# SmartSplat: Feature-Smart Gaussians for Scalable Compression of Ultra-High-Resolution Images

**Linfei Li[1], Lin Zhang[1*], Zhong Wang[2], Ying Shen[1]**

[1]School of Computer Science and Technology, Tongji University
[2] Department of Automation, Shanghai Jiao Tong University
cslinfeili@tongji.edu.cn, cslinzhang@tongji.edu.cn, cszhongwang@sjtu.edu.cn, yingshen@tongji.edu.cn

## Abstract

Recent advances in generative AI have accelerated the production of ultra-high-resolution visual content. However, traditional image formats face significant limitations in efficient compression and real-time decoding, which restricts their applicability on end-user devices. Inspired by 3D Gaussian Splatting, 2D Gaussian image models have achieved notable progress in enhancing image representation efficiency and quality. Nevertheless, existing methods struggle to balance compression ratios and reconstruction fidelity in ultra-high-resolution scenarios. To address these challenges, we propose SmartSplat, a highly adaptive and feature-aware GS-based image compression framework that effectively supports arbitrary image resolutions and compression ratios. By leveraging image-aware features such as gradients and color variances, SmartSplat introduces a Gradient-Color Guided Variational Sampling strategy alongside an Exclusion-based Uniform Sampling scheme, significantly improving the non-overlapping coverage of Gaussian primitives in pixel space. Additionally, a Scale-Adaptive Gaussian Color Sampling method is proposed to enhance the initialization of Gaussian color attributes across scales. Through joint optimization of spatial layout, scale, and color initialization, SmartSplat can efficiently capture both local structures and global textures of images using a limited number of Gaussians, achieving superior reconstruction quality under high compression ratios. Extensive experiments on DIV8K and a newly created 16K dataset demonstrate that SmartSplat significantly outperforms state-of-the-art methods at comparable compression ratios and surpasses their compression limits, exhibiting strong scalability and practical applicability. This framework can effectively alleviate the storage and transmission burdens of ultra-high-resolution images, providing a robust foundation for future high-efficiency visual content processing.

**Code** — https://github.com/lif314/SmartSplat

## Introduction

With the rapid development of generative artificial intelligence, Ultra-High-Resolution (UHR) visual content has become increasingly accessible and widely disseminated (Zhang et al. 2025; Ren et al. 2024). However, the resulting high-resolution image data poses significant challenges for storage and transmission, necessitating image representations that offer both high compression ratios and efficient decoding. Traditional formats such as PNG (Ryan 2006) and JPEG (Wallace 1992) exhibit notable limitations in this context; for instance, JPEG typically achieves a maximum compression ratio of around 50×, which falls short of meeting the demands for efficient transmission and real-time rendering of ultra-high-resolution imagery.

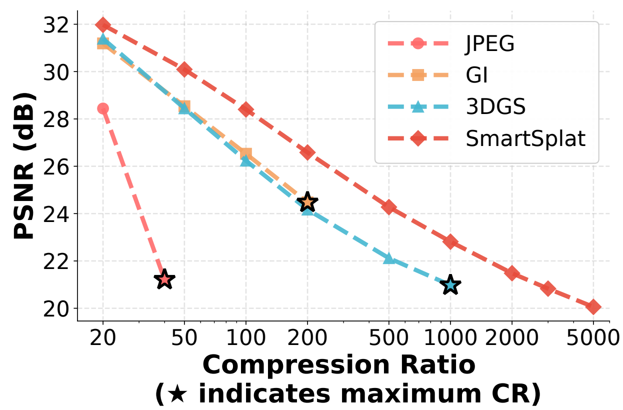


Figure 1: Comparison with baselines on $4218 \times 7350$ image. SmartSplat consistently outperforms baselines under the same Compression Ratio (CR) and surpasses the maximum compression limits achieved by previous approaches, maintaining high-fidelity reconstruction even at extreme compression levels (e.g., 1000×).

Implicit Neural Representations (INRs) have attracted substantial research interest due to their powerful compression capabilities enabled by neural networks. Nevertheless, existing INR-based methods (Sitzmann et al. 2020; Ramasinghe and Lucey 2022; Tancik et al. 2020) generally rely on fixed architectures and full-image training to preserve visual fidelity. These methods require intensive computational resources for UHR images, limiting scalability. Furthermore, the dependency on neural inference leads to slow decoding, making them less suitable for real-time applications that demand both rapid decoding and dynamic quality adjustment.

Concurrently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has recently emerged as a novel scene representa-

*Corresponding Author

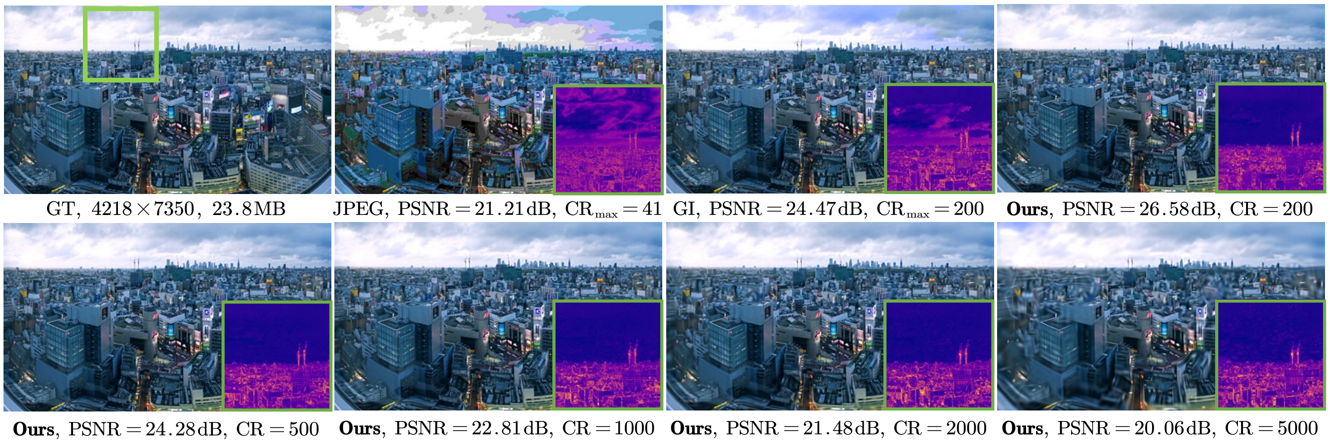

Figure 2: SmartSplat maintains visual quality under extreme high compression ratios. Under maximum compression ratio ($\mathrm{CR}_{\mathrm{max}}$), JPEG shows severe artifacts, and GI struggles with scalability. In contrast, SmartSplat outperforms GI at the same compression ratio and rivals JPEG even at $2000\times$, maintaining visually pleasing results up to $5000\times$. The insets visualize the corresponding error images, with brighter colors indicating higher errors.

