Spectral-GS: Taming 3D Gaussian Splatting with Spectral Entropy

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

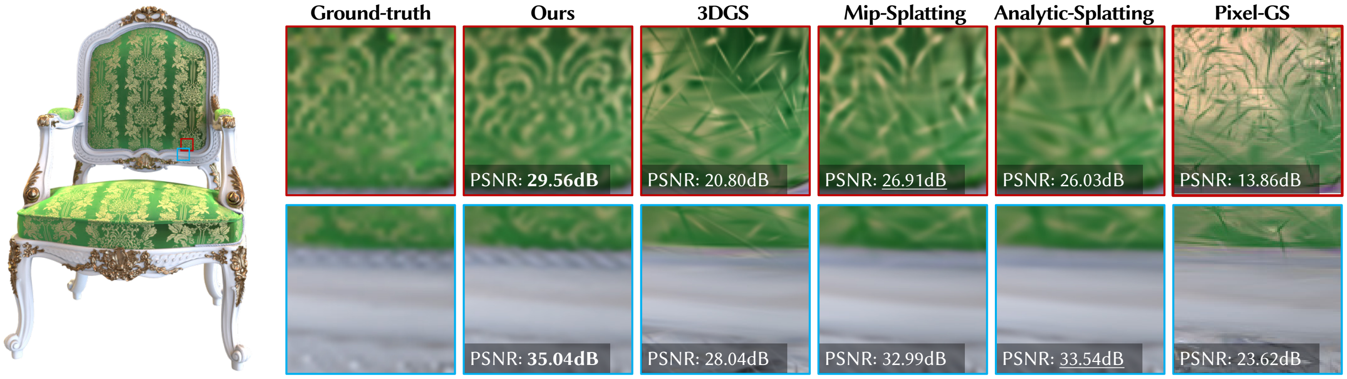

Fig. 1. Despite its high efficiency in 3D reconstruction, 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] suffers from needle-like artifacts (**red** boxes) due to undersampling or view inconsistency. Recent works [Liang et al. 2024; Yu et al. 2024; Zhang et al. 2024a] try to eliminate these artifacts. Unfortunately, they still produce needles at high-frequency regions when zooming in, and will also cause over-blurriness (**blue** boxes) since they lack shape awareness of 3D Gaussians. Through spectral analysis, we introduce Spectral-GS, which enforces shape constraints on 3D Gaussians, effectively resolving the above issues and enabling high-quality rendering without extra time cost.

Recently, 3D Gaussian Splatting (3DGS) has achieved impressive results in novel view synthesis, demonstrating high fidelity and efficiency. However, it easily exhibits needle-like artifacts, especially when increasing the sampling rate. Mip-Splatting tries to remove these artifacts with a 3D smoothing filter for frequency constraints and a 2D Mip filter for approximated supersampling. Unfortunately, it tends to produce over-blurred results, and sometimes needle-like Gaussians still persist. Our spectral analysis of the covariance matrix during optimization and densification reveals that current 3DGS lacks shape awareness, relying instead on spectral radius and view positional gradients to determine splitting. As a result, needle-like Gaussians with small positional gradients and low spectral entropy fail to split and overfit high-frequency details. Furthermore, both the filters used in 3DGS and Mip-Splatting reduce the spectral entropy and increase the condition number during zooming in to synthesize novel view, causing view inconsistencies and more pronounced artifacts. Our Spectral-GS, based on spectral analysis, introduces 3D shape-aware splitting and 2D view-consistent filtering strategies, effectively addressing these issues, enhancing 3DGS's capability to represent high-frequency details without noticeable artifacts, and achieving high-quality realistic rendering.

CCS Concepts: • **Computing methodologies** → **Rasterization**; **Image-based rendering**; **Antialiasing**; Machine learning.

Additional Key Words and Phrases: Novel view synthesis, Gaussian splatting, Anti-aliasing

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. *ACM Trans. Graph.* 44, 6, Article 1 (December 2025), 11 pages. https://doi.org/10.1145/3757377.3763907

1 Introduction

Reconstructing 3D scenes from 2D images and synthesizing novel views has been a critical task in computer graphics and vision. As the demand for real-time and photo-realistic rendering continues to rise, 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] has emerged as an efficient representation that can achieve fast reconstruction and real-time rendering [Bao et al. 2025; Chen and Wang 2024; Fei

*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

et al. 2024; Wu et al. 2024]. However, 3DGS [Kerbl et al. 2023] tends to optimize toward degraded needle-like Gaussians, resulting in unacceptable needle-like artifacts (elongated artifacts) [Hyung et al. 2024; Kerbl et al. 2023; Yu et al. 2024; Zhang et al. 2024a]. Some existing methods, such as Yu et al. [2024] and Liang et al. [2024], attempt to address these issues by employing filtering or analytic integration to mitigate aliasing. Unfortunately, when representing high-frequency textures, these methods often lead to over-blurriness or still produce artifacts.

Needle-like Gaussians correspond to 3D Gaussians with low spectral entropy and high condition number. Existing 3DGS and its variants [Kerbl et al. 2023; Yu et al. 2024] do not impose any restriction on the Gaussian's shape. The splitting strategies in 3DGS lean towards generating degenerated elongated Gaussians, due to the lack of shape-awareness, relying instead on spectral radius and view positional gradients to guide the splitting. However, needle-like Gaussians with small positional gradients are hard to split. Even after splitting, the conventional strategy leaves the Gaussian's condition number ill-conditioned, offering little improvement in mitigating needle-like artifacts. Although Hyung et al. [2024] also tries to constrain the anisotropy of Gaussians, simply introducing a covariance loss term cannot activate the densification mechanism, leading to the loss of high-frequency details. Worse still, both the EWA filter, employed in 3DGS [Kerbl et al. 2023] and in Hyung et al. [2024], and the 2D Mip filter in Mip-Splatting [Yu et al. 2024] reduce spectral entropy and increase the condition number when zooming in to synthesize novel views. Due to the view-inconsistency in filtering, needle-like artifacts become more pronounced when zooming in or when the camera moves closer to the object.

Based on the above observations, we introduce *spectral analysis* to 3D Gaussians. Specifically, we propose *3D shape-aware splitting* and *2D view-consistent filtering*, respectively, to address loss sensitivity and shape unawareness in densification, as well as view inconsistency in filtering. The splitting condition of our 3D shape-aware splitting is based on the spectral entropy of 3D Gaussians and our method ensures that the condition number after splitting reduces. The proposed 2D view-consistent filtering, combines a convolution that approximates supersampling with a view-adaptive Gaussian blur that approximates interpolation to maintain the spectral entropy consistency. Our method Spectral-GS effectively enhances 3DGS's capability to represent high-frequency details, mitigates needle-like artifacts, and achieves high-quality photorealistic rendering, as illustrated in Fig. 1. Furthermore, our method is easily implemented, requiring only few changes to the original framework. Regarding computational costs, our method achieves roughly the same performance as the original 3DGS either in training or in inference. In summary, we make the following contributions:

- We employ **spectral analysis** to examine 3DGS, revealing issues such as loss sensitivity and shape unawareness in densification, as well as view inconsistency in filtering.
- We propose **3D shape-aware splitting** to regularize needle-like Gaussians, enhancing the high-frequency detail representation for 3DGS and mitigating needle-like artifacts.
- We propose **2D view-consistent filtering** to resolve needle-like artifacts caused by view-inconsistency.

2 Related Work

In this section, we briefly review the anti-aliasing techniques in Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS).

2.1 Anti-aliasing in Neural Radiance Fields

Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] is an implicit 3D representation that employs ray casting and volume rendering. Barron et al. [2021, 2022] extend ray tracing to cone tracing and propose the integrated positional encoding (IDE) to address *aliasing and blurring induced by multi-resolution inputs*. Unlike explicit representations, NeRF relies solely on the optimization of network parameters and benefits from the interpolation capability of MLPs, eliminating both the need for densification and associated splitting artifacts. However, these same characteristics also account for its significant computational overhead. Zip-NeRF [Barron et al. 2023] and Tri-MipRF [Hu et al. 2023] combine cone tracing with an efficient grid-based representation [Müller et al. 2022] to improve performance. Rip-NeRF [Liu et al. 2024] propose to represent the faces of a Platonic solid by Ripmap Encoding, such that the projected anisotropic 2D areas can be precisely and efficiently featurized by the anisotropic area-sampling. Unfortunately, these anti-aliasing methods still cannot achieve real-time rendering.

