

NexusSplats: Efficient 3D Gaussian Splatting in the Wild

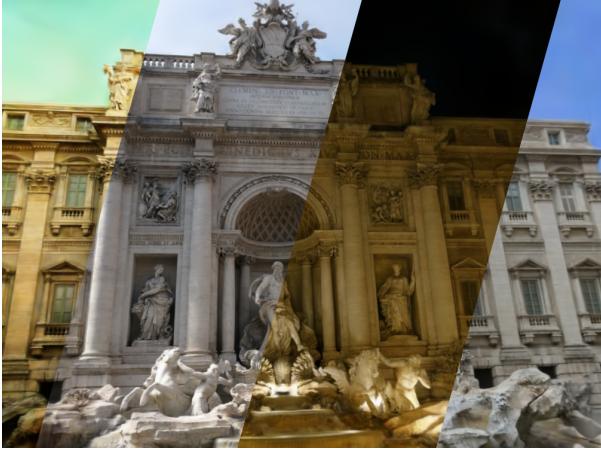
Yuzhou Tang Dejun Xu Yongjie Hou Zhenzhong Wang Min Jiang[†]

School of Informatics, Xiamen University

<https://nexus-splats.github.io/>



(a) Photos



(b) Renderings

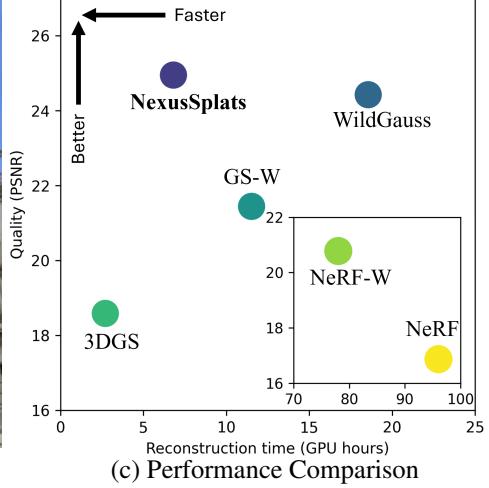


Figure 1. (a) Given photos from in-the-wild scenarios, (b) *NexusSplats* effectively decouples lighting conditions and eliminates occlusions, enabling steerable color mapping under diverse lighting conditions. (c) Our method achieves state-of-the-art results in rendering quality, with a substantial speed improvement over the current best in rendering quality.

Abstract

While 3D Gaussian Splatting (3DGS) has recently demonstrated remarkable rendering quality and efficiency in 3D scene reconstruction, it struggles with varying lighting conditions and incidental occlusions in real-world scenarios. To accommodate varying lighting conditions, existing 3DGS extensions apply color mapping to the massive Gaussian primitives with individually optimized appearance embeddings. To handle occlusions, they predict pixel-wise uncertainties via 2D image features for occlusion capture. Nevertheless, such massive color mapping and pixel-wise uncertainty prediction strategies suffer from not only additional computational costs but also coarse-grained lighting and occlusion handling. In this work, we propose a nexus kernel-driven approach, termed *NexusSplats*, for efficient and finer 3D scene reconstruction under complex lighting and occlusion conditions. In particular, *NexusSplats* leverages a novel light decoupling strategy where appearance embeddings are optimized based on nexus kernels instead of massive Gaussian primitives, thus accelerating reconstruction speeds while ensuring local color consistency for finer

textures. Additionally, a Gaussian-wise uncertainty mechanism is developed, aligning 3D structures with 2D image features for fine-grained occlusion handling. Experimental results demonstrate that *NexusSplats* achieves state-of-the-art rendering quality while reducing reconstruction time by up to 70.4% compared to the current best in quality.

1. Introduction

Reconstructing photorealistic 3D scenes from image collections is a fundamental challenge in computer vision, with applications spanning virtual reality and autonomous driving [3, 43]. The primary objective is to generate multi-view consistent 3D representations from images with known camera poses, enabling photorealistic rendering for unknown viewpoints.

As a significant advancement in this field, Neural Radiance Fields (NeRF) [1, 24, 32, 38] utilizes multilayer perceptrons (MLP) to represent 3D scenes, achieving remarkable rendering quality. While NeRF operates well in well-controlled image collections, its performance degrades significantly when presented *in-the-wild* scenarios where lighting conditions vary due to changes in weather, day-

[†]Corresponding author.

light, and camera settings while incidental occlusions appear, such as moving pedestrians and vehicles. To address these issues, NeRF-W [21] introduces per-image embeddings alongside extra MLPs to perform color mapping for decoupling lighting conditions and predict pixel-wise uncertainties for handling occlusions. Subsequent works have expanded on NeRF-W in various ways [4, 9, 18, 34, 42]. However, these approaches inherit high computational costs from NeRF, limiting their applications.

Recently, various works [3, 8, 11, 12, 19, 26, 31, 37] have developed different representations of the 3D scene to enhance the efficiency. Among these methods, 3D Gaussian Splatting (3DGS) [11, 43, 44, 48] excels in its remarkable efficiency without compromising rendering quality. Specifically, 3DGS encodes the 3D scene explicitly as a collection of 3D Gaussian primitives and uses rasterization to generate images for given viewpoints. However, similar to NeRF, 3DGS degrades its performance in rendering quality when presented *in-the-wild* scenarios with complex lighting and occlusion conditions.