tion technique. By explicitly modeling 3D Gaussian primitives and incorporating a differentiable tile-based rasterization pipeline, it achieves a compelling balance between rendering quality and real-time performance. Inspired by this paradigm, several studies (Zhang et al. 2024a,b; Zhu et al. 2025) have extended 3DGS to 2D image representation, proposing spatially-aware 2D Gaussian modeling and rendering frameworks that substantially improve training and decoding efficiency. However, these methods typically rely on a large number of Gaussian primitives to ensure reconstruction accuracy or only achieve limited compression ratios on low-resolution images (typically below 2K), thus falling short of the efficiency requirements for ultra-high-resolution image compression in practical applications.

Accordingly, our research aims to develop a high-compression-ratio representation framework tailored for ultra-high-resolution images at 8K, 16K, and beyond. Such images typically reach sizes ranging from tens to hundreds of megabytes, posing significant challenges for storage, transmission, and sharing. Therefore, there is an urgent need for a compact and efficient representation method that balances compression efficiency with reconstruction quality.

To fill this gap, we propose SmartSplat, a feature-driven 2D Gaussian image compression framework. We begin by analyzing the relationship between compression ratios and the density of Gaussians, highlighting that high compression ratios inherently constrain the number of Gaussians, thereby increasing the difficulty of faithful image reconstruction.

To mitigate this challenge, SmartSplat introduces a highly adaptive Gaussian distribution strategy guided by image features. Specifically, it introduces Gradient-Color Guided Variational Sampling and Exclusion-based Uniform Sampling to jointly optimize the means and scales of Gaussians, while a Scale-Adaptive Color Initialization scheme is proposed to enhance the expressiveness of limited Gaussian primitives in capturing both local structures and global textures. This design enables high-quality reconstruction under strict compression budgets, making it well-suited for practical applications in high-resolution scenarios.

Furthermore, to evaluate the performance of SmartSplat on ultra-high-resolution images, we construct a 16K image dataset, termed DIV16K, by leveraging the Aiarty Image Enhancer tool. As shown in Figures 1 and 2, extensive experiments conducted on both 8K and 16K images reveal that SmartSplat not only outperforms state-of-the-art methods under the same compression ratios but also surpasses their compression limits, achieving competitive reconstruction quality at significantly higher compression levels.

In summary, our main contributions are as follows:

- A unified analysis between UHR image compression ratios and GS-based representation, emphasizing the principal challenges involved.
- Development of an adaptive Gaussian sampling strategy that jointly optimizes means, scales, and colors to enable compact and efficient UHR image representations.
- Extensive experiments on DIV8K and our newly constructed DIV16K dataset demonstrate that SmartSplat achieves superior image representation quality under high compression ratios, significantly outperforming existing GS-based methods.

## Related Work

**Implicit Neural Representation.** In recent years, Implicit Neural Representations (INRs) have demonstrated significant potential in the domains of image modeling and compression. Early approaches (Sitzmann et al. 2020; Tancik et al. 2020; Ramasinghe and Lucey 2022; Fathony et al. 2021; Saragadam et al. 2023; Li et al. 2025) typically employ multilayer perceptrons (MLPs) to directly regress pixel values, leveraging positional encoding to enhance representational capacity. However, these methods often suffer from slow training and high inference costs, particularly when dealing with high-resolution images. To address these limitations, subsequent studies introduce spatial priors through structured feature grids, such as hierarchical grids (Chen et al. 2023; Martel et al. 2021; Takikawa et al. 2021) and hash-based encodings (Müller et al. 2022), which alleviate the burden on MLPs and substantially accelerate training while maintaining reconstruction quality. Nevertheless, these techniques remain memory-intensive and struggle to adapt to fine-grained and spatially varying image details.

**GS-based Image Representation.** 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has emerged as a promising paradigm for view synthesis (Yu et al. 2024; Huang et al. 2024; Lee et al. 2024) and reconstruction (Li et al. 2024; Matsuki et al. 2024), offering exceptional controllability and real-time rendering capabilities through its differentiable tile-based rasterization mechanism and explicit 3D Gaussian representations. Building on this foundation, GaussianImage (Zhang et al. 2024a) extends the principles of 3DGS to the 2D image domain by adapting Gaussian primitives to planar image space for image fitting and compression. While the method achieves satisfactory visual quality, its reliance on a two-stage optimization pipeline and computationally expensive vector quantization (Bhalgat et al. 2020) introduces significant efficiency bottlenecks. Further advancing this direction, the LIG (Zhu et al. 2025) framework employs a hierarchical Gaussian fitting strategy for high-resolution image reconstruction. However, it prioritizes fitting accuracy over compression performance and requires a large number of Gaussian components. Furthermore, ImageGS (Zhang et al. 2024b) introduces a content-aware initialization scheme along with a progressive training strategy to enhance optimization efficiency. Nevertheless, it remains suboptimal in extreme-rate image compression scenarios, particularly for ultra-high-resolution images.