2.2 Anti-aliasing in 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] is an explicit 3D representation that achieves real-time and photo-realistic rendering. However, it still suffers from a variety of artifacts, such as popping and needle-like artifacts, and faces challenges in rendering quality. StopThePop [Radl et al. 2024] systematically resorts and culls splats to alleviate the *popping artifacts induced by depth-based sorting*. Hahlbohm et al. [2025] introduce hybrid transparency to 3D Gaussians and approximate the accurate blending per-pixel. Ye et al. [2025] propose a novel representation that combines the surfel with Gaussian primitives, supporting sorting-free rendering. Popping artifacts fundamentally stem from view inconsistency as well. However, unlike our target problem, they originate from rotating views rather than zooming in. Huang et al. [2024] formally analyze the *projection errors induced by local affine approximation* and propose an optimal projection strategy to mitigate them. Tu et al. [2024, 2025] combine these techniques [Fang and Wang 2024; Huang et al. 2024; Radl et al. 2024] with an efficient foveated rendering routine, providing an immersive and artifact-free VR experience. Projection errors mainly manifest in zoom-out cases, making them also fundamentally orthogonal to our target problem.

Some studies which attempt to address the *blurring induced by densification and optimization* [Bulò et al. 2024; Kheradmand et al. 2024; Zhang et al. 2024b,a] and the *dilation and erosion induced by filtering* [Liang et al. 2024; Song et al. 2024; Yu et al. 2024] are more pertinent to our focus. Unfortunately, these approaches all neglect the anisotropy of 3D Gaussians, still causing *needle-like artifacts in high-frequency regions when zooming in*. EVSplitting [Feng et al. 2024] derives the closed form solutions for Gaussian splitting to generate plausible slicing results, but it still relies on the conventional splitting condition during training. Our 3D shape-aware splitting is dedicated to regularization during training rather than training-free

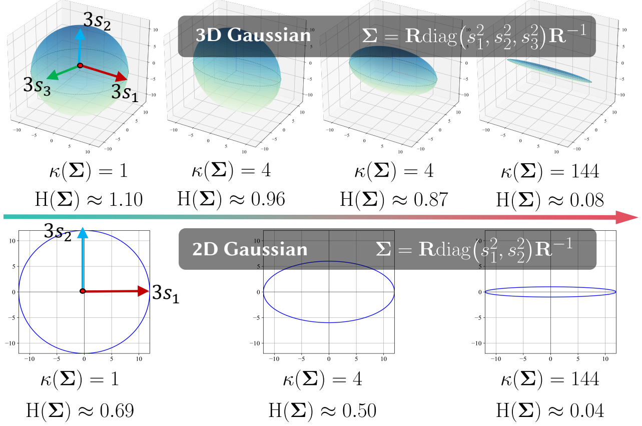
$3s_2$
$3s_3$
$3s_1$
3D Gaussian
$\Sigma = \mathbf{R}\mathrm{diag}(s_1^2, s_2^2, s_3^2)\mathbf{R}^{-1}$
$\kappa(\mathbf{\Sigma}) = 1$
$\mathrm{H}(\mathbf{\Sigma}) \approx 1.10$
$\kappa(\mathbf{\Sigma}) = 4$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.96$
$\kappa(\mathbf{\Sigma}) = 4$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.87$
$\kappa(\mathbf{\Sigma}) = 144$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.08$
$3s_2$
$3s_1$
2D Gaussian
$\Sigma = \mathbf{R}\mathrm{diag}(s_1^2, s_2^2)\mathbf{R}^{-1}$
$\kappa(\mathbf{\Sigma}) = 1$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.69$
$\kappa(\mathbf{\Sigma}) = 4$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.50$
$\kappa(\mathbf{\Sigma}) = 144$
$\mathrm{H}(\mathbf{\Sigma}) \approx 0.04$

Fig. 2. **Visualization of Gaussians with the same spectral radius but different shapes.** The spectrum of the 3D Gaussian is characterized by s_1, s_2, s_3 (top row), while the 2D Gaussian is characterized by s_1, s_2 (bottom row). From left to right, as the spectral entropy decreases and the condition number increases, the Gaussians transition from isotropic to anisotropic.

editing. Therefore, employing lightweight spectral analysis instead of the computationally expensive error function (erf) is well justified. Although Xie et al. [2024] and Hyung et al. [2024] also incorporate the shape regularization during training, our theoretical analysis and experiments reveal that covariance regularization in the loss cannot activate the densification mechanism. Consequently, while these approaches mitigate needle-like artifacts when zooming in, the results become overly smoothed and loss details.

3 Methodology

3.1 Preliminaries

Representation and projection. 3D Gaussian Splatting [Kerbl et al. 2023] constructs a scene representation using volumetric primitives $\mathcal{G}^{3D}(\cdot)$, each characterized by position $\boldsymbol{\mu}$, a covariance matrix Σ (decomposed into scale $\mathbf{S} \in \mathbb{R}^{3\times 3}$ and rotation $\mathbf{R} \in SO(3)$, *i.e.*, $\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^\top\mathbf{R}^\top$), opacity o, and spherical harmonics coefficients SH $(\cdot)$. The 3D primitives are projected to 2D image space through a Jacobian matrix $\mathbf{J} \in \mathbb{R}^{2\times 3}$ for the local affine approximation after being transformed to the camera space via a viewing transformation matrix $\mathbf{W}$. Then 2D Gaussians $\mathcal{G}^{2D}(\cdot)$, each characterized by the position $\boldsymbol{\mu}_{\text{proj}}$ and a covariance matrix $\Sigma_{\text{proj}} = \mathbf{J}\mathbf{W}\Sigma\mathbf{W}^\top\mathbf{J}^\top$, are rasterized using α-blending.

Optimization and densification. 3DGS [Kerbl et al. 2023] employs a loss function that combines $\mathcal{L}_1$ loss with a D-SSIM term:

$$\mathcal{L} = (1-\lambda_1)\mathcal{L}_1 + \lambda_1\mathcal{L}_{\text{D-SSIM}}. \tag{1}$$

For this loss can only optimize the parameters of Gaussian primitives but cannot change the number of primitives, phenomena such as “over-reconstruction” (regions where Gaussians cover large areas in the scene) and “under-reconstruction” (regions with missing geometric features) can occur. To address this, 3DGS introduces an adaptive densification scheme. For Gaussians with large view-space positional gradients $\nabla_{\boldsymbol{\mu}_{\text{proj}}}\mathcal{L}$, the scheme chooses between clone and split strategies based on the scale of Gaussians. After splitting, the shape remains unchanged, with the scale being $\frac{1}{k}$ of the original.

Filtering and Mip-Splatting. To prevent projected Gaussians from becoming too small to cover an entire pixel, 3DGS uses an EWA filter [Zwicker et al. 2002] for training stability:

$$\mathcal{G}_k^{2D}(\mathbf{u})_{\text{EWA}} = oe^{-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}})^\top(\Sigma_{\text{proj}}+\sigma\mathbf{I})^{-1}(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}})} \tag{2}$$

where $\mathbf{u}$ is the pixel coordinate, $\mathbf{I}$ is a 2D identity matrix, σ is a scalar hyperparameter to control the size of the filter, and $\mathcal{G}_k^{2D}(\cdot)_{\text{EWA}}$ is the EWA filtered Gaussian.

To limit the maximal frequency for the 3D representation, Mip-Splatting [Yu et al. 2024] applies a 3D smoothing filter $\mathcal{G}_{\text{low}}$ to the 3D Gaussians $\mathcal{G}^{3D}$, ensuring that the regularized Gaussians cover at least one pixel in all training views:

$$\begin{aligned}\mathcal{G}_k^{3D}(\mathbf{x})_{\text{reg}} &= \left(\mathcal{G}^{3D} \otimes \mathcal{G}_{\text{low}}\right)(\mathbf{x}) \\ &= o\sqrt{\frac{|\Sigma|}{\left|\Sigma + \frac{\sigma}{\hat{\nu}_k}\cdot\mathbf{I}\right|}}\,e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\left(\Sigma+\frac{\sigma}{\hat{\nu}_k}\cdot\mathbf{I}\right)^{-1}(\mathbf{x}-\boldsymbol{\mu})}.\end{aligned} \tag{3}$$

Here, the scale $\frac{\sigma}{\hat{\nu}}$ of the 3D filters for each primitive are different as they depend on the training views in which they are visible. In addition, the box filter of a function is equivalent to the integral over the corresponding region, while the convolution of Gaussians remains a Gaussian. Hense, Mip-Splatting [Yu et al. 2024] proposes a 2D Mip filter $\mathcal{G}_{\text{mip}}$ to approximate the integral or super-sampling within a pixel:

$$\begin{aligned}\mathcal{G}_k^{2D}(\mathbf{u})_{\text{Mip}} &= \left(\mathcal{G}^{2D} \otimes \mathcal{G}_{\text{mip}}\right)(\mathbf{u}) \\ &= o\sqrt{\frac{\left|\Sigma_{\text{proj}}\right|}{\left|\Sigma_{\text{proj}} + \sigma\mathbf{I}\right|}}\,e^{-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}})^\top(\Sigma_{\text{proj}}+\sigma\mathbf{I})^{-1}(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}})}\end{aligned} \tag{4}$$

where $\mathcal{G}_k^{2D}(\cdot)_{\text{Mip}}$ is the 2D Mip filtered Gaussian. To compute the Gaussian integral within a pixel more analytically, Analytic-Splatting [Liang et al. 2024] derives the cumulative distribution function (CDF) of the Gaussians, replacing the 2D Mip filter used to approximate a box filter.