Extensions of 3DGS [14, 45] attempt to address lighting variations by optimizing extra embeddings and performing color mapping on the massive Gaussian primitives, as adaptations of NeRF-based methods. To handle the occlusions, recent approaches [14, 33, 45] mainly focus on the 2D input images by introducing the latest image segmentation models [2, 29, 40]. Despite the progress, they suffer from two limitations: 1) Extensive computational costs: Typically, representing a large-scale outdoor scene involves millions of Gaussian primitives. Optimizing extra parameters of such an amount significantly slows down the training speed. 2) Rough lighting and occlusion handling: The lack of coordinated color mapping across Gaussian primitives disrupts local color consistency, degrading the quality of detailed textures. Moreover, the missing focus on the structural differences between 2D images and the 3D scene limits their effectiveness in handling occlusions. Besides, the clear boundary areas where no occlusions appear are frequently misclassified as occlusions, degrading the reconstruction results.

To address these limitations, we propose a nexus kernel-driven approach, called *NexusSplats*, for efficient and finer 3D scene reconstruction under complex lighting and occlusion conditions. Specifically, *NexusSplats* utilizes nexus kernels to divide the 3D scene into voxel segments, enabling hierarchical management for 3D Gaussian primitives. Each kernel oversees a set of Gaussian primitives within its voxel segment and optimizes an appearance embedding for coordinated color mapping on the associated Gaussian primitives. Instead of optimizing massive appearance embeddings for Gaussian primitives, this coordinated color mapping not only reduces computational costs but also ensures local color consistency when decoupling light-

ing conditions. For handling occlusions, *NexusSplats* predicts Gaussian-wise uncertainties and projects them onto pixel-wise uncertainties via rasterization, aligning the structural differences between the 3D scene and 2D images. To further reduce misidentifications on clear boundary areas, we apply a boundary penalty to the predicted pixel-wise uncertainties, relaxing the focus on clear boundary regions.

In summary, our key contributions are as follows:

- 1) We develop a novel light decoupling strategy that performs coordinated color mapping, efficiently and effectively accommodating varying lighting conditions;
- 2) We design the uncertainty splatting mechanism to predict Gaussian-wise uncertainties combined with a boundary penalty, facilitating the handling of occlusions;
- 3) Experimental results demonstrate that *NexusSplats* achieves state-of-the-art rendering quality and reduces reconstruction time in different scenes by up to 70.4% compared to the current best method in quality.

2. Related Work

2.1. 3D Scene Reconstruction

The primary objective of 3D scene reconstruction is to generate new images from viewpoints that differ from those of the known captures [10, 17]. NeRF [1, 24], which employs volume rendering [6, 16, 22], has emerged as a leading methodology in the field. However, the computationally intensive MLP optimization used by NeRF limits the reconstruction speed [5, 23, 30]. To address this, recent works have improved NeRF through enhanced scene representations [3, 8, 11, 19, 26, 37, 43]. Notably, 3DGS [11, 43] achieves remarkable results in rendering quality and efficiency by explicitly representing scenes as collections of 3D Gaussian primitives and leveraging the rasterization process over ray tracing used in NeRF. Meanwhile, various works [7, 15, 20, 25, 27, 28] further extend 3DGS to faster and more compact versions. Nevertheless, these methods often struggle with *in-the-wild* scenarios that demand additional handling for varying lighting conditions and incidental occlusions. In this paper, we investigate robust scene reconstruction techniques tailored for such challenging tasks.

2.2. Lighting Conditions Decoupling

NeRF-W [21] first addresses challenges from illumination changes by introducing per-image appearance embeddings and decoupling lighting conditions by mapping colors to adapt to different lighting conditions. Subsequent works have expanded on NeRF-W’s approach in various ways [4, 9, 18, 34, 42]. However, these methods inherit NeRF’s high computational costs, resulting in slow training and rendering speeds. Recent studies have explored using 3DGS [11, 43, 44, 48] for *in-the-wild* scenarios as a faster alternative to NeRF. Nevertheless, integrating appearance con-

ditioning with locally independent 3D Gaussian primitives presents unique challenges. Existing extensions of 3DGs [14, 45] typically optimize the appearance embeddings for each Gaussian primitive individually. However, optimizing these embeddings individually for massive Gaussian primitives incurs huge computational costs, and disrupts local color consistency due to uncoordinated color mapping across Gaussian primitives, leading to both longer training time and distortions in detailed textures.

2.3. Incidental Occlusions Handling

To mitigate the impact of incidental occlusions, such as moving pedestrians and vehicles, pioneering works [4, 21] leverage trainable embeddings to predict pixel-wise uncertainties as a filtering mask to capture and eliminate occlusions, thereby preventing their models from learning irrelevant objects from training images. Subsequent methods [14, 33, 42, 45] improve this process by introducing the latest 2D image models [2, 29, 40] to compare semantic features between training and predicted images for better occlusion capture. However, these methods neglect the structural differences between 2D semantic features and the 3D scene representation. Furthermore, they frequently misidentify clear boundary areas where no occlusions appear due to the inconsistency between training images and predicted images caused by incomplete reconstruction.

3. Methods

Figure 2 illustrates our approach, *NexusSplats*, building upon 3DGs to achieve high-fidelity scene reconstruction under varying lighting and occlusion conditions. Sec. 3.2 introduces our nexus kernels, enabling a hierarchical scene representation while performing local adaptations to different lighting and occlusion conditions. In Sec. 3.3, we detail our light decoupling module, which separates image-dependent lighting conditions, enabling coordinated color mapping. Sec. 3.4 discusses our uncertainty splatting mechanism, which handles occlusions by predicting Gaussian-wise uncertainties for a filtering mask, and refines this process with a boundary penalty.