# Methodology

## Preliminaries: Gaussian Image Splatting

3DGS (Kerbl et al. 2023) represents a 3D scene using a set of anisotropic Gaussian distributions in 3D space. Each Gaussian is parameterized by its mean position, scale, rotation, opacity, and color. During rendering, these Gaussians are projected onto the image plane through a tile-based rasterization pipeline, resulting in 2D elliptical splats. Then, a front-to-back $\alpha$-blending operation is applied at the pixel level to synthesize novel views.

Extending the 3DGS paradigm to the 2D domain allows for image representation using 2D Gaussian primitives. Specifically, each 2D Gaussian is defined by a mean vector $\boldsymbol{\mu} \in \mathbb{R}^2$, a 2D covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$, a color vector $\mathbf{c} \in \mathbb{R}^3$, and an opacity value $o \in \mathbb{R}$. The contribution of a Gaussian at a given pixel location $\mathbf{x}$ is given by:

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \quad (1)$$

where the covariance matrix $\boldsymbol{\Sigma}$ must be positive semi-definite. Direct optimization of $\boldsymbol{\Sigma}$ via gradient descent, as in LIG (Zhu et al. 2025), does not guarantee this property. GaussianImage (Zhang et al. 2024a) adopts a Cholesky decomposition (Higham 2009) approach, where the covariance matrix $\boldsymbol{\Sigma}$ is factorized as the product of a lower triangular matrix $\mathbf{L} \in \mathbb{R}^{2\times 2}$ and its transpose:

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T. \quad (2)$$

Alternatively, inspired by 3DGS, the covariance matrix can also be expressed as the product of a rotation matrix $\mathbf{R} \in \mathbb{R}^{2\times 2}$ and a scaling matrix $\mathbf{S} \in \mathbb{R}^{2\times 2}$:

$$\boldsymbol{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T. \quad (3)$$

Furthermore, since rendering on the image plane does not require depth sorting of Gaussian primitives, the color at each pixel $\hat{\mathbf{c}}(\mathbf{x})$ can be computed via a forward-style $\alpha$-blending over the $\mathcal{N}$ overlapping Gaussians:

$$\hat{\mathbf{c}}(\mathbf{x}) = \sum_{i\in\mathcal{N}} \mathbf{c}_i \cdot o_i \cdot G_i(\mathbf{x}) \prod_{j=1}^{i-1} (1 - o_j G_j(\mathbf{x})), \quad (4)$$

where $o_i$ denotes the opacity, $\mathbf{c}_i$ is the color coefficient, and $G_i(\mathbf{x})$ represents the value of the $i$-th 2D Gaussian evaluated at location $\mathbf{x}$. This formulation accumulates contributions from all overlapping Gaussians in a fixed order without requiring explicit visibility reasoning.

## Feature-Smart Gaussians

**Problem Formulation.** Assume an original $H \times W$ RGB image is encoded with 8 bits per channel (excluding transparency), resulting in an uncompressed size of $3HW$ bytes. Since pixel rendering is independent of the ordering of Gaussians, we assume a constant opacity of 1. Under this assumption, each Gaussian primitive requires seven 32-bit floating-point parameters—representing position, scale, rotation, and color—amounting to approximately 32 bytes per primitive. With vector quantization-based compression techniques (Bhalgat et al. 2020; Zhang et al. 2024a), the storage per primitive can be reduced to 7 bytes. Given a target compression ratio CR, the maximum number of Gaussian primitives $N_g$ allowed is determined by:

$$N_g = \frac{3HW}{7 \cdot \mathrm{CR}}. \quad (5)$$

**Gaussian Image Representation Decomposition.** To gain a deeper understanding of the relationship between Gaussian distributions and the image space, we adopt a physically interpretable covariance decomposition as defined in Eq. 3. Accordingly, for each Gaussian element representing the image, we normalize its color component $\mathbf{c}$ to the range $[0, 1]$, and fix its opacity $o$ to 1. The associated rotation matrix $\mathbf{R}$ can be parameterized as:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (6)$$

where $\theta \in [0, 2\pi)$ denotes the rotation angle. During initialization, $\theta$ is sampled from the interval $[0, 1)$ and scaled by $2\pi$ to span the full angular range. The scale matrix of the Gaussian is defined as a symmetric matrix:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}, \quad (7)$$

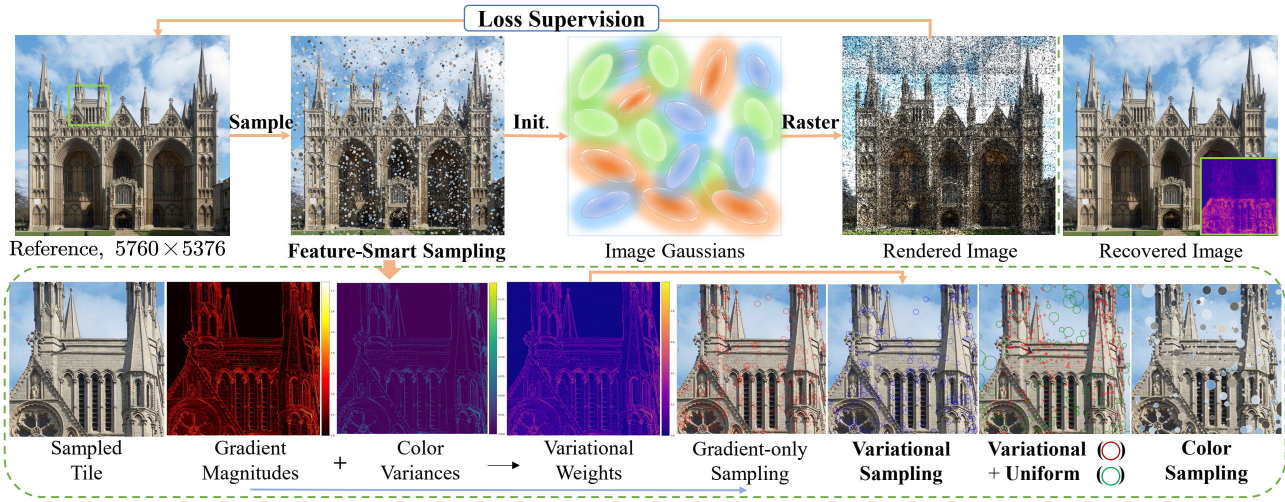


Figure 3: Pipeline of SmartSplat. Given an input image, SmartSplat initializes Gaussian primitives via feature-aware sampling and optimizes them through differentiable rasterization to learn compact, perceptually-aware representations.