3.2 Spectral Analysis of 3D Gaussian Splatting

In this section, we first perform spectral analysis to 3DGS and show loss-sensitivity and shape-unwareness in densification, and view-inconsistency in filtering.

Spectra of Gaussians. The covariance matrix Σ of a 3D/2D Gaussian is analogous to describing the configuration of an ellipsoid/ellipse. For example, the covariance matrix of 3D Gaussian can be eigendecomposed as follows:

$$\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^\top\mathbf{R}^\top = \mathbf{R}\left(\mathbf{S}\mathbf{S}^\top\right)\mathbf{R}^{-1} = \mathbf{R}\mathrm{diag}\left(s_1^2, s_2^2, s_3^2\right)\mathbf{R}^{-1} \tag{5}$$

where the rotation matrix $\mathbf{R}$ is an orthogonal matrix ($\mathbf{R}^\top = \mathbf{R}^{-1}$), $3s_1, 3s_2, 3s_3$ represent the lengths of the ellipsoid's three axes (3σ rule) [Kerbl et al. 2023], and s_1^2, s_2^2, s_3^2 are the eigenvalues (spectrum) of Σ. The spectral radius of the covariance matrix Σ is derived as:

$$\rho(\Sigma) = \max\left(s_1^2, s_2^2, s_3^2\right) \tag{6}$$

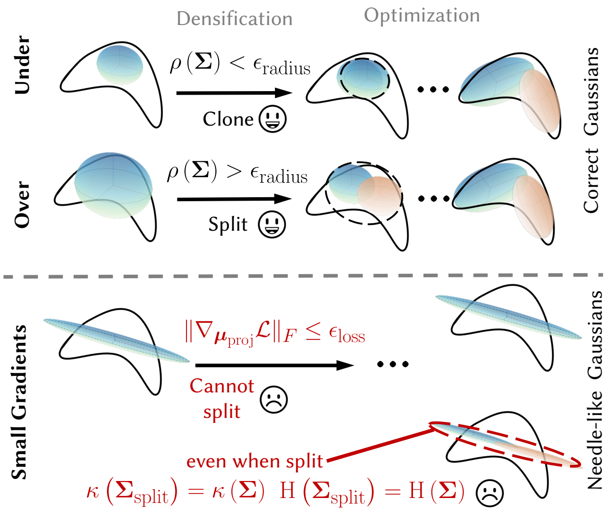
Densification
Optimization
Under
$\rho(\mathbf{\Sigma}) < \epsilon_{\text{radius}}$
Clone
Over
$\rho(\mathbf{\Sigma}) > \epsilon_{\text{radius}}$
Split
Correct Gaussians
Small Gradients
$\|\nabla_{\boldsymbol{\mu}_{\text{proj}}}\mathcal{L}\|_F \leq \epsilon_{\text{loss}}$
Cannot split
even when split
$\kappa(\mathbf{\Sigma}_{\text{split}}) = \kappa(\mathbf{\Sigma})$ $\text{H}(\mathbf{\Sigma}_{\text{split}}) = \text{H}(\mathbf{\Sigma})$
Needle-like Gaussians

Fig. 3. **Illustration of the optimization and densification of Gaussians in 3DGS.** *Correct Gaussians:* When view-positional gradients $\nabla_{\boldsymbol{\mu}_{\text{proj}}}\mathcal{L}$ exceed a certain threshold τ_{loss}, 3DGS decides to clone or split based on the Gaussian's spectral radius $\rho(\Sigma)$. *Needle-like Gaussians:* However, 3DGS does not split Gaussians with low spectral entropy but small gradients.

and can be used to measure the scale of the Gaussian. Additionally, the condition number and spectral entropy [Shannon 1948; Von Neumann 2018; Wei et al. 2024] are respectively

$$\kappa(\Sigma) = \frac{\rho(\Sigma)}{\rho_{\min}(\Sigma)} = \frac{\max\left(s_1^2, s_2^2, s_3^2\right)}{\min\left(s_1^2, s_2^2, s_3^2\right)}, \tag{7}$$

$$\text{H}(\Sigma) = \text{tr}\left(-\frac{\Sigma}{\text{tr}(\Sigma)}\ln\frac{\Sigma}{\text{tr}(\Sigma)}\right) = -\sum_{i=1}^{3}\frac{s_i^2}{\text{tr}(\Sigma)}\ln\frac{s_i^2}{\text{tr}(\Sigma)}, \tag{8}$$

where $\text{tr}(\Sigma) = s_1^2 + s_2^2 + s_3^2$. These metrics can be used to measure the shape or degree of anisotropy of the Gaussian. In Fig. 2, we visualize different 3D/2D Gaussians with the same spectral radius.

It is easily proven that when $s_1 = s_2 = s_3$, the condition number is minimized and the spectral entropy is maximized. Furthermore, the eccentricity e of the ellipse described by Σ of the 2D Gaussian satisfies $e = \sqrt{1 - \frac{1}{\kappa(\Sigma)}}$, which indicates that a higher condition number corresponds to a sharper shape of the Gaussian as shown in Fig. 2. And the needle-like artifacts correspond to Gaussians with a low spectral entropy and a high condition number. In characterizing the shape of 3D Gaussians, spectral entropy is more exact than the condition number. However, the condition number is more suitable for derivations and calculation, especially in the case of 2D Gaussians. Please refer to the detailed proofs and analysis in the supplementary materials.

Loss-sensitivity and shape-unawareness in densification. In Sec. 3.1, we have briefly introduced the optimization and densification in 3DGS [Kerbl et al. 2023]. Here, we further analyze the spectra of the Gaussians involved in the densification. As shown in Fig. 3, 3DGS employs densification to grow the quantity of Gaussians when the loss gradient $\nabla_{\boldsymbol{\mu}_{\text{proj}}}\mathcal{L}$ exceed a certain threshold τ_{loss}. While this approach effectively addresses issues of "over-reconstruction" and "under-reconstruction", it is highly sensitive to the design of the loss function $\mathcal{L}$ and the chosen threshold τ_{loss}. Specifically, when elongated Gaussians with low spectral entropy can fit high-frequency textures or geometry with small loss in the training views, the densification mechanism is not triggered. This can lead to needle-like artifacts or high-frequency over-blurriness. Furthermore, even when splitting, the spectral entropy and the condition number of the Gaussians remain consistent with that before the split and does not significantly alleviate needle-like artifacts, *i.e.*, $\rho\left(\Sigma_{\text{split}}\right) = \frac{1}{k^2}\rho(\Sigma), \kappa\left(\Sigma_{\text{split}}\right) = \kappa(\Sigma), \text{H}\left(\Sigma_{\text{split}}\right) = \text{H}(\Sigma)$.

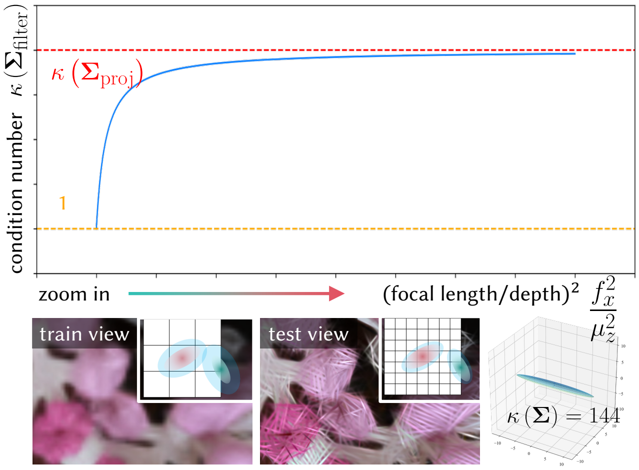

Fig. 4. **The condition number and rendering results when zooming in.** We fix the condition number $\kappa(\Sigma) = 144$ during training. Due to view-inconsistency of filtering, the train view still produces plausible rendering results, but the test view with higher $\frac{f_x^2}{\mu_z^2}$ shows needle-like artifacts.