3.1. Preliminaries: 3D Gaussian Splatting

In this section, we briefly review the process of 3DGs [11], which represents scenes using a set of anisotropic 3D Gaussian primitives, integrating the differential property of volumetric representation with the efficiency of tile-based rasterization.

Starting from Structure-from-Motion (SfM) points [35], each point is assigned as the position (mean) μ of a 3D Gaussian primitive:

$$\mathcal{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}, \quad (1)$$

where \mathbf{x} is the position in the 3D scene, and Σ denotes the covariance matrix of the 3D Gaussian primitive. To ensure Σ remains positive semi-definite, it is constructed using a scaling matrix \mathbf{S} and rotation matrix \mathbf{R} :

$$\Sigma = \mathbf{R} \mathbf{S} \mathbf{S}^\top \mathbf{R}^\top. \quad (2)$$

Each 3D Gaussian primitive is associated with a color $\hat{\mathbf{c}}$, represented by spherical harmonics (SH), and an opacity α that scales $\mathcal{G}(\mathbf{x})$ during blending.

Unlike traditional volumetric methods, 3DGs uses tile-based rasterization instead of computationally expensive ray tracing. Each 3D Gaussian primitive $\mathcal{G}(\mathbf{x})$ is projected into a 2D Gaussian primitive $\mathcal{G}'(\mathbf{x})$ on the image plane, following the projection process outlined in this work [47]. The tile-based rasterizer then sorts the 2D Gaussian primitives efficiently and applies α -blending:

$$\hat{\mathbf{C}}(\mathbf{x}') = \sum_{i \in N} \hat{\mathbf{c}}_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \quad \sigma_i = \alpha_i \mathcal{G}'_i(\mathbf{x}'), \quad (3)$$

where \mathbf{x}' is the pixel position, and N denotes the number of sorted 2D Gaussian primitives corresponding to that pixel. With the differentiable property, all attributes of 3D Gaussian primitives are learnable and optimized end-to-end via training view reconstruction. The final rendered colors $\hat{\mathbf{C}}$ are computed using a weighted combination of DSSIM [39] and L_1 losses with the reference colors \mathbf{C} :

$$\mathcal{L} = \lambda \cdot \text{DSSIM}(\hat{\mathbf{C}}, \mathbf{C}) + (1 - \lambda) \|\hat{\mathbf{C}} - \mathbf{C}\|_1, \quad (4)$$

where λ is set to 0.2 as a hyper-parameter.

3.2. Nexus Kernels

This section introduces our nexus kernels, which enable hierarchical management for Gaussian primitives in the reconstructed scene. We first divide the 3D scene into manageable voxel segments based on nexus kernels. Each kernel oversees a set of Gaussian primitives within its voxel segment, predicting the Gaussian-specific attributes for rasterization while enabling coordinated color mapping and Gaussian-wise uncertainty prediction.

Specifically, each kernel is equipped with a local appearance embedding ε_a and an uncertainty embedding ε_β . The appearance embedding ε_a captures local lighting response patterns to global illumination changes, while the uncertainty embedding ε_β estimates the likelihood of occlusion occurrence in the voxel segment.

Initialization. Following standard practices in scene representation [11, 41], we initialize nexus kernels using a sparse point cloud generated by COLMAP [35], with additional kernels positioned on a surrounding sphere to model the sky. Each kernel u is assigned a voxel segment of the 3D