where $s_x$ and $s_y$ denote the scales along the $x$- and $y$-axes, respectively. Following the $3\sigma$ rule, the maximum influence radius $R_{g\to p}$ of a Gaussian on neighboring pixels during rasterization can be approximated (ignoring rotation) by:

$$R_{g\to p} \approx 3 \cdot \max(s_x, s_y). \tag{8}$$

Based on the aforementioned decomposition analysis and guided by Eq. 5, it can be seen that the central challenge of representing an image using limited Gaussians under a given compression ratio lies in the effective design of their spatial distribution. This desired design must adaptively capture both high- and low-frequency structures within the image. To achieve this, we propose a feature-guided joint sampling strategy that simultaneously considers the position, scale, and color attributes of Gaussians, enabling efficient representation of images at arbitrary resolutions and compression ratios. Specifically, as shown in Fig. 3, image features such as gradients and color variations are leveraged through a Gradient-Color Guided Variational Sampling strategy and an Exclusion-based Uniform Sampling scheme, which collectively guide the initialization of the means and scales of Gaussians to ensure non-overlapping and content-aware spatial coverage. In addition, a Scale-Adaptive Gaussian Color Sampling strategy is introduced to initialize the color attributes of Gaussians across scales, further enhancing the fidelity of representation. This unified design allows SmartSplat to capture both fine-grained local structures and coarse global patterns using a compact set of feature-aware Gaussians, thereby achieving high-quality image reconstruction under high compression ratios.

**Gradient-Color Guided Variational Sampling.** Intuitively, high-frequency regions in an image should be represented using densely distributed Gaussians with smaller scales, while low-frequency regions are more appropriately modeled with sparsely distributed Gaussians of larger scales. A straightforward approach is to select Gaussian positions based solely on image gradients (Zhang et al. 2024b). However, since gradients primarily emphasize structural information, this may result in inadequate representation of low-frequency regions when the number of Gaussian primitives is limited. To address this issue, we propose a variational sampling strategy that jointly leverages both image gradients and color variance. Gradients are employed to guide denser sampling in structure-rich areas, while color variance is used to identify regions with high chromatic complexity. Such a joint strategy enables adaptive sampling across different frequency components of the image.

To efficiently process high-resolution images while ensuring uniform coverage during Gaussian initialization, we propose an adaptive step-size block-wise variational sampling strategy. This approach partitions the large-scale image into multiple overlapping or adjacent tiles, within which variational sampling is conducted independently. The adaptive step-size mechanism further guarantees uniform coverage across the entire image.

Specifically, within each tile sub-image $\mathbf{I}_{i,j}$, the local gradient magnitude and color variance of its pixels are computed as follows:

$$\begin{aligned} m_{i,j}(\mathbf{x}) &= \frac{1}{C}\sum_{c=1}^{C} \left\|\nabla \mathbf{I}_{i,j,c}(\mathbf{x})\right\|_2, \\ v_{i,j}(\mathbf{x}) &= \frac{1}{C}\sum_{c=1}^{C} \mathrm{Var}\big(\mathbf{I}_{i,j,c}(\mathcal{N}_{\mathbf{x}})\big), \end{aligned} \tag{9}$$

where $C$ denotes the number of channels, and $\mathcal{N}_{\mathbf{x}}$ represents the neighborhood of pixel $\mathbf{x}$. To eliminate scale discrepancies, the gradient magnitude and color variance are normalized within the tile, yielding $\tilde{m}_{i,j}(\mathbf{x})$ and $\tilde{v}_{i,j}(\mathbf{x})$, respectively. The sampling weight is then defined as a weighted combination of these normalized values:

$$w_{i,j}(\mathbf{x}) = \lambda_m \cdot \tilde{m}_{i,j}(\mathbf{x}) + (1-\lambda_m) \cdot \tilde{v}_{i,j}(\mathbf{x}), \tag{10}$$

where $\lambda_m$ denotes the weighting coefficient that balances the contributions of gradient magnitude and color variance.

Based on these sampling weights, the sampling probability of pixel $\mathbf{x}$ within tile $(i, j)$ is given by:

$$\mathbb{P}_{i,j}(\mathbf{x}) = \frac{w_{i,j}(\mathbf{x})}{\sum_{\mathbf{y} \in \mathbf{I}_{i,j}} w_{i,j}(\mathbf{y})}. \tag{11}$$

Finally, multinomial sampling is performed according to this probability distribution to select $n_{i,j}$ points within the tile:

$$\{\mathbf{x}_k^{(i,j)}\}_{k=1}^{n_{i,j}} \sim \text{Multinomial}\left(n_{i,j}, \{\mathbb{P}_{i,j}(\mathbf{x})\}_{\mathbf{x} \in \mathbf{I}_{i,j}}\right). \tag{12}$$

The proposed variational sampling strategy effectively increases the density of samples in regions exhibiting prominent gradients or significant color variation, thereby facilitating a more appropriate initialization of Gaussian primitives.

Evidently, points with higher sampling weights should be assigned smaller scales, while those with lower weights can be allocated larger scales. To ensure spatial smoothness, we adopt an exponential decay function to adaptively compute the scale. Assuming the initial scales along the $x$- and $y$-axes are equal, the scale is given by:

$$s_{i,j}(\mathbf{x}) = s_{base} \cdot \exp(-\frac{1}{2} w_{i,j}(\mathbf{x})). \tag{13}$$

To maximize the coverage of the pixel space by Gaussians, we consider the influence radius of a Gaussian. Assuming uniform coverage using circles with radius $R_{g \to p}$, the maximum non-overlapping Gaussian scale can be derived as:

$$s_{base} = \frac{1}{3} R_{g \to p} = \frac{1}{3} \sqrt{\frac{HW}{\pi N_g}}. \tag{14}$$

This scale initialization strategy adaptively represents images of arbitrary resolutions without relying on any hyperparameters or heuristic clamping.