View-inconsistency in filtering. In Sec. 3.1, we have provided a brief introduction to the EWA filter [Zwicker et al. 2002] in 3DGS and the 2D Mip filter in Mip-Splatting [Yu et al. 2024]. From Eqn. (2) and 4, we observe that the two filters have identical covariance matrices $\Sigma_{\text{filter}} = \Sigma_{\text{proj}} + \sigma\mathbf{I}$, differing only in the opacity term, *i.e.*, $o \neq o\sqrt{\frac{|\Sigma_{\text{proj}}|}{|\Sigma_{\text{proj}}+\sigma\mathbf{I}|}}$. However, when zooming in, the Jacobian matrix $\mathbf{J}$ changes, leading to a change in the projected covariance matrix Σ_{proj} while the filter kernel $\sigma\mathbf{I}$ remains constant. This causes variations in the condition number of 2D Gaussians after optimization:

$$\kappa\left(\Sigma_{\text{train}}\right) = \frac{\rho\left(\mathbf{J}_{\text{train}}\Sigma^{'}\mathbf{J}_{\text{train}}^{\top}\right) + \sigma}{\rho_{\min}\left(\mathbf{J}_{\text{train}}\Sigma^{'}\mathbf{J}_{\text{train}}^{\top}\right) + \sigma} \neq \kappa\left(\Sigma_{\text{test}}\right) \tag{9}$$

where $\Sigma^{'} = \mathbf{W}\Sigma\mathbf{W}^{\top}$ is the covariance matrix in the camera space and $\Sigma_{\text{train}} = \mathbf{J}_{\text{train}}\Sigma^{'}\mathbf{J}_{\text{train}}^{\top} + \sigma\mathbf{I}$, $\Sigma_{\text{test}} = \mathbf{J}_{\text{test}}\Sigma^{'}\mathbf{J}_{\text{test}}^{\top} + \sigma\mathbf{I}$ denote the covariance matrices during training and testing, respectively. We have derived and visualized the variation curve of $\kappa\left(\Sigma_{\text{filter}}\right)$ during

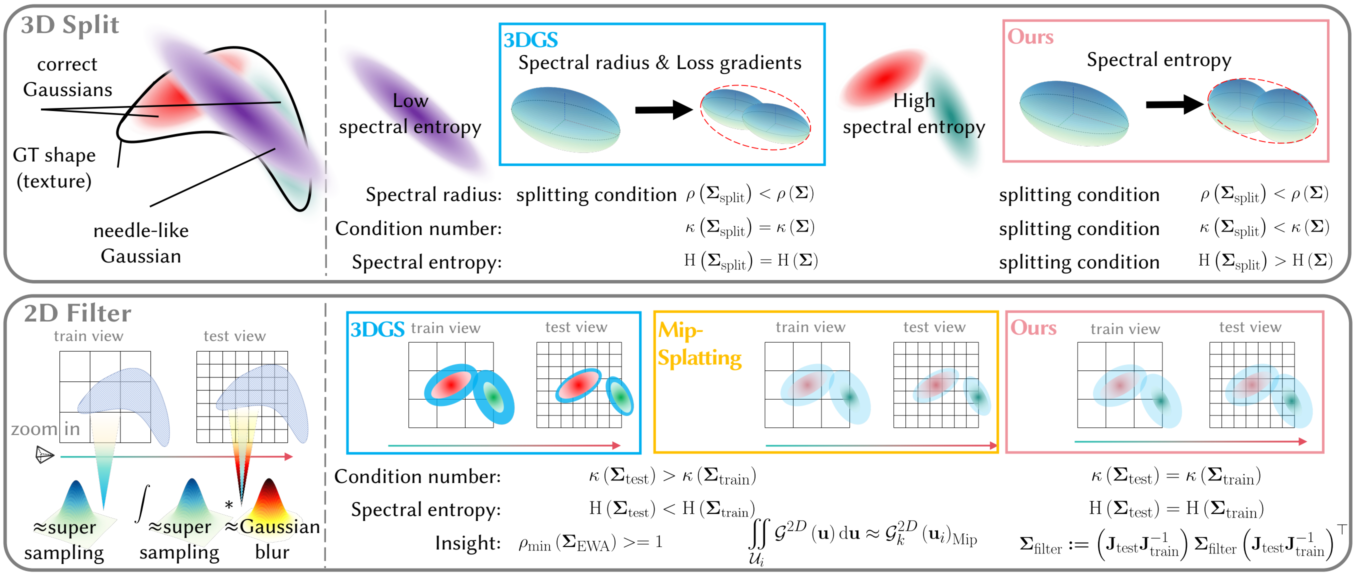
3D Split
correct Gaussians
GT shape (texture)
needle-like Gaussian
Low spectral entropy
3DGS
Spectral radius & Loss gradients
High spectral entropy
Ours
Spectral entropy
Spectral radius:
splitting condition $\rho\left(\Sigma_{\text{split}}\right) < \rho\left(\Sigma\right)$
Condition number:
$\kappa\left(\Sigma_{\text{split}}\right) = \kappa\left(\Sigma\right)$
Spectral entropy:
$\mathrm{H}\left(\Sigma_{\text{split}}\right) = \mathrm{H}\left(\Sigma\right)$
splitting condition $\rho\left(\Sigma_{\text{split}}\right) < \rho\left(\Sigma\right)$
splitting condition $\kappa\left(\Sigma_{\text{split}}\right) < \kappa\left(\Sigma\right)$
splitting condition $\mathrm{H}\left(\Sigma_{\text{split}}\right) > \mathrm{H}\left(\Sigma\right)$
2D Filter
train view
test view
zoom in
≈super sampling
≈super sampling
*
≈Gaussian blur
3DGS
train view
test view
Mip-Splatting
train view
test view
Ours
train view
test view
Condition number:
$\kappa\left(\Sigma_{\text{test}}\right) > \kappa\left(\Sigma_{\text{train}}\right)$
Spectral entropy:
$\mathrm{H}\left(\Sigma_{\text{test}}\right) < \mathrm{H}\left(\Sigma_{\text{train}}\right)$
Insight:
$\rho_{\min}\left(\Sigma_{\text{EWA}}\right) >= 1$
$\iint_{\mathcal{U}_i} \mathcal{G}^{2D}\left(\mathbf{u}\right) \mathrm{d}\mathbf{u} \approx \mathcal{G}_k^{2D}\left(\mathbf{u}_i\right)_{\text{Mip}}$
$\kappa\left(\Sigma_{\text{test}}\right) = \kappa\left(\Sigma_{\text{train}}\right)$
$\mathrm{H}\left(\Sigma_{\text{test}}\right) = \mathrm{H}\left(\Sigma_{\text{train}}\right)$
$\Sigma_{\text{filter}} := \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right) \Sigma_{\text{filter}} \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}$

Fig. 5. **Overview of Spectral-GS.** 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] decides whether to split based on the positional gradients and the spectral radius of the covariance matrix without considering the shape of primitives. We propose the 3D shape-aware splitting strategy based on the spectral analysis **(3D Split)**. In screen space, both the EWA filter [Zwicker et al. 2002] of 3DGS which attempts to cover an entire pixel, and the Mip filter of Mip-Splatting [Yu et al. 2024] which approximates supersampling, result in a reduction of spectral entropy when zooming in to synthesize novel view. Our view-consistent filter's kernel is not constant to maintain the spectral entropy consistency **(2D Filter)**.

the zoom-in process. As shown in the upper part of Fig. 4, this function increases as the camera zooms in, *i.e.*, as the $\frac{\text{focal length}}{\text{depth}}$ increases. Please refer to the supplementary materials for detailed derivations.

Due to the view-inconsistency in filtering, needle-like artifacts become more pronounced when zooming in or when the camera moves closer to the object, as illustrated by the rendering results in the lower portion of Fig. 4.

3.3 Spectral-GS

Based on the spectral analysis for 3DGS [Kerbl et al. 2023] and Mip-Splatting [Yu et al. 2024] in Sec. 3.2, we propose the 3D shape-aware splitting and the 2D view-consistent filtering, respectively, to address loss sensitivity and shape unawareness in densification, as well as view inconsistency in filtering. The overview of our method Spectral-GS is illustrated in Fig. 5.