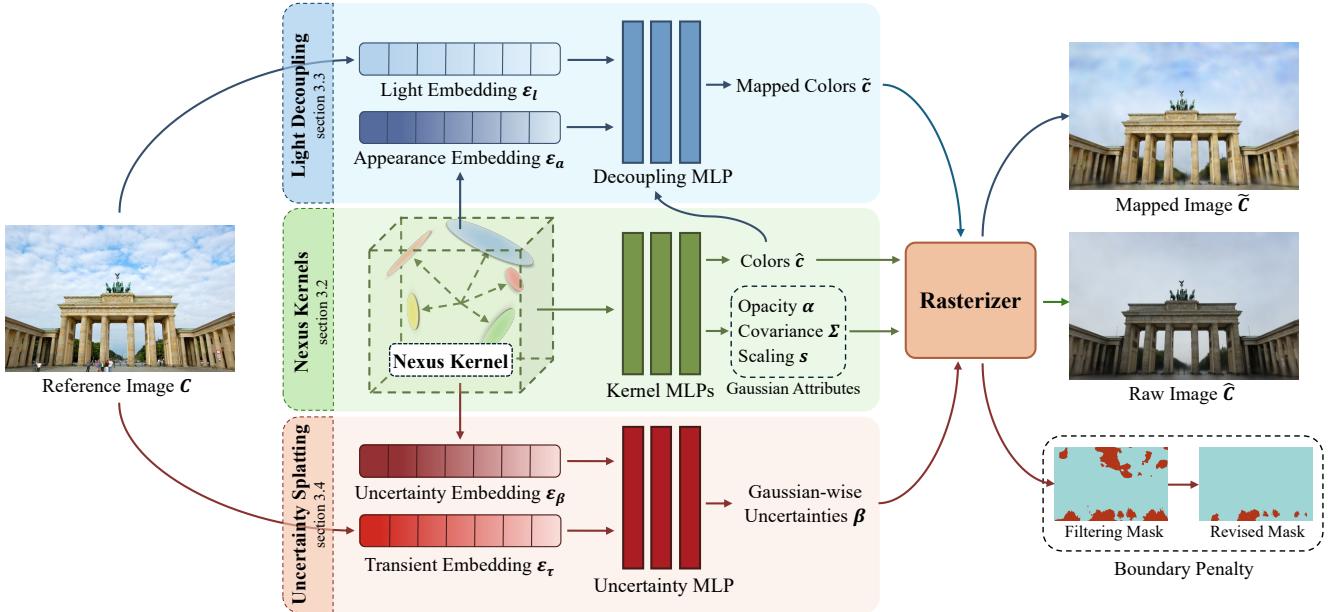


Figure 2. Overview of NexusSplats. *Left:* From the reference image \mathbf{C} , we extract light embedding ε_l and transient embedding ε_τ to capture global lighting and occlusion conditions. *Middle:* Our nexus kernels (Sec. 3.2) enable hierarchical management of Gaussian primitives, allowing efficient local adaptations to different lighting and occlusion conditions via the light decoupling module (Sec. 3.3) and the uncertainty splatting module (Sec. 3.4). *Right:* Through tile rasterization, we project raw colors $\hat{\mathbf{c}}$, mapped colors $\tilde{\mathbf{c}}$, and uncertainties β onto 2D visible planes. A boundary penalty finally refines the filtering mask in handling occlusions.

scene, overseeing h Gaussian primitives within the voxel segment. By optimizing a local context feature vector \mathbf{f}_u , a scaling factor k_u , and learnable offsets $\{\mathcal{O}_i\}_{i=1}^h$, each kernel controls the positions and other attributes of h Gaussian primitives. The positions $\{\mu_i\}_{i=1}^h$ are calculated by:

$$\mu_i = \mathbf{x}_u + \mathcal{O}_i \cdot k_u, \quad (5)$$

where \mathbf{x}_u is the central position of kernel u . Other attributes, i.e. opacity α , covariance matrix Σ , scaling s , and color $\hat{\mathbf{c}}$, are predicted by Kernel MLPs denoted as F_α , F_σ , F_s and F_c .

Color Expression. To provide the light decoupling module with a more suitable color expression, we modify the color expression in 3DGS by fixing the color channels to three and setting the SH degree to one. 3DGS employs SH to express colors accurately and ensure view consistency, progressively increasing the SH degree to a maximum level over time, thereby expanding color vector dimensions.

However, simply fixing the color channels can lead to view inconsistency because of reduced dimensions. Therefore, we additionally pass the viewing direction \vec{d}_{uv} to F_c to ensure the view consistency. For a camera located at position \mathbf{x}_v , each Gaussian primitive's color $\hat{\mathbf{c}} \in \mathbb{R}^3$ can be predicted as:

$$\{\hat{\mathbf{c}}_i\}_{i=1}^h = F_c(\mathbf{f}_u, \vec{d}_{uv}), \quad \vec{d}_{uv} = \frac{\mathbf{x}_u - \mathbf{x}_v}{\|\mathbf{x}_u - \mathbf{x}_v\|_2}, \quad (6)$$

where \mathbf{x}_u denotes the location of nexus kernel u , and \mathbf{f}_u denotes the per-kernel context feature vector. The remaining attributes of Gaussian primitives are predicted similarly as Eq. (6).

Kernel Refinement. To relieve the computational cost of optimizing extensive attributes and embeddings, we apply a pruning process to eliminate redundant kernels. By accumulating the opacity values of the Gaussian primitives associated with a nexus kernel, we eliminate it and its associated Gaussian primitives if the accumulated opacity is below a threshold. Additionally, to encourage compact scene representations with minimal overlap of Gaussian primitives, a regulation loss is introduced:

$$\mathcal{L}_{\text{reg}} = \sum_{i=1}^{N \times h} \text{Prod}(\mathbf{s}_i), \quad (7)$$

where N represents the total number of nexus kernels, $\text{Prod}(\cdot)$ denotes the product of a vector's values, and \mathbf{s} is the scaling factor of each Gaussian primitive.

3.3. Light Decoupling

Leveraging the hierarchical management for Gaussian primitives, we perform coordinated color mapping to decouple lighting conditions from training images. Our light decoupling module incorporates three components: an

image-specific light embedding ε_l , a kernel-specific appearance embedding ε_a , and a decoupling MLP F_θ . The light embedding ε_l captures the global lighting conditions of training images, while the appearance embedding ε_a learns the localized response to global lighting variations. Using these embeddings, we transform the raw colors $\{\hat{\mathbf{c}}_i\}_{i=1}^h$ of Gaussian primitives into mapped colors $\{\tilde{\mathbf{c}}_i\}_{i=1}^h$ via F_θ :

$$\{\tilde{\mathbf{c}}_i\}_{i=1}^h = F_\theta(\{\hat{\mathbf{c}}_i\}_{i=1}^h, \varepsilon_a^{(u)}, \varepsilon_l^{(v)}, \vec{d}_{uv}), \quad (8)$$

where h denotes the number of offsets associated with nexus kernel u , and v marks the specific training image. To maintain view consistency after the coordinated color mapping, we additionally input the viewing direction \vec{d}_{uv} into F_θ , similar to Eq. (6). Furthermore, we set a dropout rate of 0.2 within F_θ to mitigate overfitting in decoupling lighting conditions.

