons on the real scenes captured by ourselves.** Differences in quality highlighted by insets.

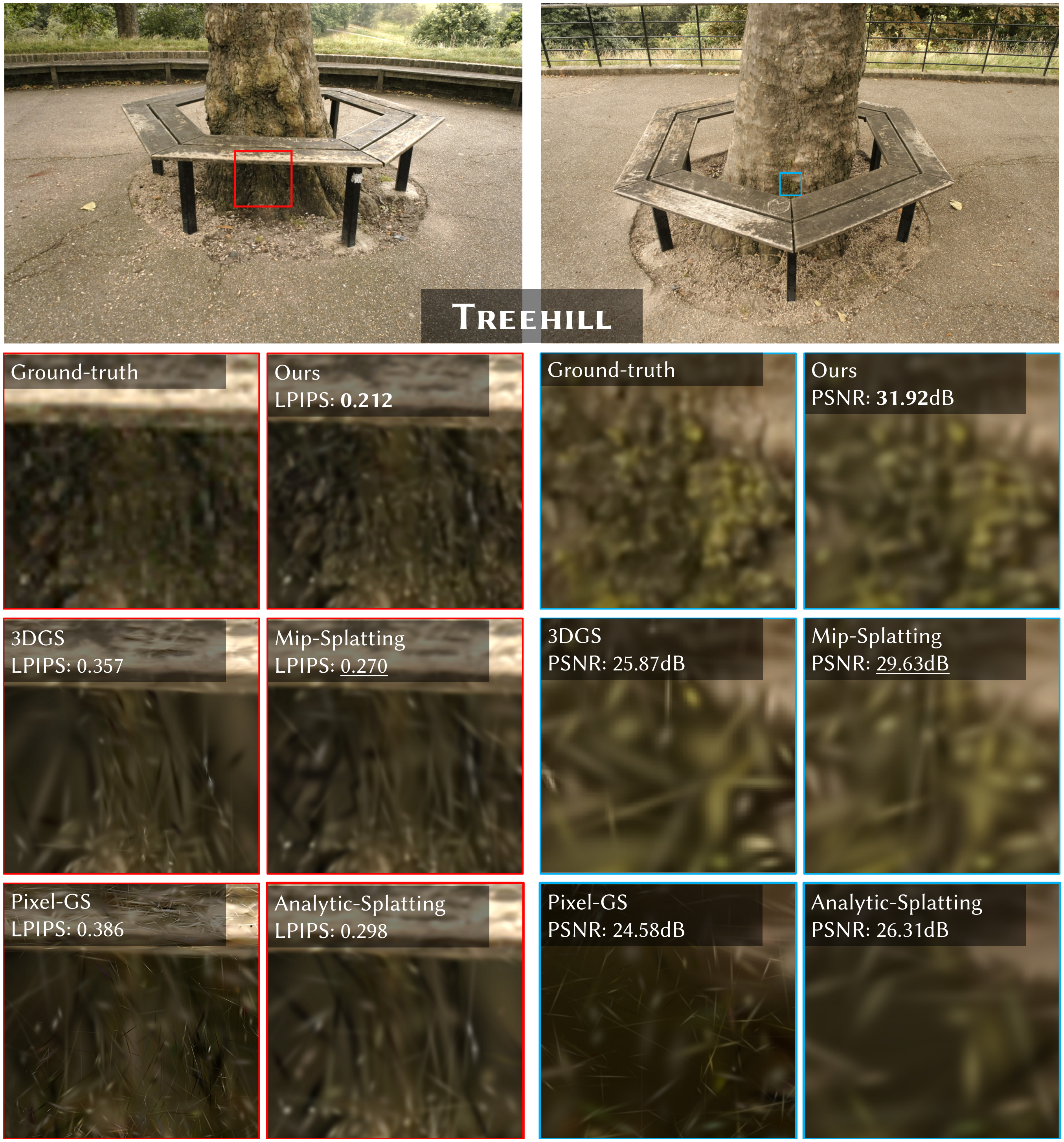


Fig. 7. **Qualitative comparisons on the Mip-NeRF 360 Dataset [Barron et al. 2022].** Differences in quality highlighted by insets.