3D shape-aware splitting. We propose the 3D shape-aware splitting to introduce shape-awareness into the optimization process. As shown in Ours of 3D Split in Fig. 5, the splitting condition of our strategy is based on the spectral entropy of 3D Gaussians $\mathrm{H}(\Sigma)$. When the spectral entropy falls below a certain threshold τ_{spectral} and may exhibit visually needle-like artifacts, K points are sampled based on the probability density function (PDF) of the old Gaussian distribution $\mathcal{G}^{3D}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$. The new Gaussian mixture distributions $\sum_K \mathcal{G}^{3D}\left(\mathbf{x}; \boldsymbol{\mu}_{\text{split}}, \Sigma_{\text{split}}\right)$ aim to fit the old Gaussian distribution $\mathcal{G}^{3D}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ as closely as possible, preserving high-frequency while increasing spectral entropy. Specifically, the reduction factor of the covariance matrix is not isotropic but anisotropic. The greedy algorithm reduces the spectral radius by the maximum extent:

$$\Sigma_{\text{split}} = \mathbf{R}\mathrm{diag}\left(\frac{1}{k_1^2}s_1^2, \frac{1}{k_2^2}s_2^2, \frac{1}{k_3^2}s_3^2\right)\mathbf{R}^{\top}, \tag{10}$$

$$k_i = k \cdot \mathbb{1}\left\{s_i^2 = \rho(\Sigma)\right\} + k_0, \tag{11}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function and $k > 0, k_0 \geq 1$. Additionally, to ensure that the condition number after splitting does not exceed that before splitting, the following condition must be satisfied:

$$k < -k_0 + \frac{k_0 \rho^{\frac{3}{2}}(\Sigma)}{\sqrt{|\Sigma|}}. \tag{12}$$

Moreover, we prune Gaussians with extremely low spectral entropy in a manner similar to how we handle those with low opacity, ensuring that the number of Gaussians does not increase excessively due to splitting.

2D view-consistent filtering. For novel view synthesis, the level of scene detail is determined by the resolution of the training images. The finer details revealed by zooming in actually correspond to views that are not seen in the training set. Due to view-inconsistent filtering, 3DGS [Kerbl et al. 2023] and Mip-Splatting [Yu et al. 2024] produce pronounced artifacts when zooming in. The needle-like artifacts amplified by zooming in resemble the pixelation artifacts typically observed when upscaling low-resolution images. A widely used and efficient strategy to ensure consistency between the original and the upscaled image is interpolation. Motivated by this, we propose the 2D view-consistent filtering, which combining a convolution that approximates supersampling with a Gaussian blur that approximates interpolation, as shown in Ours of 2D Filter in Fig. 5. We first tend to approximate the integral of projected 2D Gaussians within each pixel window area with the 2D Gaussian filter like Eqn. (4):

$$\begin{aligned}\mathcal{G}_k^{2D}\left(\mathbf{u}\right)_{\text{Box}} &= \iint_{\mathcal{U}} \mathcal{G}^{2D}\left(\mathbf{u}\right) \mathrm{d}\mathbf{u} = \left(\mathcal{G}^{2D} \otimes \mathcal{B}\right)\left(\mathbf{u}\right) \\ &\approx \mathcal{G}_k^{2D}\left(\mathbf{u}\right)_{\text{Mip}} = \left(\mathcal{G}^{2D} \otimes \mathcal{G}_{\text{mip}}\right)\left(\mathbf{u}\right)\end{aligned} \tag{13}$$

where $\mathcal{B}$ denotes the 2D box filter. Subsequently, we introduce the Gaussian blur to achieve view-consistency of the condition number, with the size of the convolution kernel given by:

$$\Sigma_{\text{blur}} = \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right) \sigma\mathbf{I} \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top} - \sigma\mathbf{I}. \tag{14}$$

Note that the matrix $\mathbf{J}_{\text{train}} \in \mathbb{R}^{2\times3}$ is not a full-rank square matrix; $\mathbf{J}_{\text{train}}^{-1} \in \mathbb{R}^{3\times2}$ represents a left/right inverse of matrix $\mathbf{J}_{\text{train}}$. Since convolving two Gaussians with covariance matrices Σ_1, Σ_2 results in another Gaussian with variance $\Sigma_1 + \Sigma_2$, we obtain the result of combining the 2D box filter with Gaussian blur $\mathcal{G}^{blur}$ as follows:

$$\begin{aligned}\mathcal{G}_k^{2D}\left(\mathbf{u}\right)_{\text{filter}} &= \left(\mathcal{G}^{2D} \otimes \mathcal{B} \otimes \mathcal{G}^{blur}\right)\left(\mathbf{u}\right) \\ &\approx o_{\text{filter}} e^{-\frac{1}{2}\left(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}}\right)^{\top}\Sigma_{\text{filter}}{}^{-1}\left(\mathbf{u}-\boldsymbol{\mu}_{\text{proj}}\right)},\end{aligned} \tag{15}$$

$$o_{\text{filter}} = o\sqrt{\frac{\left|\Sigma_{\text{proj}}\right|}{\left|\Sigma_{\text{proj}} + \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right) \sigma\mathbf{I} \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}\right|}}, \tag{16}$$

$$\Sigma_{\text{filter}} = \Sigma_{\text{proj}} + \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right) \sigma\mathbf{I} \left(\mathbf{J}_{\text{test}}\mathbf{J}_{\text{train}}^{-1}\right)^{\top}. \tag{17}$$

Compared to the 2D Mip filter, our 2D view-consistent filter does not use a constant kernel but instead employs a view-adaptive kernel. We can also approximate Eqn. (17) using the filter kernel function $\sigma\left(\text{focal length}, \text{depth}\right) = \sigma_0 \frac{\text{focal length}^2}{\text{depth}^2}$, where σ_0 is a constant. Please refer to the detailed proofs in the supplementary materials.

Compared to $\mathbf{J}_{\text{train}}$, the focal length and depth are more accessible at both training and test time. The parameters from multiple views can be reduced and moving closer corresponds to a decrease in depth. While our filter shares similarities with the 3D smoothing filter, their underlying principles are fundamentally distinct. The 3D smoothing filter in Mip-Splatting [Yu et al. 2024] is applied only during training to ensure that the highest-frequency component of any Gaussian does not exceed half of its maximum sampling rate for at least one camera. In contrast, beyond considering supersampling, our filter primarily focuses on preserving spectral entropy consistency between training and testing views.

4 Experiments

4.1 Experiment Settings

Implementation details. We implement Spectral-GS based on the PyTorch framework in 3DGS [Kerbl et al. 2023]. We use the default parameters of 3DGS to maintain consistency with the original 3DGS. For our approach, we empirically set the threshold τ_{spectral} in 3D shape-aware splitting to 0.5, with $k = 0.6$, $k_0 = 1$ and $K = 2$, and σ_0 in 2D view-consistent filtering to 0.1.

Datasets. We evaluate our approach on a total of 13 scenes, which includes eight scenes from the Blender Dataset [Mildenhall et al. 2020] and four scenes from the Tanks & Templates [Knapitsch et al. 2017] and Deep Blending [Hedman et al. 2018]. Additionally, we evaluate our method on the Ball from Verbin et al. [2022] with modified textures to validate the relationship with frequency.

Baselines and metrics. We compare Spectral-GS with 3DGS [Kerbl et al. 2023] and some current SOTA methods [Liang et al. 2024; Yu et al. 2024; Zhang et al. 2024a]. In these experiments, Analytic.+3D Filter refer to Analytic-Splatting [Liang et al. 2024] with the 3D smoothing filter from Yu et al. [2024]. Standard metrics such as PSNR, LPIPS [Zhang et al. 2018], and SSIM are used for the evaluation. Furthermore, we provide the spectral entropy to evaluate needle-like artifacts and verify the correlation between the spectral entropy of 3D Gaussians and the quality of novel view synthesis.

4.2 Results

Quantitative comparisons. We report quantiative results in Tab. 1. All methods are evaluated across four focal lengths (*i.e.*1×, 2×, 4×, 8×) to mimic zoom-in effects. These results demonstrate that our spectral analysis-based method effectively increases the spectral entropy of scenes, thereby enhancing images' quality.

Qualitative comparisons. As illustrated in Fig. 8 and Fig. 9, it can be observed that our method is capable of generating more realistic details, with fewer needle-like artifacts compared to other methods [Kerbl et al. 2023; Liang et al. 2024; Yu et al. 2024; Zhang et al. 2024a]. This is precisely the superiority brought about by our spectral analysis-based method, which results in higher spectral entropy.

Computational costs. Regarding computational costs, our method achieves roughly the same performance as the oringal 3DGS [Kerbl et al. 2023] either in training or in inference. We provide a detailed analysis as follows and report detailed timing benchmarks for both training and inference in Tab. 4.