Both raw colors $\hat{\mathbf{c}}$ and mapped colors $\tilde{\mathbf{c}}$ are projected onto 2D planes as $\hat{\mathbf{C}}$ and $\tilde{\mathbf{C}}$, respectively, using Eq. (3). Consistent with 3DGS, we train the model using a combination of DSSIM [39] and L_1 losses (Eq. (4)). Given the reference colors \mathbf{C} from the training images, we compute the L_1 loss between \mathbf{C} and mapped colors $\tilde{\mathbf{C}}$, guiding the light decoupling module to capture image-specific lighting effects. Additionally, we compute the DSSIM loss directly on raw colors $\hat{\mathbf{C}}$ to leverage its robustness to changes in appearance colors, focusing primarily on structural and perceptual similarity. The training loss for colors is formulated as:

$$\mathcal{L}_{\text{color}} = \lambda \cdot \text{DSSIM}(\hat{\mathbf{C}}, \mathbf{C}) + (1 - \lambda) \|\tilde{\mathbf{C}} - \mathbf{C}\|_1. \quad (9)$$

3.4. Uncertainty Splatting

This section elaborates on our design for handling occlusions, termed uncertainty splatting. We align the pixel-wise uncertainties with the 3D scene representation, i.e. the distributed 3D Gaussian primitives. In particular, we propagate the pixel-wise uncertainties to each Gaussian primitive and splat them onto visible planes as the filtering mask through rasterization.

Gaussian-wise Uncertainty. In parallel with the light decoupling module, we predict Gaussian-wise uncertainties $\{\beta_i\}_{i=1}^h$ using three components: an image-level transient embedding ε_τ , a kernel-level uncertainty embedding ε_β , and an uncertainty MLP F_β :

$$\{\beta_i\}_{i=1}^h = F_\beta(\varepsilon_\beta^{(u)}, \varepsilon_\tau^{(v)}). \quad (10)$$

Through rasterization (Eq. (3)), we project per-Gaussian uncertainties β onto 2D planes to produce pixel-wise uncertainties $\hat{\beta}$.

Training Objective. By modeling observed colors \mathbf{C} with an isotropic normal distribution with mean $\tilde{\mathbf{C}}$ and variance $\hat{\beta}$, we optimize predicted uncertainties $\hat{\beta}$ by minimizing the negative log-likelihood of the normal distribution:

$$\mathcal{L}_u = \frac{\|\tilde{\mathbf{C}} - \mathbf{C}\|_2^2}{2\hat{\beta}^2} + \lambda_1 \log \hat{\beta}, \quad (11)$$

where λ_1 is a regression hyper-parameter, set to 0.5 in our experiments. However, the standard L_2 loss between $\tilde{\mathbf{C}}$ and \mathbf{C} has limited effectiveness in identifying occlusions that are not part of the static scene [14, 33]. To address this, we replace the L_2 loss with a cosine similarity \mathcal{D} between the semantic feature vectors of rendered image $\tilde{\mathbf{C}}$ and reference image \mathbf{C} :

$$\mathcal{D}(\tilde{\mathbf{C}}, \mathbf{C}) = \sum_{i=1}^P \frac{\tilde{\mathcal{E}}_i \cdot \mathcal{E}_i}{\|\tilde{\mathcal{E}}_i\|_2 \|\mathcal{E}_i\|_2}, \quad (12)$$

where $\tilde{\mathcal{E}}$ and \mathcal{E} are the patch-wise feature vectors extracted by a pre-trained DINO model [2, 29], and P is the total number of patches. This adjusted loss function allows our method to better capture occlusions from training images.

Nevertheless, our method also tends to misidentify clear boundary regions as occlusions. These misidentifications can degrade reconstruction quality, bringing drawbacks to occlusion handling.

Boundary Penalty. Due to the insufficient representation of Gaussian primitives alongside the lack of observations in training images, the boundary areas of a reconstructed 3D scene often present lower rendering quality, leading to different semantic features between training and predicted images. Therefore, we apply a boundary penalty \mathcal{M} to the predicted pixel-wise uncertainties $\hat{\beta}$ to loosen the criteria at the boundaries. For a pixel at (x, y) , \mathcal{M} is modeled as a 2D Gaussian distribution:

$$\mathcal{M}(x, y) = \exp \left(-\frac{(x - \bar{x})^2}{2\sigma_x^2} - \frac{(y - \bar{y})^2}{2\sigma_y^2} \right), \quad (13)$$

where \bar{x} and \bar{y} represent the distribution center, and σ_x^2 and σ_y^2 are variances along the x and y axes. This penalty reduces sensitivity at the periphery, encouraging our model to focus on correct occlusions in central areas.

3.5. Optimization

Our final training objective combines all the loss terms as follows:

$$\mathcal{L} = \frac{\mathcal{L}_{\text{color}}}{2\hat{\beta}^2 \mathcal{M}} + \mathcal{L}_u + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \quad (14)$$

where λ_{reg} is set to 0.01 in our experiments, and the pixel-wise uncertainties $\hat{\beta}$ restricted by the boundary penalty \mathcal{M} act as filters on the color loss $\mathcal{L}_{\text{color}}$, excluding imperfect pixels that contain occlusions during optimization.