**Exclusion-based Uniform Sampling.** After variational sampling, an exclusion-based uniform sampling strategy is proposed to ensure adequate coverage of regions with low structural complexity or minimal color variation.

Specifically, let the set of variationally sampled points be denoted by $\mathcal{X}_{vs} = \{\mathbf{x}_i^{vs}\}_{i=1}^{N_g^{vs}}$, where $N_g^{vs}$ represents the number of points obtained through variational sampling. During the subsequent uniform sampling stage, the sampled point set $\mathcal{X}_{us} = \{\mathbf{x}_j^{us}\}_{j=1}^{N_g^{us}}$ must satisfy the following exclusion constraint to ensure spatial separation from the previously selected points:

$$\forall j, \quad \min_i \left\|\mathbf{x}_j^{us} - \mathbf{x}_i^{vs}\right\| \geq r_{excl}, \tag{15}$$

where $r_{excl}$ denotes the exclusion radius. To prevent excessive overlap between variationally sampled and uniformly sampled points, the exclusion radius is determined by incorporating both the Gaussian influence radius and the scale of variationally sampled points. Concretely, $r_{excl}$ is defined as the maximum of the base scale and the median scale of the variational samples:

$$r_{excl} = \max\left(s_{base}, \text{median}(\{s_i^{vs}\}_{i=1}^{N_g^{vs}})\right). \tag{16}$$

Furthermore, to ensure that Gaussian kernels adequately cover the entire pixel domain of the image, we adopt a Query-to-Reference KNN algorithm to estimate the scale of uniformly sampled points. Specifically, let the complete set of points be denoted by $\mathcal{X} = \mathcal{X}_{vs} \cup \mathcal{X}_{us}$, where $\mathcal{X}_{vs}$ and $\mathcal{X}_{us}$ represent the variationally sampled and uniformly sampled point sets, respectively. For each uniform sample $\mathbf{x}_j^{us} \in \mathcal{X}_{us}$, a $K$-nearest neighbor search is performed within the complete set $\mathcal{X}$ to determine its local scale. The scale is defined as:

$$s_j^{us} = \sqrt{\frac{1}{K} \sum_{\mathbf{q} \in \mathcal{N}_K(\mathbf{x}_j^{us}, \mathcal{X})} \left\|\mathbf{x}_j^{us} - \mathbf{q}\right\|^2}, \tag{17}$$

where $\mathcal{N}_K(\mathbf{x}_j^{us}, \mathcal{X})$ denotes the set of $K$ nearest neighbors of $\mathbf{x}_j^{us}$ within $\mathcal{X}$. The resulting scale $s_j^{us}$ reflects the local sampling density around the point. This approach adaptively enhances coverage in sparse regions, thereby improving the overall robustness and representational fidelity of the sampling distribution across the image domain.

**Scale-Adaptive Gaussian Color Sampling.** Following the initialization of 2D Gaussian positions and scales via variational and uniform sampling, we introduce a scale-adaptive Gaussian-weighted median sampling strategy to estimate the color parameters of each Gaussian element. This approach aims to enhance structural fidelity and improve robustness to local noise and outliers. Unlike traditional methods based on random initialization (Zhang et al. 2024a; Zhu et al. 2025) or pixel-center estimation (Zhang et al. 2024b), the proposed strategy effectively combines the robustness of median estimation with the spatial sensitivity of Gaussian weighting. This enables more accurate color recovery in regions with high-frequency textures or abrupt depth changes, thereby improving reconstruction quality and perceptual consistency while accelerating convergence.

Specifically, for each sampled point $\mathbf{x}_i \in \mathcal{X} = \mathcal{X}_{vs} \cup \mathcal{X}_{us}$, obtained from either variational sampling or uniform sampling, and associated with a scale parameter $s_i$, we define a circular neighborhood $\mathcal{N}_{\mathbf{x}_i}$ centered at $\mathbf{x}_i$ with radius $s_i$:

$$\mathcal{N}_{\mathbf{x}_i} = \left\{\mathbf{u} \in \mathbb{Z}^2 \,\middle|\, \|\mathbf{u} - \mathbf{x}_i\|_2 \leq s_i\right\}. \tag{18}$$

For each pixel $\mathbf{u} \in \mathcal{N}_{\mathbf{x}_i}$, a spatial weight is assigned based on a 2D isotropic Gaussian kernel centered at $\mathbf{x}_i$:

$$w_i(\mathbf{u}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{x}_i\|^2}{2\sigma_i^2}\right), \quad \text{where} \quad \sigma_i = s_i. \tag{19}$$

Then, the RGB color $\mathbf{c}_i \in \mathbb{R}^3$ corresponding to point $\mathbf{x}_i$ is estimated via a Gaussian-weighted median over the pixel intensities $\mathbf{I}(\mathbf{u})$ within its neighborhood. Specifically, for each color channel $d \in \{1, 2, 3\}$, the channel value is determined by solving the following minimization:

$$\mathbf{c}_i^{(d)} = \arg\min_{z \in \mathbb{R}} \sum_{\mathbf{u} \in \mathcal{N}_{\mathbf{x}_i}} w_i(\mathbf{u}) \cdot \left|z - \mathbf{I}^{(d)}(\mathbf{u})\right|. \tag{20}$$

This scale-adaptive color sampling strategy leverages the spatial coherence inherent in Gaussian sampling while incorporating the robustness of median filtering to reduce sensitivity to outlier scales. As a result, it enables accurate color estimation across multiple scales, even in regions affected by noise or exhibiting significant local variations.