- **Complexity analysis of 3D shape-aware splitting:** Our 3D shape-aware splitting, compared to traditional 3DGS [Kerbl et al. 2023], only adds a spectral entropy criterion and modifies the splitting results. As a result, our method does *not lead to higher computational costs.*
- **Complexity analysis of 2D view-consistent filtering:** Both 3DGS [Kerbl et al. 2023] and Mip-Splatting [Yu et al. 2024] add a constant to the covariance of the 2D Gaussian in screen space, while our method adds a variable. As a result, there is no difference in runtime. Our method remains *high performance in real-time applications.*

4.3 Discussions

Ablation study. We conduct ablation studies in the Ball scene, as shown in Fig. 6 and Tab. 3. The observed needle-like artifacts are composed of two parts: those inherent to the 3D scene and those generated by the rendering algorithm. The artifacts inherent to the scene are addressed by regulating the 3D Gaussians' spectral entropy

Table 1. **Quantitative comparisons on the Blender Dataset [Mildenhall et al. 2020].** All methods are evaluated across four focal lengths (1×, 2×, 4×, and 8×), with evaluations at higher sampling rates simulating zoom-in effects.

	PSNR ↑					SSIM ↑					LPIPS ↓					
	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	Spectral Entropy↑
3DGS [Kerbl et al. 2023]	33.76	26.13	23.09	23.88	26.72	0.969	0.879	0.795	0.843	0.872	0.039	0.138	0.261	0.274	0.178	0.164
Mip-Splatting [Yu et al. 2024]	33.97	29.89	28.51	28.63	30.25	0.969	0.924	0.906	0.929	0.932	0.031	0.100	0.161	0.165	0.114	0.363
Pixel-GS [Zhang et al. 2024a]	33.93	26.11	23.07	22.56	26.42	0.970	0.875	0.793	0.829	0.867	0.029	0.133	0.261	0.293	0.179	0.151
Analytic-Splatting [Liang et al. 2024]	34.02	29.29	27.78	27.68	29.69	0.970	0.911	0.863	0.860	0.901	0.030	0.119	0.218	0.263	0.158	0.227
Analytic.+3D Filter [Liang et al. 2024]	33.89	29.88	28.48	28.79	30.26	0.970	0.924	0.907	0.927	0.932	0.031	0.101	0.160	0.174	0.117	0.320
Ours	34.02	30.04	29.10	29.61	30.69	0.972	0.927	0.923	0.936	0.940	0.026	0.088	0.140	0.154	0.102	0.946

Table 2. **Quantitative comparisons on the Tanks & Templates [Knapitsch et al. 2017] and Deep Blending [Hedman et al. 2018].** All methods are evaluated across four focal lengths (1×, 2×, 4×, and 8×), with evaluations at higher sampling rates simulating zoom-in effects.

	PSNR ↑					SSIM ↑					LPIPS ↓					
	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	1× Foc.	2× Foc.	4× Foc.	8× Foc.	Avg.	Spectral Entropy↑
3DGS [Kerbl et al. 2023]	26.47	27.48	27.34	27.43	27.18	0.879	0.881	0.882	0.879	0.880	0.193	0.217	0.245	0.228	0.221	0.289
Mip-Splatting [Yu et al. 2024]	26.49	27.40	28.09	28.08	27.52	0.880	0.893	0.903	0.913	0.897	0.183	0.201	0.211	0.195	0.198	0.422
Pixel-GS [Zhang et al. 2024a]	26.82	27.18	27.00	27.32	27.08	0.879	0.880	0.880	0.896	0.884	0.182	0.211	0.246	0.232	0.218	0.301
Analytic-Splatting [Liang et al. 2024]	26.49	27.76	28.02	27.82	27.52	0.876	0.891	0.899	0.904	0.893	0.188	0.203	0.228	0.210	0.207	0.335
Analytic.+3D Filter [Liang et al. 2024]	26.75	27.92	28.21	28.01	27.72	0.881	0.894	0.903	0.910	0.897	0.192	0.203	0.215	0.200	0.203	0.380
Ours	26.75	27.97	28.49	28.53	27.94	0.881	0.907	0.910	0.931	0.907	0.172	0.180	0.192	0.164	0.177	0.815

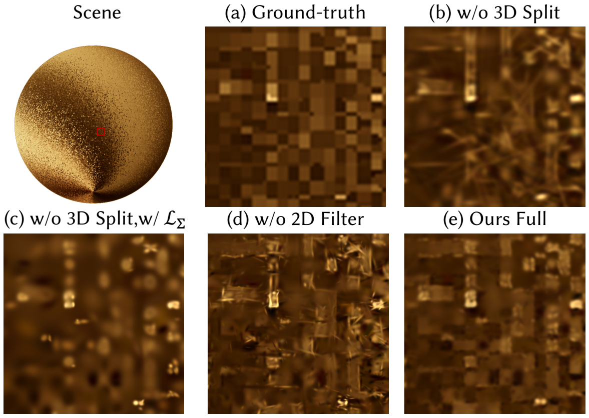

Fig. 6. **Qualitative ablation studies**. From **(a)** to **(e)**: ground truth **(a)**, our method without 3D split **(b)**, our method without 3D split but with $\mathcal{L}_\Sigma$ **(c)**, our method without 2D filter **(d)**, and the full version of our method **(e)**.

Table 3. **Quantitative ablation studies.**

3D Split	$\mathcal{L}_\Sigma$	2D Filter	PSNR↑	SSIM↑	LPIPS↓	Spectral Entropy↑
✗	✗	✓	<u>28.98</u>	<u>0.877</u>	0.259	0.444
✗	✓	✓	28.26	0.860	0.313	<u>0.817</u>
✓	✗	✗	28.45	0.865	<u>0.233</u>	**0.997**
✓	✗	✓	**29.58**	**0.894**	**0.197**	**0.997**

through our 3D splitting **(b)(e)**, while the artifacts caused by rendering are resolved by maintaining condition number consistency via our 2D filtering **(d)(e)**. Moreover, 3D splitting, which does not rely on loss gradients, enhances the representation of high-frequency details in the scene. Recall that 3DGS [Kerbl et al. 2023] is sensitive to loss and lacks shape-awareness. Another idea is to introduce a regularization term $\mathcal{L}_\Sigma$ into the loss function that accounts for 3D

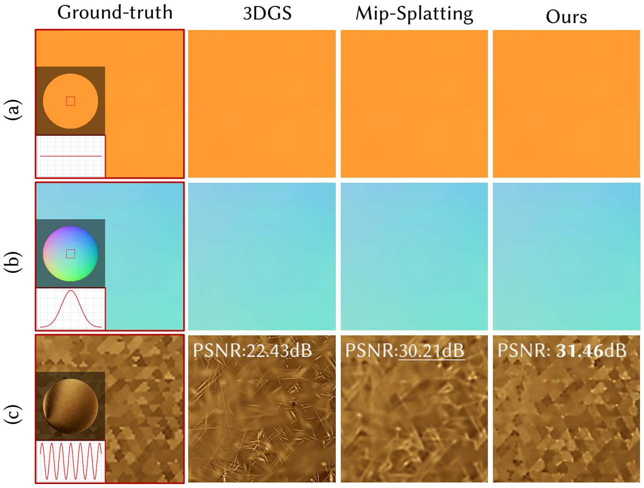

Fig. 7. **We conduct experiments on Ball with identical geometry but different textures.** From top to bottom, they are the monochrome textured **(a)**, multicolor textured **(b)**, and high-frequency textured **(c)**. The red curves intuitively represent the corresponding frequency characteristics.

Gaussians' shape:

$$\mathcal{L}_{\text{naive}} = (1 - \lambda_1)\mathcal{L}_1 + \lambda_1 \mathcal{L}_{\text{D-SSIM}} + \lambda_2 \mathcal{L}_\Sigma. \quad (18)$$

However, although this regularization term is able to effectively constrain and optimize the shape of Gaussians ($\mathrm{H}(\Sigma) = 0.817 > 0.444$), it does not affect densification since the gradient with respect to $\boldsymbol{\mu}_{\text{proj}}$ is zero:

$$\|\nabla_{\boldsymbol{\mu}_{\text{proj}}} \mathcal{L}_\Sigma\|_F = 0, \quad \|\nabla_\Sigma \mathcal{L}_\Sigma\|_F \neq 0, \quad (19)$$

$$\|\nabla_{\boldsymbol{\mu}_{\text{proj}}} \mathcal{L}_{\text{naive}}\|_F = \|\nabla_{\boldsymbol{\mu}_{\text{proj}}} \mathcal{L}\|_F, \quad (20)$$