Table 1. **Quantitative Comparison.** This table compares the performance of reconstruction quality across three scenes from the Photo Tourism dataset [36]: Brandenburg Gate, Sacre Coeur, and Trevi Fountain. Highlighted values indicate the first, second, and third scores for each metric and scene. Results for 3DGS-based methods are reproduced and evaluated based on NerfBaselines [13], with other results reported from WildGauss [14].

Method	Brandenburg Gate			Sacre Coeur			Trevi Fountain		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [24]	18.90	0.815	0.231	15.60	0.715	0.291	16.14	0.600	0.366
NeRF-W-re [21]	24.17	0.890	0.167	19.20	0.807	0.191	18.97	0.698	0.265
K-Planes [9]	25.49	0.879	0.224	20.61	0.774	0.265	22.67	0.714	0.317
3DGS [11]	20.02	0.882	0.176	17.68	0.835	0.204	18.07	0.709	0.272
GS-W [45]	24.32	0.909	0.148	19.57	0.826	0.207	20.48	0.734	0.252
WildGauss [14]	27.23	0.926	0.135	22.56	0.859	0.177	23.52	0.765	0.228
<i>NexusSplats</i> (Ours)	27.76	0.922	0.141	23.13	0.859	0.174	23.96	0.766	0.240

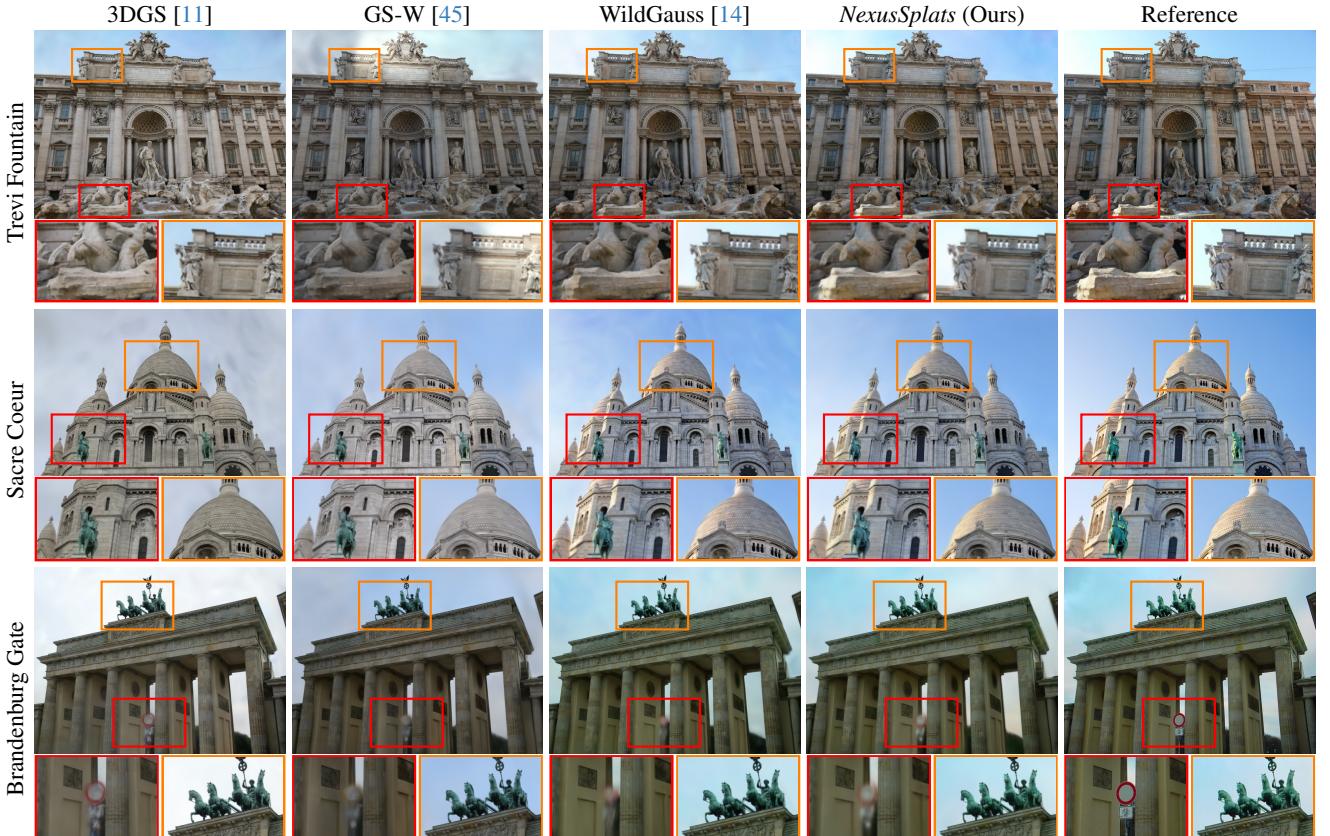


Figure 3. **Qualitative Comparison.** For each row, different architectural scenes are shown with highlighted red and orange boxes for closer inspection. *NexusSplats* presents sharper details and improved color fidelity compared to other methods, closely matching the reference images, particularly in challenging areas with intricate textures.