| Dataset | CR | *3DGS* | *LIG* | *GI (RS)* | *GI (Cholesky)* | *ImageGS* | *SmartSplat (Ours)* |
|---|---|---|---|---|---|---|---|
| DIV8K | 20 | 30.99 / 0.9636 | 28.05 / 0.9362 | 30.45 / 0.9707 | 30.33 / 0.9698 | 32.00 / 0.8680 | **33.26 / 0.9752** |
| | 50 | 28.56 / 0.9340 | 24.90 / 0.8402 | 26.99 / 0.9291 | 26.87 / 0.9271 | 29.47 / 0.8052 | **29.65 / 0.9482** |
| | 100 | 26.84 / 0.8990 | 22.91 / 0.7230 | 25.00 / 0.8827 | 24.90 / 0.8790 | 26.65 / 0.7449 | **27.49 / 0.9164** |
| | 200 | 24.92 / 0.8556 | 21.06 / 0.5792 | 23.45 / 0.8223 | 23.35 / 0.8176 | **26.80** / 0.7181 | 25.75 / **0.8745** |
| | 500 | 22.38 / 0.7874 | 17.68 / 0.3633 | Fail | Fail | **24.88** / 0.6544 | 23.82 / **0.8055** |
| | 1000 | 20.38 / 0.7068 | 12.49 / 0.2083 | Fail | Fail | **23.50** / 0.6165 | 22.66 / **0.7469** |
| DIV16K | 50 | OOM | 24.42 / 0.4561 | 29.24 / 0.7917 | 29.14 / 0.7899 | OOM | **34.34 / 0.9267** |
| | 100 | OOM | 21.37 / 0.3815 | 27.39 / 0.7648 | 27.28 / 0.7623 | OOM | **33.00 / 0.9117** |
| | 200 | OOM | 18.01 / 0.3171 | 25.63 / 0.7394 | 25.51 / 0.7365 | OOM | **31.85 / 0.8897** |
| | 500 | 28.61 / 0.8117 | 11.97 / 0.2015 | Fail | Fail | OOM | **29.40 / 0.8524** |
| | 1000 | 27.06 / 0.7854 | 6.78 / 0.1749 | Fail | Fail | OOM | **27.49 / 0.8226** |
| | 2000 | 25.54 / 0.7642 | Fail | Fail | Fail | OOM | **25.70 / 0.7966** |
| | 3000 | Fail | Fail | Fail | Fail | Fail | **24.72 / 0.7844** |

Table 1: Quantitative results on DIV8K (Avg. Res./Size: 5736 × 6120/53.56MB) and DIV16K (Avg. Res./Size: 12684 × 15898/235.52MB). Each cell reports PSNR / MS-SSIM (DIV8K) or PSNR / SSIM (DIV16K). "OOM" denotes out-of-memory, and "Fail" means training failure due to insufficient Gaussians.

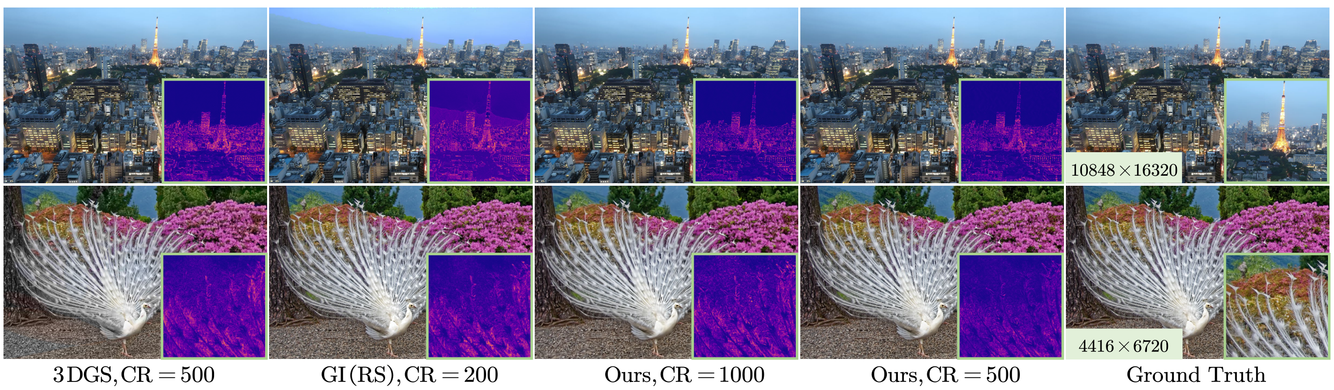


Figure 4: Qualitative comparison on DIV8K and DIV16K. More results are provided in the supplementary material.

**Optimization.** Given an input image of size $H \times W$ and a target compression ratio CR, the maximum allowable number of Gaussian elements, denoted by $N_g$, can be computed based on Eq. 5. Assuming a variational sampling ratio of $\lambda_g$, the numbers of Gaussians allocated to variational sampling and uniform sampling are determined as:

$$N_g^{vs} = \lambda_g N_g, \quad N_g^{us} = (1 - \lambda_g) N_g. \tag{21}$$

Using the aforementioned sampling strategy, a corresponding number of sample points, along with their scales and colors, are used to initialize the Gaussians. Then, the reconstruction process is optimized by minimizing a composite loss that combines the $l_1$ distance and the SSIM (Wang et al. 2004) between the rendered image and the ground-truth image. The overall loss function is defined as:

$$L = \lambda_l \|\hat{\mathbf{I}} - \mathbf{I}\|_1 + (1 - \lambda_l)(1 - \mathrm{SSIM}(\hat{\mathbf{I}}, \mathbf{I})), \tag{22}$$

where $\hat{\mathbf{I}}$ denotes the reconstructed image rendered from the set of Gaussians, $\mathbf{I}$ represents the corresponding ground-truth image, and $\lambda_l \in [0, 1]$ is a weighting factor that balances the contributions of the two loss terms.