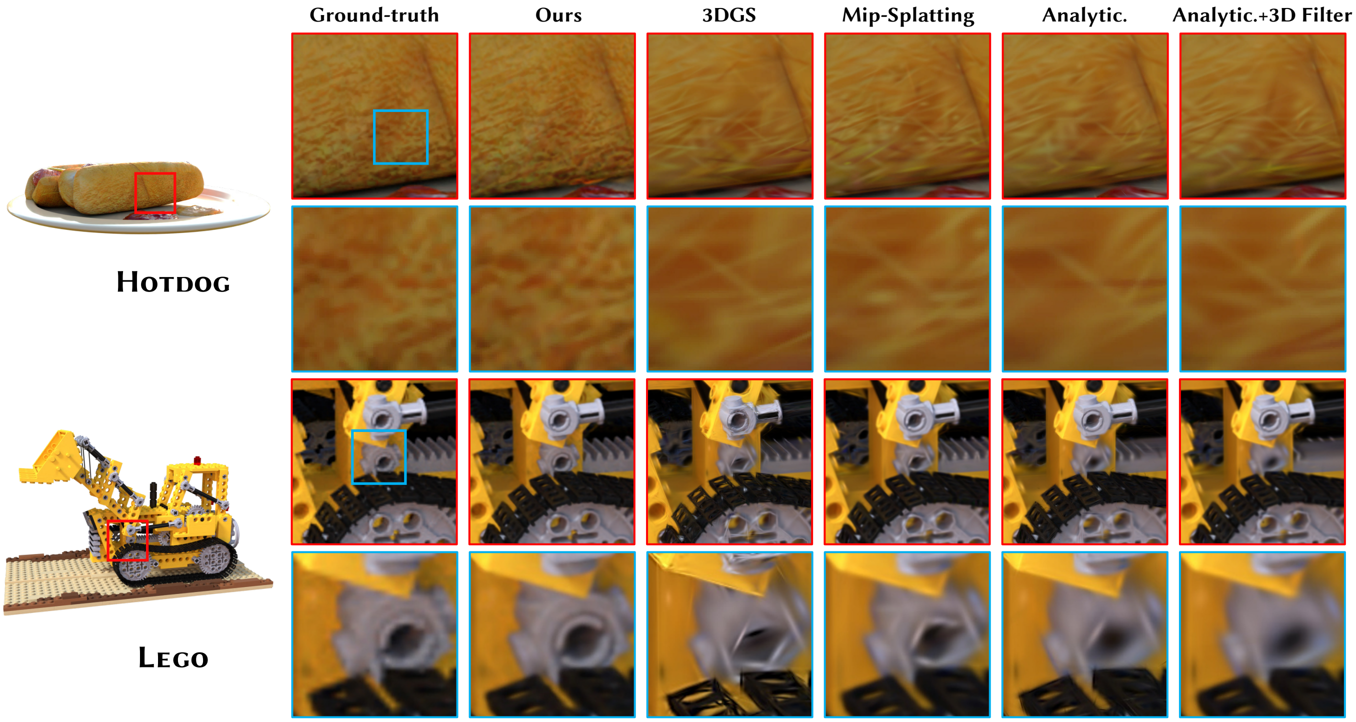

Fig. 8. **Qualitative comparisons on the synthetic scenes [Mildenhall et al. 2020].** Differences in quality highlighted by insets.

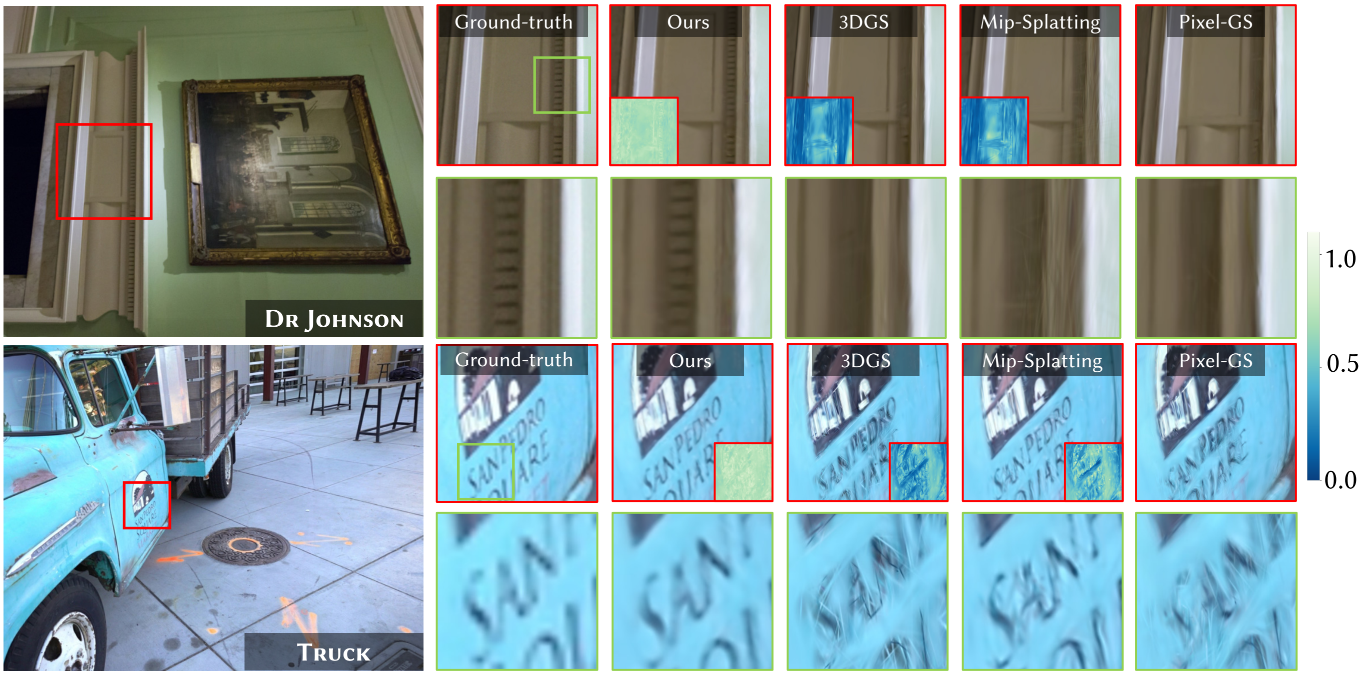

Fig. 9. **Qualitative comparisons on the real scenes [Hedman et al. 2018; Knapitsch et al. 2017].** We visualize the spectral entropy maps of 3D Gaussians. Bluer regions indicate lower spectral entropy, with more needle-like degraded Gaussians, while greener regions represent higher spectral entropy.