4. Experiments

4.1. Experimental Setup

Dataset and Metrics. We evaluate our method on three scenes from the challenging and widely-recognized dataset, Photo Tourism [36], which presents diverse lighting conditions and incidental occlusions. For quantitative comparisons,

we use PSNR, SSIM [39], and LPIPS [46] to assess the performance in rendering quality. For qualitative comparison, we inspect rendered images from matching input poses to verify visual fidelity. Additionally, we report reconstruction time in GPU hours to measure the reconstruction efficiency. Our full quantitative results are provided in the *supp. mat.*

Table 2. Efficiency Comparison. This table reports the reconstruction times (in GPU hours) for WildGauss [14] and our proposed *NexusSplats* across three scenes: I, II and III represent for Brandenburg Gate, Sacre Coeur, and Trevi Fountain, respectively. The percentage reduction in training time achieved by *NexusSplats* is also provided.

Method	Scene		
	I	II	III
WildGauss [14]	14.93h	14.2h	26.48h
<i>NexusSplats</i> (Ours)	6.17h	6.43h	7.83h
Time Reduction	58.7%	54.7%	70.4%

Table 3. Ablation Studies on the Photo Tourism dataset [36], showing the average performance across all three scenes. We assess the impact of the uncertainty splatting and light decoupling modules on reconstruction quality. Removing either module leads to a notable decrease in quality. In addition, we assess the importance of the boundary penalty by evaluating *NexusSplats* with only the boundary penalty disabled.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
w/o uncert.	24.55	0.847	0.192
w/o penalty.	24.34	0.848	0.188
w/o light.	18.44	0.799	0.245
<i>NexusSplats</i>	24.95	0.849	0.185

Baseline and Implementation. We take WildGauss [14] as the primary baseline for its state-of-the-art performance in handling illumination changes and occlusions in 3D reconstruction. For quantitative comparison, we include NeRF [24] along with its leading improvements, NeRF-W [21] and K-Planes [9], as reported in WildGauss, and evaluate the latest 3DGS-based methods, including 3DGS [11] and GS-W [45], on our machines. For qualitative comparison, we focus on 3DGS and its variants to assess visual performance across different conditions.

Notably, the image-specific lighting conditions represented by embeddings are optimized during training, but the unseen lighting conditions in test images cannot be inferred without optimization. Therefore, we optimize the image-specific embeddings of unseen images for 100 iterations with the other parameters frozen. To ensure fair comparisons, we follow the standard evaluation protocol proposed by NeRF-W [21], i.e. optimizing the embeddings only on the left half of test images while evaluating on the right half. All 3DGS-based methods, including ours, are trained for 200k iterations on a single NVIDIA A40 GPU. Evaluations are conducted using the NerfBaselines framework [13], consistent with WildGauss.



Figure 4. Uncertainty Splatting Visualization for two scenes from the Photo Tourism dataset [36]. The first row presents results from Sacre Coeur, and the second row from Trevi Fountain. Without the boundary penalty, our approach (column 3) outperforms WildGauss [14] (column 2) in capturing occlusions, though it occasionally misidentifies clear boundary areas, such as the sky and water. The full uncertainty splatting module of *NexusSplats* (column 4) captures occlusions more accurately.

4.2. Results Analysis

Rendering Quality Comparison. In our experiments, we conduct quantitative and qualitative comparisons to evaluate the performance of our approach. The quantitative results in Table 1 show that *NexusSplats* achieves the highest PSNR values across all three scenes and delivers competitive results on other metrics. Leveraging the light decoupling and uncertainty splatting module, *NexusSplats* provides state-of-the-art reconstruction results, generating more consistent and detailed views under complex lighting and occlusion conditions.

The qualitative comparison in Figure 3 further illustrates these advantages. Notably, our method surpasses other methods in handling complex textures and intricate architectural details, demonstrating its capability to consistently produce high-quality visual reconstructions with sharper details and more accurate color mapping, closely approximating the reference images.

Efficiency Comparison. Table 2 provides an efficiency comparison in terms of reconstruction time between our method and WildGauss [14] which excels other existing methods in rendering quality. Leveraging the coordinated color mapping enabled by the hierarchical management of Gaussian primitives, *NexusSplats* decouples lighting conditions more efficiently, significantly reducing training times across all scenes. This reduction highlights the efficiency of our method, enabling faster reconstruction under complex lighting and occlusion conditions without compromis-



Figure 5. Light Decoupling Visualization. *NexusSplats* maps colors from the reconstructed scene to match the target lighting conditions from reference images. We display six target lighting conditions under three different camera poses for the Trevi Fountain scene, demonstrating consistent color adaptation across lighting variations.

ing quality. The substantial decrease in training time further demonstrates our method’s suitability for large-scale tasks, which require millions of parameters for detailed 3D representation, making it a practical and scalable alternative to existing 3DGS extensions.

4.3. Ablation Studies

The ablation study in Table 3 confirms the critical contributions of both the uncertainty splatting and light decoupling module to the quality performance of our method. Removing the uncertainty splatting module results in a slight quality decrease, indicating its role in enhancing fine detail. On the other hand, disabling the light decoupling module leads to a more significant drop in rendering quality, emphasizing its importance in decoupling lighting variations and preserving color fidelity. Additional visualizations of both uncertainty prediction and color mapping can be found in the *supp. mat.*

Uncertainty Splatting Visualization. Figure 4 showcases the improved performance in handling occlusions of *NexusSplats* over WildGauss. Without the boundary penalty, however, it occasionally misidentifies clear boundary areas, such as the sky and water, as occlusions. Incorporating the boundary penalty refines the performance in handling occlusions, mitigating the drawbacks of handling occlusions. Ablation results in Table 3 further demonstrate the importance of the boundary penalty.