# Experiments

## Experimental Setup

**Dataset.** To comprehensively assess the performance of GS-based image compression on ultra-high-resolution content, the DIV8K dataset (Gu et al. 2019) was employed as the primary benchmark. In addition, a new dataset, DIV16K, was constructed by applying 8× upsampling to images from DIV2K (Agustsson and Timofte 2017) using the Aiarty Image Enhancer, thereby simulating high-resolution imagery representative of generative AI outputs. Given the significant computational demands of such data, a subset of 16 images from DIV8K and 8 images from DIV16K was selected for evaluation. All images were stored in lossless PNG format, providing a reliable testbed for examining the trade-off between compression efficiency and perceptual fidelity.

**Implementation.** To handle UHR image initialization efficiently, all sampling was performed in a tile-based manner. A CUDA-based query-to-reference KNN pipeline was introduced for exclusion sampling and scale estimation. On 16K images, the initialization stage can be completed within $2 \sim 5$ seconds. Variational and uniform sampling used $\lambda_m$=0.9 and $K$=3, respectively. During training, $\lambda_g$=0.7

| Method | Iter/s ↑ | TrainTime(s) ↓ | TrainMem.(GB) ↓ | FPS ↑ | PSNR(dB) ↑ | SSIM ↑ |
|---|---|---|---|---|---|---|
| 3DGS (10K) | 1.32 | 7841.80 | 50.19 | 10.98 | 24.42 | 0.8922 |
| GI (10K) | 7.44 | 1334.73 | 16.29 | 62.33 | 19.86 | 0.4825 |
| SmartSplat (10K) | 5.01 | 2237.52 | 19.59 | 32.35 | 31.87 | 0.9354 |
| SmartSplat (1K) | 5.03 | 336.12 | 19.38 | 33.12 | 30.52 | 0.9209 |

Table 2: Training and decoding comparison.

and $\lambda_l$=0.9 were adopted. All Gaussian parameters were jointly optimized with Adam over 50K steps using learning rates of $1e-4$, $5e-3$, $5e-2$, and $1e-3$.

**Evaluation Metrics.** PSNR measures pixel-level distortion, while MS-SSIM (Wang, Simoncelli, and Bovik 2003) evaluates perceptual and structural fidelity. For 16K images, we employed FusedSSIM (Mallick et al. 2024; Wang et al. 2004) to prevent OOM errors during evaluation.

**Baselines.** Due to the high memory cost of INR methods on UHR images, we compared only with GS-based methods, including 3DGS (Kerbl et al. 2023), GI (Zhang et al. 2024a), LIG (Zhu et al. 2025), and ImageGS (Zhang et al. 2024b).

## Evaluation

**Image Compression Performance Evaluation.** As illustrated in Table 1 and Fig. 4, the evaluation results on the DIV8K and DIV16K datasets demonstrate that SmartSplat consistently outperforms existing methods in reconstruction quality under equivalent compression ratios (CR). As the compression rate increases, the sparsity of the Gaussian distribution in existing methods often leads to the emergence of NaN values during rasterization, which disrupts the optimization process. Although ImageGS adopts an error-driven strategy by incrementally adding Gaussians, this approach tends to introduce instability when the number of Gaussians is limited. In contrast, SmartSplat employs a highly adaptive initialization strategy for Gaussian distribution, enabling stable and efficient iterative optimization across various image resolutions, even under extremely high compression ratios.

Specifically, on the DIV8K dataset, SmartSplat achieves improvements of 1.53 dB in PSNR and 0.0201 in MS-SSIM over the runner-up method, 3DGS, at the same compression ratio. It is noteworthy that 3DGS projects pixels into 3D space using an identity matrix, leading to slower training and significantly higher memory usage. Compared to the 2D Gaussian baseline GI (RS), SmartSplat achieves a 2.57dB PSNR gain and maintains similar quality at 500× compression, whereas GI (RS) requires 200× for comparable results.

On the DIV16K dataset, the advantages of SmartSplat are even more pronounced. At lower compression ratios (20/100/200), 3DGS encounters OOM issues and fails to complete training, whereas SmartSplat maintains stable optimization and achieves an average PSNR gain about 5.64 dB over GI(RS). At higher compression ratios, both GI and ImageGS fail to converge, while SmartSplat continues to deliver superior reconstruction quality and remains robust even under extremely aggressive compression ratios.

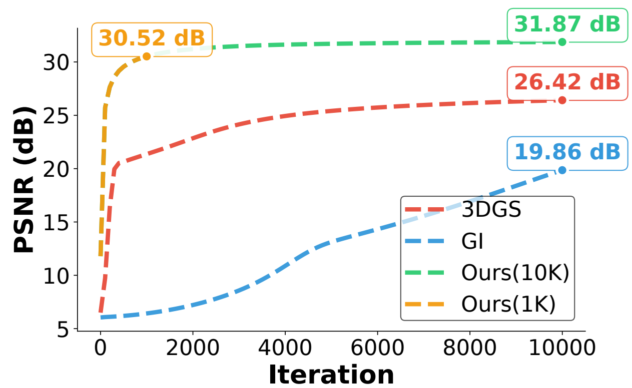


Figure 5: Training convergence speed comparison.

**Optimization Performance Evaluation.** The optimization process was evaluated on the $10848 \times 16320$ image shown in Fig. 4 under a compression ratio of $\mathrm{CR} = 200$. As illustrated in Fig. 5, SmartSplat exhibits a significantly faster convergence rate, attributed to its highly adaptive Gaussian initialization strategy. Notably, it achieves superior reconstruction quality to both 3DGS and GI within only 1K iterations—substantially outperforming their respective results even at 10K iterations. Furthermore, as reported in Table 2, although 3DGS also demonstrates strong image representation capabilities, its memory requirement is approximately $2.56\times$ that of SmartSplat, and its training time is about $3.49\times$ longer. While GI offers certain advantages in training and decoding speed, SmartSplat achieves a 10.66 dB gain in PSNR within just 1K iterations, with training time reduced to 25% of that of GI, thereby demonstrating a more favorable balance between efficiency and quality.