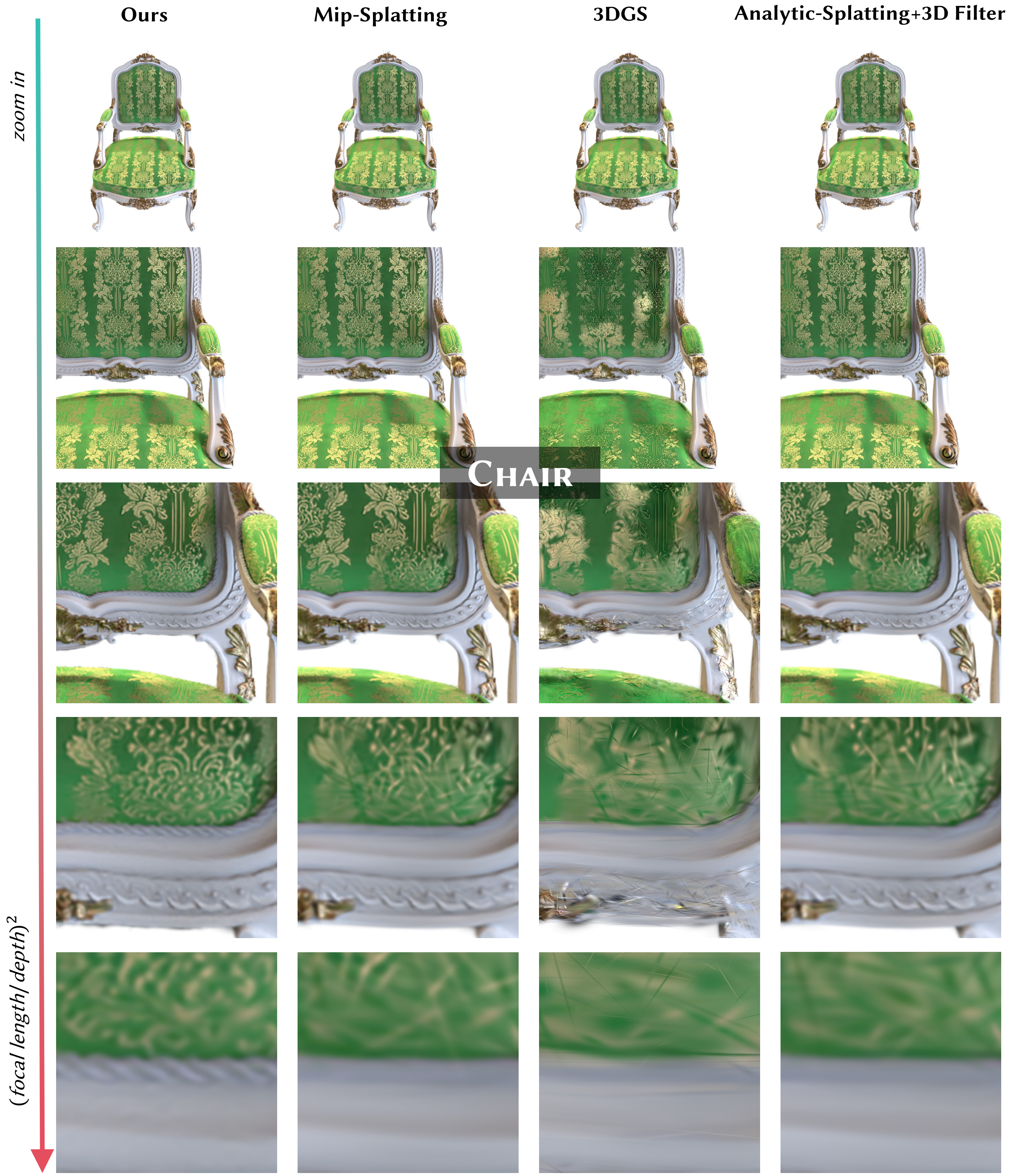

Fig. 10. **We present results from different methods at various sampling rates (focal lengths) on Chair.** The images are ordered from top to bottom, corresponding to the transition from the training focal length to larger focal lengths (zoom in).

Table 4. **Training time and FPS comparisons.** We report the corresponding image quality, spectral entropy (i.e., Ent) and efficiency metrics. Synthetic scenes are from the Blender Dataset [Mildenhall et al. 2020], and real scenes from the Tanks & Templates [Knapitsch et al. 2017] and Deep Blending [Hedman et al. 2018]. For fairness, we also report results with training time and the number of Gaussians equal to ours, marked with *.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	Ent↑	Train	FPS
Synthetic Scenes	3DGS	26.72	0.872	0.178	0.164	9m24s	292
	3DGS*	26.76	0.881	0.181	0.162	-	239
	Mip-Splatting	30.25	0.932	0.114	0.363	9m58s	249
	Mip-Splatting*	30.22	0.932	0.113	0.369	-	230
	Ours	30.69	0.940	0.102	0.946	10m09s	236
Real Scenes	3DGS	27.18	0.880	0.221	0.289	31m27s	116
	3DGS*	27.29	0.881	0.213	0.292	-	97
	Mip-Splatting	27.52	0.897	0.198	0.422	32m44s	112
	Mip-Splatting*	27.48	0.900	0.201	0.433	-	99
	Ours	27.94	0.907	0.177	0.815	32m58s	99

which can lead to loss of high-frequency details **(c)(e)**. Thus, the related work with the regularization term [Hyung et al. 2024] also tends to loss details compared to our full method.

Relationship with frequency. It is worth noting that Mip-Splatting [Yu et al. 2024] outperforms 3DGS [Kerbl et al. 2023] in both spectral entropy and image quality metrics, suggesting a relationship between the 3D smoothing filter based on frequency analysis and ours spectral analysis. Firstly, it is straightforward to prove that the 3D smoothing filter in Mip-Splatting increases the spectral entropy of the original Gaussian. Secondly, we know that Gaussian functions are closed under the Fourier transform (FT) [Nussbaumer and Nussbaumer 1982], while the covariance in the frequency domain is the inverse in the spatial domain (scaled by a coefficient):

$$\mathcal{F}\left\{\mathcal{G}\left(\mathbf{x}\right)\right\} = e^{-2\pi^2 \boldsymbol{\omega}^\top \Sigma \boldsymbol{\omega}} \tag{21}$$

where $\mathcal{F}$ denotes the Fourier transform. The eigenvalues of the covariance matrix determine the bandwidth of the frequency spectrum. Larger eigenvalues result in a narrower spectrum in the corresponding direction, while smaller eigenvalues lead to a wider spectrum in that direction. Therefore, the spectral analysis of our covariance matrix is equivalent and unified with the frequency spectrum analysis of the Gaussian. We conduct experiments with fixed geometry on Ball with different textures to demonstrate that needle-like artifacts predominantly occur in high-frequency scenes, as shown in Fig. 7. And our method effectively addresses the challenge of representing high-frequency details with Gaussians without artifacts.

Impacts of zooming in. To further validate the robustness of our method when zooming in, we conduct a series of experiments. As shown in Fig. 10, these methods produce nearly identical novel view synthesis results at the training view's focal length. However, when the focal length and sampling rate increase, 3DGS suffers from severe needle-like artifacts, leading to a notable decline in rendering quality. While Mip-Splatting and Analytic-Splatting show less degradation due to their use of a series of filters, the rendering results are blurred. In contrast, we maintain high quality similar to lower sampling rates, without needle-like Gaussians or loss of high-frequency details.

5 Conclusion

We propose Spectral-GS, a modification to 3DGS, which introduces *3D scale-aware splitting* and *2D view-consistent filtering* strategies, based on our spectral analysis, to achieve needle-like-alias-free rendering at arbitrary close-up or zoomed-in view. Our splitting strategy effectively regularizes needle-like Gaussians and increase the spectral entropy, enhancing the high-frequency details representation for 3DGS and mitigating needle-like artifacts. And the 2D view-consistent filter combines a convolution that approximates supersampling with a Gaussian blur that approximates interpolation to resolve needle-like artifacts caused by view-inconsistency.

Limitation. Since our method does not introduce additional priors, such as image super-resolution networks [Dong et al. 2015], the resolution of novel views depends on the input images.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. This work was supported by the National Natural Science Foundation of China (No. 61972194 and No. 62032011), the Natural Science Foundation of Jiangsu Province (No. BK20211147) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. KYCX25_0318).

References

Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 2025. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology* (2025).

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 5835–5844.

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.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19697–19705.

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

Guikun Chen and Wenguan Wang. 2024. A Survey on 3D Gaussian Splatting. *arXiv preprint arXiv:2401.03890* (2024).

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38, 2 (2015), 295–307.

Guangchi Fang and Bing Wang. 2024. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*. Springer, 165–181.

Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 2024. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics* (2024).

Qi-Yuan Feng, Geng-Chen Cao, Hao-Xiang Chen, Qun-Ce Xu, Tai-Jiang Mu, Ralph Martin, and Shi-Min Hu. 2024. EVSplitting: an efficient and visually consistent splitting algorithm for 3D Gaussian Splatting. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Florian Hahlbohm, Fabian Friederichs, Tim Weyrich, Linus Franke, Moritz Kappel, Susana Castillo, Marc Stamminger, Martin Eisemann, and Marcus Magnor. 2025. Efficient Perspective-Correct 3D Gaussian Splatting Using Hybrid Transparency. In *Computer Graphics Forum*. Wiley Online Library, e70014.

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.

Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. 2023. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19774–19783.

Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. 2024. 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*.

Junchen Liu, Wenbo Hu, Zhuo Yang, Jianteng Chen, Guoliang Wang, Xiaoxue Chen, Yantong Cai, Huan-ang Gao, and Hao Zhao. 2024. Rip-nerf: Anti-aliasing radiance fields with ripmap-encoded platonic solids. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

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*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

Henri J Nussbaumer and Henri J Nussbaumer. 1982. *The fast Fourier transform*. Springer.

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.

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

Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. 2024. Sa-gs: Scale-adaptive gaussian splatting for training-free anti-aliasing. *arXiv preprint arXiv:2403.19615* (2024).

Xuechang Tu, Bernhard Kerbl, and Fernando de la Torre. 2024. Fast and Robust 3D Gaussian Splatting for Virtual Reality. In *SIGGRAPH Asia 2024 Posters*. 1–3.

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).

Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. 2024. Recent advances in 3d gaussian splatting. *Computational Visual Media* 10, 4 (2024), 613–642.

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.

Keyang Ye, Tianjia Shao, and Kun Zhou. 2025. When Gaussian Meets Surfel: Ultra-fast High-fidelity Radiance Field Rendering. *ACM Transactions on Graphics (TOG)* 44, 4 (2025).

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).

Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. 2024b. Fregs: 3d gaussian splatting with progressive frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21424–21433.

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. 2024a. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European Conference on Computer Vision*.

Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. 2002. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics* 8, 3 (2002), 223–238.