Light Decoupling Visualization. Figure 5 illustrates the capability of *NexusSplats* to decouple and adapt lighting conditions from reference images through coordinated and steerable color mapping. Six distinct lighting conditions from the Trevi Fountain scene, captured at different times and under various ambient lighting scenarios, serve as the target lighting conditions. *NexusSplats* successfully maps reconstructed scene colors to these target lighting conditions across three camera poses, demonstrating its resilience to illumination variations while maintaining color consistency and detail integrity.

5. Conclusion

In this work, we introduced *NexusSplats*, an efficient framework for 3D scene reconstruction under complex lighting and occlusion conditions. Leveraging nexus kernels for hierarchical management of Gaussian primitives, our light decoupling module performs efficient and coordinated color mapping, effectively decoupling lighting conditions while accelerating the reconstruction process. By aligning the structural differences between 2D image features and the 3D scene representation, our uncertainty splatting module enhances the performance in handling occlusions. Experiments show that *NexusSplats* achieves state-of-the-art rendering quality and significantly reduces the reconstruction time, making it a practical solution for various real-world applications.

References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. [1](#) [2](#)
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pages 9650–9660, 2021. [2](#) [3](#) [5](#)
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, pages 333–350. Springer, 2022. [1](#) [2](#)
- [4] Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Ying Feng, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild. In *CVPR*, pages 12943–12952, 2022. [2](#) [3](#)
- [5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. [2](#)
- [6] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *SIGGRAPH*, 22(4):65–74, 1988. [2](#)
- [7] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *NeurIPS*, 2023. [2](#)
- [8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022. [2](#)
- [9] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, pages 12479–12488, 2023. [2](#) [6](#) [7](#)
- [10] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Semin. Graph. Pap.: Push. Bound.*, Vol. 2, pages 453–464. ACM, 2023. [2](#)
- [11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. [2](#) [3](#) [6](#) [7](#)
- [12] Jonas Kulhanek and Torsten Sattler. Tetra-nerf: Representing neural radiance fields using tetrahedra. In *CVPR*, pages 18458–18469, 2023. [2](#)
- [13] Jonas Kulhanek and Torsten Sattler. Nerfbaselines: Consistent and reproducible evaluation of novel view synthesis methods. *arXiv preprint arXiv:2406.17345*, 2024. [6](#) [7](#)
- [14] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *NeurIPS*, 2024. [2](#) [3](#) [5](#) [6](#) [7](#)
- [15] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *CVPR*, pages 21719–21728, 2024. [2](#)
- [16] Marc Levoy. Efficient ray tracing of volume data. *ACM TOG*, 9(3):245–261, 1990. [2](#)
- [17] Marc Levoy and Pat Hanrahan. Light field rendering. In *Semin. Graph. Pap.: Push. Bound.*, Vol. 2, pages 441–452. ACM, 2023. [2](#)
- [18] Peihao Li, Shaohui Wang, Chen Yang, Bingbing Liu, Weichao Qiu, and Haoqian Wang. Nerf-ms: Neural radiance fields with multi-sequence. In *ICCV*, pages 18591–18600, 2023. [2](#)
- [19] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Adv. Neural Inf. Process. Syst.*, 33:15651–15663, 2020. [2](#)
- [20] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *CVPR*, pages 20654–20664, 2024. [2](#)
- [21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. [2](#) [3](#) [6](#) [7](#)
- [22] Nelson Max. Optical models for direct volume rendering. *IEEE TVCG*, 1(2):99–108, 1995. [2](#)
- [23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. [2](#)
- [24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 65(1):99–106, 2021. [1](#) [2](#) [6](#) [7](#)
- [25] Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. *ECCV*, 2023. [2](#)
- [26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):1–15, 2022. [2](#)
- [27] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compgs: Smaller and faster gaussian splatting with vector quantization. *ECCV*, 2024. [2](#)
- [28] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *CVPR*, pages 10349–10358, 2024. [2](#)
- [29] Maxime Oquab, Timothée Darct, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. [2](#) [3](#) [5](#)
- [30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. [2](#)
- [31] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *CVPR*, pages 14335–14345, 2021. [2](#)
- [32] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, pages 12932–12942, 2022. [1](#)

- [33] Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *CVPR*, pages 8931–8940, 2024. [2](#), [3](#), [5](#)
- [34] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *ECCV*, pages 615–631. Springer, 2022. [2](#)
- [35] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. [3](#)
- [36] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, pages 835–846. ACM, 2006. [6](#), [7](#)
- [37] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, pages 5459–5469, 2022. [2](#)
- [38] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, pages 1–12, 2023. [1](#)
- [39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. [3](#), [5](#), [6](#)
- [40] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yong-dong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE TIP*, 30:1169–1179, 2020. [2](#), [3](#)
- [41] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, pages 5438–5448, 2022. [3](#)
- [42] Yifan Yang, Shuhai Zhang, Zixiong Huang, Yubing Zhang, and Mingkui Tan. Cross-ray neural radiance fields for novel-view synthesis from unconstrained image collections. In *ICCV*, pages 15901–15911, 2023. [2](#), [3](#)
- [43] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, pages 19447–19456, 2024. [1](#), [2](#)
- [44] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. [2](#)
- [45] Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. *ECCV*, 2024. [2](#), [3](#), [6](#), [7](#)
- [46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. [6](#)
- [47] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proc. Vis., 2001. VIS'01*, pages 29–538. IEEE, 2001. [3](#)
- [48] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *SIGGRAPH*, pages 371–378, 2001. [2](#)