## Conclusion and Future Work

We proposed SmartSplat, the first GS-based image compression framework that operates effectively on UHR (8K/16K) images. By introducing gradient-color guided variational sampling and exclusion-based uniform sampling, along with scale-adaptive Gaussian color initialization, SmartSplat achieves efficient, non-overlapping Gaussian coverage and strong expressiveness. It outperforms existing methods under the same compression ratios and maintains high reconstruction quality even under extreme high compression ratios. This study primarily addresses the optimization of Gaussian spatial distribution, with future work targeting advanced attribute compression for improved efficiency.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62272343 and 62476202, and in part by the Fundamental Research Funds for the Central Universities.

## References

Agustsson, E.; and Timofte, R. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*.

Bhalgat, Y.; Lee, J.; Nagel, M.; Blankevoort, T.; and Kwak, N. 2020. LSQ+: Improving Low-bit Quantization through Learnable Offsets and Better Initialization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2978–2985.

Chen, Z.; Li, Z.; Song, L.; Chen, L.; Yu, J.; Yuan, J.; and Xu, Y. 2023. NeuRBF: A Neural Fields Representation with Adaptive Radial Basis Functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4182–4194.

Fathony, R.; Sahu, A. K.; Willmott, D.; and Kolter, J. Z. 2021. Multiplicative Filter Networks. In *Proceedings of the International Conference on Learning Representations*.

Gu, S.; Lugmayr, A.; Danelljan, M.; Fritsche, M.; Lamour, J.; and Timofte, R. 2019. DIV8K: DIVerse 8K Resolution Image Dataset. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshop*, 3512–3516.

Higham, N. J. 2009. Cholesky Factorization. *WIREs Computational Statistics*, 1(2): 251–254.

Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; and Gao, S. 2024. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. *ACM Transactions on Graphics*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2024. Compact 3D Gaussian Representation for Radiance Field. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 21719–21728.

Li, L.; Zhang, L.; Wang, Z.; and Shen, Y. 2024. GS3LAM: Gaussian Semantic Splatting SLAM. In *Proceedings of the ACM International Conference on Multimedia*, MM '24, 3019–3027. New York, NY, USA: Association for Computing Machinery.

Li, L.; Zhang, L.; Wang, Z.; Zhang, F.; Li, Z.; and Shen, Y. 2025. Representing Sounds as Neural Amplitude Fields: A Benchmark of Coordinate-MLPs and a Fourier Kolmogorov-Arnold Framework. *Proceedings of the Association for the Advancement of Artificial Intelligence*, 39(23): 24458–24466.

Mallick, S. S.; Goel, R.; Kerbl, B.; Steinberger, M.; Carrasco, F. V.; and De La Torre, F. 2024. Taming 3DGS: High-Quality Radiance Fields with Limited Resources. In *Proceedings of SIGGRAPH Asia Conference*, SA '24.

Martel, J. N. P.; Lindell, D. B.; Lin, C. Z.; Chan, E. R.; Monteiro, M.; and Wetzstein, G. 2021. Acorn: Adaptive Coordinate Networks for Neural Scene Representation. *ACM Transactions on Graphics*, 40(4).

Matsuki, H.; Murai, R.; Kelly, P. H. J.; and Davison, A. J. 2024. Gaussian Splatting SLAM. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 18039–18048.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4).

Ramasinghe, S.; and Lucey, S. 2022. Beyond Periodicity: Towards Unifying Framework for Activations in Coordinate-MLPs. In *Proceedings of the European Conference on Computer Vision*, 142–158. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-19826-7.

Ren, J.; Li, W.; Chen, H.; Pei, R.; Shao, B.; Guo, Y.; Peng, L.; Song, F.; and Zhu, L. 2024. UltraPixel: Advancing Ultra-High-Resolution Image Synthesis to New Peaks. *arXiv preprint arXiv:2407.02158*.

Ryan, O. 2006. On Runlength-based Approaches for Achieving High-speed Compression of Map Images. In *Proceedings of IEEE International Conference on Signal Processing*, volume 2.

Saragadam, V.; LeJeune, D.; Tan, J.; Balakrishnan, G.; Veeraraghavan, A.; and Baraniuk, R. G. 2023. WIRE: Wavelet Implicit Neural Representations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 18507–18516.

Sitzmann, V.; Martel, J. N. P.; Bergman, A. W.; Lindell, D. B.; and Wetzstein, G. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; and Fidler, S. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 11353–11362.

Tancik, M.; Srinivasan, P. P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J. T.; and Ng, R. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Wallace, G. 1992. The JPEG still Picture Compression Standard. *IEEE Transactions on Consumer Electronics*, 38(1): 18–34.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Wang, Z.; Simoncelli, E.; and Bovik, A. 2003. Multiscale Structural Similarity for Image Quality Assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 1398–1402.

Yu, Z.; Chen, A.; Huang, B.; Sattler, T.; and Geiger, A. 2024. Mip-Splatting: Alias-free 3D Gaussian Splatting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 19447–19456.

Zhang, J.; Huang, Q.; Liu, J.; Guo, X.; and Huang, D. 2025. Diffusion-4K: Ultra-High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, X.; Ge, X.; Xu, T.; He, D.; Wang, Y.; Qin, H.; Lu, G.; Geng, J.; and Zhang, J. 2024a. GaussianImage: 1000 FPS Image Representation and Compression by 2D Gaussian Splatting. In *Proceedings of the European Conference on Computer Vision*.

Zhang, Y.; Li, B.; Kuznetsov, A.; Jindal, A.; Diolatzis, S.; Chen, K.; Sochenov, A.; Kaplanyan, A.; and Sun, Q. 2024b. Image-GS: Content-adaptive Image Representation via 2D Gaussians. *arXiv preprint arXiv:2407.01866*.

Zhu, L.; Lin, G.; Chen, J.; Zhang, X.; Jin, Z.; Wang, Z.; and Yu, L. 2025. Large Images are Gaussians: High-Quality Large Image Representation with Levels of 2D Gaussian Splatting. *arXiv preprint arXiv:2502.09039*.